

Graphic simulator of TCP communications



Universidad de Alcalá

User guide

Index

1. The application	2
1.1. Considerations.....	2
1.2. Parameters that can be entered by the user	3
1.3. User interface	4
1.3.1. The simulation.....	5
1.3.2. Information/Help	7
2. TCP	8

1. The application

This web application allows you to simulate, through parameters that the user can choose, a TCP communication between a client and a server, showing a communication diagram as a result. In section [2. TCP](#) you can learn briefly more about this protocol.

For a proper functioning of the application it is recommended to use [Google Chrome](#) or Chromium-based browsers, such as [Vivaldi](#), [Microsoft Edge](#) or [Opera](#).

1.1. Considerations

The following is taken into account for the simulation:

- The propagation time will be 'half a tick'.
- The data length of SYN, SYN ACK, ACK, FIN will be 0 bytes.
- Communications will be between two entities, 'client' (which starts the communication and sends data) and 'server' (which responds with more data).
- The confirmation policy for the reception of segments will be carried out as follows:
 - During the establishment phase and the connection closure, they settle immediately.
 - If there is data to be sent in the opposite direction, it will be settled immediately.
 - If all previous segments have been recognized and a new segment arrives with the expected acknowledgement number, a delayed ACK is sent. During 1.5 ticks the arrival of another correct segment is expected and, in case it is not received, an immediate ACK is sent.
 - A unique cumulative ACK is sent that recognizes both segments in the event that a segment with an expected sequence number is received and there is another segment in order waiting for your ACK to be transmitted.
 - If a gap is detected, i.e. a segment with a higher than expected sequence number is received, a duplicate ACK is immediately sent with the sequence number of the next expected octet (lower gap limit).

- If the segment that is received partially or totally completes the gap detected in the received data, an ACK is sent immediately if the segment begins with the lower limit of the gap.

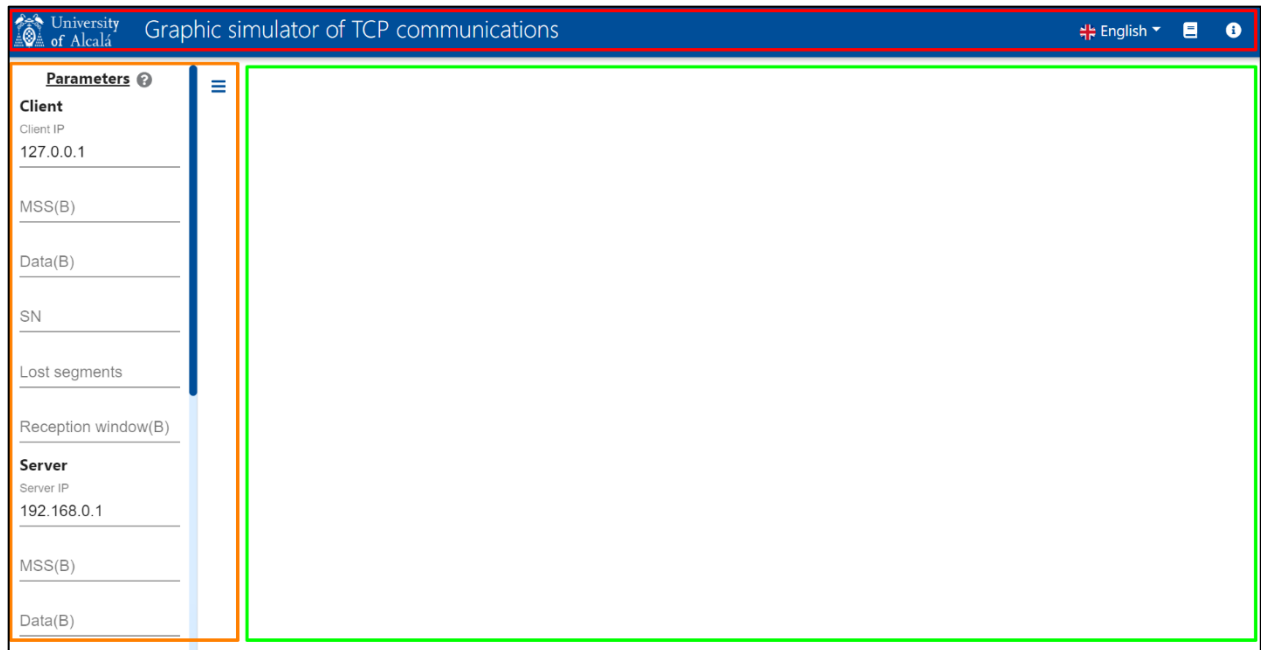
1.2. Parameters that can be entered by the user

The parameters that the user can enter are the following (* mandatory):

- **IP (Internet Protocol)***: Identifier of the network interface of a device. IPv4 (32 bits) will be used, with format [0-255].
- **MSS (Maximum Segment Size)***: Maximum size, in bytes, that a device can receive in a single segment without fragmentation.
- **Data***: Size, in bytes, of the data to be sent.
- **SN (Sequential Number)***: Initial sequence number, exchanged at the beginning by both devices, are the identifiers of the data within the byte stream in order to identify and count the bytes of the application.
- **Lost segments**: Segments of data to be sent that do not reach the destination. In the simulation there will be no loss of packets in the connection and closing phase.
- **Reception window***: Buffer, in bytes, which has the device in which to store received frames.
- **Timeout**: Maximum time, in ticks, to consider a package lost because it has not received its corresponding confirmation.
- **Congestion algorithm***: Algorithm to be used when there is congestion in the network. In this case you can choose between TCP Tahoe and TCP Reno. These are still under development, so they will not be taken into account and will be simulated only with the Congestion Avoidance algorithm.
- **Threshold/Ssthresh***: Congestion window limit from which to switch to Congestion Avoidance (CA) mode.
- **Connection closure***: Which device will initiate the end of connection.

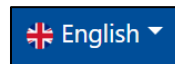
1.3. User interface

The interface has three clearly defined areas:



- **Top bar (in red):** Logo of the University of Alcalá that redirects to its website, application title (visible or hidden depending on screen size) and language selection buttons (currently available in Spanish and English), user guide (this document) and application information ("about").

- Language selection:






- User guide:



- Application information:

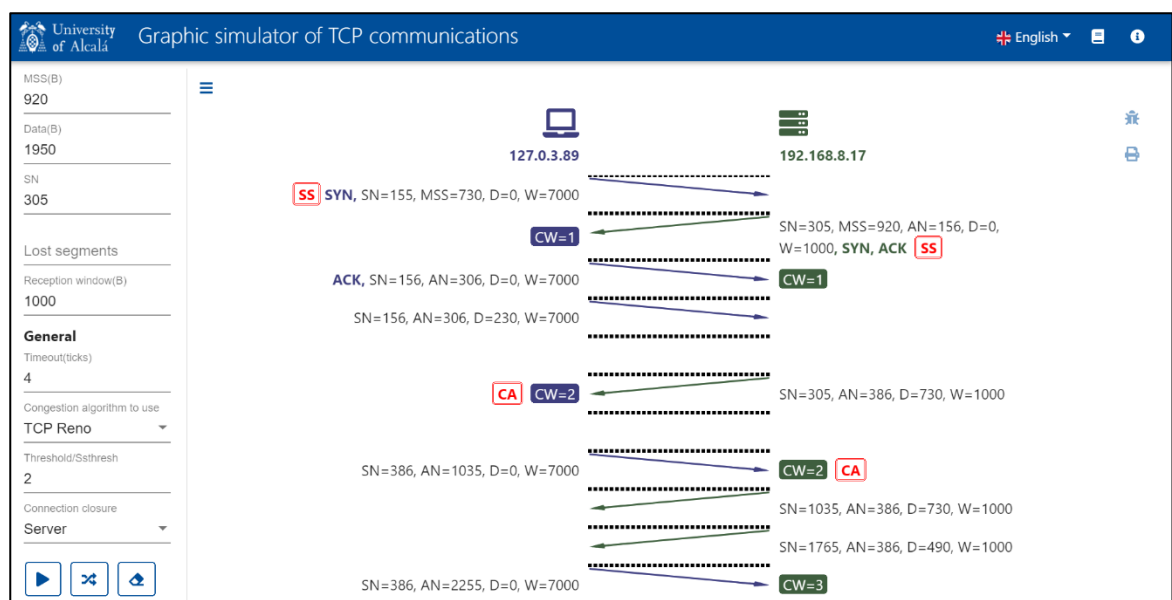


- **Parameter menu (in orange):** Next to the word "Parameters" there is a button with the considerations that have been taken into account for the simulation and information about the parameters to be entered, same as points [1.1. Considerations](#) and [1.2. Parameters that can be entered by the user](#). At the end of the menu, scrolling with the sidebar, there are three buttons:

- Simulate (): When pressed, if the parameters required to carry out the simulation have been filled in, the communications diagram is carried out and displayed. If any parameter is missing or wrong, alerts will appear indicating what is wrong or missing.
- Fill in with random data (): Fill in the parameter form with random values, except for missing segments, for both client and server.
- Clean form (): Clean form (): Returns the application to its initial state, cleaning the form and removing the simulation if one is displayed.
- **Simulation (in green):** This part shows the fins generated in case there were any when trying to run the simulation or, if everything is correct, the resulting communications diagram.

1.3.1. The simulation


If all parameters are correct, the simulation is calculated and the communications diagram is displayed next to two buttons on the right of the screen:



The data we can see in the simulation are the following:

- **SS, CA and FR:** algorithms used, which are Slow Start (SS), Congestion Avoidance (CA) and Fast Recovery (FR).
- **SYN, ACK, FIN:** Sent segment, synchronisation (SYN), acknowledgement (ACK) and finalisation (FIN) flags.
- **SN:** Sequence number.
- **AN:** Recognition number.
- **MSS:** Maximum segment size.
- **D:** Data.
- **W:** Reception window
- **VC:** Congestion window.

Regarding the two buttons on the right:

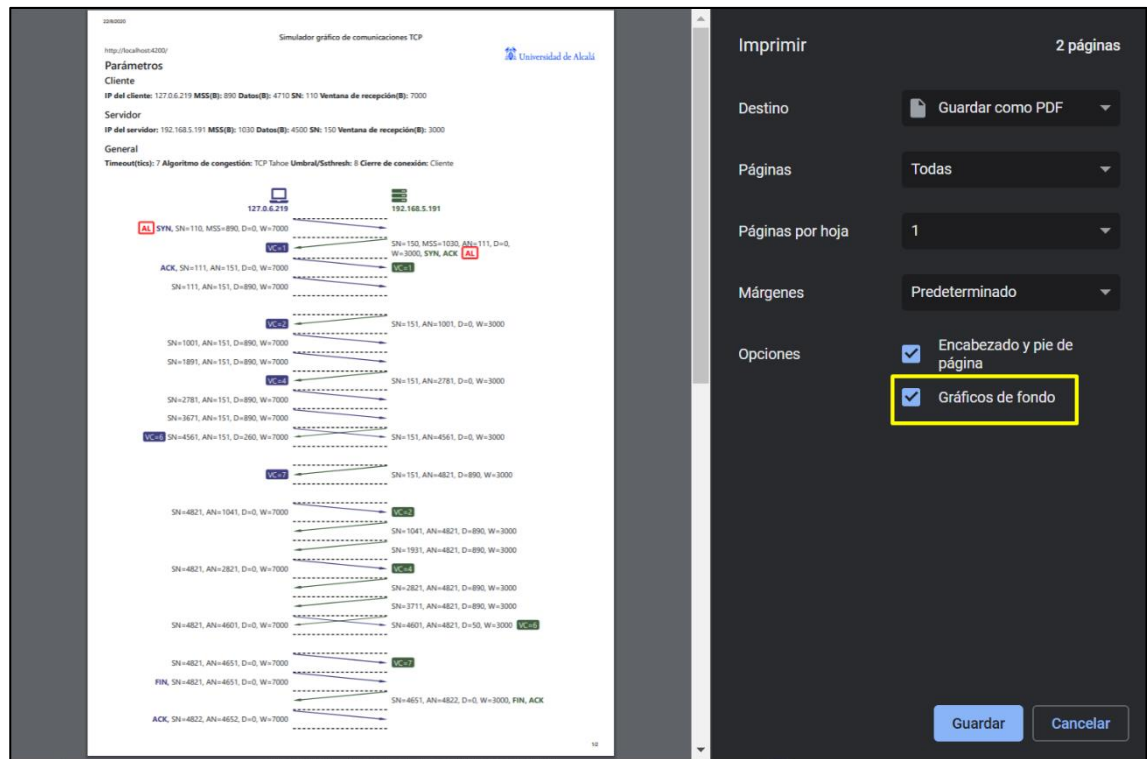
- **Report an error in the simulation** (

Asunto [Report an error in the simulation](#)

Parameters
{"ipclien":"127.0.4.189","mssclien":1580,"datosclien":4660,"snclien":425,"segperdcien":null,"wclien":4000,"ipserv":"192.168.6.0","mssse

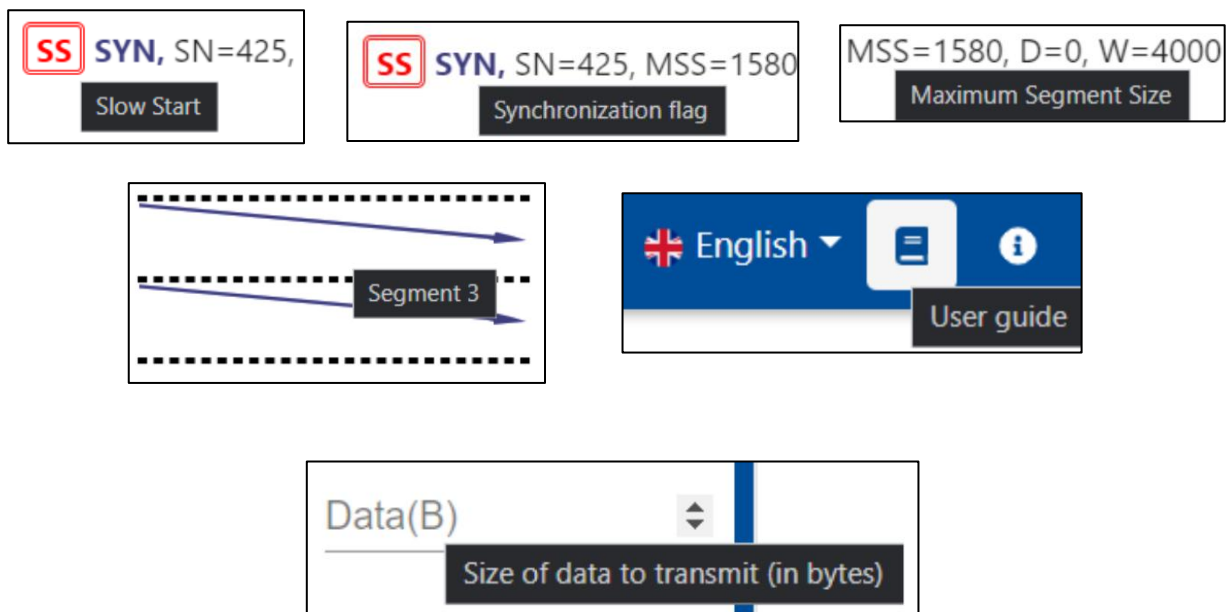
-----Write from here-----

- **Print simulation (🖨️):** It opens the browser print window with the simulation and all the parameters entered by the user. It is recommended to activate the option "Background graphics", or similar, for a correct printing.



1.3.2. Information/Help

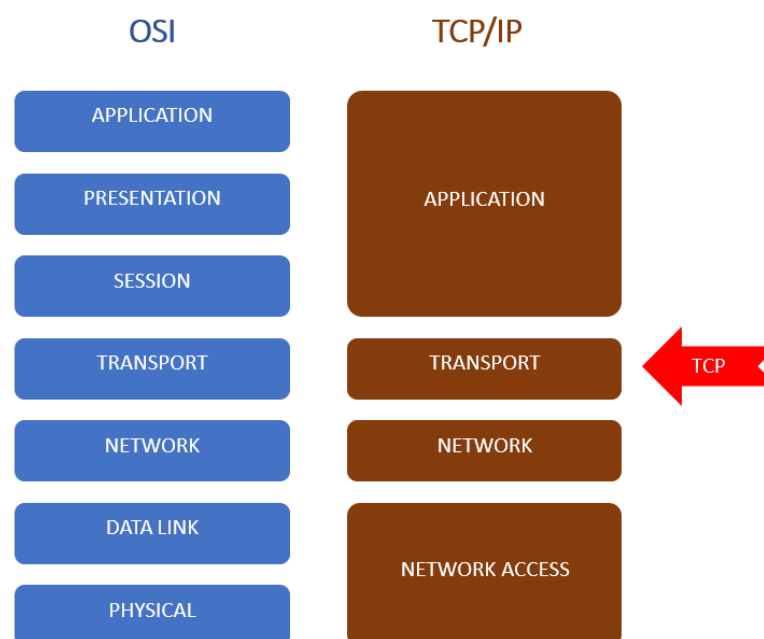
Every element in the application has a tooltip, so you only have to put the mouse pointer over it, from the buttons to the simulation segments. For example:



2. TCP

To learn about TCP, as well as how computer networks work, it is recommended that you read the book “Computer Networking: A Top-Down Approach” by James F. Kurose and Keith W. Ross, publisher in Pearson (ISBN: 9780134296159).

In summary, TCP is the acronym for Transmission Control Protocol and is one of the many protocols used in the network. It was created by the same people who created the model that takes its name, Vinton Cerf and Robert E. Kahn, in the mid-1970s and is located at the layer that communicates the network layer (IP, Internet Protocol) with the application layer, known as the transport layer.



Comparison between the OSI model and the TCP/IP model

In contrast to the UDP (User Datagram Protocol), TCP is connection-oriented, which means that both machines, sender and receiver, must recognize each other and establish a reliable communication, for which three steps are distinguished: establishment of connection, data transfer and connection closure. This involves a point-to-point connection, a single transmitter and receiver, with all the advantages and disadvantages that this entails.

An advantage could be the control over the packets, because if a packet is lost it is sent again, as well as the control of the congestion to avoid the network to be saturated. On the other hand, the disadvantage is that, being a connection-oriented protocol, data transfer is slower than in UDP, being less preferable its use in programs that require extreme immediacy or low latency.

TCP, to guarantee a correct operation, uses several congestion control algorithms. These are, if no segments are lost, Slow-start, applied in the connection phase and used until the congestion threshold is reached; and Congestion Avoidance, where, when the congestion threshold has been reached, the window increases in a linear way. In the case that the receiver receives a segment with a sequence number that it should not by the loss of a segment, the Fast Recovery algorithm is used. In addition, improving the above mentioned algorithms, a wide range of them are available, such as TCP Reno and TCP Tahoe, but they are not the only ones¹.

This document has been translated with the help of [DeepL](#).

¹ Wikipedia. (2020). TCP congestion control, from Wikipedia. Website: https://en.wikipedia.org/wiki/TCP_congestion_control#Algorithms