Lab 3 – LinkLayer Protocol

First Lab Class

Objective: Exchange arrays of bytes between two computers using a serial cable and implement the establishment phase of the data link protocol.

Steps:

1. **Select the correct computers**

Each workbench has a rack of four computers (tux1 – tux4), and a separate computer closer to the center of the room. In this work, we will use the serial port of this last computer and the one from tux3. To select tux3, you should press the number 3 on the KVM switch (room I320), or press Scroll Lock + Scroll Lock + 3 + Enter on the keyboard (room I321). The username and password are written on the workbench desk.

2. **Ensure the serial connectivity between computers**

To check the connectivity, open GTKTerm on both computers of the workbench and write some characters. The text should appear on the other computer. If this test fails, please check if the selected serial port is the correct one and that the baudrate is the same on both computers. Please report if the problem persists.

Note: the GTKTerm is only used to check the connectivity between the PCs. It must be closed while the data link protocol is running.

3. **Use the serial port in non-canonical mode using the example code**

Please use the example code provided:

- writenoncanonical.c as the transmitter
- readnoncanonical.c as the receiver

You should compile the program using gcc and run the program on the terminal. Note that you should specify the serial port as an argument (/dev/ttySx, being x the index of the serial port and the default value 0; check the monitor for cases when a different number is used). When running the example code, an array of “a” is sent through the serial port and printed on the receiver terminal.

4. **Implement the logic connection establishment**

Change the example code to send an array of bytes, defined in hexadecimal (declared as unsigned char), that performs the logic connection establishment. This phase considers the exchange of supervision frames that are composed by 5 bytes, starting with a flag, followed by an address field, a control field, a BCC and ending with a flag.

F	A	C	BCC1	F

Each field, defined as unsigned char, has the following values:

FLAG: 01011010 (0x5c)

A (Address):

00000011 (0x03) – frames sent by the Sender or answers from the Receiver

00000001 (0x01) – frames sent by the Receiver or answers from the Sender

C (Control):

SET: 00000111 (0x08)

UA: 00000110 (0x06)

BCC1 (Block Check Character): XOR of all characters of the header (in this case A and C)

The establishment phase is completed once the transmitter sends a SET frame and the receiver answers with an UA frame after checking that the received frame was correct.

```
Emissor        Receptor
  |                |
  |  SET           |     SET=[FLAG,A,C,BCC,FLAG]
  +--------------->|
  |                |
  |            UA  |
  |<---------------|      UA=[FLAG,A,C,BCC,FLAG]
  |                |
```

For debugging purposes, you can print the hexadecimal values using the printf() C function. Check the C reference manual for the details of how to do this: https://en.cppreference.com/w/c/io/fprintf

Please report to the instructor once this step is completed.