PTP vs NTP Client-Side Clock Synchronization with Servo-Clock

Students

- João Martins – up202107275
- Lourenço Carvalho – up202107707

Project Objectives

The goal of this project is to understand and measure how well we can synchronize a completely user-level virtual clock (servo-clock) when using real PTP and NTP servers running in a LAN.

Our Approach

We will set up a real PTP Grandmaster and NTP server on one Linux machine/VM using *linuxptp* and *chrony*. On a second machine we will implement, in Python, a single user-space client that maintains its own virtual clock purposefully initialized with a configurable initial offset (e.g. +10 s) and a configurable drift (e.g. +200 ppm).

This virtual clock will be corrected by using a PI servo controller (as studied in the theoretical classes). The controller's inputs are the raw offsets periodically obtained from the real servers (via *pmc* for PTP and *chronyc* tracking for NTP).

We will vary the servo parameters (*Kp, Ki*), initial offset, artificial drift and synchronization interval and record the correction behavior for both protocols. Finally, we will compare the achievable accuracy and stability of PTP and NTP protocols when both are forced to use the exact same user-level servo-clock parameters.