

15 décembre 2022

Projet PRS

Groupe Legropython
Antoine Piron et Nicolas Guy



Sommaire

**01. Rappel du
projet**

**02. Choix
techniques**

**03. Implémentations
& Difficultés**

**04. Conclusion &
Améliorations**



01.

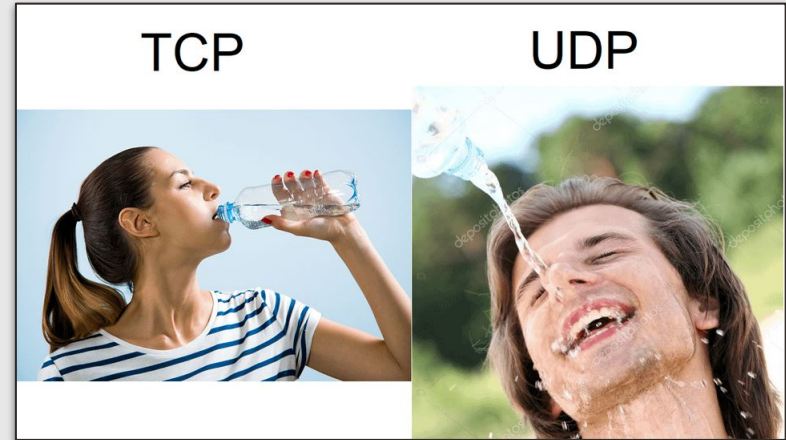
Rappel du Projet

Rebâtir TCP sur une base d'UDP

Rebatir un protocole

Objectif :

- Remettre en place des **mécanismes fondamentaux** de **TCP** sur une base d'UDP.
- Pouvoir répondre à des **scénarios** avec des **clients** dont nous ne **connaissons pas** le code
- Obtenir les **meilleures performances** possibles





02. Choix techniques

Language, protocole ...



Language : python

- Plus compétent (nous)
- Temps

Notre plan initial



Performance

- Slow start
- Fast retransmit



Basique

- Threeway handshake
- Ping pong



Améliorations

- Fast recovery
- Congestion avoidance

03.

Implémentations & Difficultés

Enfin qu'avons nous fait ? Quels problèmes avons-nous rencontrés ?

Vue d'ensemble

Threeway handshake



Gère la connection d'un nouveau client

Multithread



- Client
- Envoi / réception

Slow start



Gain très significatif



Fast retransmit

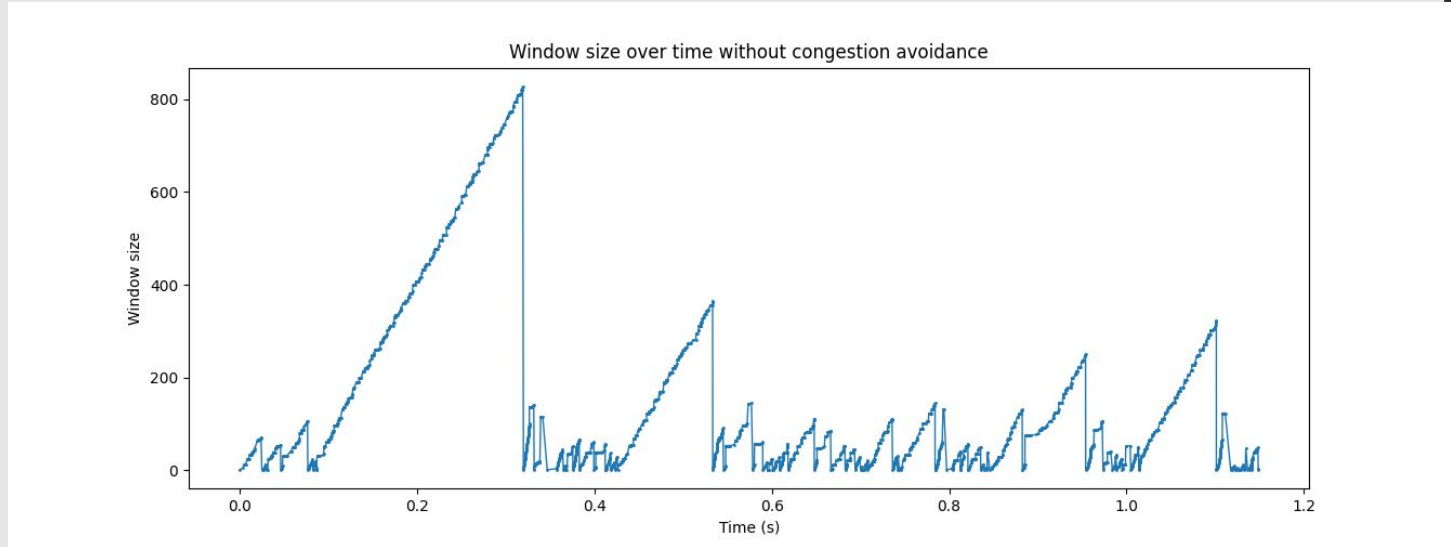
Gain de débit significatif



Congestion avoidance

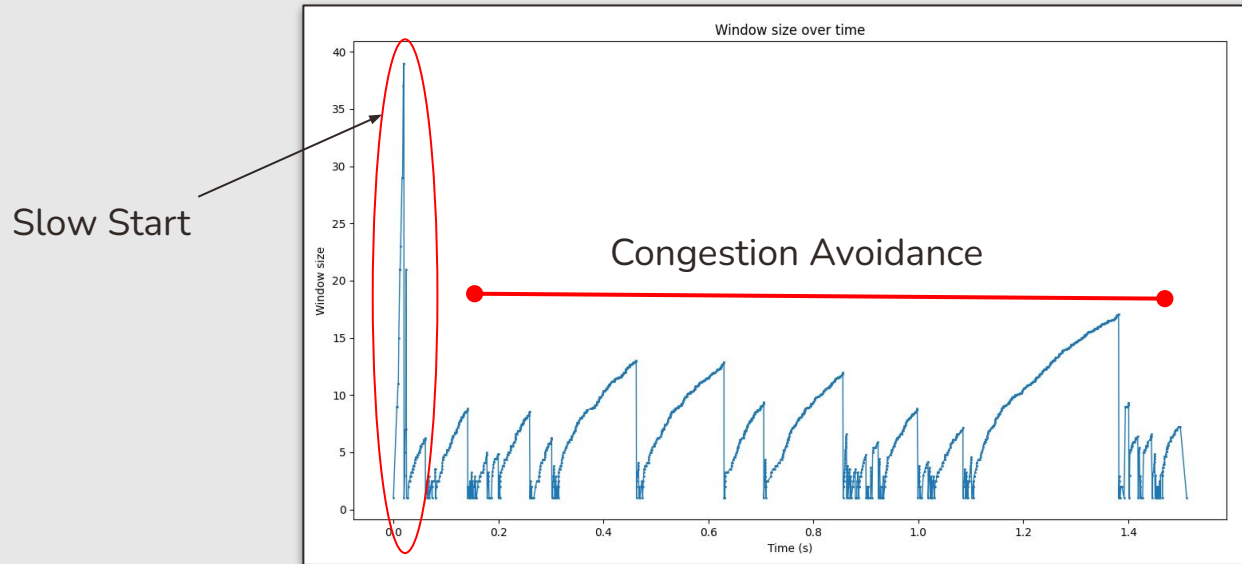
Pas de différence aussi significative

Performances obtenues



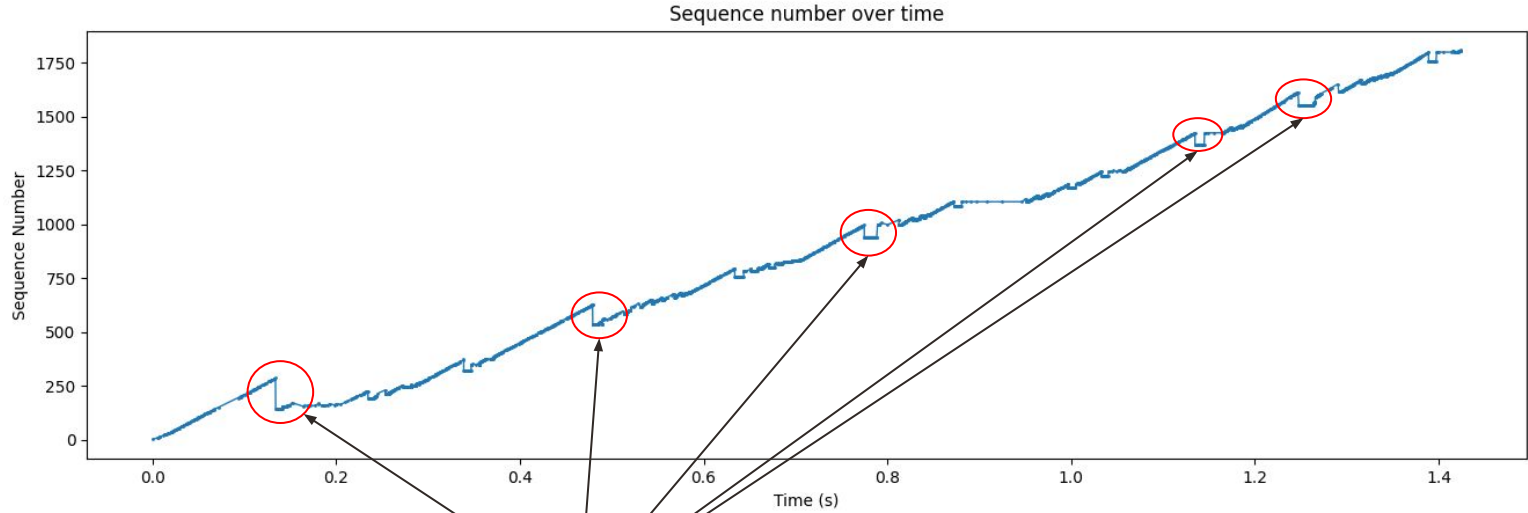
Nos **premier résultats** avant l'implémentation du *Congestion Avoidance*.

Performances obtenues



On observe ici très clairement le **premier départ en slow start** puis le **passage** progressif en **congestion avoidance** avec une évolution plus lente de la window size.

Performances obtenues



Performances obtenues

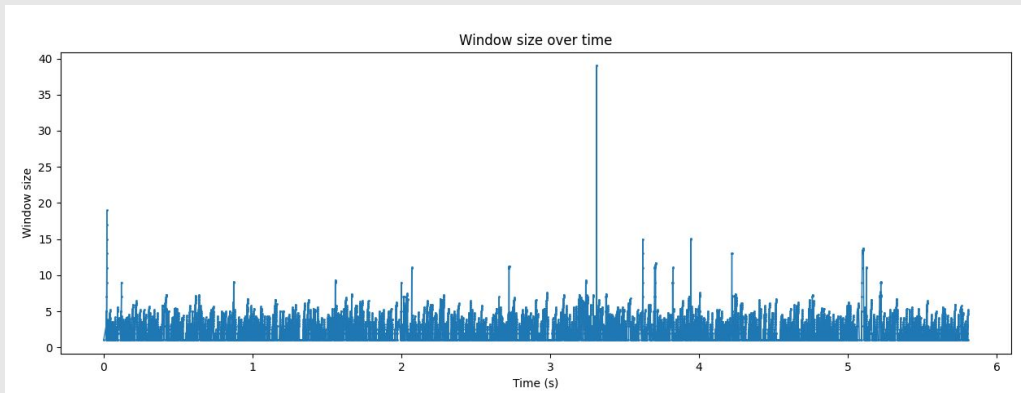
En terme de débit ?

- 10 envoies / **client 1** : ~ 1,279 secondes pour 2,7 Mo → débit de 2,11 Mo/s
- 10 envoies / **client 2** : ~ 5,452 secondes pour 2,7 Mo → débit de 0,49 Mo/s
- Pour le **troisième scénario** on obtient des performances semblables au premiers (avec 2 clients 1)



Nos problèmes

- *MVP trop rapidement atteint*
 - Cette base de code **brouillon** est finalement **resté**
 - Cela **bridé** nos **performances** sur le long terme
- *Travaillé **trop longtemps en local***
 - Pas représentatif de la réalité de l'évaluation
- *Trop **concentré** sur un **unique scénario***
 - Optimisation pour le scénario 1 mais lacune sur les deux suivants
- *Beaucoup trop de **complexité** lié aux **threads***
 - Un gros point noir : les **mutex**





04.

Conclusion & Améliorations

Conclusion



Connaissance

En terme de **compréhension** de **TCP** ce projet fut très intéressant



Performance ?

Pour un **réel gain** de performance il faudrait reprendre le projet **from scratch** avec tous les connaissances que nous avons accumulés

Ressources

Photos

- Logo Python : <https://logos-download.com/9988-python-logo-download.html>
- Memes TCP/UDP publié par u/DerpyChap sur Reddit
https://www.reddit.com/r/ProgrammerHumor/comments/9gcwqw/tcp_vs_udp/
- Photos génériques : <https://unsplash.com/fr>
- Tous les graphiques ont été produit grâce à Matplot sur python

Template

- <https://slidesgo.com/>