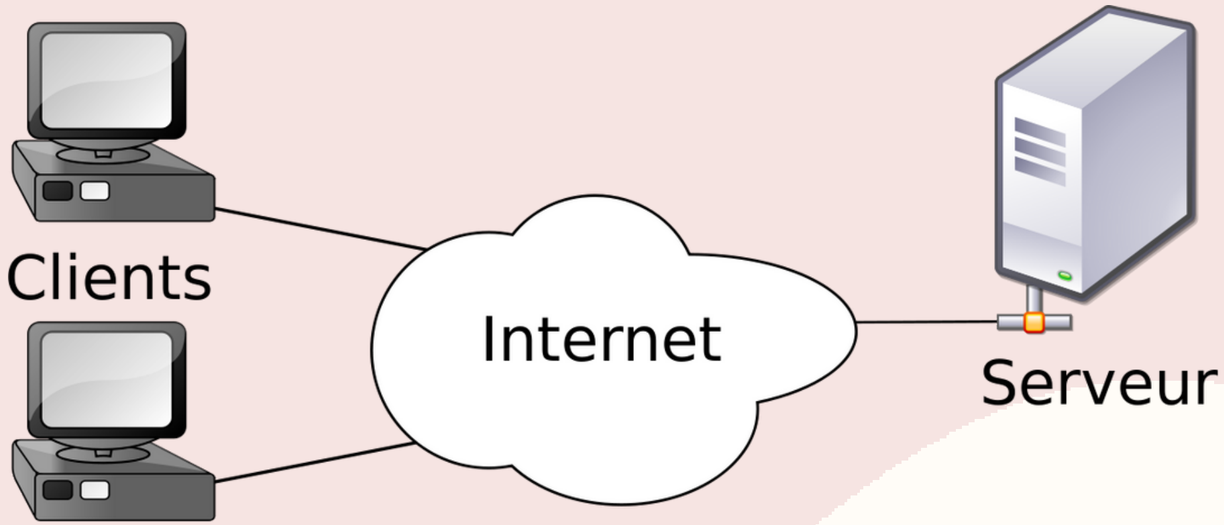
The Hunters: The best one for your Clients

PRS: Abdellah Kabbaj
Ghassen Jerbi



Sommaire

1 Présentation du projet

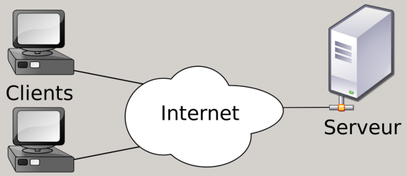
01

Scénario 1

02

Scénario 2

03



Scénario 1++

2 Notre plan de projet

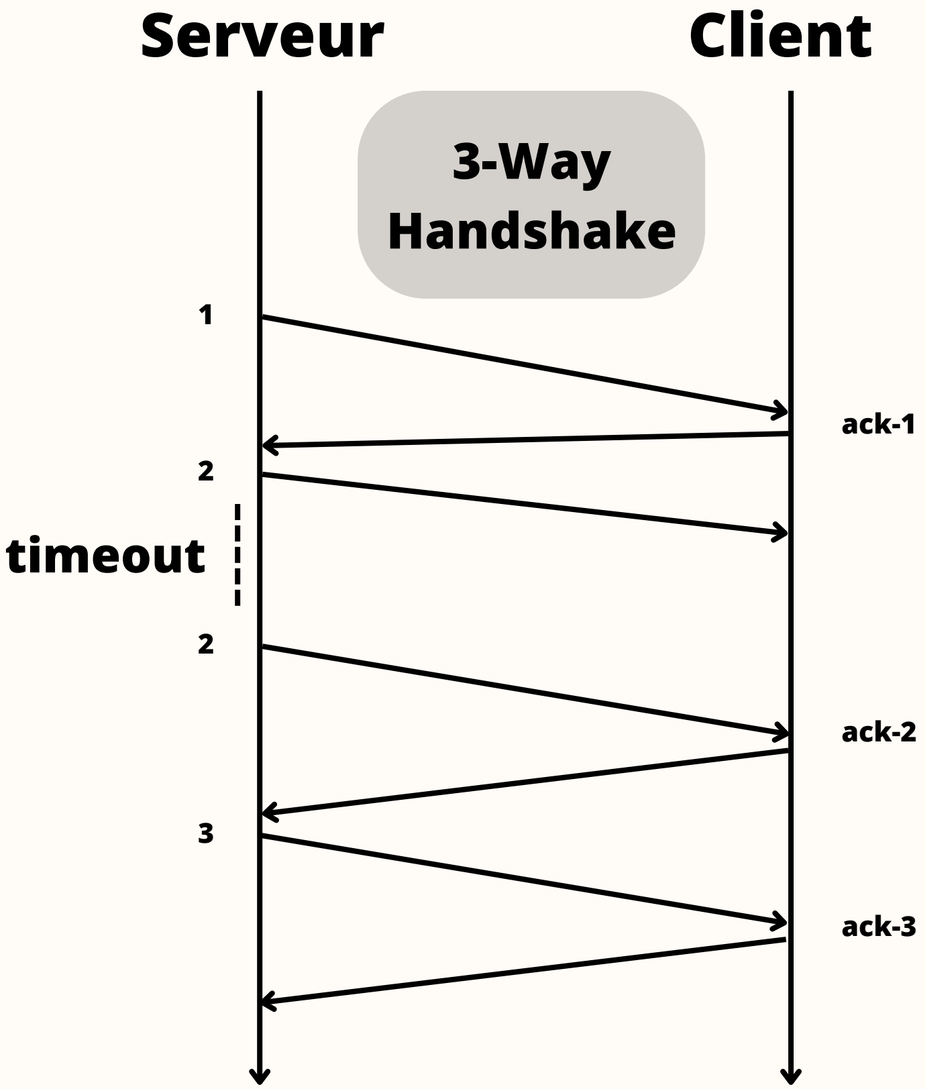
Ce qu'on prévoyait

- **Fenetre glissante**
- **Fast recovery**
- **Fast Retransmit**
- **Thread receive Ack**

Ce qu'on a pu implémenter en plus

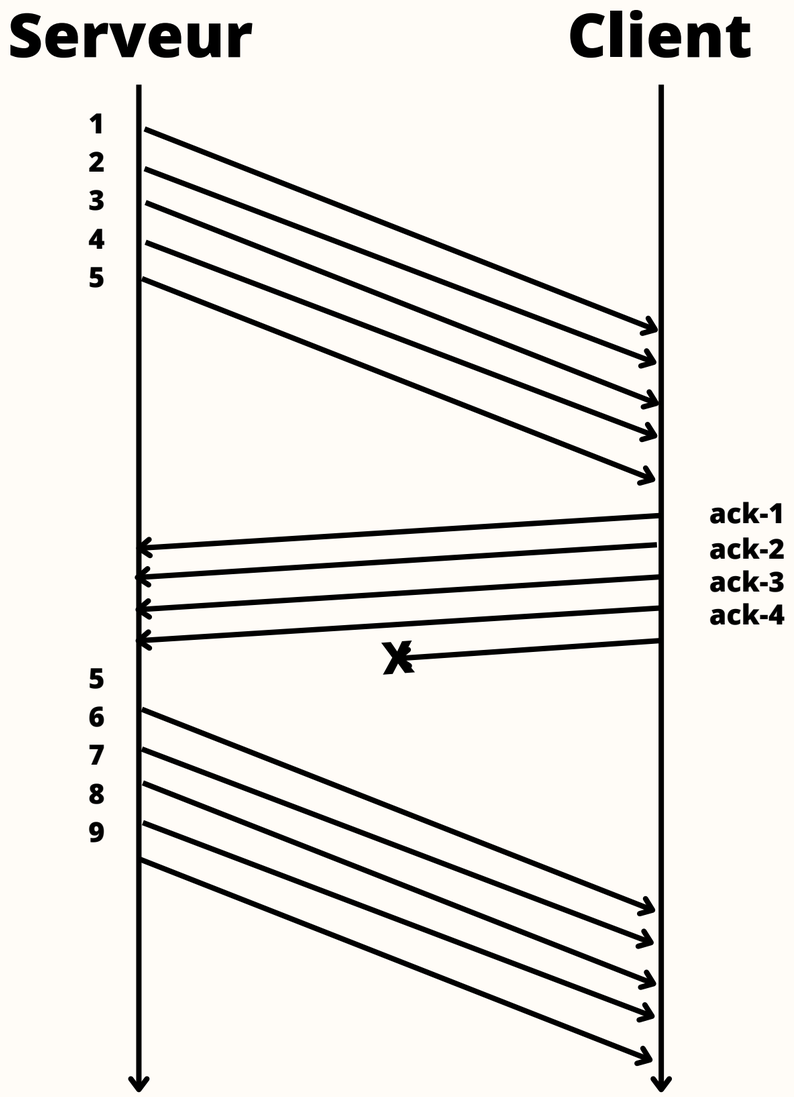
- **RTT/RTO**
- **SlowStart**
- **Congestion avoidance**

3 Première approche



Dans un premier temps nous envoyons juste un paquet en attendant son ACK

3 Première approche



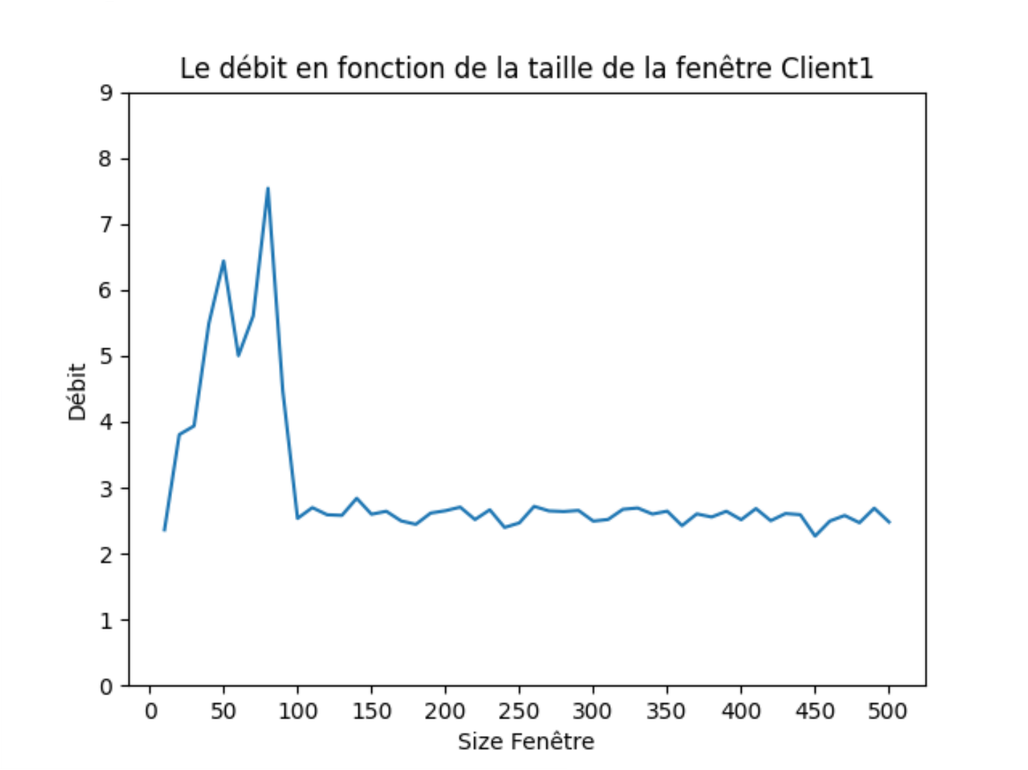
Client 1:

- Supporte une grande charge de message
- Drop de message aléatoire de 3%
- Privilégier donc un Fast Retransmit et Un Fast Recovery Rapide

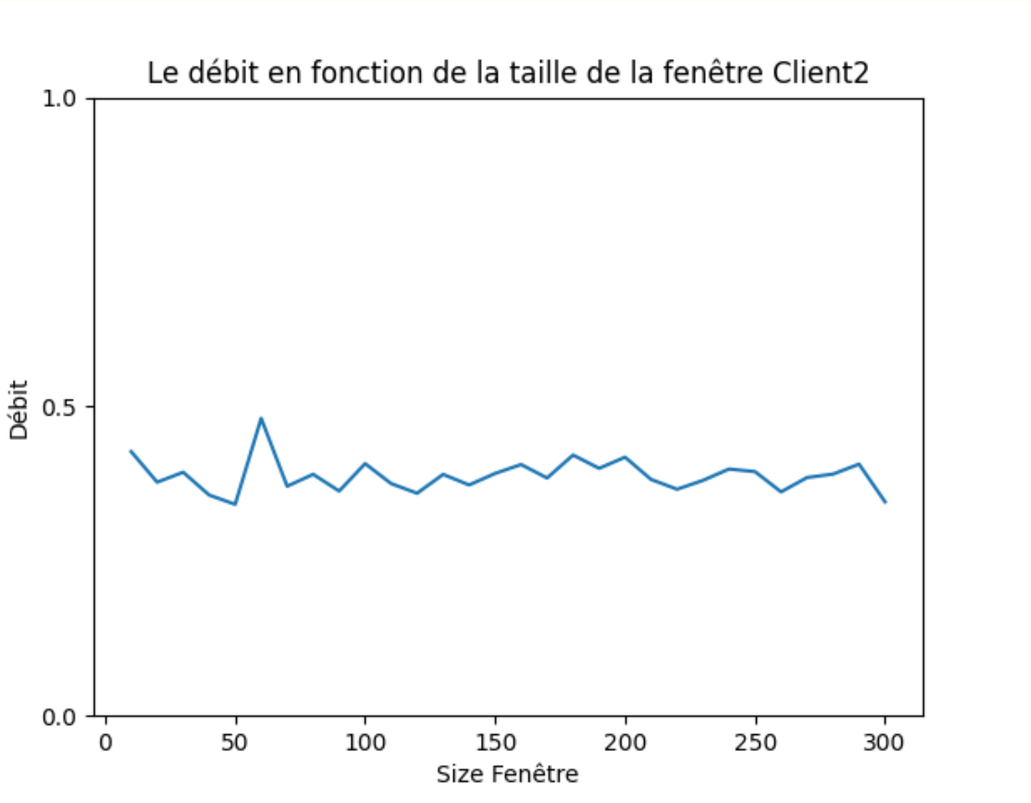
Client 2:

- Drop de message assez régulier
- Privilégier une taille de fenêtre plus petite

Évolution du débit des clients



- Taille de Fenêtre Fixe



4 Approche Finale

Utiliser C++

Nous avons décidé de passer notre projet en C++ pour pouvoir utiliser la structure <vector> pour avoir une meilleure fenêtre

Slow Start

Au vu des performances du client2 nous avons décidé d'implémenter slow start

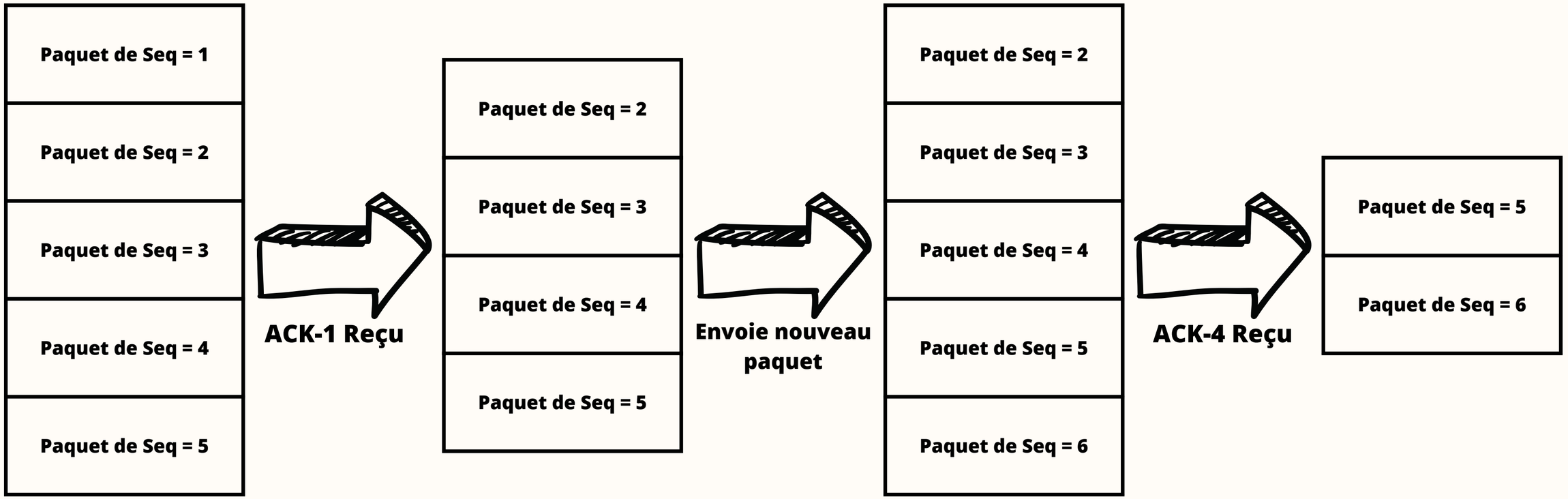
Congestion Avoidance

Au vu des performances du client2 nous avons décidé d'implémenter Congestion Avoidance

RTT/RTO

Pour essayer d'avoir un meilleur débit et faire en sorte que notre serveur fonctionne avec de différents clients

4 Approche Finale : Pourquoi C++



4 Approche Finale

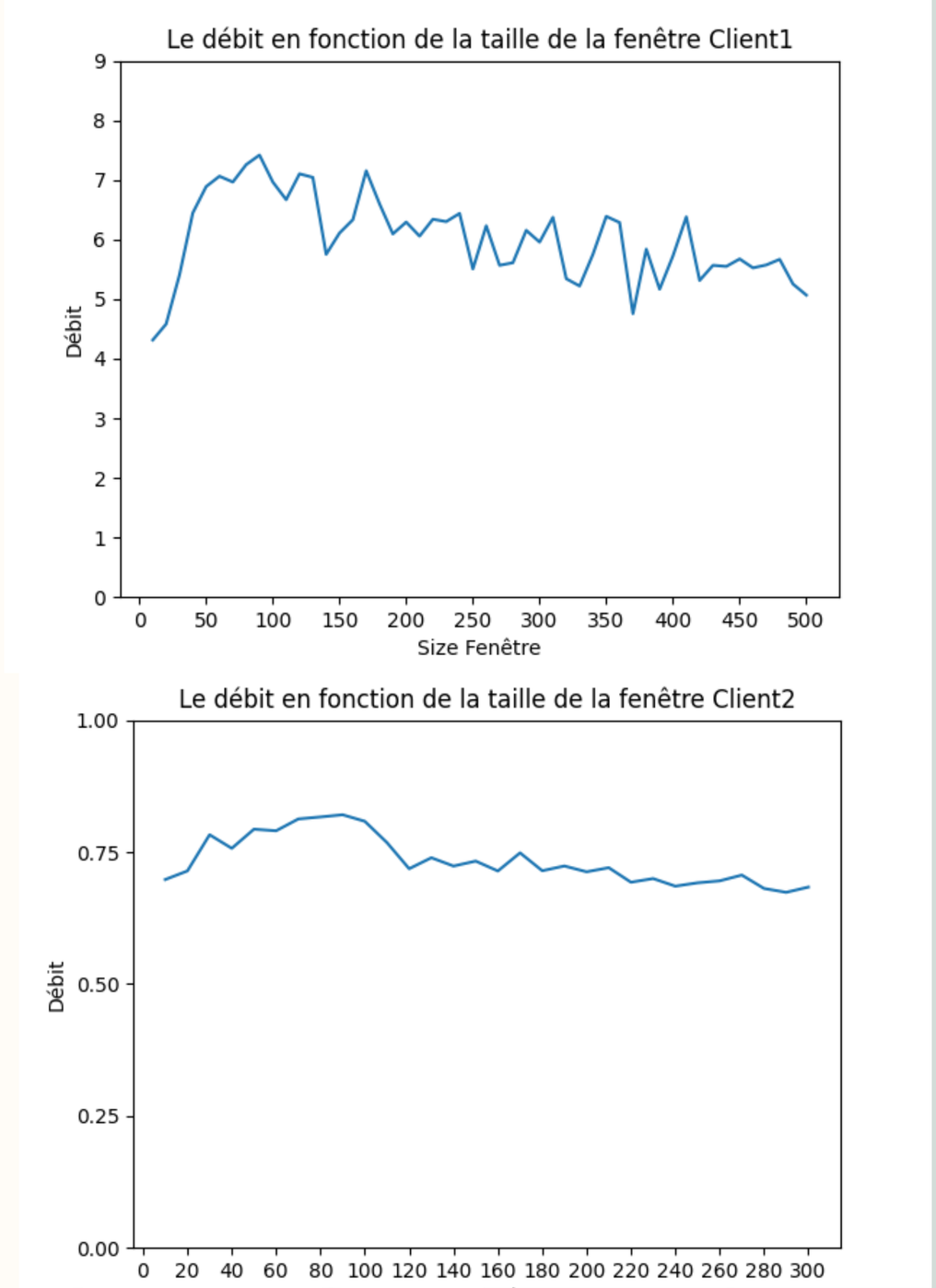
Débits:

Client 1:

- 7.32 Mo/s

Client 2:

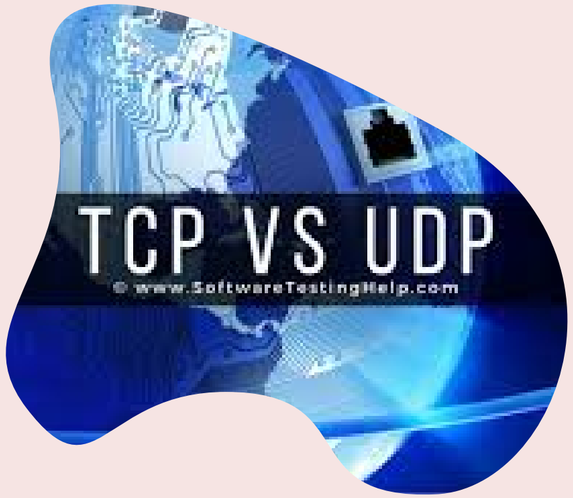
- 0.86 Mo/s



5 Conclusion

Quelques Points importants

- Thread et pas select
- Vecteurs VS Semaphores



Axes d'améliorations

- De différentes structures de données.
- De nouveaux protocoles/algorithmes
- 1 Serveur qui s'adapte à l'ensemble des clients