

Programmation Réseau et Systèmes

Hugo Courté - Clément Lagneau-Donzelle

Plan

Technologies utilisées

Partie statique

- Mise en place d'une communication
- Mise en place d'un timeout pour une transmission paquet par paquet
- Mise en place d'une fenêtre glissante

Partie dynamique

- Mise en place d'un RTT dynamique
- Recherche de paramètres optimaux

Conclusion

Technologies utilisées



Python pour le code des serveurs



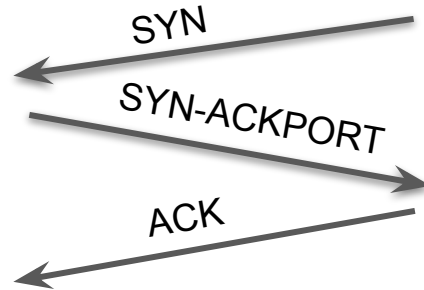
Matplotlib pour l'évaluation de performance

Mise en place d'une communication

Port de contrôle

Server

Client



Port de donnée

Server

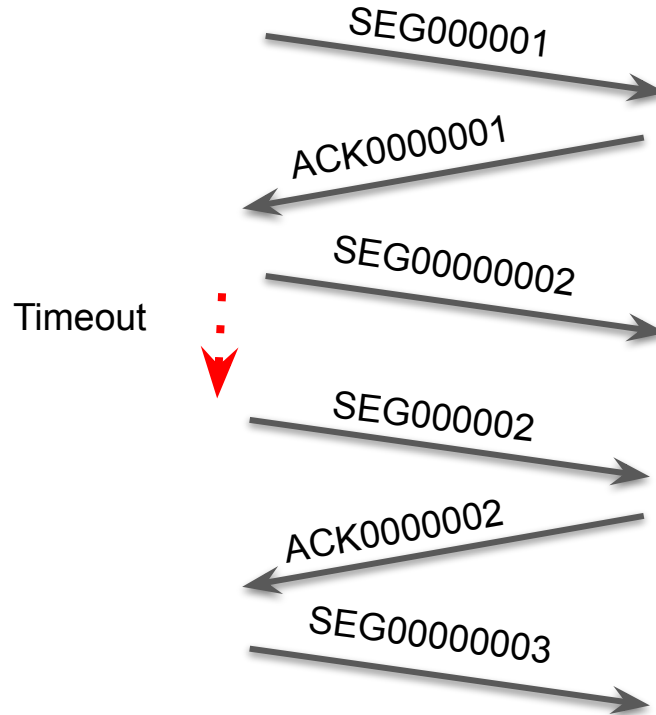
Client



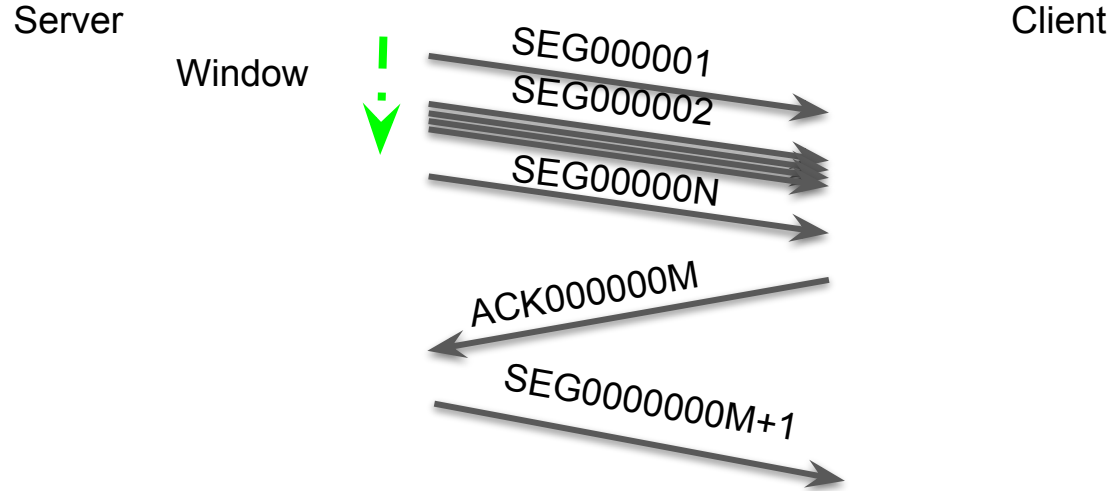
Mise en place d'un timeout

Server

Client



Mise en place d'une fenêtre



Adaptation du timeout en fonction du RTT

Au départ : timeout fixe déterminé par l'utilisateur

Mise en place du calcul de RTT

Recherche des paramètres optimaux

Recherche de la taille de fenêtre optimal et la taille de l'augmentation

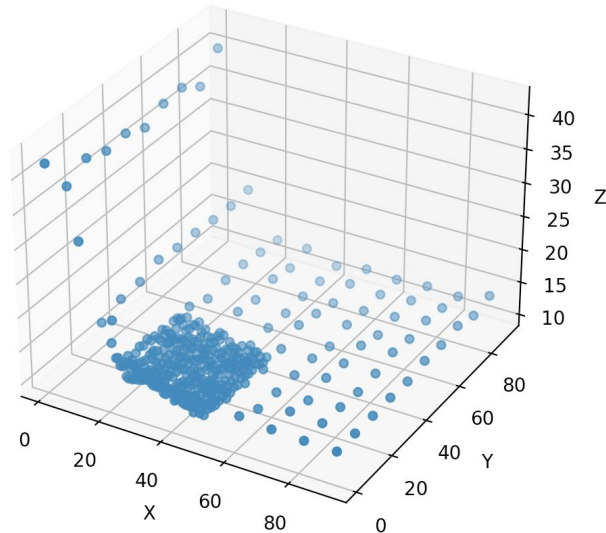


Figure 1.1 - Évaluation pour le scénario 1

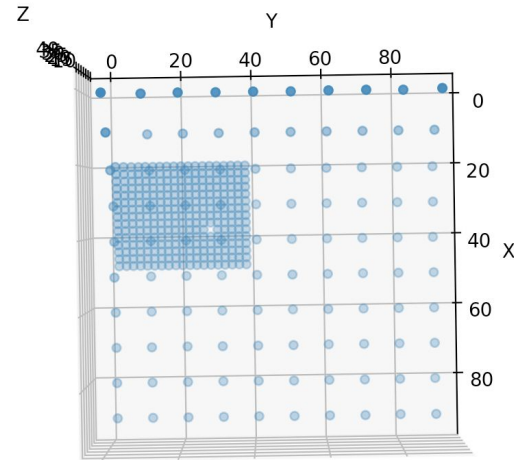


Figure 1.2 - Évaluation pour le scénario 1

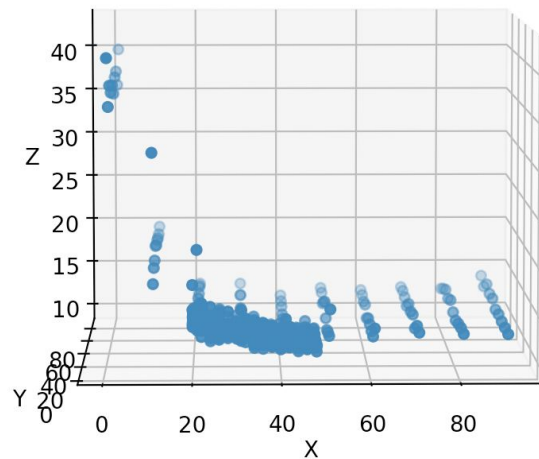


Figure 1.3 - Évaluation pour le scénario 1

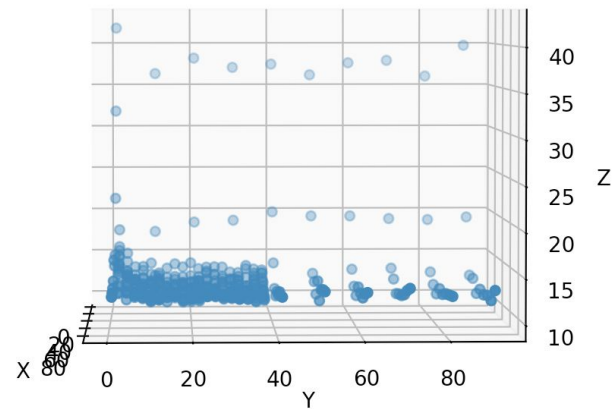


Figure 1.4 - Évaluation pour le scénario 1

Recherche des paramètres optimaux

Recherche de la taille de fenêtre optimal et la taille de l'augmentation

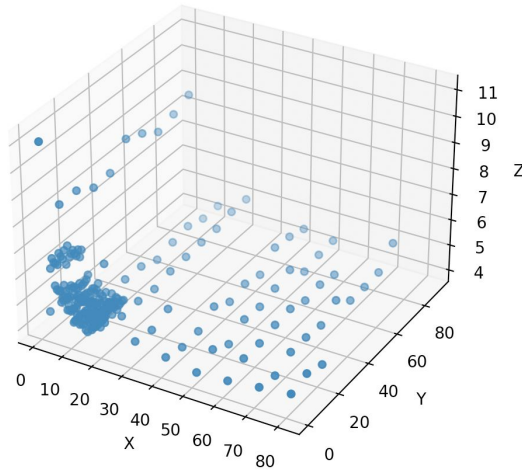


Figure 2.1 - Évaluation pour le scénario 2

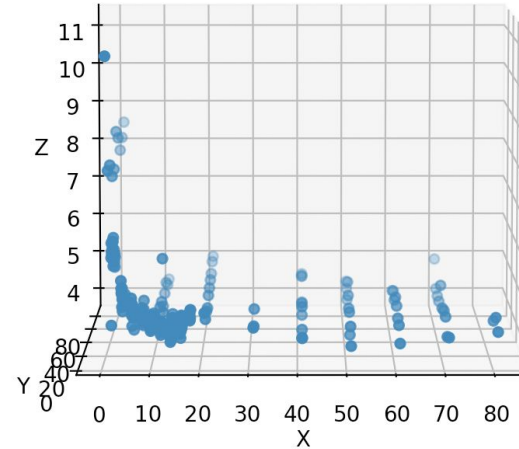


Figure 2.2 - Évaluation pour le scénario 2

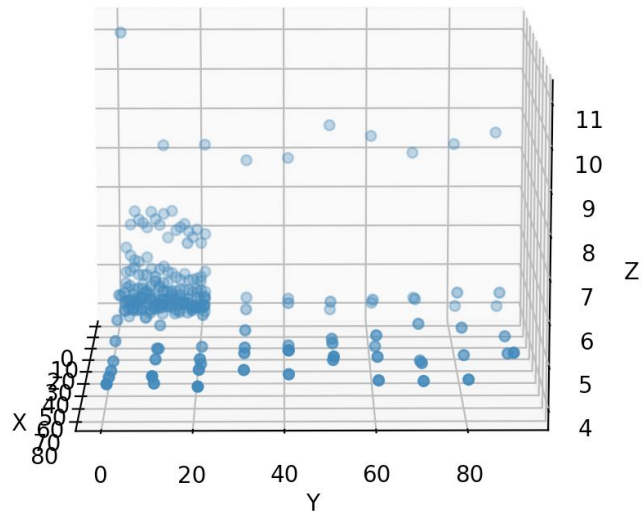


Figure 2.3 - Évaluation pour le scénario 2

Problèmes rencontrés

- Le langage C (au début)
- Problèmes d'acquittement de FIN

Résultats obtenus

Serveurs testés entre deux machines de l'INSA dans le /tmp avec un fichier de 1.2Go

Scénario 1 : 5.7Mo/s

Scénario 2 : 1.6Mo/s

Scénario 3 : 5.6Mo/s

Améliorations possible

Faire des tests sur des tailles de fichier différents

Faire des tests moyennés

Ajouter des paramètres aux batterie de test