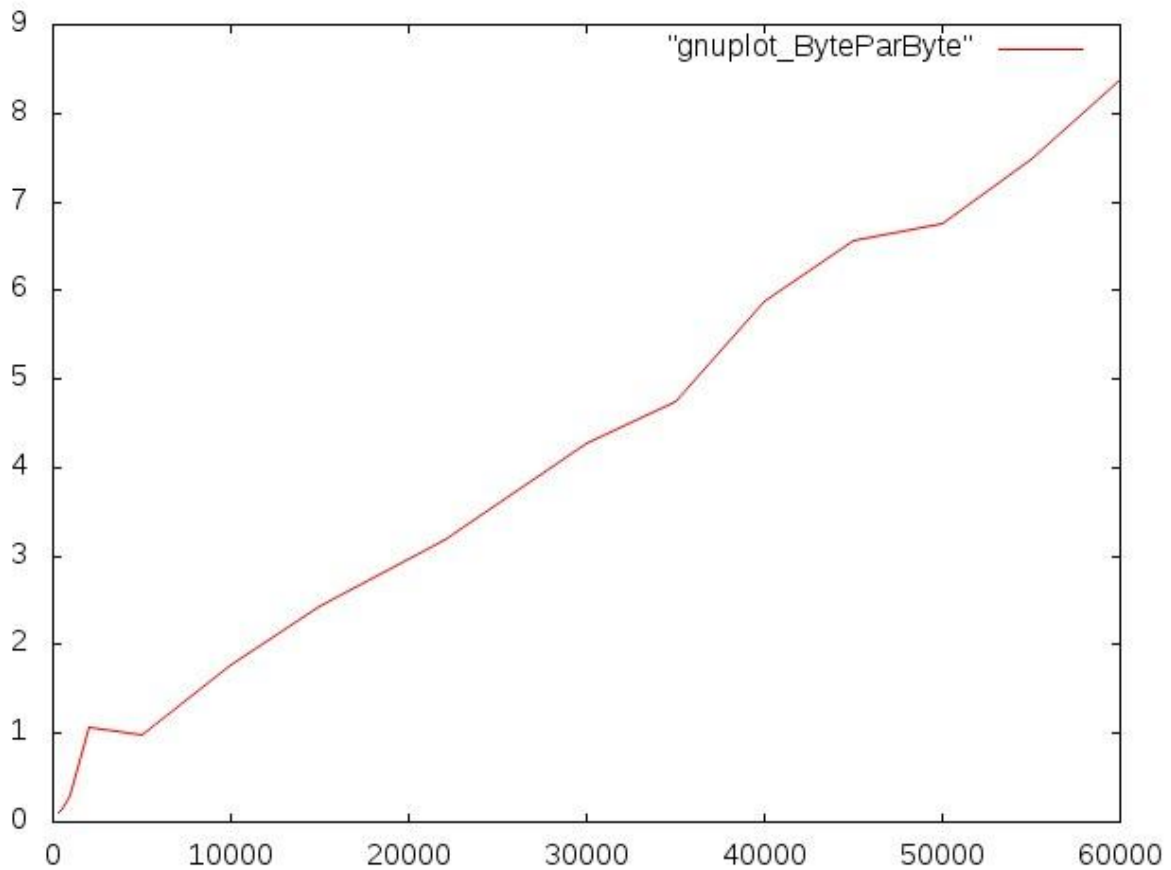


1ers test :

Envoi d'un certain nombre d'octet au serveur grâce à la commande « /echo », les octets sont envoyé un par un au serveur puis renvoyé au client.



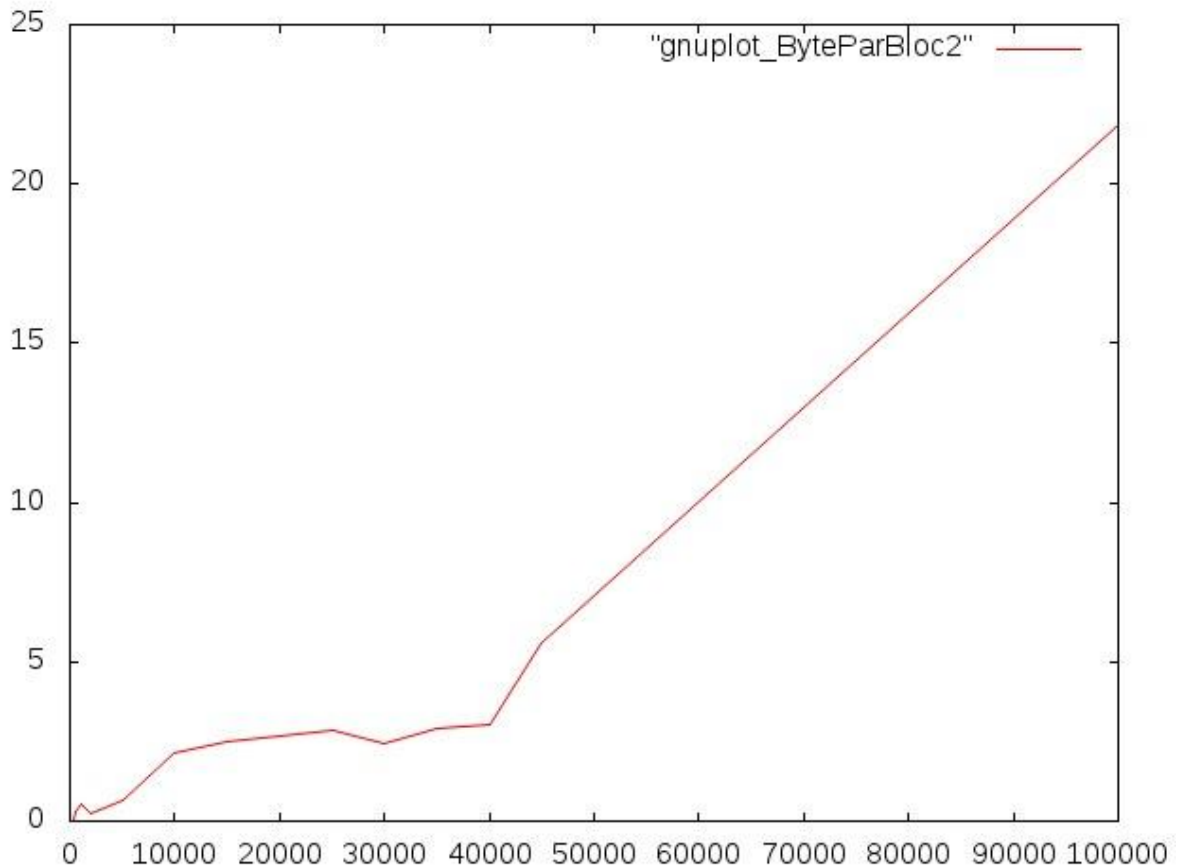
*Courbe de débit avec le nombre d'octets envoyés en abscisse (octet) et le temps de réception (sec) en ordonnée*

Note : Au-dessus de 60 000 octet envoyé le temps est très long

Par la suite on construit j'ai construit une méthode réutilisant la commande « /echo » avec un paramètre supplémentaire : la taille des blocs d'octet, les informations envoyés sont sous la forme d'un tableau de byte.

2<sup>nd</sup> test :

Envoi d'un certain nombre d'octet par bloc de 2 :

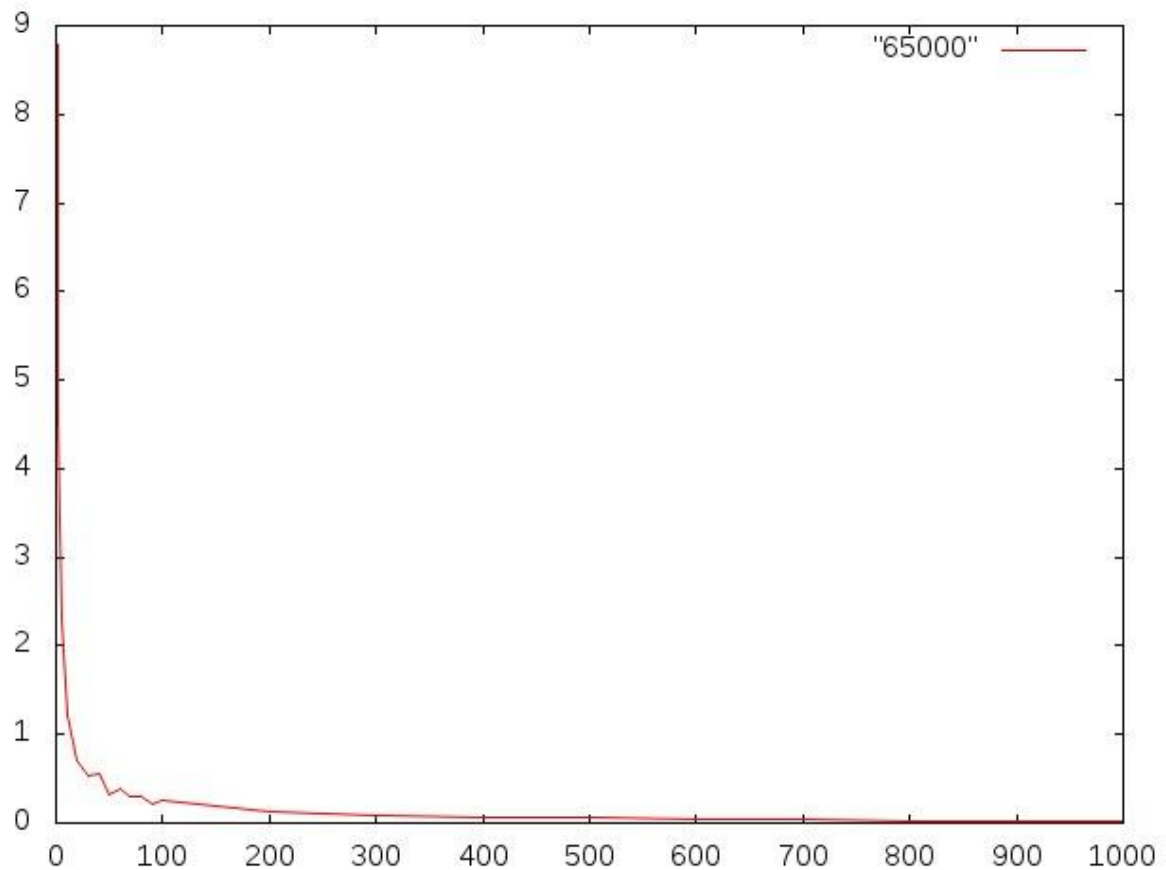


*Courbe de débit avec le nombre d'octets envoyés en abscisse (octet) et le temps de réception (sec) en ordonnée*

Note : Je remarque que cette fois il est possible d'envoyer plus 60000 octets, j'ai donc testé avec des blocs plus grand.

3<sup>e</sup> test :

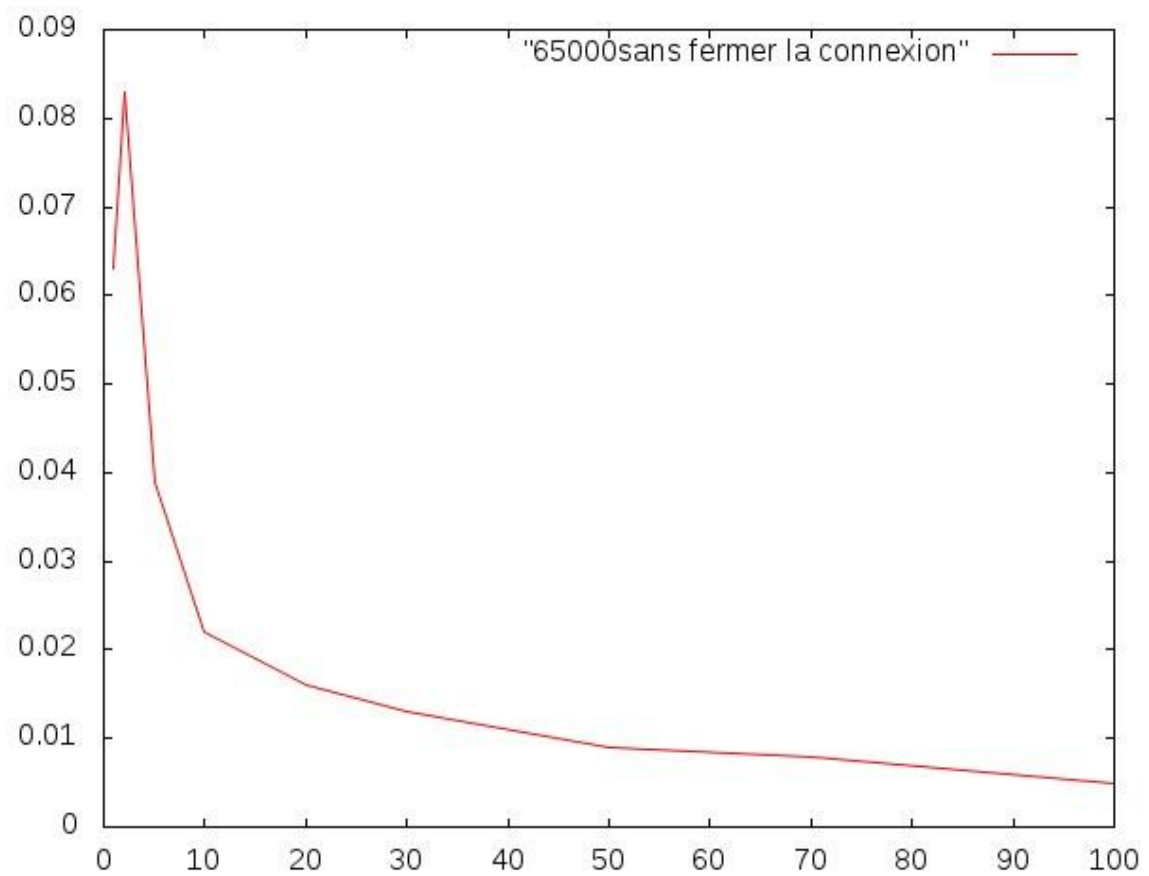
Pour finir j'ai pris un nombre d'octet fixe a envoyé pour je l'ai envoyé grâce à la commande « /echo » avec des blocs de taille de plus en plus grande.



*Temps d'envoi de 65 000 octets avec la taille des blocs pour chaque envoi en abscisse (octet) et le temps en ordonnée (sec)*

On remarque que plus les blocs sont grand plus la réception est rapide.

Pour l'instant chaque test ferme la connexion pour chaque bloc envoyé. J'ai donc effectué le même test en laissant la connexion du client ouverte.



*Taille des blocs (octet) en abscisse et temps de réception (sec) en ordonnée.*

Ressources :

Test 1 :



gnuplot\_ByteParByte

Test 2 :



gnuplot\_ByteParBloc  
2



gnuplot\_ByteParBloc  
100

Test 3 :



65000



65000sans fermer la  
connexion