



LINGI1341 Computer networks : information transfer

Rapport du projet TRTP pour le groupe 119

Aigret, Julien
8343-13-00

Schellekens, François
4257-14-00

31 octobre 2019

Table des matières

| | | |
|----------|-----------------------------------|----------|
| 1 | Implémentation | 2 |
| 1.1 | Champ Timestamp | 2 |
| 1.2 | Réception d'un NACK | 2 |
| 1.3 | Timer de retransmission | 2 |
| 1.4 | Stratégie de connexion | 2 |
| 1.5 | Partie critique | 2 |
| 1.6 | Fermeture de connexion | 2 |
| 2 | Tests et performances | 2 |
| 2.1 | Stratégie de test | 2 |
| 2.2 | Performance | 2 |
| 2.3 | Interopérabilité | 3 |

1 Implémentation

1.1 Champ Timestamp

Nous mettons une représentation en `uint32` de l'heure à laquelle le paquet a été lancé sur le réseau (en millisecondes depuis l'Époque). Il n'est pas utilisé dans l'implémentation actuelle.

1.2 Réception d'un NACK

Pour le moment, nous ne réagissons pas. Il nous a manqué le temps de régler ce paramètre, mais il est couvert par le timer de retransmission.

1.3 Timer de retransmission

Nous avons pris 2 secondes de RTT, et le timer de retransmission de 3 RTT, et nous aurions pu adapter le timer de retransmission en fonction d'un RTT calculé sur les paquets de `ACK` et de `NACK`, mais nous avons manqué de temps.

1.4 Stratégie de connexion

On signale la non-connexion et nous terminons le programme.

1.5 Partie critique

Comme nous avons eu des problèmes pour implémenter l'envoi de plusieurs paquets, nous n'avons pas pu déterminer avec certitude la partie critique de notre implémentations. Néanmoins, il nous semble que ce serait soit l'encodage en paquets, soit l'envoi sur le réseau qui serait le plus critique en terme de temps de travail et d'occupation de puissance de calcul.

1.6 Fermeture de connexion

Quand nous avons envoyé tout ce qui devait être envoyé, nous fermons la connection en appelant `socket.close`.

2 Tests et performances

2.1 Stratégie de test

Pour réaliser les test sur notre Sender nous avons utilisé le Receiver de référence, ainsi que l'outil LinkSim mis à disposition pour simuler différents scénarios du réseau. Nous avons réalisé dans un makefile une série de tests où notre Sender envoie par le biais de Linksim un fichier au Receiver de référence, et à l'aide d'un script bash nous comparons le fichier envoyé et le fichier reçu pour vérifier qu'il n'y a pas eu de corruption.

Nous testons tout d'abord le cas idéal, ensuite nous testons un à un les cas suivants : avec delai, jitter, pertes de paquets, corruption de paquets et paquets tronqués. Enfin nous réalisons un test plus robuste avec tous ces cas rassemblés.

Quand on lance la suite de test avec le fichier par default, tous les tests réussissent. Néanmoins tous les tests que nous avons réalisé avec des fichier plus volumineux que 512 octets ont échoué avec le message d'erreur "ERROR ENCODING, ERROR n° 7". Notre Sender n'arrive donc pas à envoyer plus d'un paquet, et nous n'avons toujours pas réussi à mettre le doigt sur notre erreur.

2.2 Performance

Puisque notre code ne peut pas envoyer plus d'un paquet, l'évaluation des performance n'est pas vraiment applicable.

2.3 Interopérabilité

Nous avons également testé notre Sender avec les Receiver d'autres groupes. Le premier groupe avec lequel nous avons testé notre Sender était le groupe 82 et nous avons obtenus les mêmes résultats qu'au tests, fonctionnel seulement pour un seul paquet envoyé. Le deuxième groupe était le groupe 114. Tout d'abord nous n'avons pas testé de manière classique sur deux machines séparées mais nous avons échangé nos fichiers executables et testé localement. Quand nous tentons d'envoyer un petit fichier qui devrait passer, le receiver n'arrive pas à utiliser les ports prévus 64341 ou 64342 et affiche le message "Problem with bind : Address already in use". Nous avons également tenté avec une adresse différente mais il y a toujours des erreurs au niveau du receiver :

création du client

le client est dans notre liste

free() : double free detected in tcache 2

Aborted (core dumped)

Et au niveau de notre sender il y a le message d'erreur suivant :

ERROR DECODING, ERROR N° 4

ERROR READING SOCKET

Nous concluons de ce dernier test que le receiver de ce groupe n'est pas parfaitement fonctionnel, et ne peut rien nous dire sur notre sender. Les tests d'interopérabilité ont au final confirmé le résultat de nos tests.