ÉCOLE POLYTECHNIQUE DE LOUVAIN

École Polytechnique de Louvain

Rapport TRTP

LINGI 1341 : Truncated Reliable Transport Protocol

Auteurs :
Louis De Vogeleer : 7150 1600
Baptiste Demuysère : 2395 1600

Personnes ressources :
Quentin De Coninck
Olivier Bonaventure

22 octobre 2018

Table des matières

Section 1

Architecture

L'architecture de nos 2 programmes, sender et receiver, sont tous deux composés de 3 fichiers.

- **socket**
  Socket contient les différents appels systèmes nécessaires pour pouvoir envoyer et recevoir des données entre deux ordinateurs distants.
- **packet_implement**
  Packet implement contient une interface permettant de lire et d'écrire les segments de données dans le format utilisé par le protocole.
- **sender**
  Sender devra permettre de réaliser un transfert de données unidirectionnel.
- **receiver**
  Receiver devra permettre de réaliser un transfert de données unidirectionnel

1.1 Sender

La fonction Sender envoie ce qu'il lit sur l'entrée standard. En le divisant en paquet de 528 octets maximum et envoie tout ces packets à l'adresse de destination par le biais d'un socket.

1.2 Receiver

La fonction Receiver va lire les différents paquets qu'il va recevoir et les retransmettre sur la sortie standard. En reencodant ce qu'il va recevoir en tableau de caractere comme initialement envoyé.

Section 2

Réponse aux question posées

2.1 Timestamp

Dans Timestamp, nous utilisons la fonction clock(), pour mettre une valeur de temps afin de calculer le temps qui séparent l'envoie de la réception. Et de vérifier que ce temps est bien inférieure au temps TimeOut(TimeOut est le temps au bout du quel, si aucune réception n'est faite, on décide de renvoyer un paquet)

2.2 Réception paquet de type NACK

Lorsque l'on reçoit un paquet de type NACK, si le numéro de séquence du paquet correspondant est valide dans la fenêtre d'envoie, dans ce cas la, on va renvoyer ce paquet non tronqué.
Si le numéro de séquence n'est pas dans la fenêtre d'envoi, on l'ignore.

2.3 Valeur du TimeOut

Pour le choix de la valeur du TimeOut, nous l'avons calculer de la sorte. Comme le temps maximale qu'il passe sur le réseaux pour l'envoie est de 2 secondes.
A l'envoie et au retour cela fait déjà 2+2=4 secondes auxquels on rajoute du délai de marge 1 seconde. C'est pourquoi nous avons choisi 5 secondes pour la valeur du TimeOut.

2.4 Partie critique

La partie critique de notre implémentation qui affecte la vitesse est le choix du TimeOut en effet, un timeOut trop haut diminuerai de beaucoup la vitesse de transfert.

2.5 Performance

2.6 Tests

Section 3

Piste d'amélioration

Tout d'abord nous devrions penser à améliorer notre code. En implémentant correctement le Selectieve Repeat ce qui améliorerait grandement les perte de données du à un mauvais réseaux.

Conclusion

Pour conclure, nous sommes maintenant à même de faire l'implémentation d'un protocole qui devra permettre de réaliser un transfert de données unidirectionnel sur un réseaux parfaits. Bien que notre implementation ne soit pas très robuste. Il nous a permis de nous initier à l'implémentation de transfert de données sur réseaux.