

LINGI1341 : Réseaux informatiques

Project 1 : Protocole de transfert fiable

Damiano-Joseph Siciliano

Noma : 3039-10-00

Benjamin Daubry

Noma :

1. Sender

Les variables globales clé qui communiqueront entre les thread sont les suivantes :

pkt_t * window[32] : Il s'agit du buffer dans lequel seront ajoutés/supprimés des paquets au fur et à mesure

struct timeval timeBuf [32] : il contient le temps exact

int nProd : le nombre de paquet produit mais qui n'ont pas été encore envoyé

int nSend : le nombre de paquet qui POURRAIT être encore envoyé (initialisé donc à 16)

int newPKT : représente le numéro de paquet envoyé le plus récemment

int oldPKT : représente le numéro de paquet envoyé le plus anciennement

Il y a 4 threads permettant d'obtenir un protocole type selective repeat :

a) Producer

Il va simplement rajouter des paquets dans l'ordre (numéro de séquence) en revenant au début du buffer si nécessaire. Il ne s'arrête que lorsque le contenu du buffer sur lequel il compte rajouter des paquets est déjà rempli ou simplement lorsque le fichier a été lu entièrement. A chaque rajout de paquet il incrémente nProd de 1.

b) Sender

Tout comme le producer, il va se déplacer en boucle dans le buffer à partir du début et enverra des paquets lorsqu'il saura que le producer a mis des paquets (nProd > 0) et que la limite de paquets à envoyé n'est pas atteinte (nSend > 16)

A chaque envoie, il décrémente nProd et nSend de 1 et incrémente newPKT de 1 et associe un timer à cette envoie dans timeBuf.

c) Listen

Il s'agit du thread recevant les ACK.

Si il recoit un NACK, il va simplement renvoyé le paquet et remettre un nouveau timer

Si il recoit un ACK, on verifie si il est dans la fenetre d'envoi. Si tel est le cas, on sait qu'il n'y a plus besoin du timer qui va donc s'arreter de fonctionner (via un mutex), permettant ainsi de supprimer ce paquet et son timer associé en étant sur qu'entre temps il n'y a pas une demande de renvoi (simple perte de performance dans le cas échéant).

A noter que si l'ACK est destiné à celui le plus à gauche de la fenêtre, nous allons la decaller de 1 + un bonus selon le nombre d'ACK recuit à la suite de ce dernier. Ce chiffre sera incrémenté dans oldPKT et nSend (en effet on peut envoyer d'autre paquet).

d) CheckTime

Il va constamment regarder le timer associé au plus ancien paquet en le comparant au temps actuel, si il depasse les 3 secondes. On entre dans la section critique vu précédemment pour renvoyer le paquet et mettre à jour le timer.

Problème actuel : il connaît le plus ancien paquet par rapport au positionnement de la fenetre, il recoit un NACK (14), puis un NACK(12), il se peut qu'il regarde le 12 en premier.

Solution : afin de prioriser les timers les plus anciens, nous allons creer une liste FIFO accompagner d'un nouveau MUTEX qui bloquera le check si l'on rajoute/enleve/deplace les elements de cette liste.

Exemple d'un envoie parfait sans NACK : 1->2->3 , 2->3->4

Exemple d'un envoie avec NACK : 1->2->3 , 1->3->2

2. Receiver

3. Test interopabilité