Implémentation d'un protocole de transfert sans pertes

Baurain Aymeric

22 Octobre 2017

1 Introduction

Ce rapport décrira l'ensemble du projet demandé dans le cadre du cours LINGI1341 Computer networks : information transfer. Il nous a été demandé d'implémenter dans le langague C un protocole de transport via segment UDP. La connection devra néanmoins être fiable avec un selective repeat mise en place du coté du récepteur.

le protocole devra prendre en compte la troncation de payload et devra pouvoir fonctionner avec IPV6 et IPV4. Il n'y a d'ailleurs aucune garantie que les paquets arrivent au récepteur dans le bon ordre et sans erreur.

2 Architecture du réseau

2.1 Gestion des erreurs

Il est tout d'abord important de préciser que le protocole de transport utilisé est UDP. Celui-ci étant un protocole non fiable et non orienté connection, il nous faudra alors implémenter des solutions permettant de géré la perte/corruption de paquet. De ce fait, deux checksums seront mise un place afin de géré la corruption de paquet :

Le premier est calculé via l'entête UDP et placé juste après celui-ci.

Le deuxième sera calculé juste après l'envoi du paquet sur le réseau, en fonction de l'éventuel payload et placé après celui-ci.

Lors de la réception du paquet le champ TR doit être mise à 0, le serveur devra recalculé les CRC du paquets et les comparés avec le champ TR. Si les deux valeurs diffèrent le paquet sera ignoré et un paquet de type NACK sera renvoyé à l'émetteur du paquet. Pour ce qui est de la perte de paquet, deux implémentations seront mise en place :

1. Le retransmission timer qui sera le temps accordé à l'émetteur du packet avant de le considérer comme perdu et donc le renvoyer. Il ne devra pas être

plus grand que le round-trip-time représentant le délai entre l'émission de la frame et la réception du ACK correspondant.

2. Le numéro de séquence indiquant le "numéro" de la frame correspondante de manière à permettre au receveur d'ordonnancer correctement les frames et de pouvoir annoncer à l'émetteur lors d'un ACK si une frame a été reçu hors séquence afin qu'il renvoie la frame manquante.

2.2 Gestion des paquets

Du coté de la réception, les paquets devront être géré via un selective repeat c'est à dire que chaque paquet reçu hors séquence devra être stocké dans un buffer jusqu'a réception du (ou des) paquet(s) manquant(s). Lors de la réception d'un paquet hors séquence un ACK devra renvoyé le numéro de séquence de la dernière frame dans l'ordre ainsi que le numéro de séquence de la frame stockée en mémoire. Il sera toutefois nécessaire de pouvoir placé les paquets ainsi que les erreurs reçus dans un fichier. Avant cela un décodage devra être mise en place pour décortiquer l'header, vérifier les CRC et placé le payload (si il existe) en mémoire.

Pour ce qui est de l'émission, une phase d'encodage permettra de placer dans le header les valeurs correspondante au bon champs, que ce soit le numéro de séquence correspondant à la frame, la taille de la fenêtre d'émission des deux entités, la longueur des paquets variant selon le payload et aussi le type de paquet. Toute ces informations seront écrite en fonction de l'état du réseau et de la machine émettrice.

3 réponses aux questions

Que mettez-vous dans le champs Timestamp, et quelle utilisation en faites-vous ? Comment avez-vous choisi la valeur du retransmission timeout ?

Pour rappel le Timestamp est une option du protocole ip enregistrant (lors du "voyage" d'un paquet sur le réseau) l'heure à laquelle chaque machine du réseau a recu le paquet. Afin de pouvoir récuperer le timestamp côté receveur, l'option SO_TIMESTAMP doit être activé lors de la réception du packet.

Pour ce qui est de l'utilisation du Timestamp, il permettra aux deux entités de pouvoir régler leur retransmission timer. Sachant que le Timestamp prend comme valeur le temps mis pour allez sur chaque noeud du réseau, il suffira de prendre une valeur plus grande pour le retransmission timer que deux fois le temps parcouru entre les deux entités.

Comment réagissez-vous à la réception de paquets PTYPE_NACK ?

Quand on reçoit un paquet de type PTYPE_NACK on peut en déduire que le paquet envoyé (dont le numéro de séquence est dans le NACK) a été corrompu. Il va donc falloir répérer quel est ce paquet perdu pour ensuite pouvoir le renvoyé.

Quelle est la partie critique de votre implémentation, afectant la vitesse de transfert ?

Le problème pourrait venir lors de l'envoi de gros fichier. Ceux-ci ne seront pas optimisé pour la mise en place sur le réseau que ce soit au niveau de l'encodage du fichier (aucun algorithme de compression n'est mise en place) mais aussi l'utilisation de UDP qui n'est pas optimisé pour l'envoi de fichier. En effet, la segmentation du packet sera beaucoup plus grande, ce qui ajoutera du temps de travail considérable à la machine émettrice.

4 remarque

Il est important de préciser que cela ne représente pas encore l'état actuel du projet. Ce manque d'avancement est due à un travail de groupe réduit à un travail individuel ainsi qu'un manque total d'expérience en langague Les tests n'ont d'ailleurs pas encore été mise en place.