

Projet de téléinformatique et protocoles

IUP 2ème année GMI / Maîtrise d'informatique

Le protocole UDP utilisé pour l'échange de données n'est pas très fiable. En effet, les pertes de paquets sont monnaie courante lors de transfert de fichiers de taille importante par exemple. L'objectif du projet est de proposer un programme réalisant un transfert de fichier fiable basé sur UDP en point à point, c'est-à-dire utilisant les sockets en mode datagramme. Dans ce projet on aura un processus émetteur qui envoie un fichier et un autre processus récepteur qui reçoit ce fichier. Les données ne circulent donc que dans un sens. Les erreurs possibles sont : les pertes de trames et les trames erronées (pas de déséquencement).

Pour simuler la liaison point à point, nous fournissons un programme « médium » qui simule le support de transmission non-fiable. Ce médium pourra introduire soit des erreurs sur les trames ou perdre des trames, avec des probabilités diverses.

Le protocole utilisé n'est pas imposé. On pourra choisir un protocole parmi les suivants :

- Protocole 3 : protocole de type Send & Wait. Une fois que l'émetteur a envoyé une trame, il attend de recevoir un acquittement avant d'envoyer la trame suivante. S'il ne reçoit pas d'acquiescement (trame ou acquiescement perdu ou erroné), il retransmet la même trame
- Protocole 5 : protocole de type retransmission continue. L'émetteur a une fenêtre d'anticipation en émission de taille supérieure à 1. L'émetteur pourra alors envoyer toute sa fenêtre sans recevoir d'acquiescement. A chaque réception d'un acquiescement, l'émetteur déplace sa fenêtre d'émission du nombre de trames acquiescées. Lorsqu'un timeout se déclenche, l'émetteur devra renvoyer toutes les trames de sa fenêtre d'émission. Le récepteur attend les trames dans l'ordre. Les trames pourront être numérotées modulo 8.
- Protocole 6 : protocole à fenêtre d'émission et à fenêtre de réception. L'émetteur a une fenêtre d'anticipation en émission de taille supérieure à 1 et le récepteur a une fenêtre d'anticipation en réception de taille supérieure à 1. L'émetteur pourra envoyer toute sa fenêtre sans recevoir d'acquiescement. Le récepteur pourra stocker les trames reçues et demander la retransmission d'une trame spécifique (retransmission sélective). Les trames pourront être numérotées modulo 8.

Le processus médium fourni fonctionne se lance de la façon suivante :

```
médium nom_machine1 port_machine1 nom_machine2 port_machine2  
proba_perte_trame proba_erreur_trame
```

- port_medium représente le numéro de port du médium (éventuellement 0).
- nom_machine1 représente le nom d'une des machines qui communiquent à travers ce médium
- port_machine1 représente le numéro de port de la machine1.
- nom_machine2 représente le nom de l'autre machine.
- port_machine2 représente le numéro de port de l'autre machine.

- `proba_perte_trame` (≥ 0 et ≤ 1) est le taux de perte de trames. Par exemple, pour `proba_perte_trame=0.5`, en moyenne une trame sur deux sera perdue. Par contre pour `proba_perte_trame=0.9`, c'est 90% des trames qui seront perdues.
- `proba_erreur_trame` (≥ 0 et ≤ 1) est le taux des trames erronées. Chaque trame pourra comporter une rafale d'erreurs e 16 bits. Par exemple pour `proba_erreur_trame=0.5`, en moyenne une trame sur deux aura cette rafale d'erreurs.

Avant de lancer le programme médium, il faut lancer les deux processus sur les machines 1 et 2 et connaître leurs adresses IP et leurs numéros de port.

⇒ Pour lancer l'émetteur, il faudra taper : `emetteur port_local fichier_a_transmettre`

⇒ Pour lancer le récepteur, il faudra taper : `recepteur port_local fichier_a_recevoir`

Ensuite, on pourra lancer le programme médium avec les arguments adéquats.

Ensuite, le médium envoie, à intervalles réguliers, une trame de type texte "MEDIUM_ONLINE" qui permettra aux processus de connaître l'adresse IP et le numéro de port du médium. Par la suite, c'est le médium qui se chargera d'envoyer à la machine2 les trames venant de la machine1 et vice-versa. La taille maximale des trames qui peuvent transiter par le médium est de 256 octets. Un exemple de programme Emetteur et Recepteur est fourni avec le code médium

CE QUE VOUS DEVEZ FAIRE

- Définir le protocole qui doit permettre aux machines 1 et 2 de communiquer de façon fiable à travers le support de transmission fourni. Cette spécification sera à faire en langage promela. Il faudra expliquer les choix que vous aurez fait dans le rapport. Le code promela sera également à inclure dans le rapport.
- La partie des processus qui gère la réception et l'émission des trames devra effectuer un contrôle d'erreur en implémentant le code polynomial. On pourra choisir l'un des deux polynômes générateurs suivants :
 - $X^{16} + X^{15} + X^2 + 1$
 - $X^{16} + X^{12} + X^5 + 1$
- Les deux processus devront générer chacun un fichier log qui permettra d'analyser leur comportement effectif :
 - Ecrire les trames reçues et envoyées en clair (de façon explicite pour l'utilisateur)
 - Afficher les timeouts, les ré-émissions etc.
 - Le cas échéant indiquer les états du processus (connexion, en cours de transfert, `attente_de_connexion` etc.)
 - Bonus : statistiques sur les trames perdues et/ou erronées
- A chaque réception d'une trame MEDIUM_ONLINE, remettre à jour l'adresse du médium (on suppose qu'il pourra éventuellement changer en cours de transmission).
- Inclure dans le rapport l'explication de l'implémentation en C du protocole de gestion du médium (format des trames, taille des fenêtres, gestion de la temporisation) et une partie tests qui inclura quelques fichiers log commentés.

Note :

- Si la gestion des trames erronées vous pose des problèmes, essayer en premier lieu de gérer uniquement les pertes de trame (mettre à 0 l'argument `proba_erreur_trame` lors du lancement du médium)
- Une fois le choix du protocole fixé, il faut le respecter. Il faut donc implémenter le même protocole en C que celui que vous aurez vérifié en promela.

Installation du médium :

Sous Linux : `tar -xzvf Medium.tgz`

Sous les Machines solaris : `gzip -cd Medium.tgz | tar xvf -`

Ensuite dans le répertoire `Medium` : `make linux` ou `make solaris` suivant l'architecture.

Ce projet sera effectué en **binômes**. Le rapport devra faire maximum 6 pages. Il faut rendre le rapport et les sources commentées pour le 14 janvier 2002, ainsi que l'ensemble des programmes (en C et promela) par mail aux chargés de TP.