Université de Versailles-St-Quentin-en-Yvelines

LICENCE 2 : Concepts avancés de programmation (IN407)

2023-2024

Projet Python :

Stratégies de gestion de flux à l'entrée d'un un réseau de communication

On considère un réseau de communication à commutation de paquets. Les paquets arrivant au réseau de transmission peuvent provenir de plusieurs sources en entrée de ce réseau (voir Figure 1). Si le taux d'arrivée (en bits/s) de ces paquets au réseau dépasse le taux de transmission du lien, alors les paquets sont stockés dans un buffer (ou file d'attente) noté B, avant d'être transmis sur le lien. Comme ce buffer est de capacité finie, notée C, si à l'arrivée d'un paquet le buffer est plein, ce paquet sera rejeté, c'est-à-dire perdu. On suppose que les paquets arrivent selon un processus de Poisson de paramètre λ.

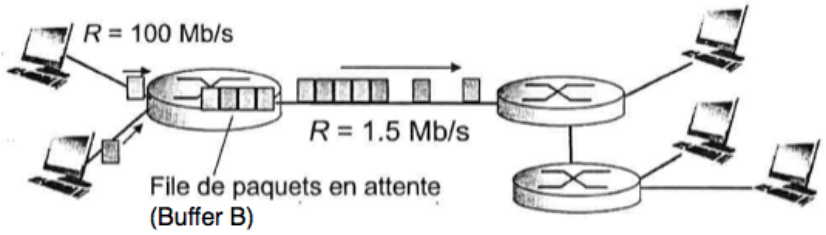

FIGURE 1 – Réseau de communication

L'objectif principal de ce projet est de développer une application qui permet de comparer les stratégies de gestion de flux à l'entrée d'un réseau de communication. Ce projet comporte deux parties.

1 Partie 1 : Structures abstraites de données

On considère le système composé des sources, de la file d'attente et du lien de transmission. L'objectif de cette première partie est de proposer une implémentation orientée objet de ce système. Il s'agira de :

1. Proposer trois classes : la classe $Source$ dont chaque objet est un source de paquets, la classe $Buffer$ dont chaque objet est une file d'attente, et enfin la classe $Paquet$ dont chaque objet est un paquet de données.
2. Implémenter les opérations d'arrivée, d'insertion et de retrait d'un paquet du buffer, et enfin de sa transmission. Pensez à surcharger les opérateurs adéquats.
3. Proposer un programme de test démontrant les fonctionnalités de votre application.
4. Développer une interface graphique qui permet de voir la dynamicité du système, c'est-à-dire l'arrivée d'un paquet dans le buffer, son retrait du buffer et sa transmission.
5. Analyser les performances du réseau en terme de taux de perte des paquets en fonction du taux d'arrivée des paquets λ (le faire varier).

2 Partie 2 : Stratégies de gestion

Pour limiter le nombre de paquets perdus, on associe à chaque source S_i, $i = 1, \ldots N$, un buffer B_i de capacité C_i dans lequel tous les paquets émis par la source S_i sont préalablement stockés avant d'être acheminés vers la file d'attente B (voir Figure 2). On suppose que les paquets d'une source S_i arrivent au buffer B_i selon un processus de Poisson de paramètre λ_i.

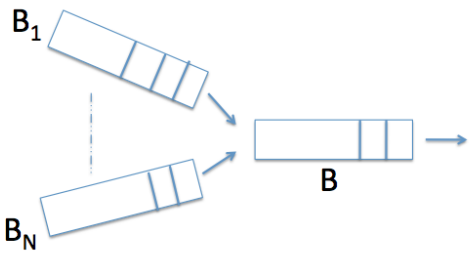

FIGURE 2 – Le système de files d'attente

L'objectif de cette seconde partie du projet est de comparer les performances de plusieurs stratégies de traitement des paquets des différentes files d'attente. En effet, plusieurs stratégies peuvent être considérées, en terme du choix et de l'ordre de traitement des paquets des différentes files d'attente.

- La file d'attente choisie est celle contenant le plus grand nombre de paquets.
- Un paquet est pris de chaque file d'attente, à tour de rôle.
- La file d'attente est choisie de manière aléatoire

Compléter votre programme en :

1. intégrant les différentes files d'attente à votre réseau,
2. implémentant les 3 stratégies décrites ci-dessus,

3. visualisant le retrait d'un paquet d'une file d'attente source B_i et son insertion dans la file d'attente B,
4. comparant les performances des 3 stratégies, en terme de taux de perte des paquets et de leur temps moyen d'attente.

3 Remise du projet

Ce travail est à réaliser en binôme. **Aucun projet en mônome, trinôme,.. ne sera accepté.** Vous devez fournir le programme source commenté (un fichier compressé) et un mécanisme d'installation. Un compte rendu de 3 pages expliquant, entre autres, l'architecture de votre application vous est aussi demandé.

Dates à retenir :

- Remise du projet : vendredi 26 avril 2024
- Présentations : dernière séance de TD.