Projet Réseaux Charleroi

Modélisation d'un file d'attente

Stéphane Delire 241808

Guy Gillain 241973

19 décembre 2023

Construction et exécution

Dans le cadre du cours de laboratoire réseaux, il nous est demandé d'élaborer un simulateur informatique pour modéliser l'envoi de paquets entre deux hôtes avec l'intermédiaire d'un routeur. Cette application doit être en capacité de simuler divers scénarios d'envoi de données en tenant compte des facteurs tels que la distance, la vitesse de propagation, le débit de transmission, ainsi que les délais engendrés par la file d'attente du routeur, qui dispose d'une capacité limitée. La gestion des paquets doit se faire en utilisant la stratégie de tail drop.

Le point d'entrée du simulateur est le fichier Simulator.py. c'est à partir lui qu'il est possible d'explorer les différentes hypothèses requises par le projet.

Pour ajuster les paramètres liés aux liens et à la mémoire du routeur ; il est possible d'agir lors de l'initialisation ou par les fonctions scénarios.

Modification à l'initialisation

Pour les liens, il faut modifier les lignes 13 à 15 du fichier.

Dans notre exemple, nos liens l1 et l2 font 1 mètre et transmettent à 1024 m/s

Listing 1 – Variables pour les liens

```
self.l1 = Link(length=1, speed=1024)
self.l2 = Link(length=1, speed=1024)
```

Pour modifier la taille de la mémoire du routeur, il vous suffit de changer la variable `queue_max_size` à la ligne 15.

Présentation des différents scénarios

Les scénarios sont modélisés sous la forme de fonctions, où vous procédez aux ajustements nécessaires afin de faire varier les données.

Prenons comme exemple le scénario 1 : Avec les paramètres de base,
nous obtenons les valeurs suivantes :

1 0.000 2.000 4.000 5.000 0 False
2 1.000 3.000 9.000 10.000 1 False

Si je modifie les propriétés des liens L1 avec 10000 m pour la distance et $\frac{2}{3}$ vit. lum. pour la vitesse et pour L2, 20000 m et $\frac{2}{3}$ vit lum. Nous obtenons les valeurs suivantes :

1 0.000 1.051 3.051 3.154 0 False
2 1.000 2.051 7.102 7.205 1 False

Il en va de même pour les autres scénarios. En modifiant les paramètres tels que le nom et la taille des paquets, ainsi que la bande passante du lien, vous pouvez personnaliser chaque scénario selon les besoins spécifiques de votre simulation.

Lancement des différents scénarios.

Pour lancer les divers scénarios, la gestion est des plus simples. Il vous suffit de décommenter le scénario que vous souhaitez tester et de commenter les autres. Bien qu'il soit possible de lancer plusieurs simulations simultanément, cela peut entraîner une perte de lisibilité.

Impémentation

Nous avons choisi d'utiliser le langage Python pour la programmation et l'exécution de notre application, en parallèle avec la création d'un référentiel Git afin d'optimiser le processus de développement. Le programme a été organisé en cinq classes distinctes.

La classe Host est dédiée à l'instanciation des hôtes du réseau. La classe Link est spécialement conçue pour créer les divers liens entre les hôtes et le routeur. La classe Packet facilite la création des paquets, tandis que la classe Router est employée pour la conception de notre routeur. Enfin, la classe Simulator a été mise en place pour créer et exécuter les divers scénarios requis.

Le processus de codage en lui-même ne s'est pas révélé particulièrement complexe. Il a toutefois nécessité une attention particulière pour bien comprendre les divers scénarios demandés.

Suite à des conseils recueillis auprès de l'assistant, nous avons décidé d'opter pour le codage en dur des différentes données nécessaires à l'exécution des scénarios. Cette approche a été adoptée pour assurer une gestion plus directe et précise des paramètres, permettant ainsi une meilleure maîtrise des résultats escomptés dans notre simulation réseau.

Application du modèle

Pour l'ensemble des tests, nous allons utiliser une longueur de 10.000 m pour L1, 20.000 m pour L2 et $\frac{2}{3}$ de la vitesse de la lumière comme base de travail..

Sim 1 "Illustration du goulot d'étranglement" :

Voici nos résultats :

1 0.000 1.051 3.051 3.154 0 False
2 1.000 2.051 7.102 7.205 1 False

Sim 2 "Saturation de L2 avec goulot d'étranglement" :

Voici nos résultats, nous constatons que le paquet 2 est rejetté :

1 0.000 1.051 3.051 3.154 0 False
2 1.000 2.051 False False 0 True

Sim 3 "Saturation de L2 sans congestion" :

Voici nos résultats, nous constatons aucune perte de paquets :

0 0.000 1.051 3.051 3.154 0 False
1 2.000 3.051 5.051 5.154 0 False
2 4.000 5.051 7.051 7.154 0 False
3 6.000 7.051 9.051 9.154 0 False
4 8.000 9.051 11.051 11.154 0 False
5 10.000 11.051 13.051 13.154 0 False
6 12.000 13.051 15.051 15.154 0 False
7 14.000 15.051 17.051 17.154 0 False
8 16.000 17.051 19.051 19.154 0 False
9 18.000 19.051 21.051 21.154 0 False

Sim 4 “Rafales de paquets avec L2 comme goulot d’étranglement” :

Nos résultats :

0 0.000 1.051 3.051 3.154 0 False
1 1.000 2.051 5.051 5.154 1 False
2 2.000 3.051 7.051 7.154 2 False
3 6.000 7.051 9.051 9.154 0 False
4 7.000 8.051 11.051 11.154 1 False
5 8.000 9.051 13.051 13.154 2 False
6 12.000 13.051 15.051 15.154 0 False
7 13.000 14.051 17.051 17.154 1 False
8 14.000 15.051 19.051 19.154 2 False
9 18.000 19.051 21.051 21.154 0 False
10 19.000 20.051 23.051 23.154 1 False
11 20.000 21.051 25.051 25.154 2 False

sim 5 : “Envoi aléatoire de paquets (bonus)” :

Pour le bonus voici les résultats que nous obtenons :

0 0.000 0.938 2.723 2.815 0 False
1 2.723 3.737 5.664 5.763 0 False
2 5.664 6.716 8.716 8.818 0 False
3 8.716 9.169 10.033 10.077 0 False
4 10.033 10.605 11.695 11.751 0 False
5 11.695 12.524 14.100 14.181 0 False
6 14.100 14.946 16.555 16.638 0 False
7 16.555 17.369 18.918 18.997 0 False
8 18.918 19.918 21.820 21.918 0 False
9 21.820 22.602 24.091 24.167 0 False
10 24.091 25.014 26.770 26.859 0 False
11 26.770 27.655 29.341 29.427 0 False
12 29.341 30.205 31.850 31.934 0 False
13 31.850 31.981 32.231 32.244 0 False
14 32.231 32.865 34.070 34.131 0 False
15 34.070 34.889 36.448 36.527 0 False
16 36.448 37.495 39.487 39.589 0 False
17 39.487 40.250 41.701 41.775 0 False
18 41.701 42.630 44.397 44.488 0 False
19 44.397 45.272 46.936 47.021 0 False