

Evaluation de performances

Simulations avancées et développement

Nicolas Renard

Exercice 2

Question 1

J'utilise la fonction de répartition de Zipf, grâce à la fonction `np.random.zipf(a, 1)` de numpy, une bibliothèque Python spécialisée dans le calcul mathématique.

Cette fonction prend deux paramètres:

- `a`: Le paramètre de distribution, supérieur à 1. Pour ma part, je l'ai fixé à 1,2 afin d'obtenir une distribution correcte. En effet, si `a` est plus petit, trop de grands nombres sont tirés, et le cas est inversé si `a` est trop grand.
- `1`: le nombre de valeurs retournées. Une seule valeur est retournée car la fonction est intégrée dans une boucle.

Liens

D'après le fichier `topo.top`, la capacité de chaque lien est donnée en Gb. Cet ordre de grandeur est gardé pour la génération des liens. Conséquemment, les files d'attente des nœuds n'ont pas à être très grande, d'où une taille choisie de 10 paquets.

Flux

Dû à l'utilisation de la répartition de Zipf, des nœuds TCP (et donc également des nœuds NULL) sont indispensables. Les segments TCP ont une taille fixe de 1500 octets, limite due à la taille d'une trame Ethernet. Une application FTP est configurée par dessus le nœud TCP afin d'avoir la possibilité de fixer précisément la taille de paquet et son heure d'envoi.

La taille du paquet (en bits) correspond à la valeur tirée par la fonction de répartition de Zipf et est additionnée à la précédente et ce jusqu'à atteindre la valeur correspondante au volume donné dans le fichier `traff.traf`. Selon l'encadrant, le volume est donné en Mo. Il faut donc multiplier cette valeur par 1024×1024 , ainsi que par 8, parce que la taille du paquet est indiquée en bits, et la valeur de volume en octets.

L'heure d'envoi du paquet correspond à une valeur aléatoire tirée entre 0 et 240. La simulation durant 5 minutes (300 secondes), le paquet pourra donc être envoyé jusqu'à 4 minutes après le lancement de celle-ci afin de lui donner le temps de parcourir le réseau.

La numérotation des variables est réalisée comme suit:

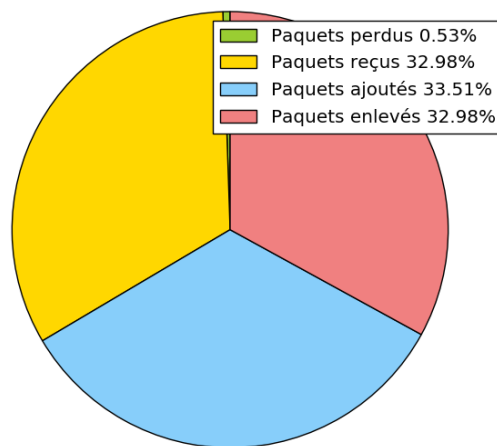
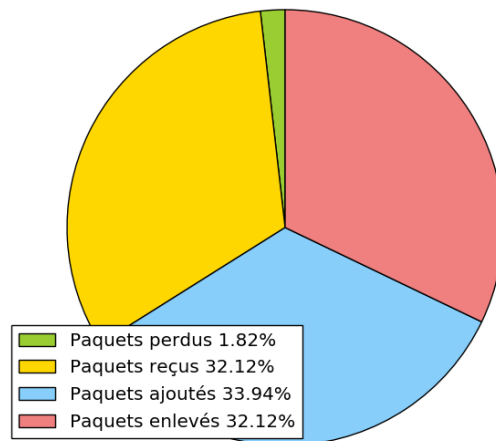
`$tcp(0_2_12)`

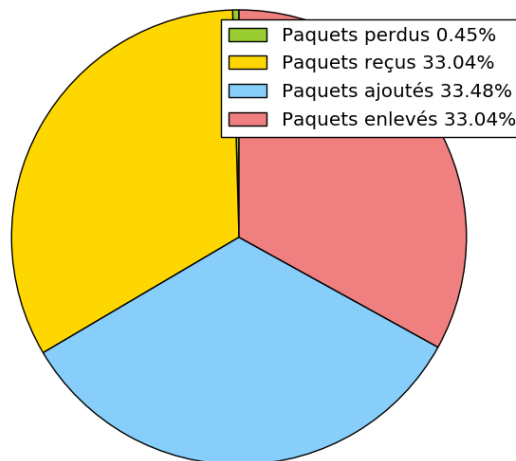
- 0 => nœud source
- 2 => nœud destination
- 12 => itérateur du sous flux

Question 2

J'ai réalisé un graphe en camembert afin de visualiser le pourcentage de pertes pour un lien donné. (0-19, 5-22, 18-22) On peut voir sur les graphes suivants que très peu de pertes sont observées (moins de 2%). Les trois graphes suivants représentent les liens ayant le plus de pertes.

Lien	Pourcentage de pertes
0 -> 19	1,85 %
5 -> 22	0,53 %
18 -> 22	0,45 %





On remarque grâce à ces graphes que le réseau est surdimensionné.

Question 3

Le réseau étant surdimensionné, on peut le mettre à l'épreuve en diminuant la capacité des liens ou en augmentant le volume de données envoyées. Une autre manière de mettre le réseau à l'épreuve serait de désactiver des nœuds afin de simuler des pannes. Les autres nœuds seront donc plus sollicités et mécaniquement, leur charge augmentera.

Les trois graphes suivants représentent à nouveau les mêmes liens ayant le plus de pertes, après avoir adapté le réseau pour en provoquer. On y remarque que les pertes sont beaucoup plus présentes.¹

Lien	Pourcentage de pertes
0 -> 19	31,06 %
14 -> 15	27,7 %
5 -> 22	22,83 %

