

Master 1
Sciences et Ingénierie du Réseaux, de l'Internet et des Système

Évaluation de performance

Rapport

Travaux Pratique
Analyse de données de simulation

Constantin DIVRIOTIS

Tables des Matières

- 1. Manuel d'utilisation**
- 2. Résultats obtenus et performance**
- 3. Analyse des données**

1. Manuel d'utilisation

Le programme `my_eval` nécessite obligatoirement deux arguments : la trace à évaluer et la matrice d'adjacence. Un affichage des données globales est présenté sur la sortie standard à l'utilisateur à la fin de l'exécution du programme.

Pour obtenir un affichage plus précis, il est possible de demander au programme d'afficher les données d'un paquet, d'un flux ou d'un noeud. La condition obligatoire est de passer l'identifiant du paquet, du flux ou du noeud pour obtenir les informations le concernant, c'est-à-dire :

- pour un paquet, l'option -p suivi de son <PID> ,
- pour un flux, l'option -f suivi de son <FID> ,
- pour un noeud, l'option -n suivi de son <NID> .

Enfin, une dernière option permet de générer des fichiers .dat et de créer des graphiques GNUplot : c'est l'option -g. Cette option ne nécessite pas d'argument supplémentaire.

Les informations demandés dans le sujet sont quasiment toutes affichés.

2. Résultats obtenus et performance

Le premier résultat ci-dessous concerne le test portant sur les données générales de l'analyse du fichier de test `trace2650.txt`. Ce test est effectué sans aucune option et avec la commande `time`. Le temps pris par le programme est toujours d'environ 1min40 sur Turing. Il est à noter que ce temps est différent sur des ordinateurs personnels (proche de la minute, environ 1min - 1min10).

```
*****Affichage DATA*****
Nombre de Paquets           : 750979
Paquets traités             : 1000147
Paquets émis                : 750979
Paquets reçues              : 715815
Paquets perdus              : 35164
Paquets actifs              : 0
Nombre de Flux               : 3018
Nombre de Noeuds            : 26
Nombre d'Evenements          : 3502252
Délai moyen                 : 0.006921
Temps attente moyen files    : 0.006016
Temps transmission moyen liens : 0.000904
Taux perte moyen            : 4.682421
*****

real    1m44.171s
user    1m43.589s
sys     0m0.328s
```

Le deuxième résultat est le test précédent avec toutes les options d'affichage et l'option GNUplot -g. L'affichage d'un noeud, d'un paquet (et son chemin) et l'affichage d'un flux sont présents. Pour ces options, comme dit précédemment, il est obligatoire de placer les ID, cela implique donc d'aller dans le fichier de trace et choisir les ID que l'on souhaite afficher sur la sortie standard.

```
*****Affichage Noeud*****
Noeud           : N7
Nombre de Paquets traités : 172772
Nombre de Paquets recus  : 58520
Nombre de Paquets émis   : 55732
Nombre de Paquets perdus : 58520
Nombre de Paquets passés : 117040
*****
*****Affichage Flux*****
FID             : 5
```

```

Début Flux      : 0.020683
Fin Flux        : 6.058257
Source          : N2
Destination     : N19
Nombre de Paquets : 2672
Paquets traités  : 1669
Paquets émis    : 334
Paquets reçues  : 334
Paquets perdus  : 0
Délai moyen     : 0.781434
*****
*****Affichage Paquet*****
PID            : 3018
Source         : N23
Destination    : N16
Temps attente files : 0.000000
Temps chemin   : 0.001894
Taille Paquet  : 0.002525
-----
Chemin emprunté par le paquet :
Numero      Noeud      Code      Date
1           N23        0        0.000000
2           N19        2        0.000000
3           N19        1        0.000631
4           N1         2        0.000631
5           N1         1        0.001263
6           N16        2        0.001263
7           N16        1        0.001894
8           N16        3        0.001894
*****
*****Affichage DATA*****
Nombre de Paquets      : 750979
Paquets traités        : 1000147
Paquets émis           : 750979
Paquets reçues         : 715815
Paquets perdus         : 35164
Paquets actifs         : 0
Nombre de Flux         : 3018
Nombre de Noeuds       : 26
Nombre d'Evenements    : 3502252
Délai moyen            : 0.006921
Temps attente moyen files : 0.006016
Temps transmission moyen liens : 0.000904
Taux perte moyen       : 4.682421
*****
real    1m39.034s
user    1m36.943s

```

Un test (sans option) avec Valgrind a été nécessaire pour vérifier la gestion des malloc et des free. Suite à l'exécution, on obtient :

```

==2474== Memcheck, a memory error detector
==2474== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==2474== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==2474== Command: ./my_eval trace2650.txt matrice_adjacence.txt
==2474==
Chargement...
*****Affichage DATA*****
Nombre de Paquets           : 750979
Paquets traités             : 1000147
Paquets émis                : 750979
Paquets reçues              : 715815
Paquets perdus              : 35164
Paquets actifs              : 0
Nombre de Flux              : 3018
Nombre de Noeuds            : 26
Nombre d'Evenements         : 3502252
Délai moyen                 : 0.006921
Temps attente moyen files   : 0.006016
Temps transmission moyen liens : 0.000904
Taux perte moyen            : 4.682421
*****
==2474==
==2474== HEAP SUMMARY:
==2474==    in use at exit: 0 bytes in 0 blocks
==2474==   total heap usage: 7,758,791 allocs, 7,758,791 frees, 252,672,100
bytes allocated
==2474==
==2474== All heap blocks were freed -- no leaks are possible
==2474==
==2474== For counts of detected and suppressed errors, rerun with: -v
==2474== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

real    9m27.256s
user    9m23.460s
sys     0m2.885s

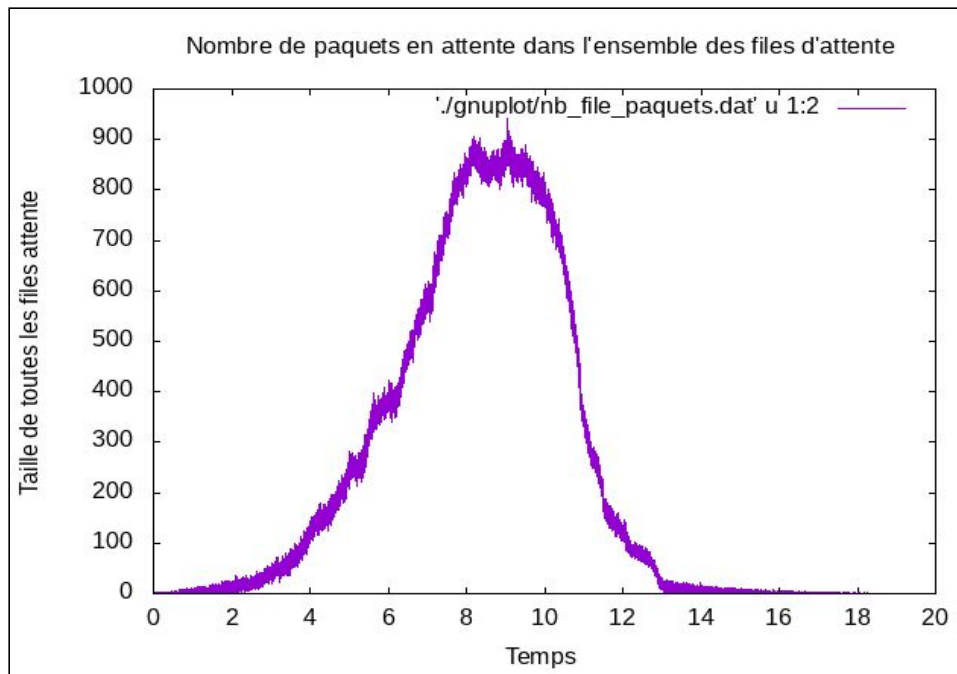
```

Le programme pourrait être amélioré au niveau de la recherche des éléments. Une fonction de recherche optimale des éléments avec une fonction de tri des éléments de la liste chaînée (basé sur les ID) permettrait d'améliorer considérablement le temps du programme. Malheureusement, le temps ne me permettait pas de créer ces deux fonctions de façon optimale. Le programme permet tout de même de répondre à toutes les attentes du sujet.

Remarque : Les tests ont été effectués sur Turing.

3. Analyse des données

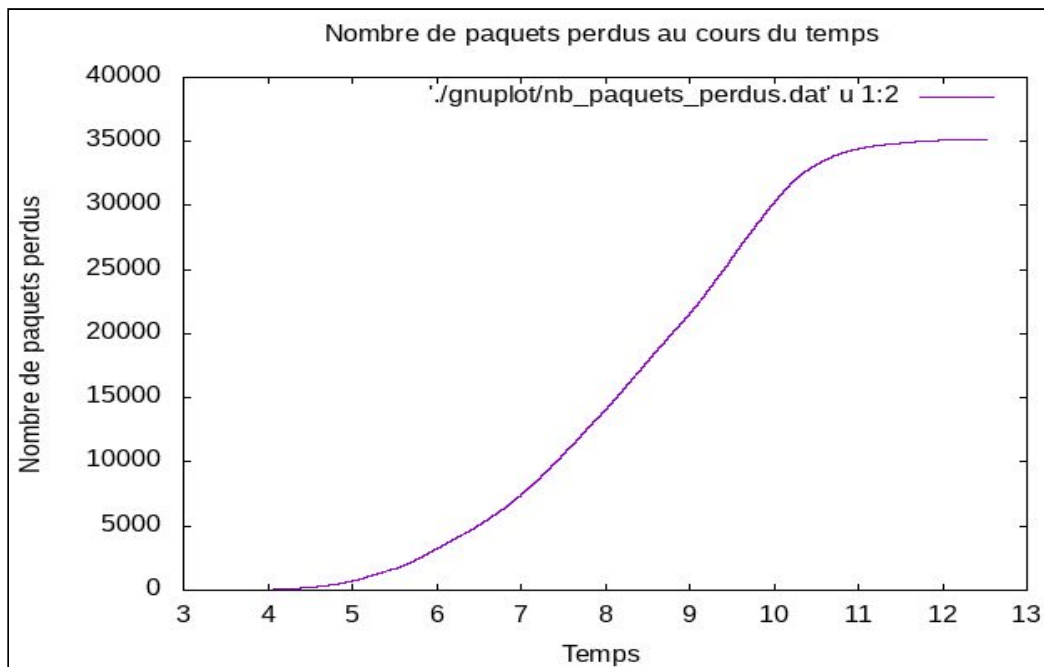
Le programme permet avec l'option -g de placer des fichiers .dat et des graphiques GNUplot sur certains paramètres proposés dans le sujet.



Graphique 1

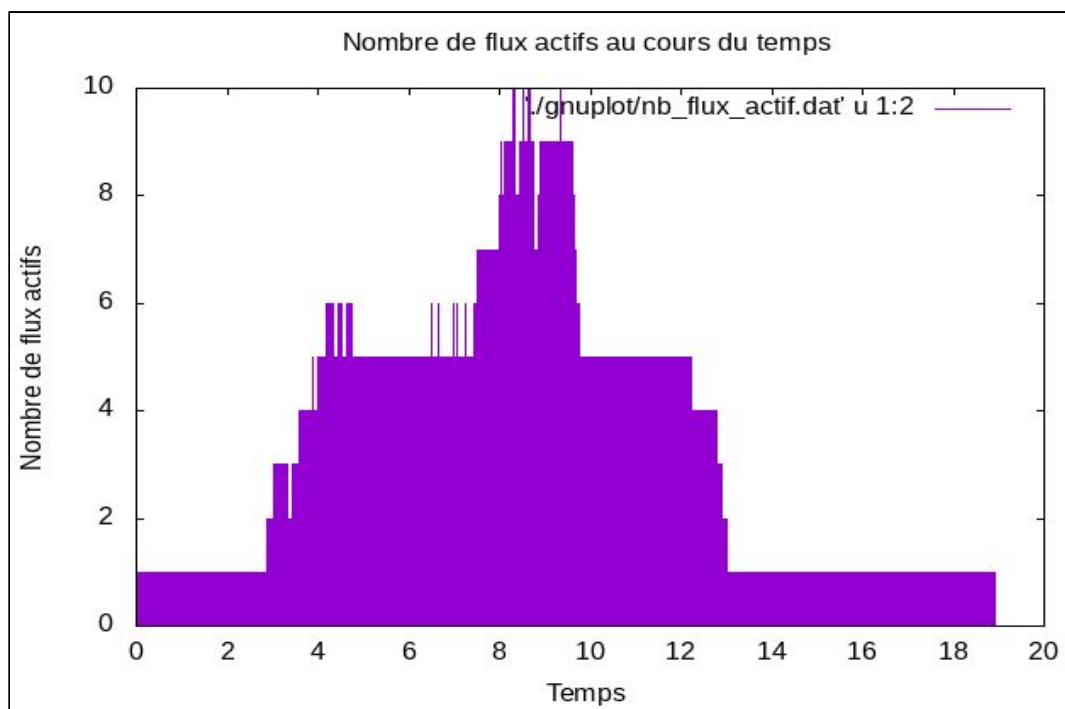
Le premier graphique présenté ci-dessus est celui du nombre de paquets en attente dans toutes les files d'attente c'est-à-dire dans l'ensemble des files d'attente dans les noeuds. On constate que le nombre de paquets en file d'attente est le plus grand au milieu de la trace. On peut envisager l'idée que c'est à cette période que le plus grand nombre de paquets est en transit et ainsi augmente la taille des files d'attente. On peut supposer que la période entre le temps $t=0$ et $t=8$ est une période de warm-up du à la montée exponentielle de la taille de toutes les files d'attente. Le temps $t=12$ jusqu'à la fin de la trace correspond au warm-down. La période entre ces deux phases est une période "stable" correspondant à la période où le nombre de paquets est au maximum. On peut supposer que c'est ce grand nombre de paquets qui remplit les files d'attente.

Le deuxième graphique présenté ci-dessous est celui du nombre de flux actifs présent sur toute la durée de la trace. On constate que le nombre de flux est au plus haut lors de la phase "stable". On peut également observer le même phénomène que précédemment c'est-à-dire une période de warm-up et warm-down.

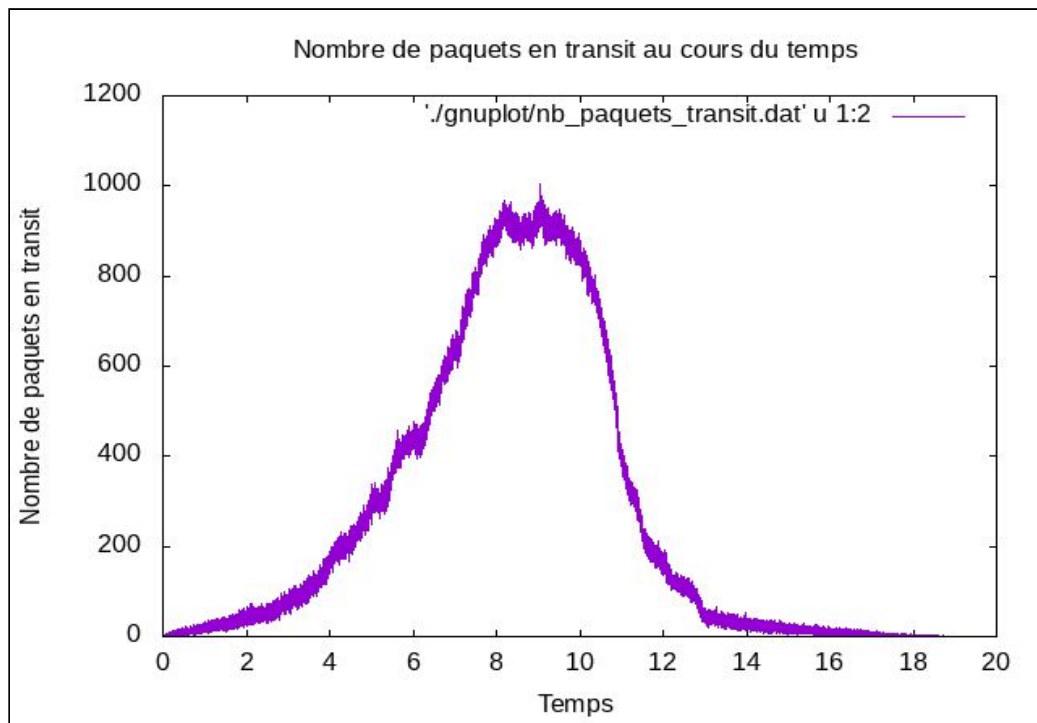


Graphique 2

Le troisième graphique présenté ci-dessous est celui du nombre de paquets perdus au cours de la trace. On observe que ce nombre ne cesse d'augmenter au cours de la trace. Il augmente considérablement à partir du temps $t=6$, car la phase de warm-up est presque terminée. Lors de la phase de warm-down correspondant au temps $t=10$, le nombre de paquets cesse d'augmenter d'où la réduction du nombre de paquets. Entre ces deux temps, le nombre de paquets perdus passe de 5000 à 35 000 et on peut supposer que cela est en lien direct avec le nombre de paquets présents.



Graphique 3



Graphique 4

Le dernier graphique présenté ci-dessus est celui du nombre de paquets en transit. Ce dernier graphique permet d'appuyer l'hypothèse que les tailles de files d'attente et le nombre de paquets perdus augmentent considérablement si le nombre de paquet présent est assez grand. On peut clairement observer le même phénomène que sur les graphiques précédents c'est-à-dire une zone de warm-up et de warm-down intercalant entre ces deux zones, une zone "stable" où le paramètre (ici, le nombre de paquets en transit) est élevé et le reste pendant un certain temps.

Remarque : Ces graphiques sont à retrouver dans le dossier `gnuplot` après exécution du programme avec l'option `-g`.