

M1 RISE	Evaluation de performances
<p style="text-align: center;"><b>TP</b></p> <p style="text-align: center;"><b>Analyse de données de simulation</b></p>	

*Objectifs* : analyse de données de simulation a posteriori ; programmation de l'outil d'analyse en C ; utilisation de GNUplot

## Généralités

Lorsque l'on exécute un scénario de simulation, on peut collecter un ensemble de résultats immédiats (i.e. des résultats directement interprétables à la fin de la simulation). On peut également générer une trace de la simulation, c'est-à-dire une liste d'événements survenus pendant l'exécution, qui sera traitée *a posteriori*.

Souvent, les traces de simulation sont des fichiers plats, dont la taille peut être très grande : leur analyse doit donc être effectuée en utilisant des algorithmes de faible complexité, et minimisant les accès au disque.

## Un cas concret

On a lancé une simulation discrète décrivant un ensemble de paquets circulant sur un réseau IP, à partir de laquelle on a généré une trace décrivant l'ensemble des changements d'état du système simulé. En d'autres termes, chaque ligne de la trace correspond à un événement traité dans l'échéancier discret. Ces événements correspondent aux changements d'états de chacun des paquets IP simulés.

### Topologie du réseau

La topologie du réseau est connue :

- le réseau comporte 26 nœuds et 50 liens bidirectionnels (l'existence du lien (i,j) implique celle du lien (j,i)) et symétriques (les débits sur les liens (i,j) et (j,i) sont égaux),
- le délai de propagation est nul sur chacun des liens : le délai de transmission d'un paquet sur un lien dépend donc uniquement de sa taille et du débit du lien,
- le temps de traitement dans les nœuds (lecture des données d'en-tête, lecture de la table de routage, mise à jour TTL et checksum...) est nul,
- les nœuds disposent d'une file d'attente pour chacune de leurs interfaces (liens) de sortie. Les files sont FIFO, de taille finie. Si l'on veut placer un paquet dans une file d'attente et que celle-ci est pleine, le paquet est simplement détruit.

La matrice d'adjacence est fournie : res26.txt (matrice des bandes passantes en Mbits/s) - voir moodle.

### Trafic

Le trafic est composé d'un ensemble de flux UDP. Un flux représente un ensemble de paquets devant être transmis d'une source vers une destination. Chacun des flux démarre et se termine à des instants donnés. A un instant donné, il peut y avoir plusieurs flux actifs entre un même couple (source,destination). Par simplicité, tous les paquets d'un même flux sont de même taille ; en revanche, les tailles des paquets issus de différents flux sont différentes.

La matrice de trafic et les caractéristiques des flux ne sont pas fournis : ils pourront être déduits de la trace.

### Trace de simulation

La trace de simulation est un fichier texte qui énumère l'ensemble des événements traités durant la simulation. Chaque ligne est de la forme :

t	code	pid	fid	tos	<bif>	s	d	pos
---	------	-----	-----	-----	-------	---	---	-----

- **t** indique l'heure où est survenu l'événement. La simulation débute à t=0, et se termine lorsque tous les paquets émis sont arrivés à destination (ou ont été détruits dans des files pleines), c'est-à-dire quand l'échéancier est vide. Dans l'analyse de la trace, on ne se préoccupe pas des périodes de warmup et de cooldown.
- **code** précise le type de l'événement traité
  - o 0 : départ de la source
  - o 1 : arrivée dans un nœud intermédiaire
  - o 2 : départ d'une file d'attente
  - o 3 : arrivée à destination
  - o 4 : destruction d'un paquet (placement dans une file pleine)
- **pid** : identifiant (unique) du paquet

- **fid** : identifiant du flux auquel appartient le paquet
- **tos** : type de service (vous n'utiliserez pas cette information)
- **bif** : opérateur de bifurcation (vous n'utiliserez pas cette information) ; cette information n'apparaît pas dans les événements dont le code est égal à 4
- **s** : nœud source du paquet
- **d** : nœud destination du paquet
- pos
  - o pour les codes 0, 1, 3, 4 : nœud dans lequel se trouve le paquet à l'instant t
  - o pour le codes 2 : prochain saut = nœud qui se trouve à l'extrémité du lien pour lequel le paquet a été placé en file d'attente.

### Travail à effectuer

Le fichier de trace à analyser (136Mo, 3.502.252 événements) est disponible sur moodle (trace2650.txt.zip).

L'objectif est de programmer un outil d'analyse permettant de tirer de cette trace le maximum d'informations intéressantes en un minimum de temps. Vous programmerez cela en langage C, en vous efforçant de minimiser les complexités (en temps et en mémoire) et les accès disque.

Lorsque vous calculerez des valeurs moyennes, vous appliquerez impérativement les consignes vues en cours (calcul de l'écart-type ou de la variance, intervalle de confiance, ...)

Ci-dessous est proposée une liste de résultats ou de fonctionnalités qui peuvent s'avérer intéressants. Cette liste n'est pas exhaustive, vous pouvez en proposer d'autres.

### Données globales

Ce sont les résultats qui sont certainement les plus simples à obtenir. Elles concernent l'ensemble de la simulation.

- nombre de paquets traités/émis/reçus
- nombre de flux
- nombre de paquets perdus, taux de perte
- localisation des pertes : proportion des paquets perdus dans chaque nœud
- délai moyen de bout en bout
- temps d'attente dans les files / temps de transmission sur les liens
- taille des files d'attente
- etc.

### Traçage

Ce sont les données qui concernent un seul paquet ou un seul flux.

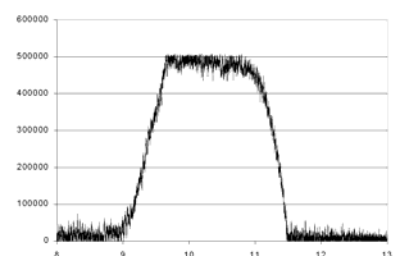
- parcours complet d'un paquet
  - o sa source
  - o sa destination
  - o sa taille
  - o la liste des routeurs traversés, à quel instant
  - o son délai d'acheminement de bout en bout, composé de
    - temps d'attente dans des files (où ? combien de temps ?)
    - temps de transmission sur des liens
  - o etc.
- analyse d'un ou plusieurs flux (par exemple tous les flux entre deux nœuds donnés)
  - o nombre de paquets /émis/reçus/perdus
  - o taux de perte
  - o durée de vie (un flux est actif, ou en vie, durant la période qui sépare l'émission de son premier paquet et la réception ou la destruction du dernier)
  - o délai moyen d'acheminement des paquets de ce(s) flux
  - o débit moyen à l'émission/à la réception
  - o loi qui régit l'émission des paquets (et ses paramètres)
  - o etc.

### Echantillonnage

L'échantillonnage consiste à tracer une courbe représentant les valeurs prises par une variable au cours du temps. Vous pouvez générer ces courbes en créant des fichiers de données pour GNUplot.

Ainsi, vous pouvez représenter graphiquement la variation au cours du temps :

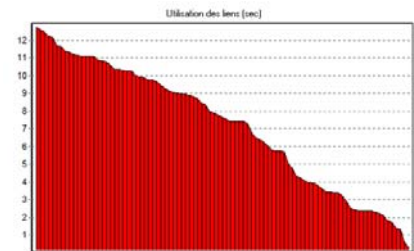
- du nombre de paquets en attente dans une ou l'ensemble des files d'attente (ce qui permet de repérer les congestions)



- du nombre de paquets en transit dans le réseau
- du délai de bout en bout pour l'ensemble des paquets d'un flux
- du nombre de flux actifs (utile pour repérer les périodes warmup et cooldown)
- du nombre de paquets perdus
- etc.

On peut également tracer des courbes représentant la distribution d'un ensemble de valeurs (densité de probabilité d'une ou plusieurs variables).

- délais de bout en bout de l'ensemble des paquets d'un même flux
- utilisation de l'ensemble des liens du réseau (temps ou proportion du temps durant lequel ils transmettent des paquets) : cela permet d'avoir une représentation graphique de l'utilisation des ressources, etc.



### **Résultat final**

Avant la fin de la période précisée pendant la séance, vous déposerez sur moodle :

- l'ensemble de votre code C (+ makefile obligatoire)
- un rapport au format pdf comportant :
  - o une petite documentation utilisateur de votre outil d'analyse, avec des exemples d'utilisation,
  - o l'ensemble des résultats que vous avez obtenus sur la trace fournie