

MIF27 : Rapport TP3

Exercice 1 : Communication Multi-sauts

1. Récupérez l'exécutable "setdest":

```
cd /home/pers/mohammed.haddad/ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen  
cp setdest/ /home/etu/m/p0604950/Desktop
```

usage:

<original 1999 CMU version (version 1)>

```
./setdest -v <1> -n <nodes> -p <pause time> -M <max speed>  
-t <simulation time> -x <max X> -y <max Y>
```

OR

<modified 2003 U.Michigan version (version 2)>

```
./setdest -v <2> -n <nodes> -s <speed type> -m <min speed> -M <max speed>  
-t <simulation time> -P <pause type> -p <pause time> -x <max X>
```

2. Générez une scène avec 10 nœuds (Fixez la vitesse à une valeur proche de 0).

On lance :

```
./setdest -v 1 -n 10 -M 0.001 -t 10 -x 500 -y 300 > script.tcl
```

Le Wifi porte à 250 mètre, on prend une zone de 500*300 (rectangulaire).

Ce qui crée une partie du script.

Avant l'utilisation de la variable \$god, on doit la créer, on ajoute la ligne :

```
set god_ [create-god $val(nn)]
```

3. Etablissez un flux FTP entre deux nœuds séparés de 2, ensuite, 3 sauts (jouez sur les dimensions x et y pour assurer cela ; une forme rectangulaire de l'arène sera plus apte à générer un réseau avec un diamètre plus important).

On a quelques nœuds distants de 2 et 3 sauts.

```
$god_ set-dist 2 6 3
```

```
$god_ set-dist 2 7 2
```

4. Dans la configuration a 2 sauts, ajoutez un flux CBR-UDP entre deux autres nœuds voisins. Que se passe-t-il ?

Le flux UDP perd tout ses paquets à cause des deux autres flux TCP. Ce qui est normal puisque l'UDP ne se régule pas par rapport aux flux TCP.

5. Pourquoi un flux UDP causerait-il plus de gêne a un flux TCP qu'un autre flux TCP ?

Avec un câble Ethernet, les flux UDP prennent le dessus sur les flux TCP, mais avec du WIFI ils perdent la main sur la connexion. TCP renvoie des ACK est régule sa connexion pour un meilleur partage.

Exercice 2 : Evaluation de la mobilité

1. Comment interpréteriez-vous ces formules ?

Formule 1: Moyenne des distances par rapport a un nœud a un instant t.

Cette formule fait la somme des distances des nœuds par rapport a un nœud précis, puis divise le résultat par le nombre de nœuds total, ce qui nous donne une moyenne.

Formule 2: Mobilité moyenne d'un nœud durant un temps T.

Cette formule est calculé pour nœud.

On calcul la distance D séparant un nœud de tous les autres entre l'instant $[t + \text{pas}]$ et t. On fait évoluer t de 0 à 20 et on somme tous les D. Une fois la somme effectuée, on divise par le temps total pour obtenir la mobilité moyenne du nœud.

Formule 3: Mobilité moyenne du réseau.

Elle fait simplement la moyenne des mobilités de tout les nœuds.

2. Dans votre script TCL, définissez une fonction Record qui enregistre dans un fichier les coordonnées X et Y de chaque nœud (une ligne aura la forme : i X Y). La fonction Record se appellera chaque seconde (t).

#appel de notre fonction

set traceCoord [open traceCoord.tr w]

\$ns at 0.0 "appelRecord"

Ecris la coordonnée des points pour chaque noeud au temps t

proc appelRecord {} {

 global ns val node_ traceCoord

 set time 1.0

 set now [\$ns now]

```
for {set i 0} {$i < $val(nn)} { incr i } {  
  puts $traceCoord "$i [$node_($i) set X_] [$node_($i) set Y_]"  
}  
$ns at [expr $now+$time] "appelRecord"  
}
```

3. Développez un programme (dans le langage de votre choix) qui, a partir du chier log obtenu par la question précédente, calcule la mobilité moyenne du réseau.

Mobilite.cpp

4. Générez des scènes avec des facteurs de mobilité respectifs (Mob) : 1, 3 puis 5 (faible, moyen et élève)

Fichiers tcl:

mobilite_s1.tcl

mobilite_s2.tcl

mobilite_s5.tcl

Fichiers trace:

1_out.tr

2_out.tr

5_out.tr