

AbdelMourhit MAZIANE  
Swan ROCHER  
Master 1 - MOCA

# Simulation de réseaux IP avec *ns*

May 14, 2012

Université Montpellier 2

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Topologie du Réseau</b>	<b>2</b>
<b>3</b>	<b>Code Source</b>	<b>3</b>
<b>4</b>	<b>Résultats</b>	<b>6</b>
4.1	Débits moyens . . . . .	6
4.2	Taux de perte . . . . .	7
4.3	Taille de la file d'attente . . . . .	7
4.4	Taille de la fenêtre TCP . . . . .	7
<b>5</b>	<b>Conclusion</b>	<b>8</b>

## 1 Introduction

Il est souvent difficile et long d'effectuer réellement des tests de réseaux, c'est pourquoi il est préférable d'utiliser des outils de simulation. Ici, nous nous intéressons au logiciel *ns*, et nous étudions les débits moyens, taux de perte de paquets ainsi que la taille de la file d'attente et des différentes fenêtres TCP. Le but de ce TP étant de nous familiariser avec cet outil, les différents tests effectués sont très simples et ont un intérêt pratique discutable.

## 2 Topologie du Réseau

Le réseau simulé est composé de deux sites avec un lien du premier vers le second d'une capacité de 2Mb/s et de latence 20ms. Sa file d'attente de type *DropTail* peut supporter 100 paquets.

Quatre trafics sont mis en place :

- la première connexion suit le protocole UDP, son débit est exponentiel, la durée moyenne des périodes d'activité est de 10ms, tandis que celles d'inactivité durent 5 ms ;
- les trois autres suivent un protocole TCP, et sont de débits constants, leurs latences respectives sont de 50ms, 100ms et 150ms.

Le débit moyen théorique de chacune de ces connexions est de 1Mb/s, il était donc nécessaire de calculer le débit crête du lien UDP de manière à pouvoir obtenir un débit moyen théorique juste malgré les périodes d'inactivité.

$$\bar{\lambda} = \frac{\lambda_{crete} \times ON}{ON + OFF}$$

$$\text{On a donc : } \lambda_{crete} = \frac{(ON + OFF) \times \bar{\lambda}}{ON}.$$

$$\text{On en déduit } \lambda_{crete} = 1.5 Mb/s.$$

De plus il fallait également calculer la taille des paquets des connexions TCP de manière à assurer le débit moyen théorique de 1Mb/s.

$$\bar{\lambda} = \frac{taille_{paquet}}{intervalle}$$

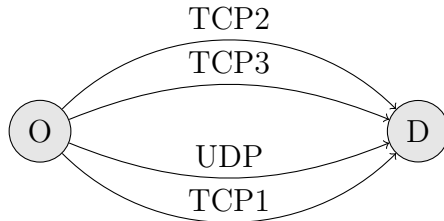
$$\text{On a donc : } taille_{paquet} = \bar{\lambda} \times intervalle.$$

Les tailles de paquets sont donc respectivement :

- 6250 octets,

- 12500 octets,
- 18750 octets.

Les quatre connexions sont lancées simultanément au démarrage, et sont stoppées à 10 secondes du démarrage.



### 3 Code Source

```

1  #####
2  # tp1.tcl
3  # TP de simulation de reseaux avec ns
4  # 2012/04/20
5  # Maziane Abdel-Mourhit
6  # Rocher Swan
7  # UM2 - M1 MOCA
8  #####
9  set ns [new Simulator]
10
11 # creation des noeuds
12 set O [$ns node]
13 set D [$ns node]
14
15 # ouverture des fichiers
16 set f [open res/tp1.tr w]
17 set queueFile001 [open res/tp1_queue001.tr w]
18 set queueFile005 [open res/tp1_queue005.tr w]
19 set windowSizeFile1 [open res/tp1_winsize1.tr w]
20 set windowSizeFile2 [open res/tp1_winsize2.tr w]
21 set windowSizeFile3 [open res/tp1_winsize3.tr w]
22 $ns trace-all $f
23
24 # creation et configuration du lien "physique"
25 $ns duplex-link $O $D 2Mb 20ms DropTail

```

```

26 $ns queue-limit $O $D 100
27
28 # creation d'un moniteur sur la file (afin de mesurer sa taille)
29 set queueMonitor [$ns monitor-queue $O $D $queueFile001]
30
31 # lien 1 : UDP
32 set udp0 [new Agent/UDP]
33 $udp0 set fid_ 1
34 $ns attach-agent $O $udp0
35 set null0 [new Agent/Null]
36 $ns attach-agent $D $null0
37 $udp0 set class_ 1
38 $ns connect $udp0 $null0
39 set Expo0 [new Application/Traffic/Exponential]
40 $Expo0 attach-agent $udp0
41 $Expo0 set rate_ 1.5Mb
42 $Expo0 set burst_time_ 10ms
43 $Expo0 set idle_time_ 5ms
44 $Expo0 set packet_size_ 100
45
46 # lien 2 : TCP
47 set tcp0 [new Agent/TCP]
48 $tcp0 set fid_ 2
49 $ns attach-agent $O $tcp0
50 set sink0 [new Agent/TCPSink]
51 $ns attach-agent $D $sink0
52 $ns connect $tcp0 $sink0
53 set cbr0 [new Application/Traffic/CBR]
54 $cbr0 set packet_size_ 6250
55 $cbr0 set interval_ 50ms
56 $cbr0 attach-agent $tcp0
57
58 # lien 3 : TCP
59 set tcp1 [new Agent/TCP]
60 $tcp1 set fid_ 3
61 $ns attach-agent $O $tcp1
62 set sink1 [new Agent/TCPSink]
63 $ns attach-agent $D $sink1
64 $ns connect $tcp1 $sink1
65 set cbr1 [new Application/Traffic/CBR]
66 $cbr1 set packet_size_ 12500
67 $cbr1 set interval_ 100ms

```

```

68 $cbr1 attach-agent $tcp1
69
70 # lien 4 : TCP
71 set tcp2 [new Agent/TCP]
72 $tcp2 set fid_ 4
73 $ns attach-agent $O $tcp2
74 set sink2 [new Agent/TCPSink]
75 $ns attach-agent $D $sink2
76 $ns connect $tcp2 $sink2
77 set cbr2 [new Application/Traffic/CBR]
78 $cbr0 set packet_size_ 18750
79 $cbr2 set interval_ 150ms
80 $cbr2 attach-agent $tcp2
81
82 # instructions de demarrages
83 $ns at 0.0 "recordQueue 0.01 $queueFile001"
84 $ns at 0.0 "recordQueue 0.05 $queueFile005"
85 $ns at 0.0 "recordWinSize 0.01 $windowSizeFile1 $tcp0"
86 $ns at 0.0 "recordWinSize 0.01 $windowSizeFile2 $tcp1"
87 $ns at 0.0 "recordWinSize 0.01 $windowSizeFile3 $tcp2"
88 $ns at 0.1 "$Expo0 start"
89 $ns at 0.1 "$cbr0 start"
90 $ns at 0.1 "$cbr1 start"
91 $ns at 0.1 "$cbr2 start"
92 $ns at 10 "finish"
93
94
95 # Procedures
96
97 # effectue les logs de la taille de file
98 proc recordQueue {delai qfile} {
99     global ns queueMonitor
100     set now [$ns now]
101     $queueMonitor instvar size_ pkts_ barrivals_ bdepartures_ parrivals_
102     puts $qfile "$now [$queueMonitor set pkts_]"
103     $ns after $delai "recordQueue $delai $qfile"
104 }
105
106 # effectue les logs des tailles de fenetre
107 proc recordWinSize {delai wfile tcpAgent} {
108     global ns
109     set now [$ns now]

```

```

110     puts $wfile "$now [$tcpAgent set cwnd_]"
111     $ns after $delai "recordWinSize $delai $wfile $tcpAgent"
112 }
113
114 # ferme les differents fichiers
115 proc finish {} {
116     global ns f queueFile001 queueFile005 windowSizeFile1 windowSizeFile
117     $ns flush-trace
118     close $f
119     close $queueFile001
120     close $queueFile005
121     close $windowSizeFile1
122     close $windowSizeFile2
123     close $windowSizeFile3
124     exit 0
125 }
126
127 $ns run

```

## 4 Résultats

### 4.1 Débits moyens

La première étape consistait à déterminer les débits moyens respectifs des quatre liens. Ainsi nous avons obtenu les résultats suivants :

- UDP : 121545 octets/s = 950 kb/s
- TCP 1 : 112032 octets/s = 875 kb/s
- TCP 2 : 10917 octets/s = 85 kb/s
- TCP 3 : 7426 octets/s = 58 kb/s

Nous remarquons bien que les débits sont décroissants, ce qui est bien le résultat attendu : en effet, l'UDP envoie sans attendre de réponse, tandis que les trois TCP sont ordonnés par temps de latence croissant. De plus, il est visible que les deux premiers liens utilisent presque toute la bande passante disponible.

## 4.2 Taux de perte

La deuxième étape était de mesurer les taux de perte respectifs de chacun des liens. Nous avons procédé de deux manières différentes, la première valeur est le taux de pertes de paquets, tandis que la seconde est le taux de pertes en quantité d'information (octets).

- UDP : 3.58% ; 3.58% (taille de paquets fixée)
- TCP1 : 1.86% ; 3.52%
- TCP2 : 3.34% ; 6.31%
- TCP3 : 2.12% ; 4.07%

Le calcul des taux de pertes revient à diviser le nombre de paquets (resp. leur taille) perdus par le nombre envoyé. On remarque que les taux de perte ne sont pas si élevés du fait que les agents TCP attendent une réponse avant de recommencer à envoyer.

## 4.3 Taille de la file d'attente

Comme nous pouvons le constater sur la figure suivante 4.3, la taille de la file d'attente du lien augmente puis diminue le temps que les messages soient traités.

## 4.4 Taille de la fenêtre TCP

Sur la figure suivante 4.4 où les tailles des différentes fenêtres TCP sont représentées, nous observons que la première connexion TCP utilise la majorité de la bande passante restante après le passage du trafic UDP. Ainsi les deux dernières connexions ont une fenêtre qui diminue peu (étant donné que la fenêtre est réinitialisée à la réception d'un accusé et que ceux-ci ne peuvent pas circuler sur le lien surchargé).



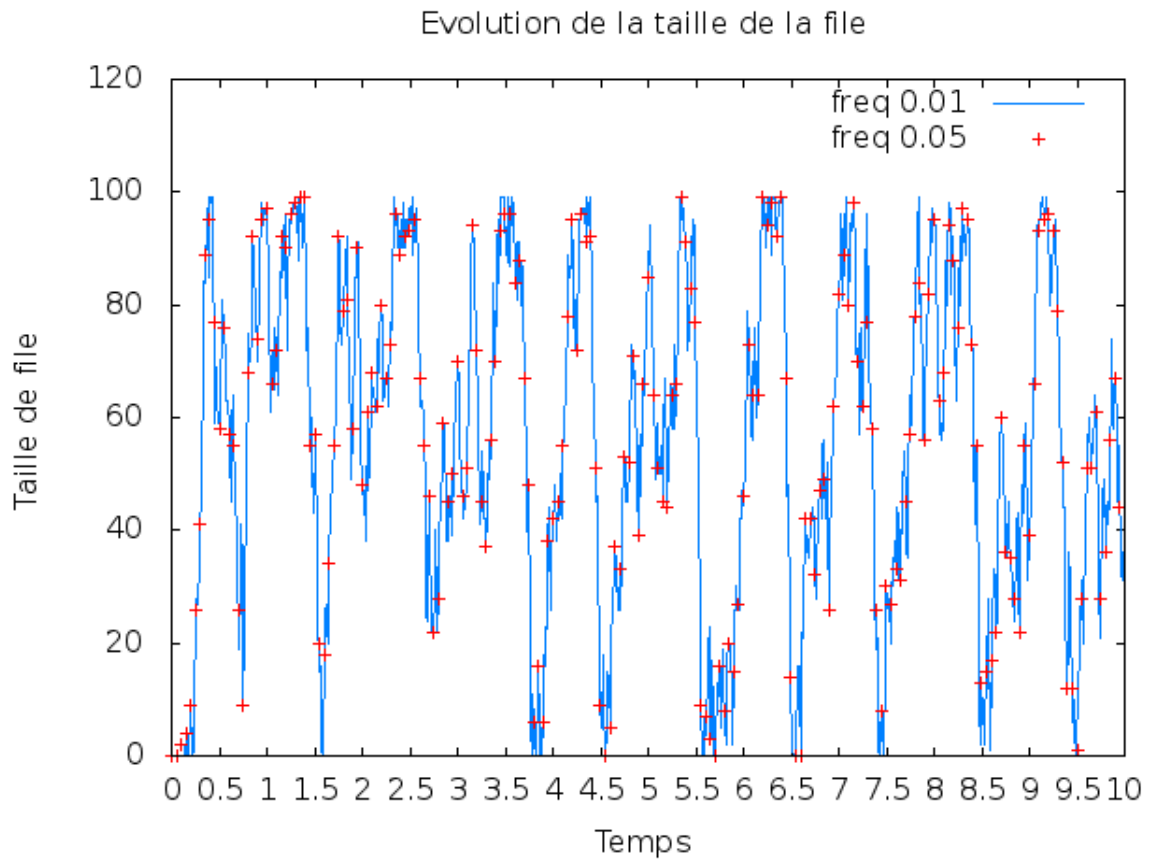


Figure 1: Evolution de la taille de la file d'attente au cours du temps

## 5 Conclusion

Pour conclure, nous pouvons dire que nous avons rencontré peu de difficultés pour l'implémentation, mais qu'il a été moins évident d'analyser les résultats, particulièrement la taille de la fenêtre TCP dont l'utilité était obscure. Mais l'ensemble des tests s'est bien déroulé, et évidemment, les connexions TCP ne peuvent que difficilement communiquer, puisque la connexion UDP utilise la majorité de la bande passante. En effet, quatre débits de 1Mb/s ne peuvent circuler sans encombre sur un lien ayant une capacité de 2Mb/s.

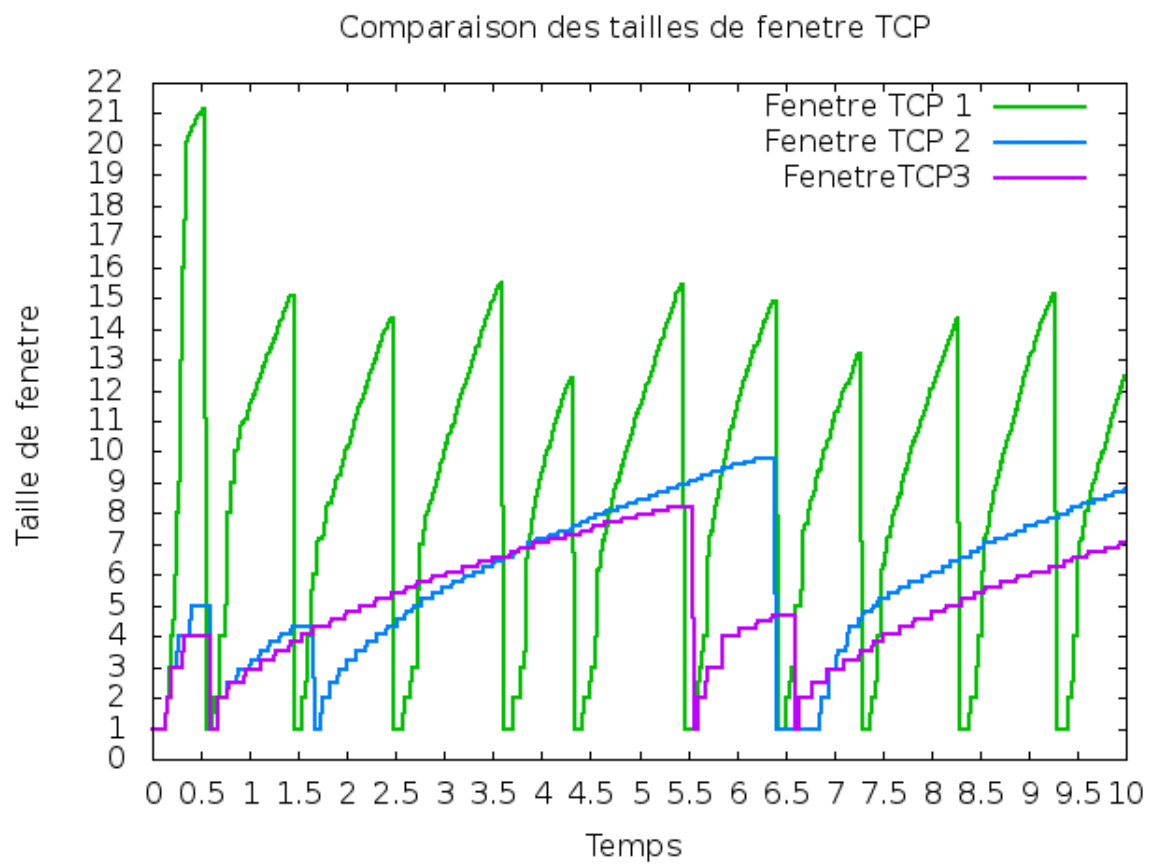


Figure 2: Evolution de la taille de la fenetre TCP