

Service et qualité des réseaux - TP1

23 mars 2012

Auteurs :

Chloé DESDOUITS
Guillaume DUVILLIE

Professeur :

Anne-élisabeth BAERT

Table des matières

1	Introduction	3
2	Paramètres de la simulation	3
2.1	UDP	3
2.2	TCP	4
3	Code source du script TCL	4
4	Analyse et résultats	6
4.1	Le fichier de trace	6
4.2	Taille de la fenêtre TCP	7
4.3	Longueur de la file d'attente	7

1 Introduction

L'objectif principal de ce TP est une prise en main d'un logiciel de simulation de réseaux libre : *ns-2*. Ce logiciel est basé sur un langage de script nommé *Tcl* et le langage orienté objet *C++*. C'est sur *Tcl* que repose l'essentiel de la simulation, puisque c'est dans ce langage que sont écrits les scripts de simulation définissant, de manière séquentielle, la topologie du réseau, les échanges entre les noeuds du réseau ainsi que les protocoles et paramètres utilisés par ces échanges¹.

Un second objectif repose sur l'étude du fichier de trace. Il s'agit d'un fichier texte, généré par *ns-2*, représentant un historique de tous les échanges du réseau simulé. Il sauvegarde entre autre la date d'envoi des paquets, la taille de ceux-ci, le protocole utilisé ou encore s'il s'agit d'un paquet perdu ou non.

Le premier objectif apparaît clairement dans la mise en place de la problématique étudiée : il s'agit de simuler un réseau dont la topologie est décrite à la figure 1. Le médium utilisé est un câble bidirectionnel de capacité $c_m = 2Mb.s^{-1}$ et de latence $l = 20ms$, utilisé par quatre connexions différentes, dont les caractéristiques désirées sont détaillées dans le tableau suivant :

Connexion	Débit	Trafic	Périodes OFF	Périodes ON
UDP	$1Mb.s^{-1}$	Loi exponentielle	$5ms$	$10ms$
TCP ₁	$1Mb.s^{-1}$	Constant ($f_1 = 20Hz$)	Variable	Variable
TCP ₂	$1Mb.s^{-1}$	Constant ($f_2 = 10Hz$)	Variable	Variable
TCP ₃	$1Mb.s^{-1}$	Constant ($f_3 = 6.66Hz$)	Variable	Variable

Considérons le débit *désiré* total donné par :

$$d_{dt} = \sum_{i \in \text{Connexions}} d_i = 4Mb.s^{-1}$$

. On remarque alors que $d_{dt} = 2 \times c_m$, ce qui implique qu'un phénomène de perte se produira au cours de la simulation. L'étude de ce phénomène repose sur le second objectif énoncé précédemment.

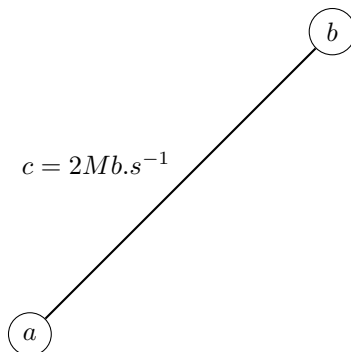


FIGURE 1 – Topologie du réseau

2 Paramètres de la simulation

2.1 UDP

Le débit moyen de la connexion étant fixé, il est nécessaire de calculer le débit crête d_c de cette dernière en fonction de la durée des périodes ON et périodes OFF. Ceci se fait à l'aide de la formule

1. Il est possible par exemple de définir si les échanges utilisent *TCP* ou *UDP*, le temps d'accès au médium de communication ou même la loi d'arrivée des paquets à envoyer.

suivante :

$$\begin{aligned}\bar{d}(Mb.s^{-1}) &= \frac{d_c(Mb.s^{-1}) \times ON(s)}{ON(s) + OFF(s)} \\ 1 &= \frac{d_c \times 10}{10 + 5} \\ d_c &= 1.5\end{aligned}$$

Le débit d'envoi des paquets est donc fixé à $1.5Mb.s^{-1}$ pour atteindre un débit moyen théorique de $1Mb.s^{-1}$. Les autres paramètres de cette connexion peuvent se configurer directement dans le script.

2.2 TCP

Le trafic de chacune des connexions est constant et se fait à fréquence donnée dans le tableau présenté en introduction. Il est donc nécessaire de fixer la taille des paquets à envoyer afin d'obtenir le débit théorique de chacune des connexions TCP. Ceci se fait ainsi :

$$\begin{aligned}\bar{d}(Mb.s^{-1}) &= \frac{t_p(Mb)}{i(s)} \\ 1 &= \frac{t_p}{50.10^{-3}} \\ t_p &= 1 \times 50.10^{-3} \\ t_p &= 50.10^{-3} \\ t_p &= 6,25Ko\end{aligned}$$

De la même manière, il est possible de calculer la taille des paquets pour chacune des autres connexions TCP dont les intervalles sont donnés ainsi : $i_2 = 100ms$ et $i_3 = 150ms$. On obtient alors : $t_{p2} = 12,5Ko$ et $t_{p3} = 18,75Ko$.

3 Code source du script TCL

```

1  set ns [new Simulator]
2
3
4  $ns color 0 blue
5  $ns color 1 red
6  $ns color 2 white
7
8  puts "Mise en place des noeuds"
9  set 0 [$ns node]
10 set D [$ns node]
11
12 set windowSize [open ./windowSize.txt w]
13 set queue1 [open ./queueSize1.txt w]
14 set queue5 [open ./queueSize5.txt w]
15 set f [open ./tp1.tr w]
16 $ns trace-all $f
17 set nf [open ./tp1.nam w]
18 $ns namtrace-all $nf
19
20 puts "Mise en place des liens"
21 $ns duplex-link $0 $D 2Mb 20ms DropTail
22 $ns queue-limit $0 $D 100
23 set spy [$ns monitor-queue $0 $D $queue1]
24
25 #Mise de la connection UDP
26 set udp [new Agent/UDP]
27 $udp set fid_ 1
28 set trappe [new Agent/LossMonitor]
29 $ns attach-agent $0 $udp
30 $ns attach-agent $D $trappe
31 $ns connect $udp $trappe
32 set appUDP [new Application/Traffic/Exponential]
33 $appUDP attach-agent $udp
34 # $udp set packetSize_ 100
35 $appUDP set burst_time_ 10ms
36 $appUDP set idle_time_ 5ms
37 $appUDP set rate_ 1.5Mb

```

```

38 $ns at 0.0 "$AppUDP start"
39
40
41 puts "Mise de la premiere connection TCP"
42 set tcp1 [$ns create-connection TCP $0 TCPSink $D 2]
43 set appTCP1 [new Application/Traffic/CBR]
44 $appTCP1 attach-agent $tcp1
45 $appTCP1 set interval_ 50ms
46 $tcp1 set packetSize_ 6250
47 $ns at 0.0 "$AppTCP1 start"
48
49 puts "Mise de la seconde connection TCP"
50 set sinkTCP2 [new Agent/TCPSink]
51 set tcp2 [new Agent/TCP]
52 $tcp2 set fid_ 3
53 $ns attach-agent $0 $tcp2
54 $ns attach-agent $D $sinkTCP2
55 $ns connect $tcp2 $sinkTCP2
56 set appTCP2 [new Application/Traffic/CBR]
57 $appTCP2 attach-agent $tcp2
58 $appTCP2 set interval_ 100ms
59 $tcp2 set packetSize_ 12500
60 $ns at 0.0 "$AppTCP2 start"
61
62 puts "Mise de la troisieme connection TCP"
63 set sinkTCP3 [new Agent/TCPSink]
64 set tcp3 [new Agent/TCP]
65 $tcp3 set fid_ 4
66 $ns attach-agent $0 $tcp3
67 $ns attach-agent $D $sinkTCP3
68 $ns connect $tcp3 $sinkTCP3
69 set appTCP3 [new Application/Traffic/CBR]
70 $appTCP3 attach-agent $tcp3
71 $appTCP3 set interval_ 150ms
72 $tcp3 set packetSize_ 18750
73 $ns at 0.0 "$AppTCP3 start"
74
75 set timeFinish 20.0
76
77 $ns at 0.0 "printTCPwin"
78 $ns at 0.0 "printQueueSize1"
79 $ns at 0.0 "printQueueSize5"
80 $ns at $timeFinish "finish"
81
82 proc printTCPwin {} {
83     global ns tcp1 tcp2 tcp3 timeFinish windowSize
84
85     set now [$ns now]
86     set delai 0.05
87
88     puts $windowSize "$now [$tcp1 set cwnd_] [$tcp2 set cwnd_] [$tcp3 set cwnd_]"
89
90     $ns after $delai printTCPwin
91 }
92
93 proc printQueueSize1 {} {
94     global ns spy queue1
95
96     set now [$ns now]
97     set delai 0.01
98
99     if {[ $spy set pkts_] > 0} {
100         puts $queue1 "$now [ $spy set pkts_]"
101     }
102
103     $ns after $delai printQueueSize1
104 }
105
106 proc printQueueSize5 {} {
107     global ns spy queue5
108
109     set now [$ns now]
110     set delai 0.05
111
112     if {[ $spy set pkts_] > 0} {
113         puts $queue5 "$now [ $spy set pkts_]"
114     }
115
116     $ns after $delai printQueueSize5
117 }
118
119 proc finish {} {
120     global ns f nf filename
121     $ns flush-trace
122     close $f
123     close $nf
124 }

```

```

125 #puts "running nam..."
126 #exec nam "tpl.tr" &
127
128     exit 0
129 }
130
131 $ns run

```

4 Analyse et résultats

4.1 Le fichier de trace

Afin d'étudier et ensuite de quantifier le phénomène de perte annoncé, nous avons été amenés à analyser le fichier de trace généré lors de la simulation. Ce dernier est formaté selon une syntaxe particulière :

<evt> <tps> <orig> <dest> <pkt> <size> — <id> <src id> <dest id> <seq> <attr>

Les champs nous intéressant sont les suivants :

- <evt> : il décrit l'action effectuée. Lorsqu'un paquet est envoyé depuis un noeud il prend la valeur -, lorsqu'il est reçu, il prend la valeur r, il prend la valeur d pour un paquet perdu, enfin il prend la valeur + pour un paquet qui s'apprête à être envoyé².
- <size> : il s'agit de la taille du paquet et est utilisé pour calculer le débit
- <src id> : représente l'identité de l'émetteur du paquet. Dans le réseau actuel les identités suivent le format suivant : num_noeud.num_prot. num_prot étant égal à 1 pour UDP, 2 pour TCP₁, 3 pour TCP₂ et 4 pour TCP₃.
- <dest id> : représente l'identité du destinataire du message et suit le même format que <src id>.

Un petit programme C a été rédigé afin de parcourir le fichier de trace et de calculer les différentes valeurs recherchées, qui sont, pour chacun des noeuds : le débit de chargement (d_{ch}), le débit envoyé (d_s), le débit de perte (d_p) et le débit reçu (d_r). À l'aide de ces valeurs, nous pourrions calculer le taux de pertes τ_p et le taux de réception τ_r donnés par :

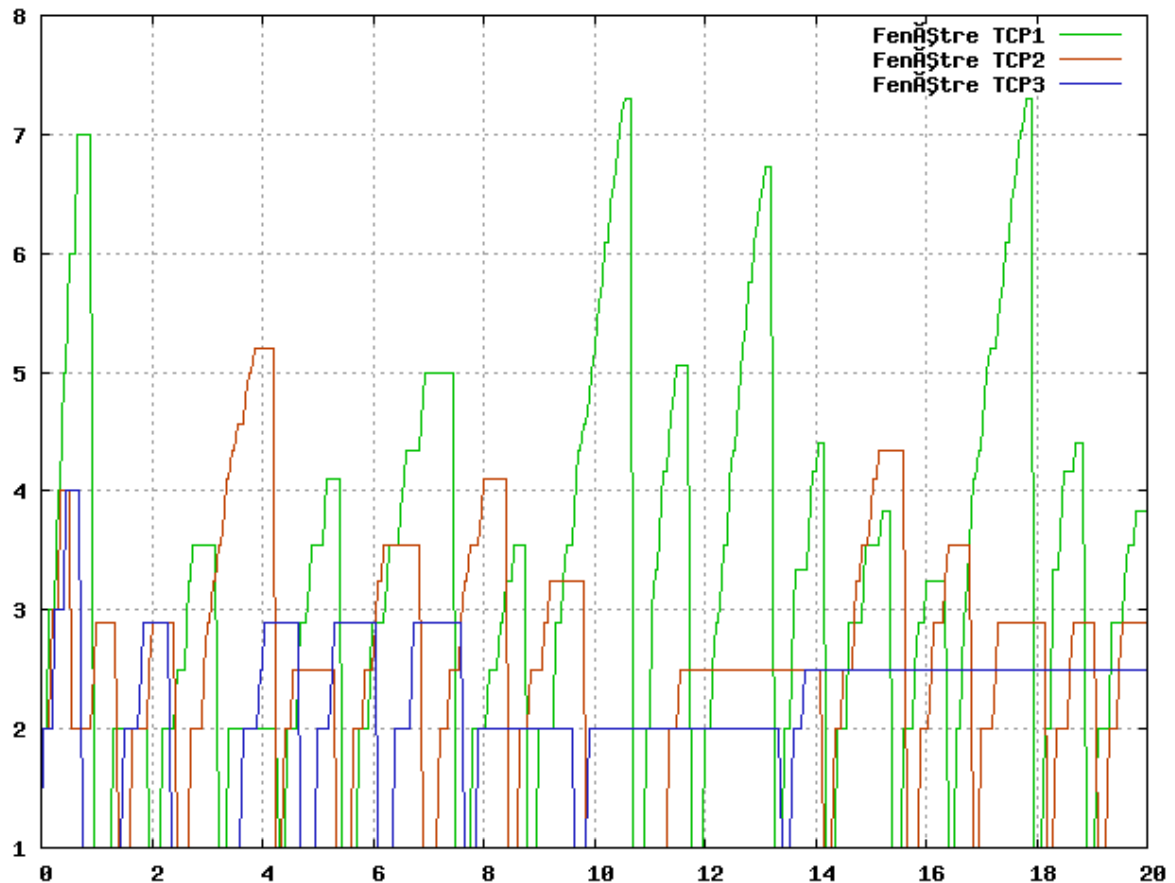
$$\tau_p = \frac{d_p}{d_{ch}} \quad \text{et} \quad \tau_r = \frac{d_r}{d_{ch}}$$

Les résultats obtenus sont décrits dans le tableau suivant, tous les débits étant donnés en $Kb.s^{-1}$:

Connexion	d_{ch}	d_e	d_r	d_p	τ_p	τ_r
UDP	1010,23	775,74	773,64	28,57	0,23	0,77
TCP ₁	616,44	543,47	543,47	70,45	0,12	0,88
TCP ₂	672,16	491,58	491,58	170,54	0,27	0,73
TCP ₃	323,20	187,92	187,92	135,29	0,42	0,58
Total	2622,03	1998,71	1996,61	604,84	0,24	0,76

On peut remarquer plusieurs choses :

- le total des débit envoyés n'excède pas $2Mb.s^{-1}$, ce qui paraît logique puisqu'il s'agit de la limite physique du medium
- le total des débits de chargements n'est pas égal à $4Mb.s^{-1}$, ce qui est expliqué par le fait que le protocole TCP attend un ACK avant d'envoyer un second paquet. La taille des paquets étant fixée en fonction de la fréquence d'envoi, ce temps de latence réduit considérablement le débit réel.
- le débit reçu est différent du débit envoyé pour UDP. Ceci s'explique par la nature même de UDP qui ne garantit pas l'acheminement du paquet envoyé, il existe donc une certaine quantité de paquets perdus sur le médium qui est représentée par la différence de ces deux valeurs.



4.2 Taille de la fenêtre TCP

4.3 Longueur de la file d'attente

On remarque que la taille de la fenêtre TCP augmente lorsque la file d'attente diminue. La taille de la fenêtre TCP étant lié aux accusés de réception, le nombre de ces derniers augmente lorsque les traitement se font plus importants et donc lorsque la taille de la file d'attente diminue.

Un des exemples les plus flagrants a lieu à la 11^e seconde, où l'on observe une chute de la taille de la file d'attente et une fenêtre TCP₁ atteignant un pic.

5 Conclusion

Ce TP a permis la découverte, plus ou moins laborieuse, du langage Tcl ainsi que la mise en avant du comportement relativement aléatoire du simulateur ns-2³.

De plus, bien que ne sachant pas exactement ce à quoi correspond la taille de la fenêtre TCP, ce TP nous a permis de mettre en évidence une relation directe entre cette dernière et la taille de la file d'attente du nœuds destinataire.

Réalisé avec L^AT_EX

2. En cas de surcharge du medium, le paquet ne pourra pas être envoyé et contribuera à la quantité de perte. Le débit relatif à cette quantité sera appelé débit de chargement

3. On peut par exemple citer le fait que la taille des paquets TCP peut être fixée arbitrairement par le simulateur, ou le fait que certains conflits de paramètres des connexions ne sont pas signalés.

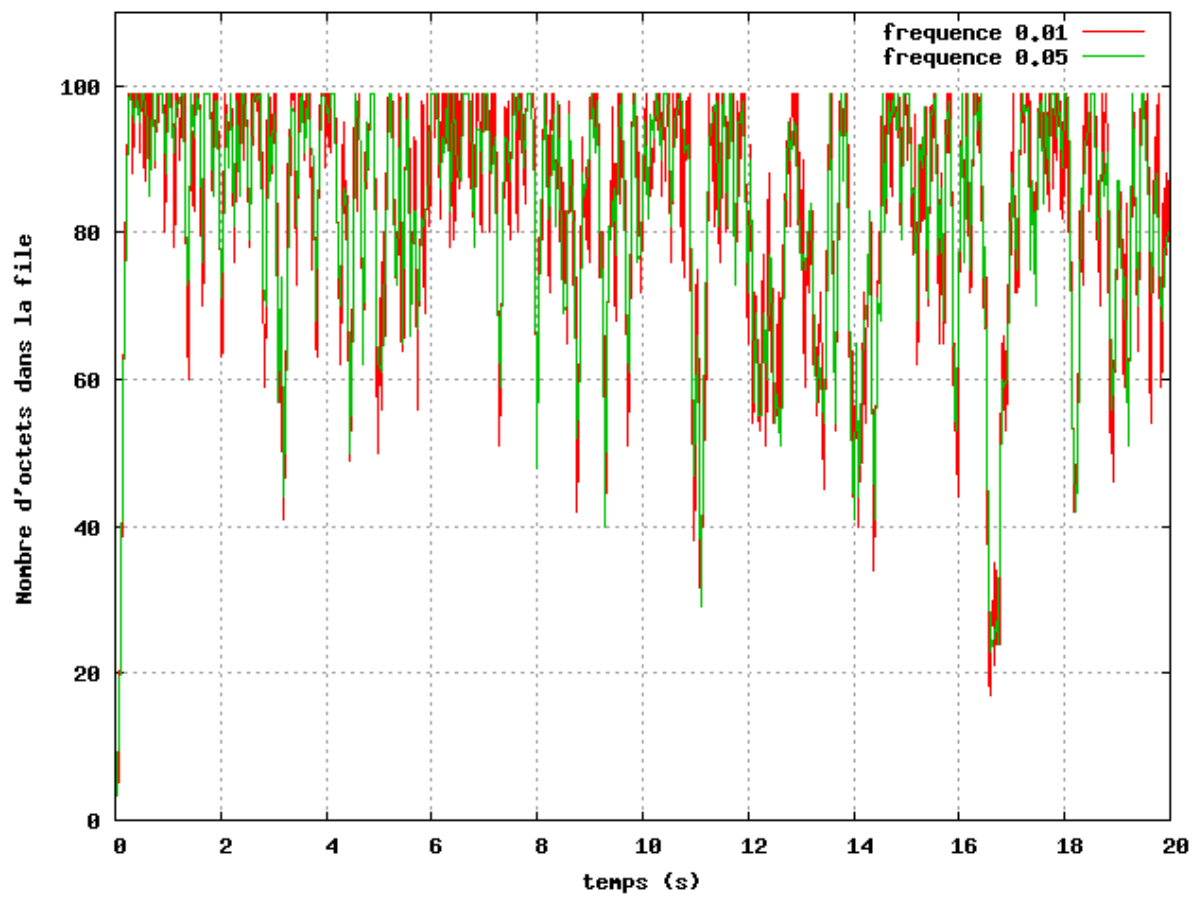


FIGURE 2 – Taille de la file d'attente