

QNAP

Boulic Guillaume, Émeric Tosi

21 mars 2016

# Sommaire

<b>1</b>	<b>RTC : Réseau Téléphonique Commuté</b>	<b>2</b>
1.1	Analyse sur un lien . . . . .	2
1.2	Analyse sur un réseau de trois commutateurs . . . . .	5
<b>2</b>	<b>Commutation de paquets</b>	<b>9</b>
2.1	Un commutateur de paquets . . . . .	9
<b>A</b>	<b>Annexes</b>	<b>17</b>
A.1	Exemple(s) . . . . .	17
A.2	Analyse sur un lien . . . . .	18
A.3	Analyse sur un réseau de trois commutateurs . . . . .	21
A.4	Un commutateur de paquets . . . . .	24

# Chapitre 1

## RTC : Réseau Téléphonique Commuté

### 1.1 Analyse sur un lien

#### 1.1.1 Énoncé

Considérons un lien d'un réseau à commutation de circuits permettant de véhiculer de la voix téléphonique.

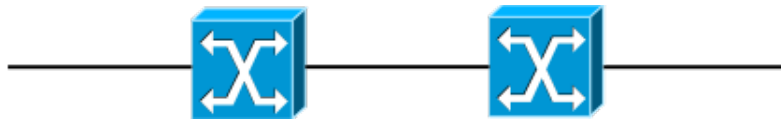


FIGURE 1.1 – Schéma du réseau à commutation de circuit étudié

Chacune des connexions nécessite un débit de  $64 \text{ Kb.s}^{-1}$  de façon bi-directionnel. On peut multiplexer simultanément  $C$  appels téléphoniques sur ce lien.

Le nombre d'utilisateurs est suffisamment grand pour supposer que les arrivées des nouveaux appels suivent une loi de paramètre  $\lambda$ , les durées des appels sont supposées suivre une loi exponentielle de paramètre  $\mu$  avec ( $\frac{1}{\mu} = 3$  minutes).

### 1.1.2 Probabilité de blocage d'appel en fonction de la charge $\rho$ et de la capacité $C$

$$P(\text{blocage}) = \frac{\frac{\rho^C}{C!}}{\sum_{i=0}^C \frac{\rho^i}{i!}}$$

avec  $\rho$  la charge et  $C$  la capacité.

Voici les résultats de ce calcul, obtenus par notre script Python inclus en annexe A.2

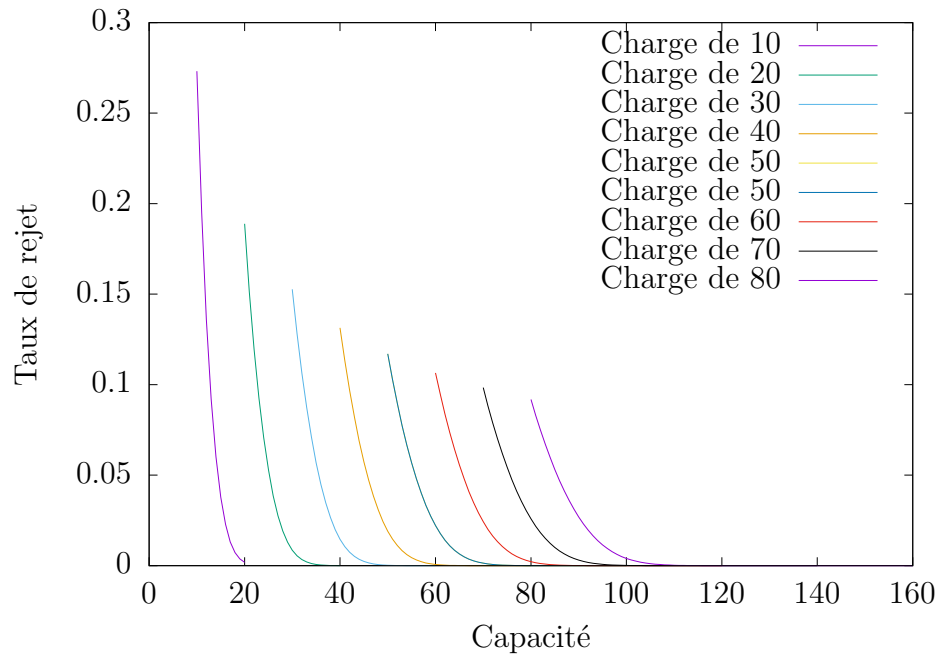


FIGURE 1.2 – Graphique des résultats de l'étude théorique.

### 1.1.3 Simulation de cette probabilité de blocage

Pour une charge  $10 < \rho < 70$  et pour une capacité  $\rho < C < 2 * \rho$ , on obtient les résultats suivants.

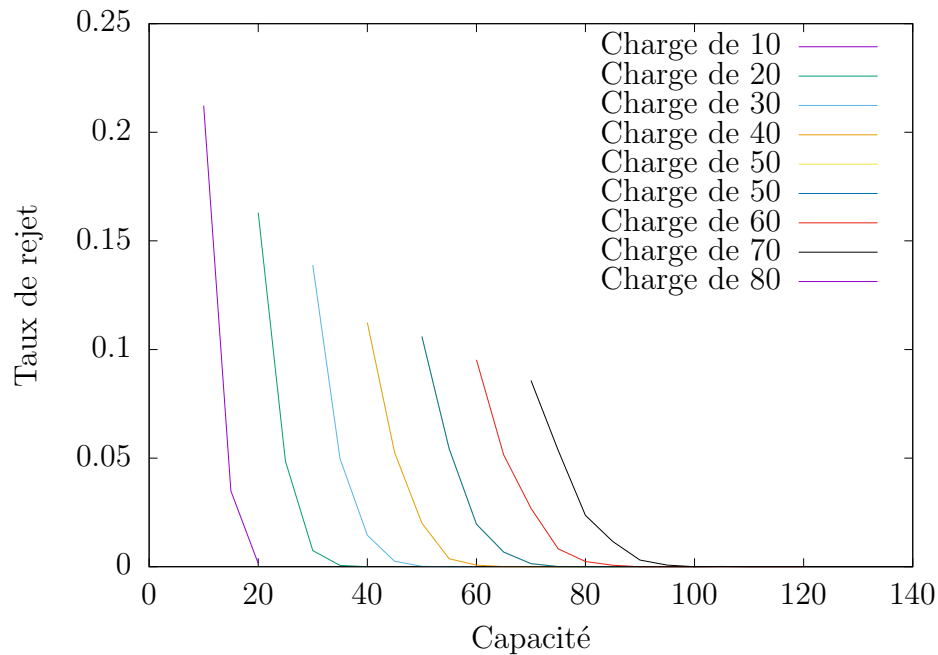


FIGURE 1.3 – Graphique des résultats de la simulation.

### 1.1.4 Comparaison des taux de blocage expérimentaux et théoriques

On constate que les résultats sont similaires, les erreurs de simulation sont gommées par le nombre de simulations important (cela réduit les écarts aléatoires) et leur longue durée (ce qui assure un régime stable).

## 1.2 Analyse sur un réseau de trois commutateurs

### 1.2.1 Énoncé

Désormais, nous considérons le réseau composé de 3 noeuds.

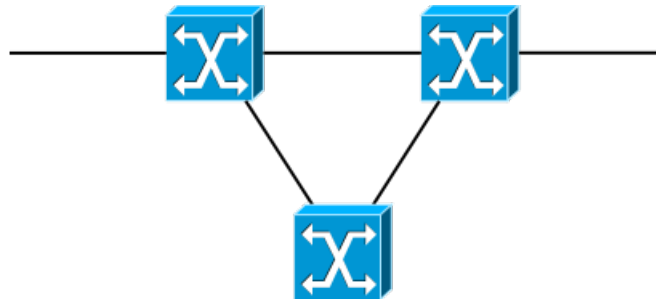


FIGURE 1.4 – Schéma du réseau à 3 commutateurs de circuit étudié

Les arrivées sont supposées Poissonniennes sur chacun des noeuds et le trafic se répartit équiprobablement entre les différents noeuds. Les durées des appels sont supposées exponentielles de même paramètre que dans la première partie (1-a). Nous ne considérons pas les appels locaux ni les appels qui n'aboutissent pas (absence).

### 1.2.2 Probabilités de blocage avec le chemin de débordement en cas de saturation du chemin direct

Chaque routeur est représenté par une source qui émet alternativement sur ces deux liens à destination des deux autres routeurs. Lorsqu'un lien est surchargé, un routeur tente d'utiliser le chemin de débordement qu'il a à disposition. On imagine déjà qu'en cas de forte charge un problème de saturation du système sera inévitable.

### 1.2.3 Comparaison des résultats avec la partie 1.1

La simulation se termine brutalement avant d'atteindre une charge de 40, le réseau sature totalement.

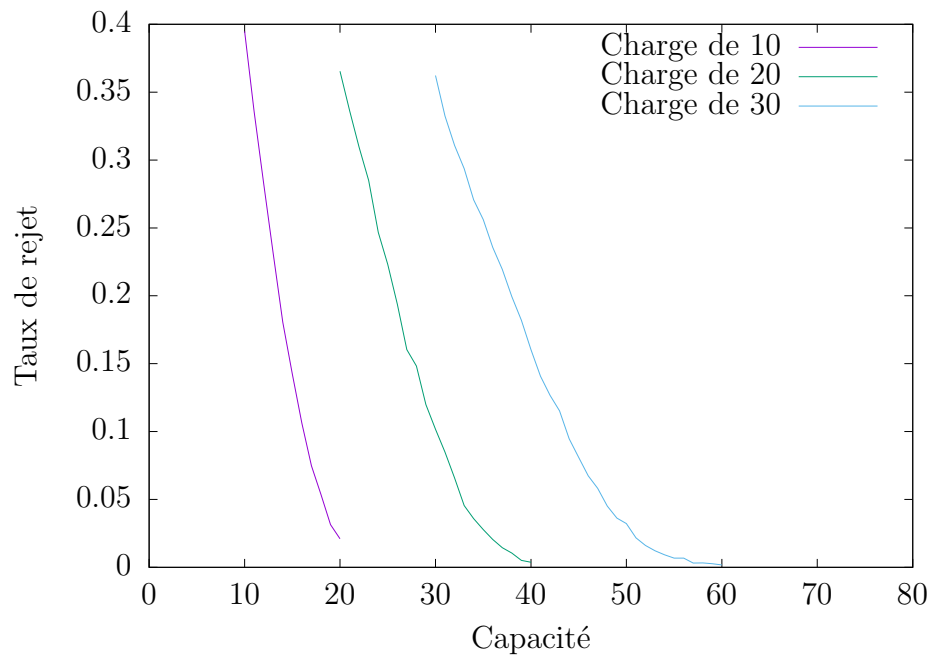


FIGURE 1.5 – Graphique des résultats de la simulation.

### 1.2.4 Problèmes à très forte charge !

Une solution consiste à n'utiliser le chemin de débordement que lorsque celui-ci n'est pas très encombré, c'est à dire en dessous d'un certain seuil d'occupation sur chacun des liens. Cela revient donc à laisser une marge  $M$  aux appels directs.

#### Commentaires

On imagine que cela évite de bloquer les stations en débordant sur elles puisqu'elles auront chacune une marge de sécurité pour transmettre les appels directs.

#### Simulation en prenant une marge comprise entre 1 et 3

Plus on augmente la marge, moins l'effet d'auto-saturation provoqué par les débordements est présent, cependant il est normal de constater une légère augmentation du taux de rejet : la marge n'est pas toujours totalement utilisée (l'augmenter ne fera qu'accroître ce phénomène).

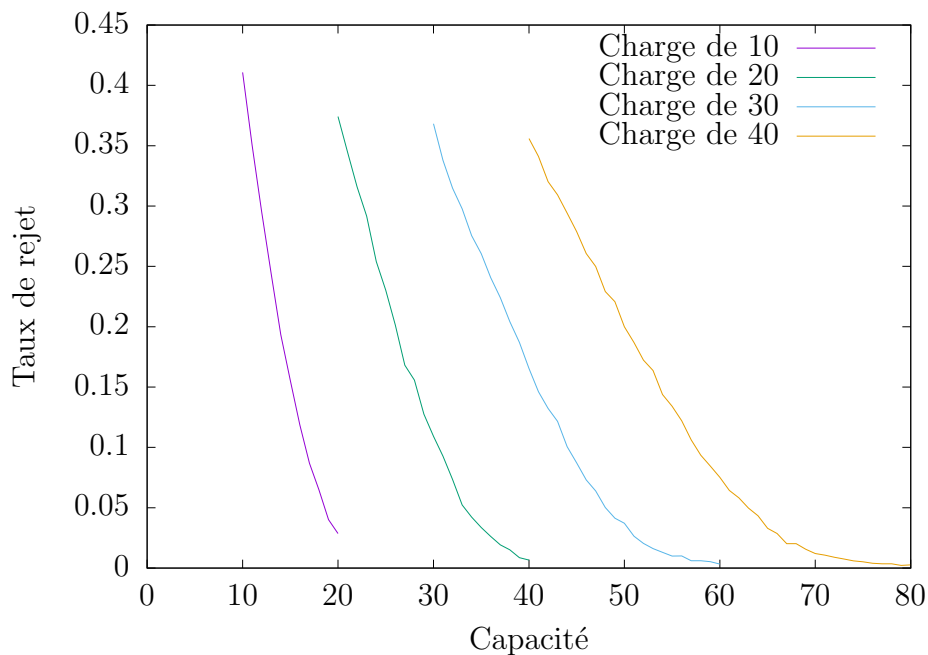


FIGURE 1.6 – Résultats de la simulation pour une Marge de 1.



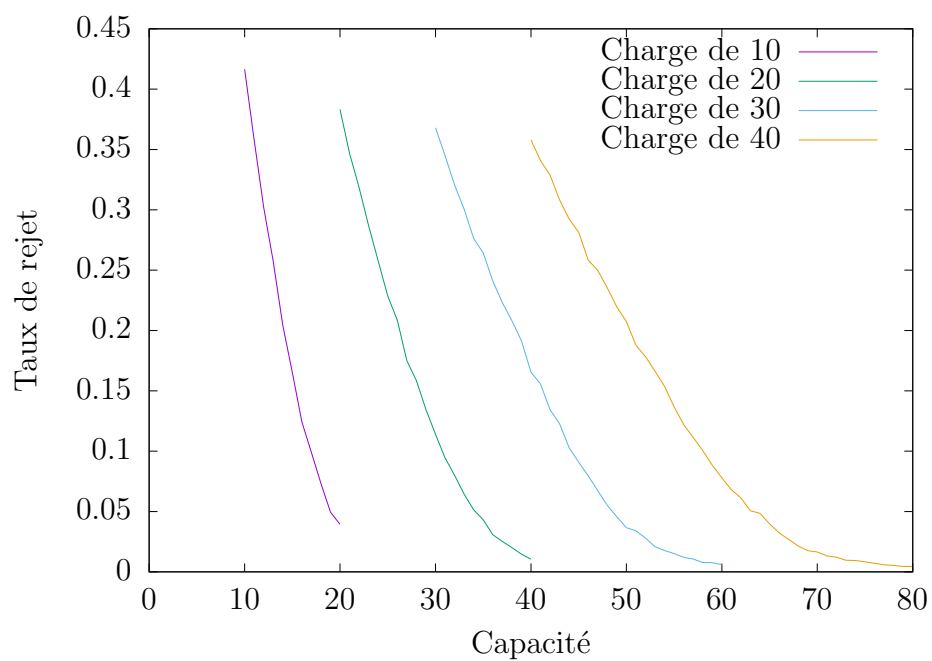


FIGURE 1.7 – Résultats de la simulation pour une Marge de 2.

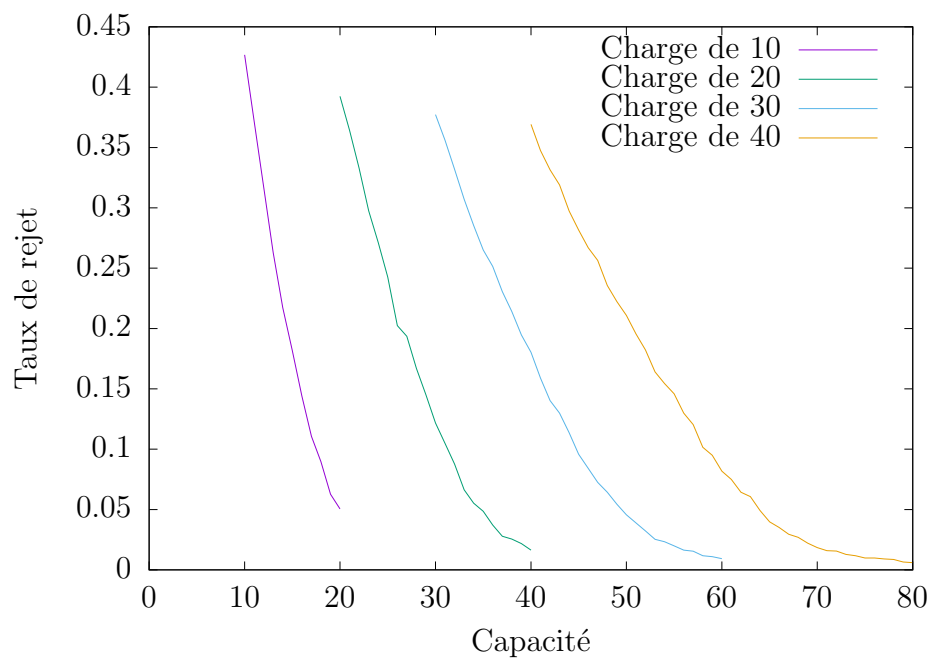


FIGURE 1.8 – Résultats de la simulation pour une Marge de 3.

# Chapitre 2

## Commutation de paquets

### 2.1 Un commutateur de paquets

#### 2.1.1 Énoncé

Nous cherchons à simuler un lien de sortie d'un commutateur de paquets.



FIGURE 2.1 – Schéma du système à commutateur de paquets étudié

L'arrivée des paquets est supposée suivre une loi exponentielle de paramètre  $\lambda$ . Nous positionnons une file en sortie du commutateur pour stocker les différents paquets. Les paquets ont une longueur exponentiellement distribuée de paramètre  $\frac{1}{\nu} = 10$  Kb. Le lien de sortie a un débit de  $10 \text{ Mb.s}^{-1}$ .

#### 2.1.2 Calcul analytique du temps moyen de service $\frac{1}{\mu}$

On obtient le même temps moyen de service suivant :

$$\begin{aligned}\text{Temps moyen de service} &= \frac{1}{\mu} \\ \iff \frac{1}{\nu} * \frac{1}{D} &= 10 * 10^3 * \frac{1}{10^7} \\ \iff 10^4 * 10^{-7} &= 10^{-3} \text{ seconde}\end{aligned}$$

### 2.1.3 Déterminer le nombre moyen de paquets dans la file et le temps moyen de réponse en fonction du taux d'arrivée pour différentes durées de simulation

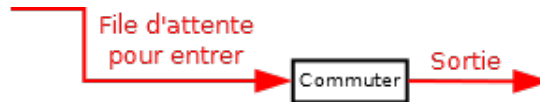


FIGURE 2.2 – Schéma de fonctionnement d'un commutateur de paquets

$$\lambda = \rho * \mu$$

$$\text{Charge de trafic } \rho = \frac{\lambda}{\mu}$$

$$\text{Nombre moyen de client(s) en file d'attente } \bar{N} = \frac{\rho}{(1 - \rho)}$$

$$\text{Temps moyen de réponse } \bar{W} = \frac{1}{(\mu - \lambda)}$$

$$\bar{N} = \lambda \bar{W}$$

Quelques résultats sous forme de tableau :

$\rho$	0.1	0.5	0.9
$\lambda$	$10^2$	$5 * 10^2$	$9 * 10^2$
Nombre moyen de client(s) en file d'attente	0.11111	1	9
Temps moyen de réponse [s]	0.00111	0.002	0.01

### 2.1.4 Comparaison du résultat de la simulation avec la théorie

Blablabla 1

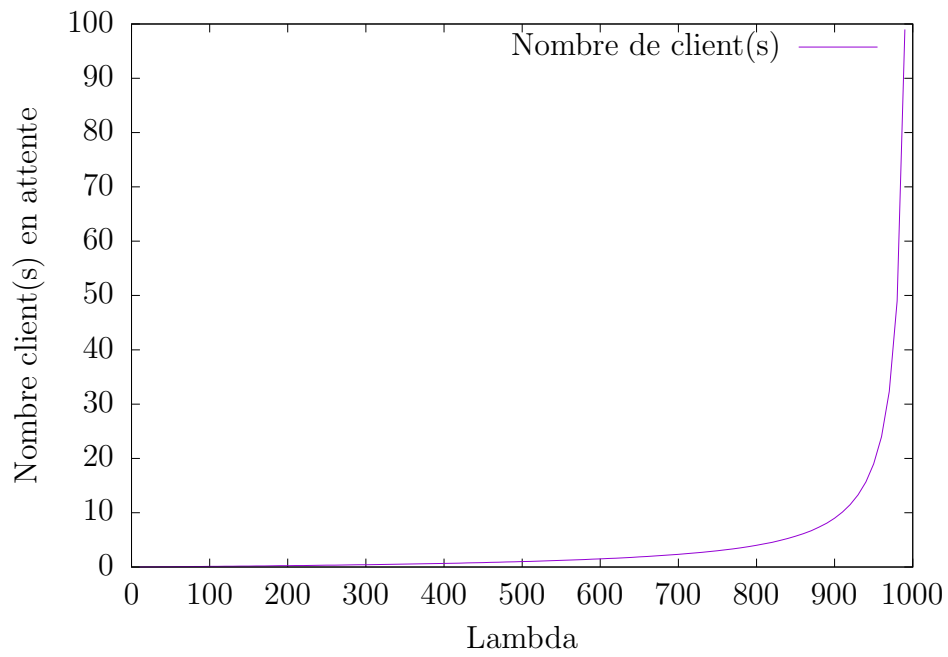


FIGURE 2.3 – Résultats du calcul Théorique.

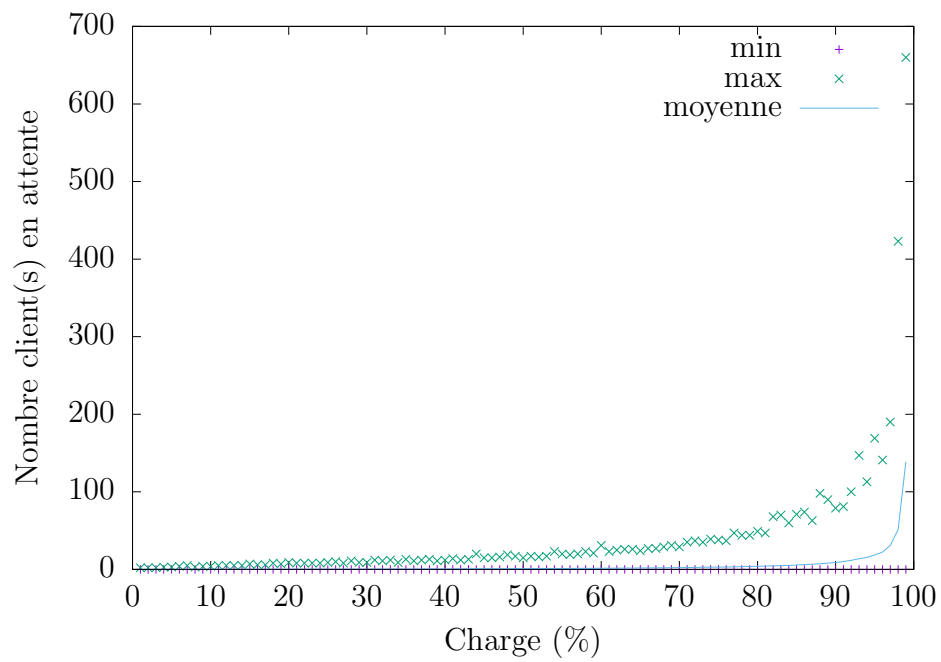


FIGURE 2.4 – Résultats de la simulation.

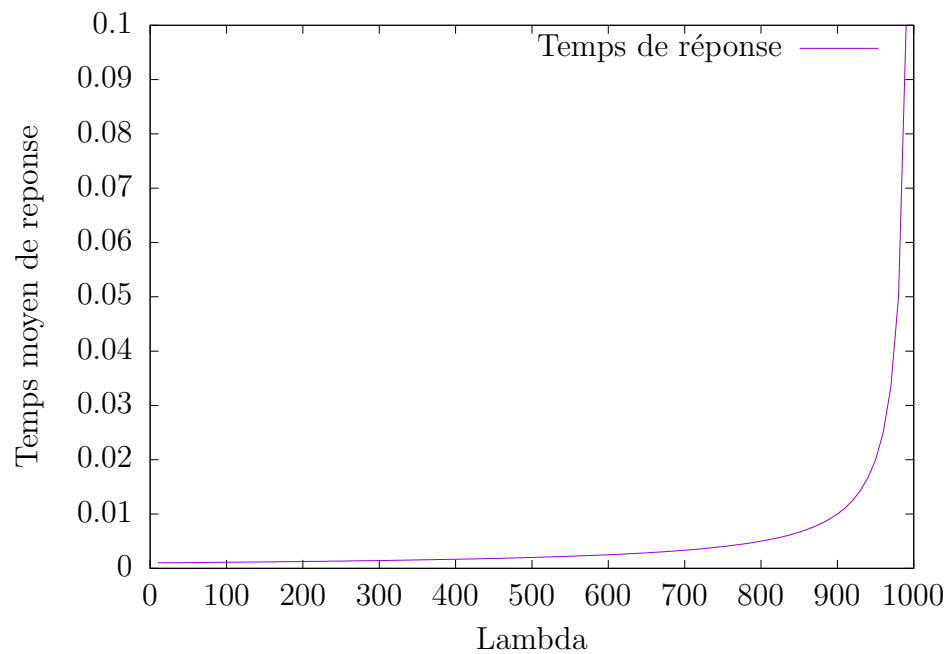


FIGURE 2.5 – Résultats du calcul Théorique.

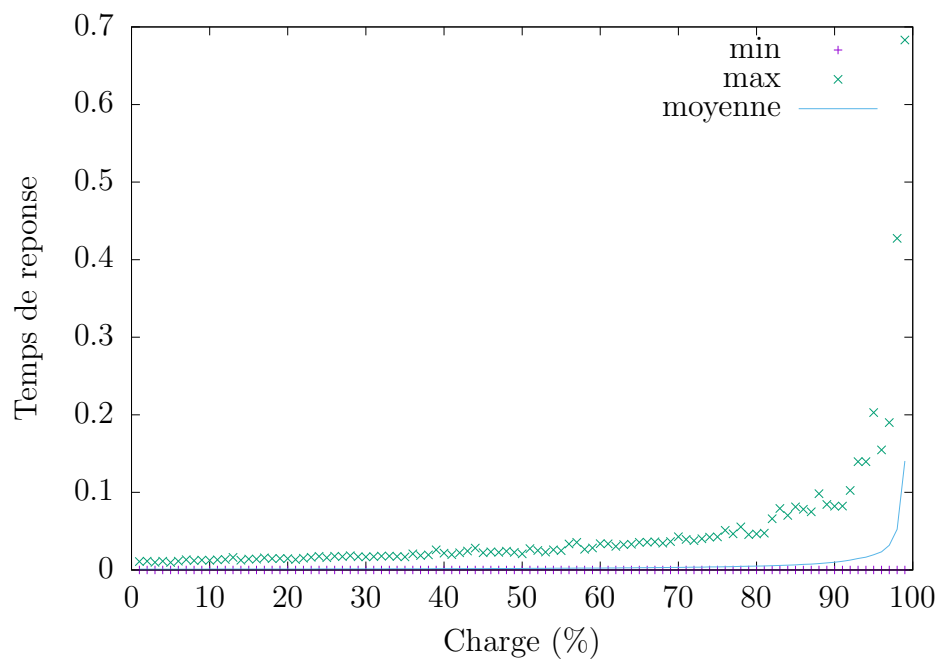


FIGURE 2.6 – Résultats de la simulation.

### 2.1.5 Cas où les paquets ont une taille fixe de 10 Kb

Calculer analytiquement le temps moyen de service  $\frac{1}{\mu}$

On obtient le même temps moyen de service que précédemment :

$$\text{Temps moyen de service} = \frac{1}{\mu}$$

$$\Longleftrightarrow \frac{1}{\nu} * \frac{1}{D} = 10 * 10^3 * \frac{1}{10^7}$$

$$\Longleftrightarrow 10^4 * 10^{-7} = 10^{-3} \text{ seconde}$$

### Résultats en fonction du taux d'arrivée pour différentes durées de simulation

Les résultats sont divisés par deux puisque l'exponentielle n'est plus là, on obtient une file de type M/D/1. Les paquets sont tous de même taille, ils s'agencent parfaitement bien dans le processus de commutation.

Temps moyen de réponse et nombre moyen de paquets dans la file d'attente :

$$\lambda = \rho * \mu$$

$$\text{Charge de trafic } \rho = \frac{\lambda}{\mu}$$

$$\text{Nombre moyen de client } \bar{N} = \frac{\rho}{2 * (1 - \rho)}$$

$$\text{Temps moyen de reponse } \bar{W} = \frac{1}{2 * (\mu - \lambda)}$$

$$\bar{N} = \lambda \bar{W}$$

### Analyse et comparaison des résultats

Blablabla 2

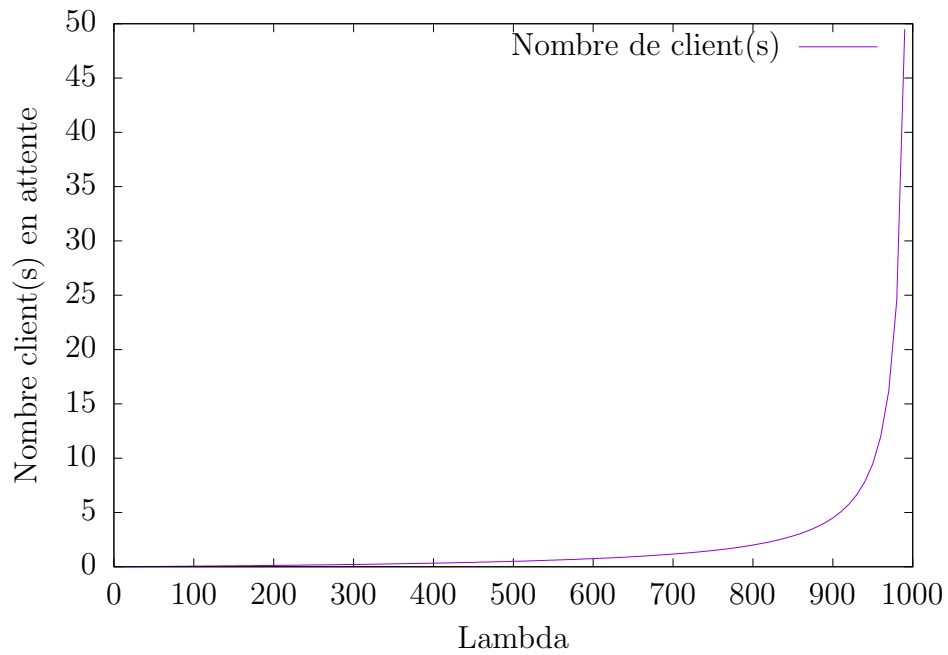


FIGURE 2.7 – Résultats du calcul Théorique.

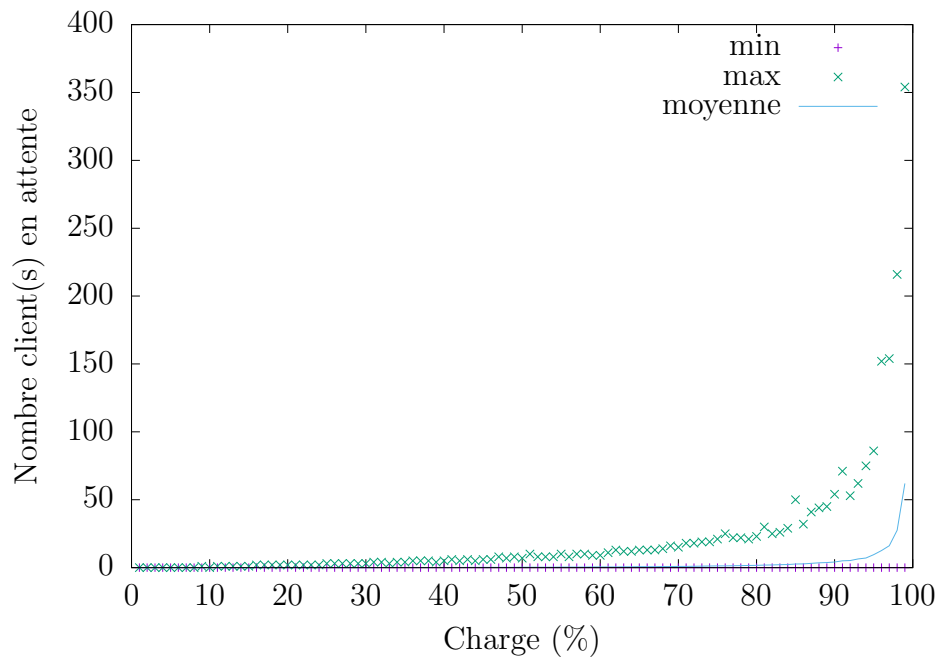


FIGURE 2.8 – Résultats de la simulation pour des paquets à la taille fixe de 10 Kb.



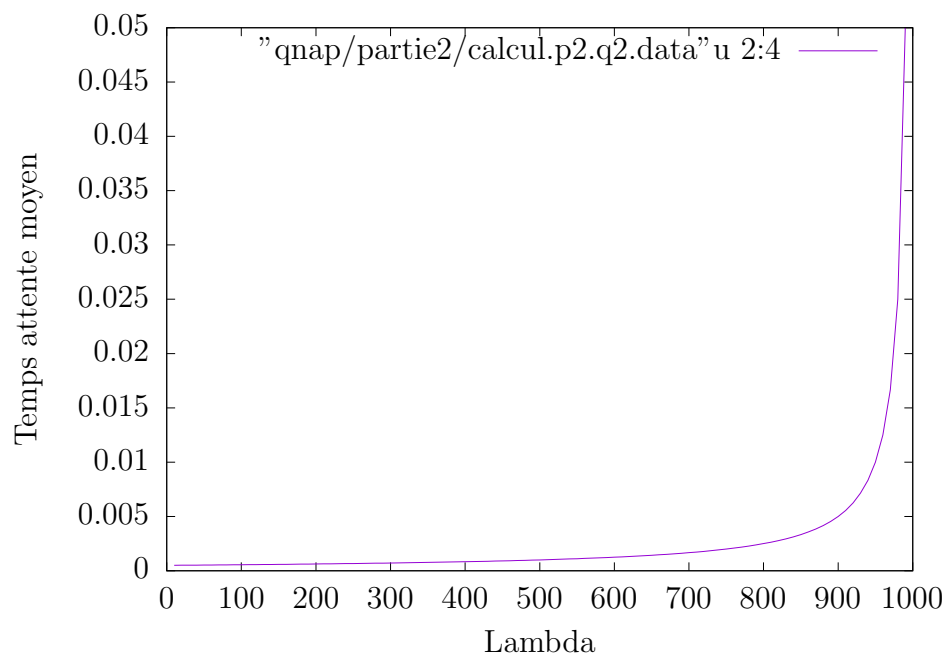


FIGURE 2.9 – Résultats du calcul Théorique.

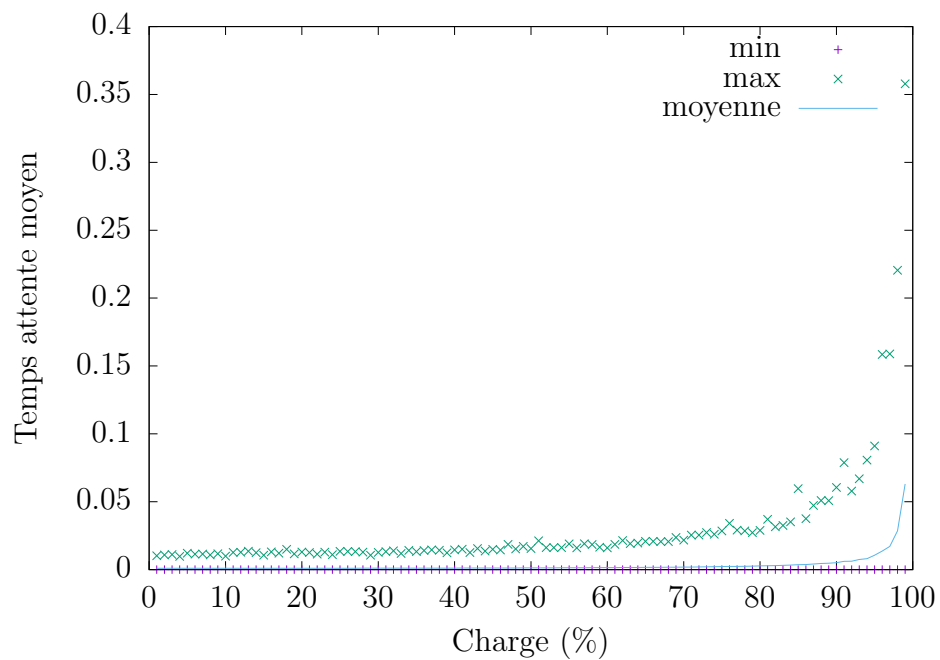


FIGURE 2.10 – Résultats de la simulation pour des paquets à la taille fixe de 10 Kb.

# Annexe A

## Annexes

### A.1 Exemple(s)

*emphatique* **gras** machine à écrire *incliné* PETITES MAJUSCULES

The foundations of the rigorous study of *analysis* were laid in the nineteenth century, notably by the mathematicians Cauchy and Weierstrass. Central to the study of this subject are the formal definitions of *limits* and *continuity*.

Let  $D$  be a subset of  $\mathbf{R}$  and let  $f: D \rightarrow \mathbf{R}$  be a real-valued function on  $D$ . The function  $f$  is said to be *continuous* on  $D$  if, for all  $\epsilon > 0$  and for all  $x \in D$ , there exists some  $\delta > 0$  (which may depend on  $x$ ) such that if  $y \in D$  satisfies

$$|y - x| < \delta$$

then

$$|f(y) - f(x)| < \epsilon.$$

One may readily verify that if  $f$  and  $g$  are continuous functions on  $D$  then the functions  $f+g$ ,  $f-g$  and  $f.g$  are continuous. If in addition  $g$  is everywhere non-zero then  $f/g$  is continuous.

## A.2 Analyse sur un lien

Script Python de calcul de la probabilité de blocage d'appel  $P$  en fonction de la charge  $\rho$  et de la capacité  $C$ .

```
1  # -*- coding: utf-8 -*-
2
3  # module maths
4  import math
5
6  # boucle de variation de la charge
7  for p in range(1, 80+1):
8
9      # boucle de variation de la capacite
10     for c in range(p, p*2+1):
11
12         # calcul du numerateur
13         numerateur = pow(p, c) * 1.0 / math.factorial(c)
14
15         # calcul du denominateur
16         denominateur = 0
17         for i in range(0, c):
18             denominateur += pow(p, i) * 1.0 / math.factorial(i)
19
20         # ecriture des resultats sur la sortie standard
21         print(str(c) + " " + str(p) + " " + str(numerateur /
                denominateur))
```

Simulation d'un lien d'un réseau à commutation de circuits.

```
1 &
2 /CONTROL/
3     OPTION = NSOURCE;
4     ACCURACY = ALL QUEUE;
5     TMAX = 100000;
6 &
7 /DECLARE/
8     QUEUE SRC, LIEN;
9     INTEGER NbRejets, Erlang, I, C, RHO, NBSIMU;
10    REAL NbOK, DEBIT, LAMBDA, MU, REJET, X;
11    FILE OUTFILE;
12 &
13 /STATION/
14     NAME = SRC;
15     TYPE = SOURCE;
16     SERVICE = EXP(1./LAMBDA);
17     TRANSIT = LIEN;
18 &
19 /STATION/
20     NAME = LIEN;
21     TYPE = MULTIPLE(C);
22     SERVICE = BEGIN
23         EXP(1./(MU));
24         NbOK := NbOK + 1;
25     END;
26     CAPACITY = C;
27     REJECT = BEGIN
28         NbRejets := NbRejets + 1;
29         TRANSIT(OUT);
30     END;
31     TRANSIT = OUT;
32 &
33 /EXEC/
34     BEGIN
35         & output data file
36         FILASSIGN(OUTFILE , "p1.q2.data");
37         OPEN(OUTFILE);
38         & var init
39         DEBIT := 64000;
40         MU := 1. / 180;
41         Erlang := 70;
42         OUTFILE := 5;
```

```

43      & process simulations
44      FOR RHO := 10 STEP 5 UNTIL Erlang DO
45          BEGIN
46              LAMBDA := MU * RHO;
47              FOR C := RHO STEP 5 UNTIL (RHO * 2) DO
48                  BEGIN
49                      NbOK := 0;
50                      NbRejets := 0;
51                      FOR I := 1 STEP 1 UNTIL NBSIMU DO
52                          BEGIN
53                              SIMUL;
54                              END;
55                              NbOK := NbOK / NBSIMU;
56                              NbRejets := NbRejets / NBSIMU;
57                              WRITELN(OUTFILE , " " , RHO , " " , C , " " ,
                                  NbRejets , " " , NbOK , " " , NbRejets / (
                                  NbOK + NbRejets));
58                      END;
59              END;
60      END;
61 /END/

```

## A.3 Analyse sur un réseau de trois commutateurs

Simulation d'un lien d'un réseau à 3 commutateurs de circuits.

```
1  &
2  /CONTROL/
3      OPTION = NSOURCE;
4      TMAX = 100000;
5      ACCURACY = ALL QUEUE;
6  &
7  /DECLARE/
8      QUEUE SRC, LIEN_H, LIEN_B_1, LIEN_B_2;
9      INTEGER TOTAL, TOTALTX, TOTALRJ, LOAD, I, J, C, RHO, NBSIMU;
10     REAL DEBIT, LAMBDA, MU, PRCT;
11     CUSTOMER REAL TIME;
12     FILE OUTFILE;
13 &
14 /STATION/
15     NAME = SRC;
16     TYPE = SOURCE;
17     SERVICE = BEGIN
18         TIME := ( 1. / MU );
19         TOTAL := TOTAL + 1;
20         EXP( 1. / LAMBDA );
21     END;
22     TRANSIT = LIEN_H;
23 &
24 /STATION/
25     NAME = LIEN_H;
26     TYPE = MULTIPLE(C);
27     SERVICE = BEGIN
28         CST(TIME);
29         TRANSIT(OUT);
30     END;
31     CAPACITY = C;
32     REJECT = BEGIN
33         TOTALRJ := TOTALRJ + 1;
34         TRANSIT(LIEN_B_1);
35     END;
36 &
37 /STATION/
38     NAME = LIEN_B_1;
```

```

39     TYPE = MULTIPLE(C);
40     SERVICE = BEGIN
41         CST(TIME);
42         TRANSIT(LIEN_B_2);
43     END;
44     CAPACITY = C;
45     REJECT = BEGIN
46         TOTALRJ := TOTALRJ + 1;
47         TRANSIT(OUT);
48     END;
49 &
50 /STATION/
51     NAME = LIEN_B_2;
52     TYPE = MULTIPLE(C);
53     SERVICE = BEGIN
54         CST(TIME);
55         TRANSIT(OUT);
56     END;
57     CAPACITY = C;
58     REJECT = BEGIN
59         TOTALRJ := TOTALRJ + 1;
60         TRANSIT(OUT);
61     END;
62 &
63 /EXEC/
64     BEGIN
65         & output data file
66         FILASSIGN(OUTFILE , "p1.q3.data");
67         OPEN(OUTFILE);
68         & var init
69         NBSIMU := 5;
70         DEBIT := 64000;
71         MU := 1. / 180;
72         LOAD := 70;
73         & process simulations
74         FOR RHO := 1 STEP 1 UNTIL LOAD DO
75             BEGIN
76                 FOR C := RHO STEP 1 UNTIL 2 * RHO DO
77                     BEGIN
78                         LAMBDA := MU * RHO;
79                         TOTAL := 0;
80                         TOTALRJ := 0;
81                         TOTALTX := 0;

```

```

82         FOR I := 1 STEP 1 UNTIL NBSIMU DO
83             BEGIN
84                 SIMUL;
85             END;
86             TOTAL := TOTAL / NBSIMU;
87             TOTALRJ := TOTALRJ / NBSIMU;
88             TOTALTX := TOTALTX / NBSIMU;
89             PRCT := TOTALRJ * 1. / TOTAL;
90             WRITELN(OUTFILE, RHO , " " , C , " " , TOTAL ,
                    " " , TOTALRJ , " " , PRCT );
91         END;
92     END;
93 END;
94 /END/

```



## A.4 Un commutateur de paquets

Simulation d'un lien de sortie d'un commutateur de paquets.

```
1
2 &
3 /DECLARE/
4
5 QUEUE SRC, Q;
6
7 REAL LAMBDA, RO;
8 REAL MU = 1000;
9 INTEGER I;
10
11 QUEUE fakeOUT;
12
13 REAL temps;
14
15 REAL minA;
16 REAL maxA;
17 REAL sumA;
18
19 REAL minT;
20 REAL maxT;
21 REAL sumT;
22
23 CUSTOMER INTEGER nbpsg;
24 CUSTOMER REAL Tdebut;
25
26 FILE cbQ;
27
28
29 &
30 /STATION/
31
32 NAME = SRC;
33 TYPE = SOURCE;
34 SERVICE = BEGIN
35     EXP(1. / LAMBDA);
36     Tdebut := TIME;
37 END;
38 TRANSIT = Q;
39
40
```

```

41  &
42  /STATION/
43
44  NAME = Q;
45  SERVICE = BEGIN
46      EXP(1. / MU);
47  END;
48  TRANSIT = fakeOUT;
49
50
51  &
52  /STATION/
53
54  NAME = fakeOUT;
55  SERVICE = BEGIN
56      temps := (TIME - Tdebut);
57
58      sumT := sumT + temps;
59
60      IF minT > temps THEN
61      BEGIN
62          minT := temps;
63      END;
64
65      IF maxT < temps THEN
66      BEGIN
67          maxT := temps;
68      END;
69
70      sumA := sumA + CUSTNB(Q);
71
72      IF minA > CUSTNB(Q) THEN
73      BEGIN
74          minA := CUSTNB(Q);
75      END;
76
77      IF maxA < CUSTNB(Q) THEN
78      BEGIN
79          maxA := CUSTNB(Q);
80      END;
81  END;
82  TRANSIT = OUT;
83

```

```

84
85 &
86 /CONTROL/
87
88 TMAX = 1000;
89
90
91 &
92 /EXEC/
93
94 BEGIN
95     & On fait varier Rho pour qu'il prenne 99 valeurs definies et qu
          'on realise nos mesures (0.01 , ... , 0.99)
96     FILASSIGN(cbQ,"p2.q3.data");
97     OPEN(cbQ);
98     & boucle des simulations
99     FOR I := 1 STEP 1 UNTIL 99 DO
100     BEGIN
101         & initialisation des variables
102         RO := I * 0.010;
103         LAMBDA := RO * MU;
104         minA := 10000000;
105         maxA := 0;
106         sumA := 0;
107         minT := 10000000;
108         maxT := 0;
109         sumT := 0;
110         & simulation
111         SIMUL;
112         & affichage de la progression
113         WRITELN(I);
114         & ecriture resultats dans le fichier
115         WRITELN(cbQ,I," ",minT," ",maxT," ",sumT/SERVNB(fakeOUT)," "
          ,minA," ",maxA," ",sumA/SERVNB(fakeOUT));
116     END;
117 END;
118
119
120 &
121 /END/
122
123
124 &

```

```
125 & GNUPLOT
126 &
127 & plot "p2.q3.data" u 1:2 t "minT", "p2.q3.data" u 1:3 t "maxT", "
    p2.q3.data" u 1:4 w l t "moyenne temps traitement", "p2.q3.data"
    u 1:5 t "minA" axes x1y2, "p2.q3.data" u 1:6 t "maxA" axes x1y2
    , "p2.q3.data" u 1:7 w l t "mayyenne nb packet att" axes x1y2
128 &
```