

QNAP

Boulic Guillaume, Émeric Tosi

22 mars 2016

Sommaire

1	RTC : Réseau Téléphonique Commuté	2
1.1	Analyse sur un lien	2
1.2	Analyse sur un réseau de trois commutateurs	5
2	Commutation de paquets	9
2.1	Un commutateur de paquets	9
A	Annexes	17
A.1	Analyse sur un lien	17
A.2	Analyse sur un réseau de trois commutateurs	20
A.3	Un commutateur de paquets	29

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.

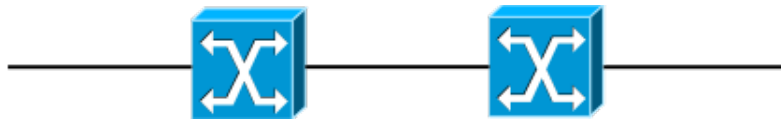


FIGURE 1.1 – Schéma du réseau à commutation de circuit étudié

Chacune des connexions nécessite un débit de 64 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 λ , les durées des appels sont supposées suivre une loi exponentielle de paramètre μ avec ($\frac{1}{\mu} = 3$ minutes).

1.1.2 Probabilité de blocage d'appel en fonction de la charge ρ et de la capacité C

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

avec ρ la charge et C la capacité.

Voici les résultats de ce calcul, obtenus par notre script Python inclus en annexe A.1

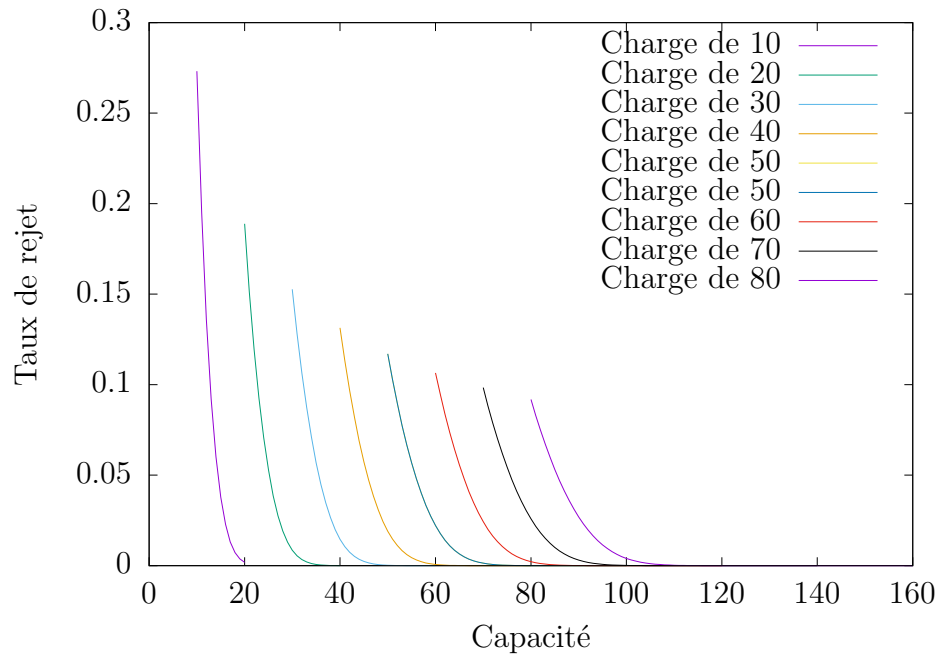


FIGURE 1.2 – Graphique des résultats de l'étude théorique.

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.

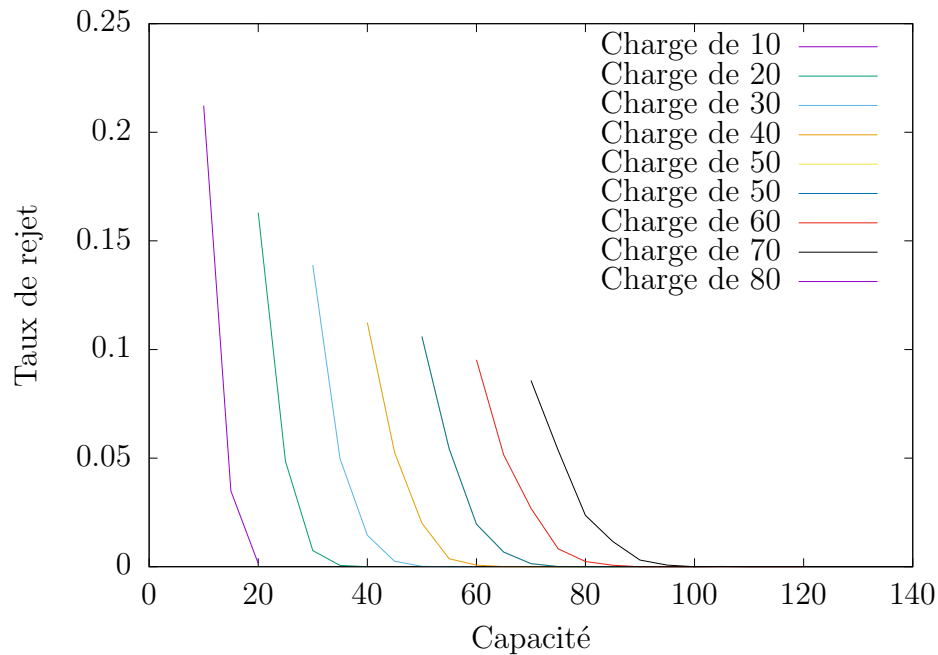


FIGURE 1.3 – Graphique des résultats de la simulation.

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.

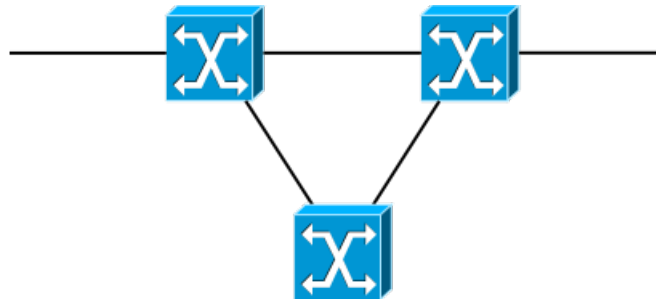


FIGURE 1.4 – Schéma du réseau à 3 commutateurs de circuit étudié

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. Toutefois il est parfois, sur un coup de chance pendant la simulation, possible de s'en rapprocher mais le simulateur n'arrive pas à terminer la simulation par manque de mémoire (limitation de ce vieux programme sûrement compilé en 16 bits).

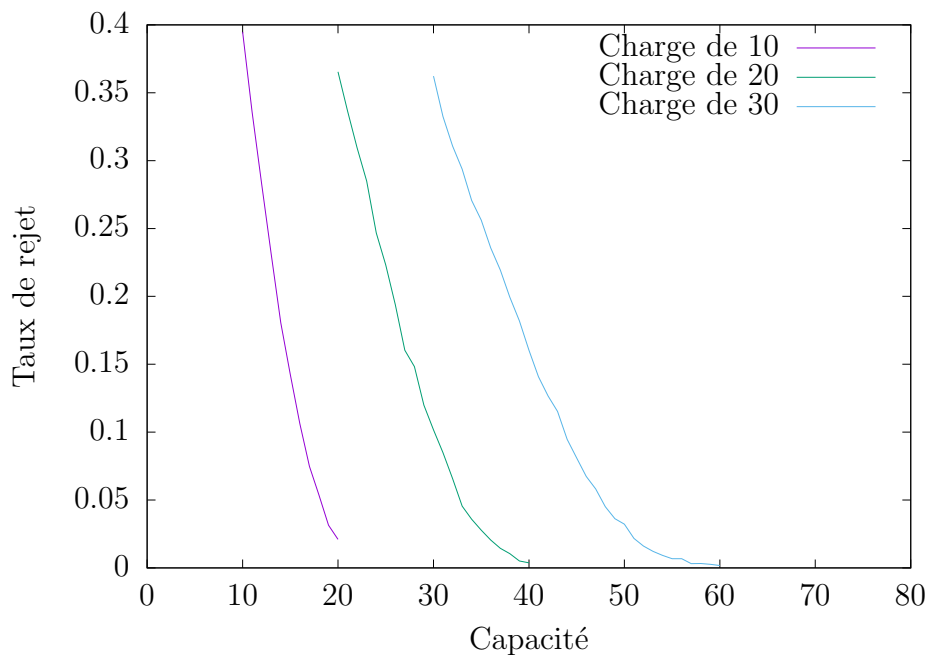


FIGURE 1.5 – Graphique des résultats de la simulation.

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).

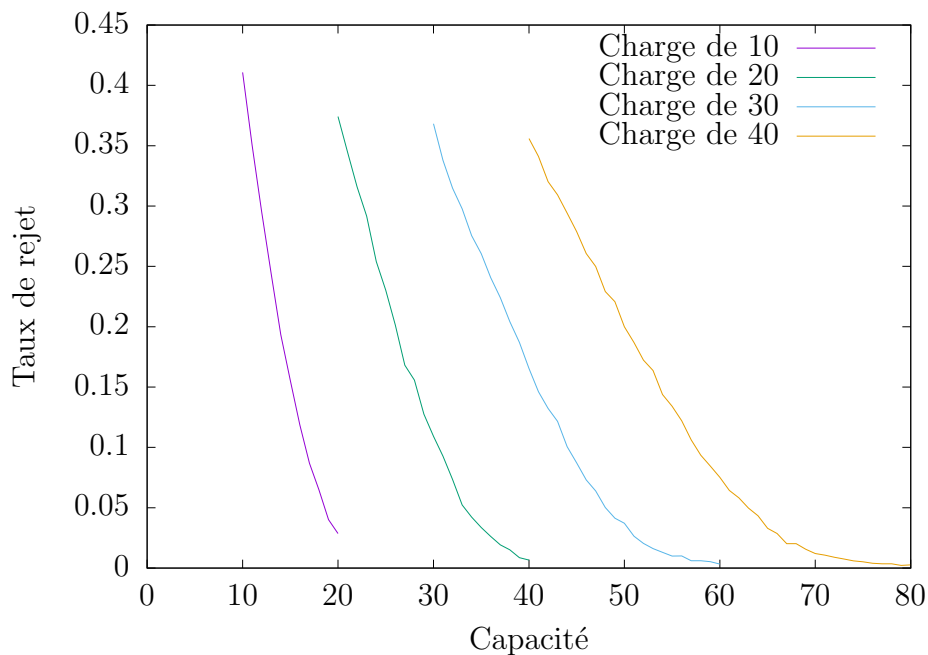


FIGURE 1.6 – Résultats de la simulation pour une Marge de 1.

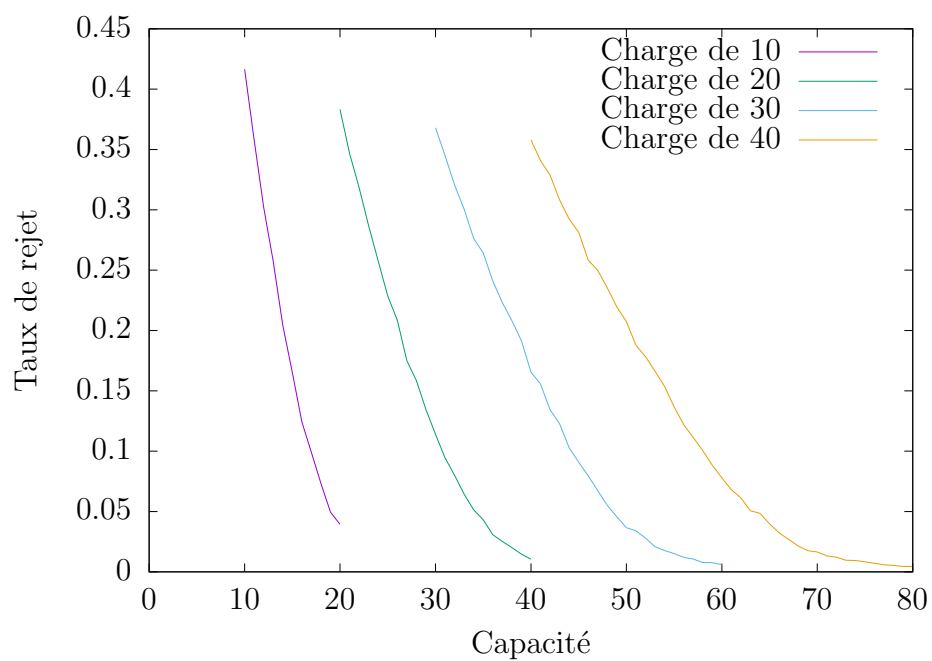


FIGURE 1.7 – Résultats de la simulation pour une Marge de 2.

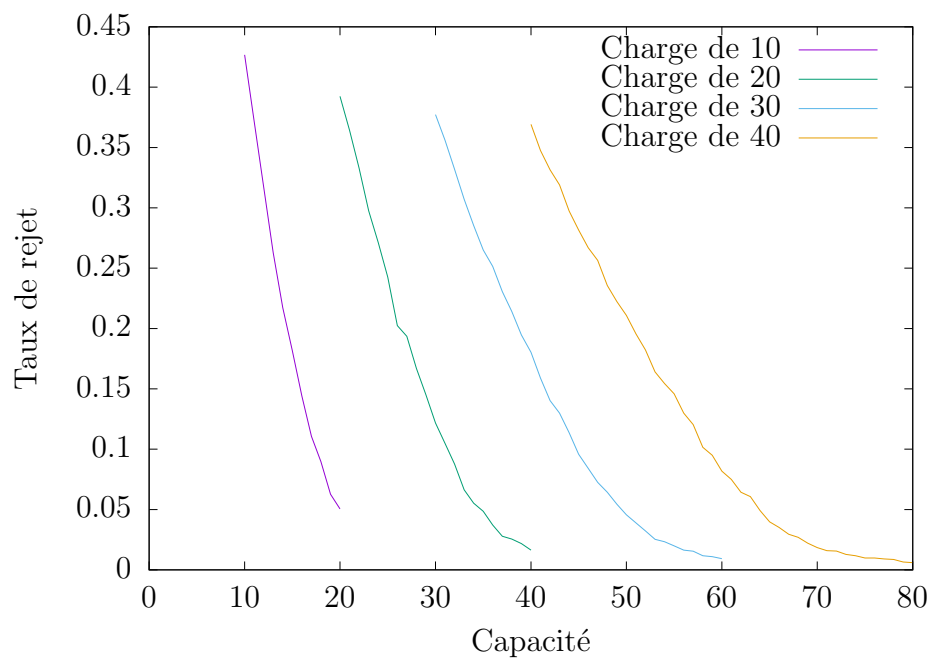


FIGURE 1.8 – Résultats de la simulation pour une Marge de 3.

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 λ . 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 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

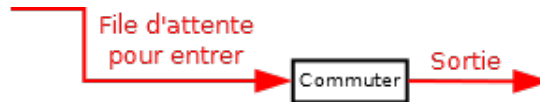


FIGURE 2.2 – Schéma de fonctionnement d'un commutateur de paquets

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

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

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

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

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

Quelques résultats sous forme de tableau :

ρ	0.1	0.5	0.9
λ	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

Lambda représente la charge multipliée par 10. Les résultats sont similaires, on observe toutefois les maximums des simulations qui atteignent de grandes valeurs par rapport aux moyennes.

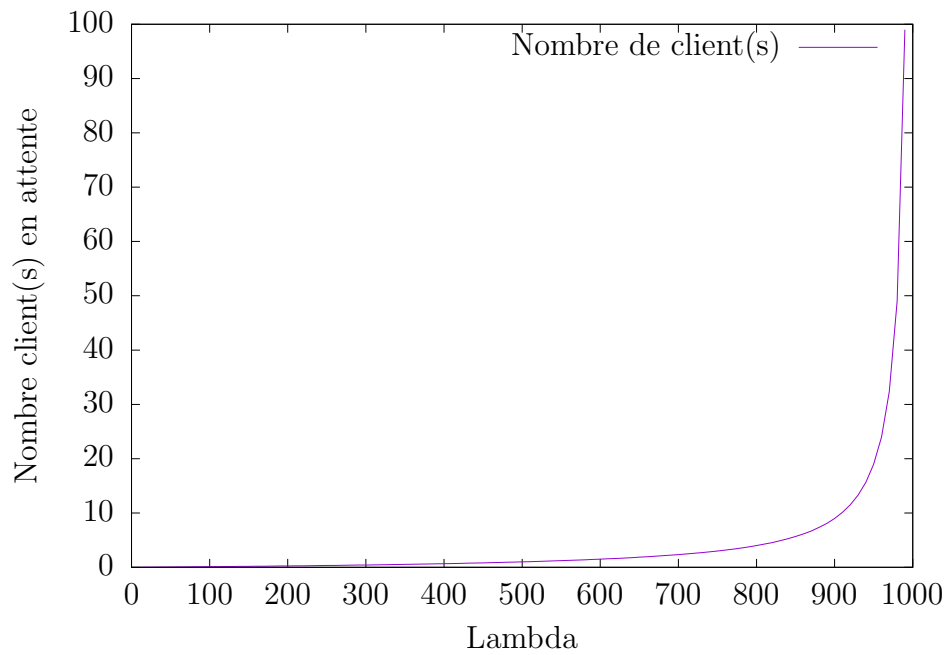


FIGURE 2.3 – Résultats du calcul Théorique.

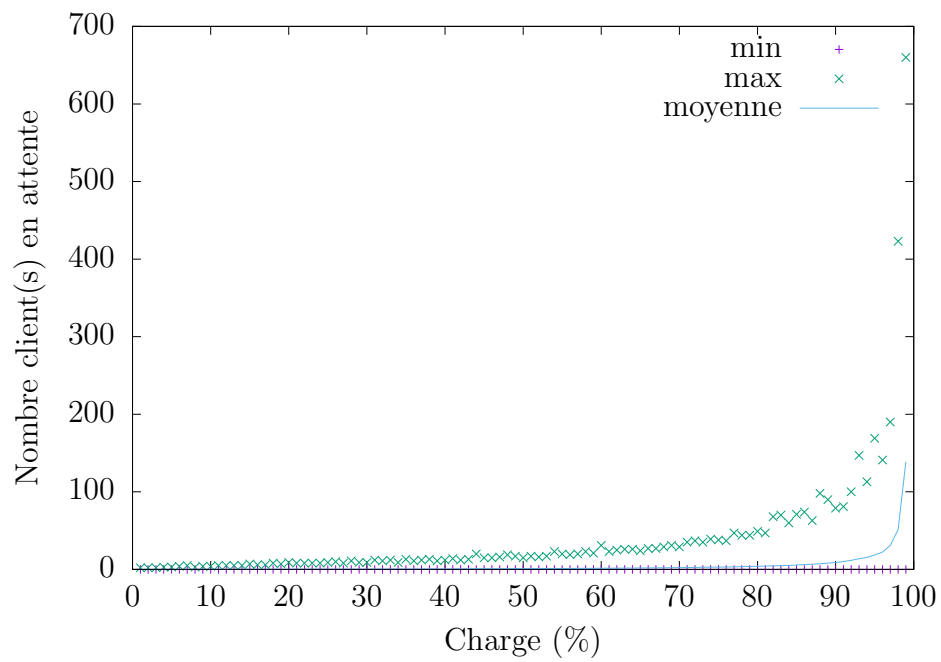


FIGURE 2.4 – Résultats de la simulation.

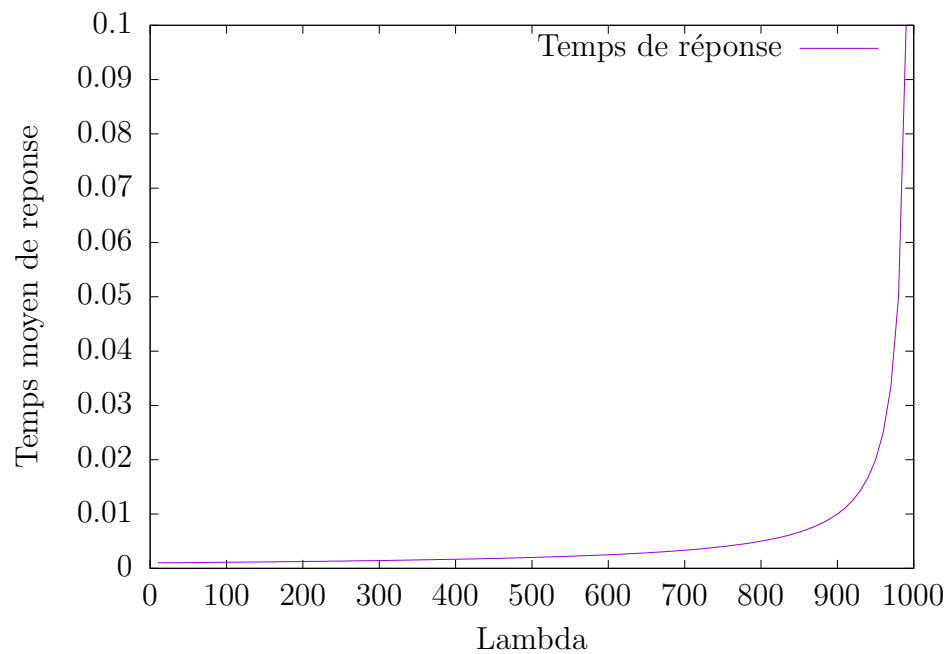


FIGURE 2.5 – Résultats du calcul Théorique.

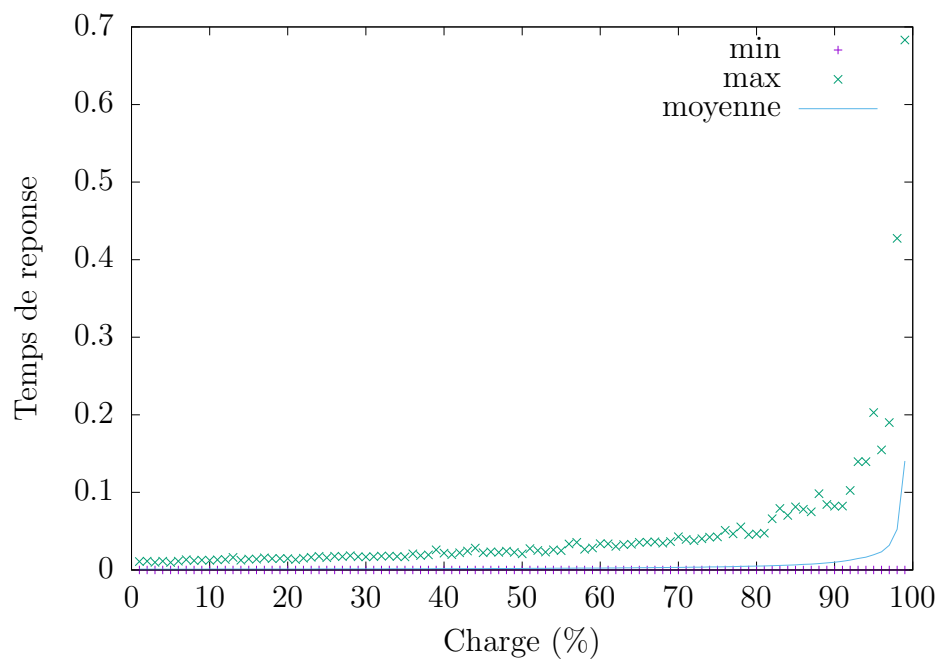


FIGURE 2.6 – Résultats de la simulation.

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 :

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

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

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

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

Lambda représente la charge multipliée par 10. Les résultats sont similaires, on observe toutefois les maximums des simulations qui atteignent de grandes valeurs par rapport aux moyennes.

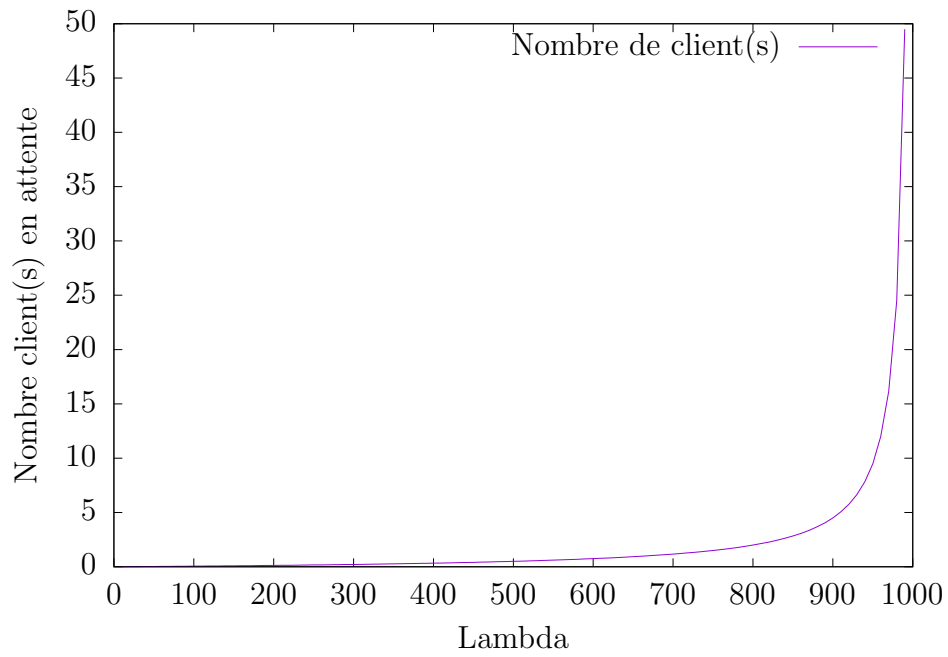


FIGURE 2.7 – Résultats du calcul Théorique.

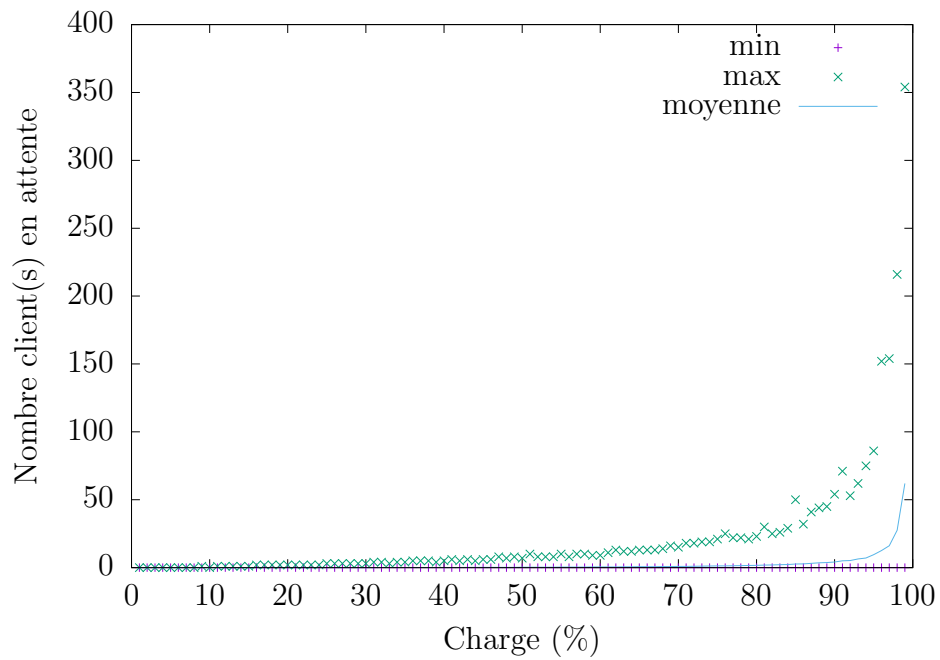


FIGURE 2.8 – Résultats de la simulation pour des paquets à la taille fixe de 10 Kb.

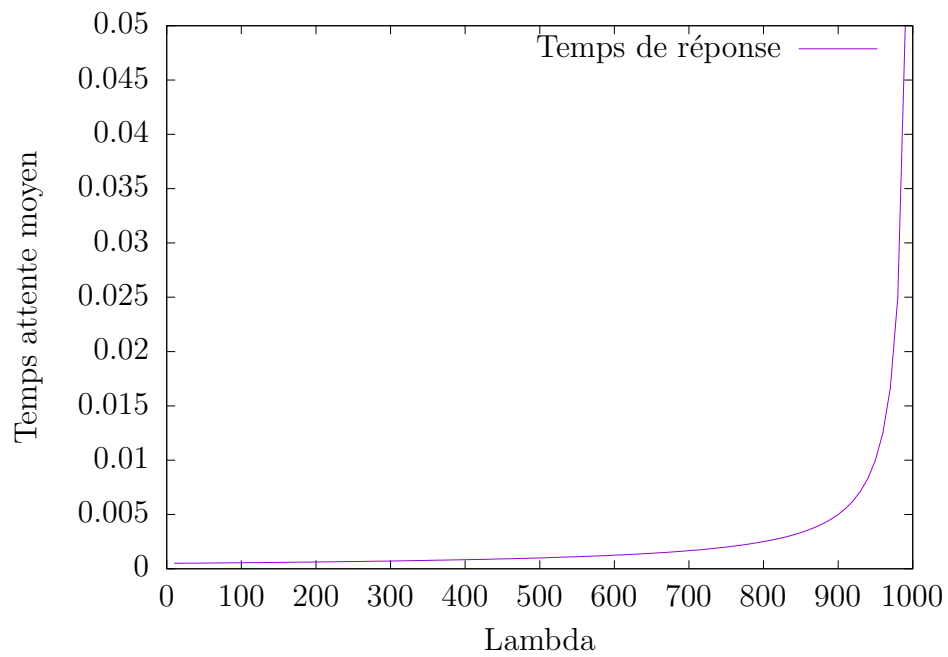


FIGURE 2.9 – Résultats du calcul Théorique.

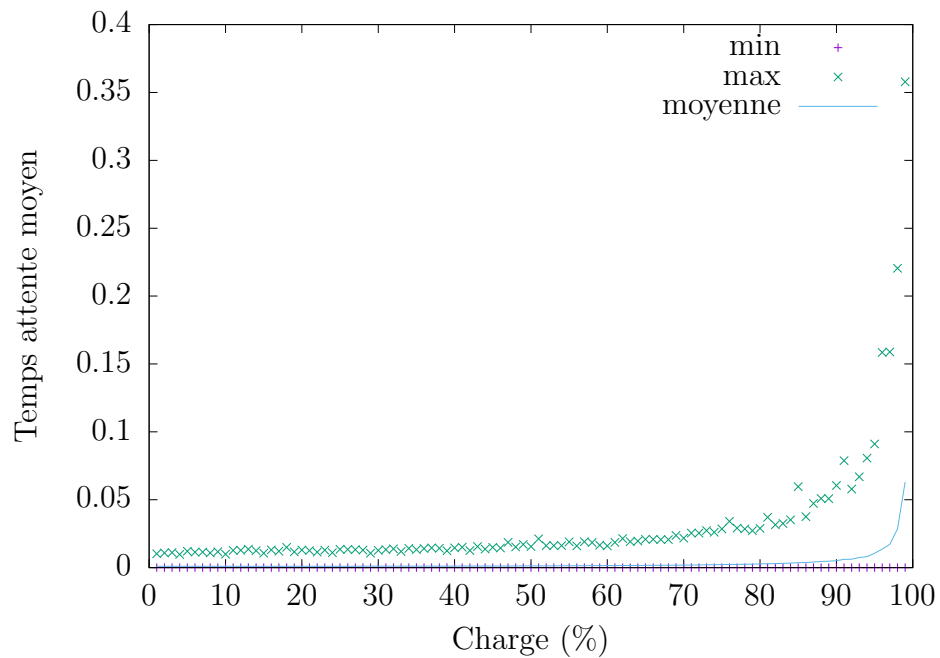


FIGURE 2.10 – Résultats de la simulation pour des paquets à la taille fixe de 10 Kb.

Annexe A

Annexes

A.1 Analyse sur un lien

Script Python de calcul de la probabilité de blocage d'appel P en fonction de la charge ρ 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.2 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 SRC1, SRC2, SRC3, LIEN1, LIEN2, LIEN3;
9     INTEGER TOTAL, NBTX, NBRJ, LOAD, I, C, RHO, RDRB, NBSIMU;
10    REAL DEBIT, LAMBDA, MU, PCT;
11    CUSTOMER REAL DUREE;
12    FILE fichier;
13 &
14 /STATION/
15     NAME = SRC1;
16     TYPE = SOURCE;
17     SERVICE = BEGIN
18         DUREE := ( 1. / MU );
19         TOTAL := TOTAL + 1;
20         EXP( 1. / LAMBDA );
21         & round robin
22         IF (RDRB = 0) THEN
23             BEGIN
24                 RDRB := 1;
25                 TRANSIT(LIEN1);
26             END
27         ELSE
28             BEGIN
29                 RDRB := 0;
30                 TRANSIT(LIEN2);
31             END;
32     END;
33     TRANSIT = OUT;
34 &
35 /STATION/
36     NAME = SRC2;
37     TYPE = SOURCE;
38     SERVICE = BEGIN
```

```

39     DUREE := ( 1. / MU );
40     TOTAL := TOTAL + 1;
41     EXP( 1. / LAMBDA );
42     & round robin
43     IF (RDRB = 0) THEN
44         BEGIN
45             RDRB := 1;
46             TRANSIT(LIEN2);
47         END
48     ELSE
49         BEGIN
50             RDRB := 0;
51             TRANSIT(LIEN3);
52         END;
53     END;
54     TRANSIT = OUT;
55 &
56 /STATION/
57     NAME = SRC3;
58     TYPE = SOURCE;
59     SERVICE = BEGIN
60         DUREE := ( 1. / MU );
61         TOTAL := TOTAL + 1;
62         EXP( 1. / LAMBDA );
63         & round robin
64         IF (RDRB = 0) THEN
65             BEGIN
66                 RDRB := 1;
67                 TRANSIT(LIEN3);
68             END
69         ELSE
70             BEGIN
71                 RDRB := 0;
72                 TRANSIT(LIEN1);
73             END;
74         END;
75     TRANSIT = OUT;
76 &
77 /STATION/
78     NAME = LIEN1;
79     TYPE = MULTIPLE(C);
80     SERVICE = BEGIN
81         CST(DUREE);

```

```

82     TRANSIT(OUT);
83 END;
84 CAPACITY = C;
85 REJECT = BEGIN
86     IF (CUSTNB(LIEN2) < C) AND (CUSTNB(LIEN3) < C) THEN
87         BEGIN
88             TRANSIT(LIEN2);
89             TRANSIT(LIEN3);
90         END;
91     NBRJ := NBRJ + 1;
92     TRANSIT(OUT);
93 END;
94 &
95 /STATION/
96     NAME = LIEN2;
97     TYPE = MULTIPLE(C);
98     SERVICE = BEGIN
99         CST(DUREE);
100     TRANSIT(LIEN3);
101 END;
102 CAPACITY = RHO;
103 REJECT = BEGIN
104     IF (CUSTNB(LIEN1) < C) AND (CUSTNB(LIEN3) < C) THEN
105         BEGIN
106             TRANSIT(LIEN1);
107             TRANSIT(LIEN3);
108         END;
109     NBRJ := NBRJ + 1;
110     TRANSIT(OUT);
111 END;
112 &
113 /STATION/
114     NAME = LIEN3;
115     TYPE = MULTIPLE(C);
116     SERVICE = BEGIN
117         CST(DUREE);
118     TRANSIT(OUT);
119 END;
120 CAPACITY = C;
121 REJECT = BEGIN
122     IF (CUSTNB(LIEN1) < C) AND (CUSTNB(LIEN2) < C) THEN
123         BEGIN
124             TRANSIT(LIEN1);

```

```

125         TRANSIT(LIEN2);
126     END;
127     NBRJ := NBRJ + 1;
128     TRANSIT(OUT);
129 END;
130 &
131 /EXEC/
132 BEGIN
133     & output data file
134     FILASSIGN(fichier , "p1.q3-2.data");
135     OPEN(fichier);
136     & var init
137     NBSIMU := 3;
138     RDRB := 0;
139     DEBIT := 64000;
140     MU := 1. / 180;
141     LOAD := 40;
142     & process simulations
143     FOR RHO := 1 STEP 1 UNTIL LOAD DO
144         BEGIN
145             FOR C := RHO STEP 1 UNTIL 2 * RHO DO
146                 BEGIN
147                     LAMBDA := MU * RHO;
148                     TOTAL := 0;
149                     NBRJ := 0;
150                     NBTX := 0;
151                     FOR I := 1 STEP 1 UNTIL NBSIMU DO
152                         BEGIN
153                             SIMUL;
154                         END;
155                     TOTAL := TOTAL / NBSIMU;
156                     NBRJ := NBRJ / NBSIMU;
157                     NBTX := NBTX / NBSIMU;
158                     PCT := NBRJ * 1. / TOTAL;
159                     WRITELN(fichier, RHO , " " , C , " " , TOTAL ,
160                             " " , NBRJ , " " , PCT );
161                 END;
162             END;
163         END;
164     END;
165 /END/

```


Simulation d'un lien d'un réseau à 3 commutateurs de circuits avec marge pour le débordement.

```

1  &
2  /CONTROL/
3      OPTION = NSOURCE;
4      TMAX = 100000;
5      ACCURACY = ALL QUEUE;
6  &
7  /DECLARE/
8      QUEUE SRC1, SRC2, SRC3, LIEN1, LIEN2, LIEN3;
9      INTEGER TOTAL, NBTX, NBRJ, LOAD, I, C, RHO, RDRB, NBSIMU, MARGE;
10     REAL DEBIT, LAMBDA, MU, PCT;
11     CUSTOMER REAL DUREE;
12     FILE fichier;
13 &
14 /STATION/
15     NAME = SRC1;
16     TYPE = SOURCE;
17     SERVICE = BEGIN
18         DUREE := ( 1. / MU );
19         TOTAL := TOTAL + 1;
20         EXP( 1. / LAMBDA );
21         & round robin
22         IF (RDRB = 0) THEN
23             BEGIN
24                 RDRB := 1;
25                 TRANSIT(LIEN1);
26             END
27         ELSE
28             BEGIN
29                 RDRB := 0;
30                 TRANSIT(LIEN2);
31             END;
32     END;
33     TRANSIT = OUT;
34 &
35 /STATION/
36     NAME = SRC2;
37     TYPE = SOURCE;
38     SERVICE = BEGIN
39         DUREE := ( 1. / MU );
40         TOTAL := TOTAL + 1;
41         EXP( 1. / LAMBDA );

```

```

42      & round robin
43      IF (RDRB = 0) THEN
44          BEGIN
45              RDRB := 1;
46              TRANSIT(LIEN2);
47          END
48      ELSE
49          BEGIN
50              RDRB := 0;
51              TRANSIT(LIEN3);
52          END;
53      END;
54      TRANSIT = OUT;
55  &
56  /STATION/
57      NAME = SRC3;
58      TYPE = SOURCE;
59      SERVICE = BEGIN
60          DUREE := ( 1. / MU );
61          TOTAL := TOTAL + 1;
62          EXP( 1. / LAMBDA );
63          & round robin
64          IF (RDRB = 0) THEN
65              BEGIN
66                  RDRB := 1;
67                  TRANSIT(LIEN3);
68              END
69          ELSE
70              BEGIN
71                  RDRB := 0;
72                  TRANSIT(LIEN1);
73              END;
74          END;
75          TRANSIT = OUT;
76  &
77  /STATION/
78      NAME = LIEN1;
79      TYPE = MULTIPLE(C);
80      SERVICE = BEGIN
81          CST(DUREE);
82          TRANSIT(OUT);
83      END;
84      CAPACITY = C;

```

```

85 REJECT = BEGIN
86     IF (CUSTNB(LIEN2) < C - MARGE) AND (CUSTNB(LIEN3) < C -
      MARGE) THEN
87         BEGIN
88             TRANSIT(LIEN2);
89             TRANSIT(LIEN3);
90         END;
91     NBRJ := NBRJ + 1;
92     TRANSIT(OUT);
93 END;
94 &
95 /STATION/
96     NAME = LIEN2;
97     TYPE = MULTIPLE(C);
98     SERVICE = BEGIN
99         CST(DUREE);
100     TRANSIT(LIEN3);
101 END;
102 CAPACITY = RHO;
103 REJECT = BEGIN
104     IF (CUSTNB(LIEN1) < C - MARGE) AND (CUSTNB(LIEN3) < C -
      MARGE) THEN
105         BEGIN
106             TRANSIT(LIEN1);
107             TRANSIT(LIEN3);
108         END;
109     NBRJ := NBRJ + 1;
110     TRANSIT(OUT);
111 END;
112 &
113 /STATION/
114     NAME = LIEN3;
115     TYPE = MULTIPLE(C);
116     SERVICE = BEGIN
117         CST(DUREE);
118     TRANSIT(OUT);
119 END;
120 CAPACITY = C;
121 REJECT = BEGIN
122     IF (CUSTNB(LIEN1) < C - MARGE) AND (CUSTNB(LIEN2) < C -
      MARGE) THEN
123         BEGIN
124             TRANSIT(LIEN1);

```

```

125         TRANSIT(LIEN2);
126     END;
127     NBRJ := NBRJ + 1;
128     TRANSIT(OUT);
129 END;
130 &
131 /EXEC/
132 BEGIN
133     & output data file
134     FILASSIGN(fichier , "p1.q4.data");
135     OPEN(fichier);
136     & var init
137     NBSIMU := 3;
138     RDRB := 0;
139     DEBIT := 64000;
140     MU := 1. / 180;
141     LOAD := 40;
142     & process simulations
143     FOR MARGE := 1 STEP 1 UNTIL 3 DO
144     BEGIN
145         FOR RHO := 1 STEP 1 UNTIL LOAD DO
146         BEGIN
147             FOR C := RHO STEP 1 UNTIL 2 * RHO DO
148             BEGIN
149                 LAMBDA := MU * RHO;
150                 TOTAL := 0;
151                 NBRJ := 0;
152                 NBTX := 0;
153                 FOR I := 1 STEP 1 UNTIL NBSIMU DO
154                 BEGIN
155                     SIMUL;
156                 END;
157                 TOTAL := TOTAL / NBSIMU;
158                 NBRJ := NBRJ / NBSIMU;
159                 NBTX := NBTX / NBSIMU;
160                 PCT := NBRJ * 1. / TOTAL;
161                 WRITELN(fichier, MARGE, " ", RHO , " " , C
162                     , " " , TOTAL , " " , NBRJ , " " , PCT
163                     );
164             END;
165         END;
166     END;
167 END;

```

166 | /END/

A.3 Un commutateur de paquets

Script Python de calcul pour un lien de sortie d'un commutateur de paquets.

```
1  # -*- coding: utf-8 -*-
2
3  # module maths
4  import math
5
6  paquet = 10 * pow(10, 3)
7  debit = 10 * pow(10, 6)
8
9  tps_moy_serv = paquet * 1.0 / debit
10
11 for p in range(1, 100):
12     p = p / 100
13
14     lambd = p * 1.0 / tps_moy_serv
15     n = p / (1 - p)
16     w = 1 / (1 / tps_moy_serv - lambd)
17
18     print(str(tps_moy_serv) + ' ' + str(lambd) + ' ' + str(n) + ' '
           + str(w))
```

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

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

```

42         END;
43     sumA := sumA + CUSTNB(Q);
44     IF minA > CUSTNB(Q) THEN
45         BEGIN
46             minA := CUSTNB(Q);
47         END;
48     IF maxA < CUSTNB(Q) THEN
49         BEGIN
50             maxA := CUSTNB(Q);
51         END;
52     END;
53     TRANSIT = OUT;
54 &
55 /EXEC/
56     BEGIN
57         FILASSIGN(cbQ,"p2.q3.data");
58         OPEN(cbQ);
59         & boucle des simulations
60         FOR I := 1 STEP 1 UNTIL 99 DO
61             BEGIN
62                 & initialisation des variables
63                 RO := I * 0.010;
64                 LAMBDA := RO * MU;
65                 minA := 10000000;
66                 maxA := 0;
67                 sumA := 0;
68                 minT := 10000000;
69                 maxT := 0;
70                 sumT := 0;
71                 & simulation
72                 SIMUL;
73                 & affichage de la progression
74                 WRITELN(I);
75                 & ecriture resultats dans le fichier
76                 WRITELN(cbQ,I," ",minT," ",maxT," ",sumT/SERVNB(
77                     fakeOUT)," ",minA," ",maxA," ",sumA/SERVNB(
78                     fakeOUT));
79             END;
80         END;
81     /END/

```


Script Python de calcul pour un lien de sortie d'un commutateur de paquets avec la taille des paquets constante.

```
1  # -*- coding: utf-8 -*-
2
3  # module maths
4  import math
5
6  paquet = 10 * pow(10, 3)
7  debit = 10 * pow(10, 6)
8
9  tps_moy_serv = paquet * 1.0 / debit
10
11 for p in range(1, 100):
12     p = p / 100
13
14     lambd = p * 1.0 / tps_moy_serv
15     n = p / (2 * (1 - p))
16     w = 1 / (2 * (1 / tps_moy_serv - lambd))
17
18     print(str(tps_moy_serv) + ' ' + str(lambd) + ' ' + str(n) + ' '
           + str(w))
```

Simulation d'un lien de sortie d'un commutateur de paquets avec la taille des paquets constante.

```

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

```

```

41         END;
42     sumA := sumA + CUSTNB(Q);
43     IF minA > CUSTNB(Q) THEN
44         BEGIN
45             minA := CUSTNB(Q);
46         END;
47     IF maxA < CUSTNB(Q) THEN
48         BEGIN
49             maxA := CUSTNB(Q);
50         END;
51     END;
52     TRANSIT = OUT;
53 &
54 /EXEC/
55     BEGIN
56         FILASSIGN(cbQ,"p2.cst.data");
57         OPEN(cbQ);
58         & boucle des simulations
59         FOR I := 1 STEP 1 UNTIL 99 DO
60             BEGIN
61                 & initialisation des variables
62                 RO := I * 0.010;
63                 LAMBDA := RO * MU;
64                 minA := 10000000;
65                 maxA := 0;
66                 sumA := 0;
67                 minT := 10000000;
68                 maxT := 0;
69                 sumT := 0;
70                 & simulation
71                 SIMUL;
72                 & affichage de la progression
73                 WRITELN(I);
74                 & ecriture resultats dans le fichier
75                 WRITELN(cbQ,I," ",minT," ",maxT," ",sumT/SERVNB(
76                     fakeOUT)," ",minA," ",maxA," ",sumA/SERVNB(
77                     fakeOUT));
78             END;
79         END;
80     /END/

```