

QNAP

Boulic Guillaume, Émeric Tosi

8 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	4
2	Commutation de paquets	5
2.1	Un commutateur de paquets	5
	Conclusion	7
A	Annexes	8
A.1	Exemple(s)	8
A.2	Analyse sur un lien	9
A.3	Analyse sur un réseau de trois commutateurs	12
A.4	Un commutateur de paquets	16

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.

Chacune des connexions nécessite un débit de 64 Kb/s bi-directionnels. 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, les durées des appels sont supposées suivre une loi exponentielle de paramètre μ ($\mu \approx 3$ min).

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 en Erlang et C la capacité.

1.1.3 Simulation de la probabilité de blocage d'appel pour une charge comprise entre 10 et 70 Erlangs

Blablabla 1

1.1.4 Variation de la capacité C pour une variation de la charge normalisée entre 0.5 et 1

Blablabla 2

1.1.5 Comparaison des taux de blocage expérimental et théorique

Blablabla 3

1.2 Analyse sur un réseau de trois commutateurs

1.2.1 Énoncé

Désormais, nous considérons le réseau composé des 3 nœuds suivant :

Les arrivées sont supposées Poissonniennes sur chacun des nœuds et le trafic se répartit équiprobablement entre les différents nœuds. 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

Blablabla 4

1.2.3 Comparaison des résultats avec la partie 1.1

Blablabla 5

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 (en dessous d'un certain seuil d'occupation sur chacun des liens). Cela revient donc à laisser une marge M aux appels directs.

Commentaires

Blablabla 6

Simulation en prenant une marge comprise entre 1 et 3

Blablabla 7

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.

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 $\frac{1}{\nu} = 10000 \text{bits}$. Le lien de sortie a un débit de 10 Mbit/s.

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

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

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

$$\begin{aligned} \lambda &= \rho\mu \\ \text{Charge de trafic } \rho &= \frac{\lambda}{\mu} \end{aligned}$$

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

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

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

ρ	0.1	0.5	0.9
λ	10^2	510^2	910^2
Nombre moyen de client	0.11111	1	9
Temps moyen de réponse	0.00111	0.002	0.01

2.1.4 Comparaison du résultat de la simulation avec la théorie

Blablabla 1

2.1.5 Cas où les paquets ont une longueur constante (10000 bits)

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

On obtient la même chose que précédemment :

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

$$\Longleftrightarrow \frac{1}{\nu} \frac{1}{D} = 10^4 \frac{1}{10^7} = \frac{1}{10^3} = 10^{-3} \text{ seconde}$$

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

Temps moyen de réponse Blablabla 3

Nombre moyen de paquets dans la file d'attente Blablabla 4

Analyse et comparaison des résultats

Blablabla 5

Conclusion

Too much bullshit here :P

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 de calcul théorique

```
1 import math
2
3 for p in range(1,80+1):
4     for c in range(p,p*2+1):
5         numerateur = pow(p,c) * 1.0 / math.factorial(c)
6         denominateur = 0
7         for i in range(0,c):
8             denominateur += pow(p,i) * 1.0 / math.factorial(i)
9         print( str(c) + " " + str(p) + " " + str(numerateur /
            denominateur))
```

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

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

```

43      DEBIT := 64000;
44      MU := 1. / 180;
45      Erlang := 70;
46      NbSimu := 5;
47      &
48      FOR RHO := 10 STEP 5 UNTIL Erlang DO
49      BEGIN
50          &
51          LAMBDA := MU * RHO;
52          &
53          FOR C := RHO STEP 5 UNTIL (RHO * 2) DO
54          BEGIN
55              &
56              NbOK := 0;
57              NbRejets := 0;
58              &
59              FOR I := 1 STEP 1 UNTIL NbSimu DO
60              BEGIN
61                  SIMUL;
62              END;
63              &
64              NbOK := NbOK / NbSimu;
65              NbRejets := NbRejets / NbSimu;
66              &
67              WRITELN(cbQ , " " , RHO , " " , C , " " , NbRejets ,
68                  " " , NbOK , " " , NbRejets / (NbOK + NbRejets));
69          END;
70          &
71      END;
72
73      &
74      /END/
75
76      &
77      & plot "p1.q2.data" u 1:2 w l t "taux de rejet"
78      &

```

A.3 Analyse sur un réseau de trois commutateurs

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

```
1 /DECLARE/
2 /DECLARE/
3 QUEUE SRC, LIEN1, LIEN2, LIEN3;
4 INTEGER I, C, ERL = 70, NbSimu = 20;
5 REAL RHO, LAMBDA, DUREE, App10K, App1Bloc, App20K, App2Bloc, App30K
6 , App3Bloc, MU = 1./180;
7 CUSTOMER REAL DUREE;
8 REF CUSTOMER CUST;
9 FILE courbe;
10
11 & Station emettrice des appels de duree exponentielle
12 /STATION/
13 NAME = SRC;
14 TYPE = SOURCE;
15 SERVICE = BEGIN
16     EXP(1./LAMBDA);
17     & Verification que le LIEN1 ne soit pas surcharge pour envoyer
18     sur le 1er chemin
19     IF (CUSTNB(LIEN1)<C) THEN
20         BEGIN
21             DUREE := EXP(1./MU);
22             TRANSIT(LIEN1);
23         END
24     & Sinon on envoit sur le chemin de debordement
25     ELSE
26         BEGIN
27             DUREE := EXP(1./MU);
28             CUST := NEW (CUSTOMER);
29             CUST.DUREE := DUREE;
30             TRANSIT(CUST,LIEN2);
31             TRANSIT(CUST,LIEN3);
32         END;
33     END;
34 TRANSIT = OUT;
35
36 & Station receptrice simulant le premier lien (donc 1er chemin)
```

```

37 /STATION/
38 NAME = LIEN1;
39 TYPE = MULTIPLE(C);
40 & Si l'appel peut etre décroché, alors on compte le nombre d'appel
    reçu
41 SERVICE =
42 BEGIN
43     EXP(1./MU);
44     App1OK := App1OK +1;
45 END;
46 CAPACITY = C;
47 & En cas de rejet, on compte le nombre d'appel bloqué
48 REJECT =
49 BEGIN
50     App1Bloc := App1Bloc + 1;
51     TRANSIT(OUT);
52 END;
53 TRANSIT = OUT;
54
55
56 & Station receptrice simulant le second lien (1ere partie du 2nd
    chemin)
57 /STATION/
58 NAME=LIEN2;
59 TYPE= MULTIPLE(C);
60 SERVICE =
61 BEGIN
62     EXP(1./MU);
63     App2OK := App2OK +1;
64 END;
65 CAPACITY = C;
66 & En cas de rejet, on compte le nombre d'appel bloqué
67 REJECT =
68 BEGIN
69     App2Bloc := App2Bloc + 1;
70     TRANSIT(OUT);
71 END;
72 TRANSIT=OUT;
73
74
75 & Station receptrice simulant le second lien (2nd partie du 2nd
    chemin)
76 /STATION/

```

```

77 NAME=LIEN3;
78 TYPE= MULTIPLE(C);
79 SERVICE =
80 BEGIN
81     EXP(1./MU);
82     App3OK := App3OK +1;
83 END;
84 CAPACITY = C;
85 & En cas de rejet, on compte le nombre d'appel bloqué
86 REJECT =
87 BEGIN
88     App3Bloc := App3Bloc + 1;
89     TRANSIT(OUT);
90 END;
91 TRANSIT=OUT;
92
93
94 /CONTROL/
95 TMAX = 10000;
96
97
98 /EXEC/
99 BEGIN
100     FILASSIGN(courbe , "Partie 1 - 3 courbes.data");
101     OPEN(courbe);
102     & Variation de la charge (RHO) de 10 a 70 (ERL) par pas de 10
103     FOR RHO:=10 STEP 10 UNTIL ERL DO
104     BEGIN
105         LAMBDA := MU * RHO;
106         & Variation de la capacite (C) pour posseder une charge
            normalisee (RHO) comprise entre 0.5 et 1
107         FOR C := RHO STEP (RHO/2) UNTIL 2 * RHO DO
108         BEGIN
109             App1Bloc := 0;
110             App1OK := 0;
111             App2Bloc := 0;
112             App2OK := 0;
113             App3Bloc := 0;
114             App3OK := 0;
115             FOR I := 1 STEP 1 UNTIL NbSimu DO
116             BEGIN
117                 SIMUL;
118             END;

```

```

119      App1Bloc := App1Bloc / NbSimu;
120      App1OK := App1OK / NbSimu;
121      App2Bloc := App2Bloc / NbSimu;
122      App2OK := App2OK / NbSimu;
123      App3Bloc := App3Bloc / NbSimu;
124      App3OK := App3OK / NbSimu;
125      WRITE(courbe, " ", RHO, " ", C, " ", App1Bloc, " "
126            , App1OK, " ", App1Bloc/(App1OK+App1Bloc));
127      WRITE(courbe, " ", App2Bloc, " ", App2OK, " ");
128      IF((App2OK < 1) AND (App2Bloc <1)) THEN
129          BEGIN
130              WRITE(courbe, "0");
131          END
132      ELSE
133          BEGIN
134              WRITE(courbe, App2Bloc/(App2OK+App2Bloc));
135          END;
136
137      WRITE(courbe, " ", App3Bloc, " ", App3OK, " ");
138
139      IF((App3OK < 1) AND (App3Bloc <1)) THEN
140          BEGIN
141              WRITE(courbe, "0");
142          END
143      ELSE
144          BEGIN
145              WRITE(courbe, App3Bloc/(App3OK+App3Bloc));
146          END;
147      WRITELN(courbe, " ");
148  END;
149 END;
150
151
152 /END/

```


A.4 Un commutateur de paquets