74-617

# ETUDE STATISTIQUE D'UN SYSTEME DE TELECOMMUNICATION BASE SUR UN MODE D'AFFECTATION PRIORITAIRE ET SERVANT UN GROUPE LIMITE D'UTILISATEURS

S. MORISSETTE LE PHUNG

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

FACULTÉ DES SCIENCES APPLIQUÉES

UNIVERSITÉ DE SHERBROOKE

LKC
P
91
.C654
M67
1975

IC

SHERBROOKE, QUÉBEC, CANADA

Checked 11/8

P
91
C654
M67f
1975

Industry Canada
Library - Queen

AVR
APR - 2 2013

Industrie Canada
Bibliothèque - Queen

# ETUDE STATISTIQUE D'UN SYSTEME DE TELECOMMUNICATION BASE SUR UN MODE D'AFFECTATION PRIORITAIRE ET SERVANT UN GROUPE LIMITE D'UTILISATEURS

rapport final

S. MORISSETTE LE PHUNG

RAPPORT FINAL

Soumis au Centre de Recherche en Communication (CRC) pour la réalisation finale du contrat No. 0SU4-0118, No. de dossier MAS 02SU 36100-4-0239, pour la période du 1er avril 1974 au 31 mars 1975.

Directeur du Projet S. Morissette
S. Morissette

Chercheur principal Le Phung
Le Phung

Université de Sherbrooke

31 Mars 1975

COMMUNICATIONS CANADA
OCT 1 1984
LIBRARY – BIBLIOTHÈQUE

RECEIVED
DUP
DEC 15 1975
1632

P
91
C654
M67f
1975

DD4807162
DL4819840

RECEIVED
DEC 15 1975
COMMUNICATIONS CENTER
LIBRARY

# ABSTRACT

On s'intéresse dans ce rapport à une analyse statistique d'un réseau de télécommunication destiné à:

a- desservir une population d'utilisateurs formée par des groupes de faible dimension

b- chaque groupe génère son propre trafic

c- l'affectation des lignes se fait suivant un protocole prioritaire avec

d- réjection d'un appel de plus basse priorité à l'arrivée d'un autre de plus haute priorité quand tous les canaux sont occupés

En admettant que les distributions des temps inter-arrivée des appels et des temps de services soient exponentiels, on démontre que le réseau peut être représenté par un processus de mort et de naissance multidimensionnel avec coefficients variables suivant l'état du système. On calcule à partir de la distribution des états en régime permanent les indices de performance du système tels que:

a- le trafic écoulé

b- la probabilité de congestion des lignes pour des appels d'une priorité donnée

c- les probabilités de réjection et

d- de non admissibilité d'un appel de priorité prédéterminée reliant deux sous-populations quelconques.

Le modèle obtenu englobe ceux connues dans la littérature comme cas particulier.

Un programme pour le calcul numérique des résultats a été développé. Le programme est conçu pour économiser au maximum le temps de calcul étant donné la complexité du modèle. Il se base sur deux sous-programmes qui assurent la génération rapide des vecteurs d'état. Un exemple numérique illustre les résultats théoriques et leur exploitation numérique.

# REMERCIEMENTS

J'aimerais d'abord remercier le Centre de Recherche en Communication (CRC) d'avoir bien voulu nous suggérer cette étude et lui accorder son aide financière. Cette étude a sensiblement accru nos connaissances dans le domaine concerné.

Cette étude n'aurait pu être réalisée sans la participation très active et brillante du docteur Le Phung qui a agi en tant que chercheur principal pour nous aider à mener à bien ce travail. Nous l'en remercions et espérons obtenir sa collaboration dans des projets futurs.

Des remerciements aussi à monsieur Le Thang Thoi qui a réussi à programmer avec beaucoup d'ingéniosité les équations d'état que nous avions obtenus.

On ne peut oublier la participation du docteur Roger Goulet en tant que Directeur du Département de Génie Electrique ainsi que chercheur.

Enfin nos remerciements les plus sincères à madame Louise Morissette pour la dactylographie et la présentation soigneuse de ce document ainsi que du rapport préliminaire.

Sarto Morissette, ing.

# TABLE DES MATIERES

ABSTRACT---------- ii

REMERCIEMENTS---------- iv

TABLE DES MATIERES---------- v

LISTE DES FIGURES---------- vii

LISTE DES TABLEAUX---------- viii

LISTE DES PRINCIPALES NOTATIONS---------- ix

INTRODUCTION---------- 1

PREMIERE PARTIE - DEVELOPPEMENT THEORIQUE---------- 5

1. HISTORIQUE---------- 7

2. DEFINITION DU PROBLEME---------- 9

2.1 Sous-populations d'utilisateurs---------- 9

2.2 Caractérisation d'un appel---------- 10

2.3 Caractérisation du trafic des sous-populations---------- 10

2.4 Protocole des lignes---------- 11

2.5 Vecteur d'état---------- 11

2.6 Dimension des vecteurs d'état---------- 13

2.7 Caractéristiques d'un vecteur d'état---------- 14

3. EQUATION D'ETAT---------- 16

3.1 Taux de génération des appels---------- 16

3.2 Equations d'état en régime transitoire et sans réjection---------- 17

3.3 Equations d'état en régime transitoire avec réjection---------- 19

3.4 Equations d'état en régime permanent---------- 22

3.5 Equations d'état du système sans réjection---------- 23

3.6 Equations d'état du système sans réjection desservant une population infinie---------- 24

4. RESOLUTION DES EQUATIONS D'ETAT EN REGIME PERMANENT---------- 25

4.1 Système sans réjection---------- 25

4.2 Système avec réjection---------- 26

4.3 Cas d'une population infinie---------- 30

5. INDICE DE PERFORMANCE---------- 33

5.1 Trafic écoulé---------- 33

5.2 Probabilité de congestion d'un appel de priorité $k_o$---------- 33

5.3 Probabilité de réjection d'un appel de priorité s et reliant les sous-populations i,j---------- 34

5.4 Probabilité de non admissibilité d'un appel $(k_o;i^*,j^*)$---------- 37

5.5 Taux d'appels acceptés---------- 39

5.6 Cas d'une population infinie---------- 39

5.6.1 Trafic écoulé. Efficacité---------- 39

5.6.2 Probabilité de congestion des canaux pour un appel de priorité k---------- 41

DEUXIEME PARTIE - EXPLOITATION DES RESULTATS---------- 43

6. TECHNIQUES DE CALCUL NUMERIQUE DES INDICES DE PERFORMANCE---------- 44

6.1 Génération de l'ensemble de vecteur d'état---------- 44

6.2 La génération des partitions d'un nombre entier n---------- 46

6.3 La permutation d'un multi-ensemble---------- 47

6.4 Le programme principal---------- 53

7. RESULTATS NUMERIQUES---------- 61

7.1 Exemple numérique ---------- 61

7.2 Courbes représentatives de la formule d'Erlang généralisée---------- 67

CONCLUSION---------- 71

REFERENCES---------- 75

# LISTE DES FIGURES


Fig. 1 L'arbre de partition de N = 4---------- 47

Fig. 2 L'organigramme de la génération des partitions $B_i$ d'un nombre n---------- 48

Fig. 3 L'arbre représentant la permutation d'un multi-ensemble pour la génération des vecteurs d'état ayant D = 4 et associé avec la partition B = (1 1 3) du nombre n = 5---------- 50

Fig. 4 L'organigramme de l'opération de génération de l'ensemble des vecteurs d'état associé avec une partition donnée $B_i$---------- 52

Fig. 5a L'organigramme du programme principal---------- 54

Fig. 5b L'organigramme du programme principal (suite)---- 55

Fig. 5c L'organigramme du programme principal (suite et fin)---------- 56

Fig. 6 Schéma du système hypothétique étudié dans l'exemple numérique---------- 63

Fig. 7 Courbes représentatives des pertes des appels de différentes priorités en fonction du trafic total---------- 68

# LISTE DES TABLEAUX


Tableau 1 Les seules différences entre les composantes et paramètres des vecteurs d'état adjacents---------- 17

Tableau 2 Liste des partitions de N = 4---------- 46

Tableau 3 Liste des vecteurs d'état de dimension D = 4 associés avec la partition B = 1 1 3 du nombre 5---------- 49

Tableau 4 Paramètres à calculer et leurs équations correspondantes---------- 58

Tableau 5 Répartition des utilisateurs parmi les sous-populations---------- 61

Tableau 6 Probabilités de congestion des lignes pour les appels de priorité k = 1 et 2---------- 66

Tableau 7 Probabilités de réjection des lignes pour les appels de priorité $k_o$ = 2 ---------- 66

Tableau 8 Probabilités de non admissibilité pour les appels de priorité k = 1---------- 66

# LISTE DES PRINCIPALES NOTATIONS

| | |
|---|---|
| $C$ | nombre total des canaux du système |
| $D$ | dimension du vecteur d'état |
| $F$ | nombre total des vecteurs d'état |
| $M$ | nombre total d'utilisateurs dans le système |
| $N_{k;i,j}(n)$ | vecteur d'état ayant n appels en service |
| $N_{k,s;i,j}(n)$ | vecteur d'état ayant n appels dont la priorité la plus basse des appels est égale à s |
| $P$ | nombre total de priorités différentes |
| $P\left[N_{k;i,j}(n);t\right]$ | probabilité de trouver à l'instant t le système à l'état $N_{k;i,j}(n)$ |
| $P\left[N_{k;i,j}(n)\right]$ | probabilité de trouver, en régime permanent le système à l'état $N_{k,i,j}(n)$ |
| $Q_s(i^*,j^*)$ | probabilité de réjection d'un appel de priorité s reliant les sous-populations $(i^*,j^*)$ |
| $R_{k_o}(i^*,j^*)$ | probabilité de non admissibilité d'un appel de priorité $k_o$ reliant les sous-populations $(i^*,j^*)$ |
| $S$ | nombre des sous-populations |
| $S_i$ | ième sous-population |
| $m_i$ | nombre d'utilisateurs dans la sous-population $S_i$ |
| $m_i'$ | nombre d'utilisateurs qui sont en service dans la sous-population $S_i$ |
| $m_i''$ | nombre d'utilisateurs libres dans la sous-population $S_i$ |
| $n$ | nombre total d'appels en service |
| $n_k(i,j)$ | nombre d'appels en service de priorité k, d'origine $S_i$ et de destination $S_j$ |

| | |
|---|---|
| $\mu_k(i,j)$ | taux de service des appels de priorité k, d'origine $S_i$ et de destination $S_j$ |
| $\lambda_k(i,j)$ | taux de génération d'appels de priorité k, d'origine $S_i$ et de destination $S_j$ |
| $\rho_k(i,j)$ | intensité du trafic de priorité k, d'origine $S_i$ et de destination $S_j$ |
| $k_0$ | lowest priority of calls in progress |

# INTRODUCTION

Dans l'ensemble d'une étude de faisabilité d'un nouveau réseau gouvernemental de communication, entreprise par le Centre de Recherche en Communication, nous nous proposons d'étudier le comportement statistique d'un réseau spécial destiné à:

a- desservir une population d'utilisateurs formée par des groupes de faible dimension

b- chaque groupe a un trafic qui lui est propre

c- l'affectation des lignes se fait suivant un protocole prioritaire

d- la réjection d'un appel en cours à l'arrivée d'un appel de plus haute priorité est permise

L'étude statistique d'un réseau de télécommunication possédant les quatre caractéristiques mentionnées ci-dessus constitue l'essence de ce contrat entre le CRC et l'Université de Sherbrooke, Sherbrooke, Québec.

Le présent rapport final présente le travail, en entier, réalisé pendant la période du début d'avril 1974 jusqu'à mars 1975. Le travail rapporté se compose de deux parties:

Partie I : Développements théoriques

Partie II: Exploitation des résultats

La partie I est essentiellement une mise au point des résultats théoriques qui ont déjà été présentés au CRC dans un rapport préliminaire, au mois d'octobre 1974. On y a ajouté une brève note sur l'historique du problème et un résumé succinct des travaux ayant trait au problème sous considération.

En admettant que chaque liaison entre deux sous-populations est un processus de Poisson, il est clair que les appels entre deux sous-populations sont générés suivant un processus de Poisson dont le paramètre est proportionnel au nombre de liaisons inactives. Cette observation nous permet de tenir compte de la dimension des groupes d'utilisateurs dans le modèle. Comme d'habitude on admet aussi que les temps de service des appels sont exponentiellement distribués. Sous ces hypothèses on obtient pour le système un modèle multi-dimentionnel de mort et de naissance.

Il est évident que le comportement d'un système sans réjection des appels de basse priorité à l'arrivée d'un autre de plus haute priorité ne se différencie de celui d'un système avec réjection que quand tous les canaux sont occupés. Alors, on démontre que la probabilité de trouver le système avec réjection dans un état donné, est identique à celle d'un système sans réjection, dans le même état, en admettant que les paramètres définissant ces deux systèmes sont identiques. Pourtant, on démontre que le système avec réjection doit satisfaire à une condition d'équilibre, qui exprime la probabilité qu'un appel de

priorité donnée soit admis dans le système par suite de la réjection d'un appel de plus basse priorité présent dans le système à son arrivée.

On trouve facilement que notre modèle inclut aussi le cas d'une population de dimension infinie comme un cas particulier. Les résultats obtenus sont identiques avec ceux trouvés récemment par Brandt [4].

On définit quatre indices de performances caractérisant le système:

a- le trafic écoulé

b- la probabilité de congestion des canaux

c- la probabilité de réjection d'un appel de priorité donnée reliant deux sous-populations quelconques

d- la probabilité de non-admissibilité d'un appel de priorité donnée reliant deux sous-populations quelconques

Clairement ces indices de performance intéressent les utilisateurs de réseaux aussi bien que ceux qui les exploitent ou ceux qui les concoivent. Ces paramètres caractéristiques sont exprimés en fonction de la distribution des états du système. Pour le cas d'un système desservant une population infinie, les résultats obtenus sont comparables avec ceux connus dans la littérature technique.

Dans la deuxième partie, on s'intéresse aux problè-

relatifs au calcul numérique des indices de performance du système. Le problème principal dans ce calcul est la génération de l'ensemble des vecteurs d'état. Comme le nombre de ces vecteurs croît excessivement vite avec les dimensions du système, il est nécessaire de trouver un moyen de générer sans répétition des vecteurs d'état pour minimiser le temps de calcul et la longueur de la mémoire du calculateur. Cette technique se base sur deux algorithmes:

a- la génération des partitions d'un nombre donné

b- la permutation des multi-ensembles.

Les principes généraux de ces algorithmes sont introduits par des exemples simples. Finalement, on donne des exemples numériques illustrant les résultats théoriques obtenus dans la première partie en travaillant sur un réseau hypothétique de dimension réduite.

On termine le rapport en suggérant deux problèmes intéressants et en passant en revue leur méthode de solution qui est une généralisation du modèle multi-dimensionnel développé dans ce travail.

# PREMIERE PARTIE

# DEVELOPPEMENT THEORIQUE

*Sommaire*

La première partie de ce rapport se compose de cinq chapitres.

Le chapitre I présente brièvement l'historique du problème et une revision très succincte des travaux publiés traitant des questions ayant des traits communs avec notre problème.

La définition analytique du problème est donnée dans le chapitre II. Le système à l'étude est décrit par un ensemble de C canaux desservant S sous-populations qui génèrent chacune ses propres trafics tant du point de vue des priorités que des destinations. L'état du système est défini par un vecteur, dont le nombre des composantes est égal au nombre de liaisons de différentes priorités entre les sous-populations et même entre deux utilisateurs d'une même sous-population. Supposons que le système se trouve dans un état correspondant à un vecteur donné. La valeur d'une composante de ce vecteur indique le nombre de canaux occupés dans le système par les appels d'une priorité donnée entre deux sous-populations déterminées qui sont associées avec la dite composante.

La mise en équation du système est discutée dans le

chapitre III. On tient compte de l'influence de la dimension des sous-populations en admettant que:

a- les distributions des temps inter-arrivée entre les appels et des temps de service de ces appels sont exponentielles, et

b- le paramètre du processus de Poisson entre deux sous-populations données est proportionnel au nombre des liaisons non occupées

Le système est ainsi modelé par un processus multidimentionnel de mort et de naissance dont les coefficients sont fonctions de l'état du système. On s'intéresse particulièrement au système avec réjection d'un appel de plus basse priorité à l'arrivée d'un appel de plus haute priorité.

Le chapitre IV traite en détail la résolution des équations d'état en régime permanent. Le cas d'un système sans réjection et le cas d'une population infinie d'utilisateurs ne sont que des cas particuliers de notre modèle.

Les indices de performance du système sont définis dans le chapitre V et sont des indices intéressant l'utilisateur et l'exploiteur du système. On étudie en détail le calcul de:

a- trafic écoulé

b- la probabilité de congestion des canaux

c- la probabilité de réjection d'un appel

d- la probabilité de non-admissibilité d'un appel

La simplification des résultats obtenus conduit aux résultats connus de cas particuliers d'un système servant une population infinie.

# CHAPITRE 1

# HISTORIQUE

L'étude statistique d'un système de télécommunication basé sur un mode d'affectation prioritaire et desservant un groupe limité d'utilisateurs fait partie d'une étude de faisabilité d'un réseau de communication privé qu'entreprend le Centre de Recherche en Communication (CRC). Ce problème a été reconnu comme un problème nouveau [2]. En effet, notre recherche bibliographique montrait que ce problème a été attaqué partiellement dans les cas particuliers plus simples.

L'étude de l'occupation des canaux d'un système de communication desservant une population infinie est identifiable au problème de "l'Interférence des machines" (3). L'affectation prioritaire des lignes est cependant ignorée en utilisant ce dernier modèle.

Plus récemment, le cas d'une population finie d'utilisateurs, tenant compte de l'affectation prioritaire des lignes, était l'objet d'une étude publiée par Brandt (4) en appliquant la théorie des promenades aléatoires. Le modèle de Brandt n'est pas non plus utilisable pour le problème en cours, car d'une part la population est supposée infinie et d'autre part il n'y a pas la réjection d'un appel de basse priorité à l'arrivée d'un appel de plus haute priorité. On verra plus tard que les résultats de Brandt peuvent être obtenus comme un cas particulier de notre modèle.

La théorie des processus multi-dimensionnels de mort et de naissance a été appliquée avec succès à la résolution des "problèmes d'échange", dans l'étude des systèmes téléphoniques [5]. Cette technique sera appliquée pour la résolution de notre modèle.

Finalement, d'autres travaux théoriques concernant l'étude de la superposition des trafics ont été publiés par Palm [6], Cox et Smith [7], Newland [8] et O'Dell [9]. Ces travaux formaient la base mathématique de la théorie de congestion multi-dimentionnelle.

# CHAPITRE 2

# DEFINITION DU PROBLEME

L'étude entreprise dans ce travail, telle que spécifiée dans le contrat, se situe dans le contexte d'une analyse mathématique d'un réseau de communication par satellite ayant les particularités suivantes:

a- un nombre restreint d'utilisateurs

b- les utilisateurs forment des groupes de faible dimension, chaque groupe générant un trafic qui lui est propre

c- les appels possèdent différentes priorités de service et cette priorité permet la réjection d'un appel de plus faible priorité lorsque nécessaire.

L'objet de ce chapitre est le développement d'un modèle mathématique représentant le système spécifié ci-dessus.

## 2.1 *Sous-populations d'utilisateurs*

Soit M le nombre total d'utilisateurs du réseau. Ces utilisateurs se groupent en S sous-populations, désignées respectivement par $S_i$, $i = 1,2,\ldots,S$. Le nombre d'utilisateurs faisant partie de la sous-population $S_i$ est indiqué par $m_i$, $i = 1, 2,\ldots,S$. On a évidemment:

$$M = \sum_{i=1}^{S} m_i \tag{1}$$

## 2.2 *Caractérisation d'un appel*

Chaque appel est caractérisé par trois paramètres:

| | | |
|---|---|---|
| - sa priorité | k | k = 1,2,...,P |
| - son origine | i | i = 1,2,...,S |
| - sa destination | j | j = 1,2,...,S |

Il y a P différentes catégories de priorités, la catégorie k = 1 étant la priorité de plus grand poids. Plus généralement, lorsque nécessaire, un appel de priorité p peut interrompre le service d'un appel de priorité q, à condition que:

$$q > p$$

La réjection d'un appel de basse priorité à l'arrivée d'un appel de plus haute priorité aura lieu quand tous les canaux du système sont occupés.

## 2.3 *Caractérisation du trafic des sous-populations*

On base tout ce qui suit sur les deux hypothèses suivantes:

A-1 Chaque liaison libre entre les clients des sous-populations $S_i$ et $S_j$ génère des appels, de priorité k, suivant un processus de Poisson de paramètre

$$\lambda_k(i,j)$$

$$k = 1,2,\ldots,P \quad ; \quad i,j = 1,2,\ldots,S$$

A-2 Les temps de service des appels de priorité k, générés par des clients de la sous-population i et destinés à un utilisateur de la sous-population j, sont exponentiellement distribués avec paramètre

$$\mu_k(i,j)$$

$k = 1,2,\ldots,P$ $\qquad$ $i,j = 1,2,\ldots,S$

Note: La condition i = j , dans la définition du trafic d'une sous-population, indique le fait qu'il y a des connections via-satellite entre deux clients d'une même sous-population.

## 2.4 *Protocole de lignes*

A-3 Le protocole de lignes en vigueur dans le système est la priorité absolue. En d'autres termes, un appel de priorité k, sans distinction de son origine et de sa destination, est admis dans le système quand:

a- il existe un canal disponible

b- tous les canaux sont occupés mais il y a au moins un canal occupé par un appel de priorité plus basse, k'(k' > k)

c- s'il y a plusieurs appels de même priorité, sans distinction de leurs origines et destinations, qui sont susceptibles d'être rejetés à l'arrivée d'un appel de plus haute priorité, ces appels ont la même probabilité d'être rejetés.

## 2.5 *Vecteur d'état*

Sur le plan opérationnel, le système étudié n'est ni

plus ni moins qu'un ensemble de C canaux bilatéraux(duplex). L'état du système, à chaque instant, est défini par un ensemble de nombre de canaux reliant deux utilisateurs appartenant à deux sous-populations distinctes i et j et qui communiquent entre eux avec une priorité k. On désigne par:

$$n_k(i,j)$$

$$k = 1,2,\ldots,P \qquad\qquad i,j = 1,2,\ldots,S$$

le nombre de canaux occupés par des messages de priorité k, reliant $n_k(i,j)$ utilisateurs de la sous-population i au même nombre d'utilisateurs de la sous-population j.

Posons:

$$n = \sum_{k=1}^{P} \sum_{j=1}^{S} \sum_{i=1}^{S} n_k(i,j) \tag{2}$$

n désigne ainsi le nombre total des canaux du réseau occupés par des appels sans distinction de priorités, d'origines et de destinations.

Pour décrire mathématiquement l'état du système, on introduit le vecteur d'état:

$$\begin{aligned} N_{k;i,j}(n) &= \left\{n;\ldots,n_k(i,j),\ldots\right\} \\ k &= 1, 2,\ldots,P \;;\; i,j = 1, 2,\ldots,S \end{aligned} \tag{3}$$

Soit C le nombre des canaux du système. On voit nettement que:

$$0 \leqslant n \leqslant C$$

de même

$$n_k(i,j) = 0, 1, \ldots, C$$

Pour $n_k(i,j) = 0$, il n'y a pas d'appel de priorité k entre les clients des sous-populations i et j. Par contre pour $n_k(i,j) = C$, tous les canaux sont occupés par les C appels de priorité k entre les clients des sous-populations i et j.

Quand n = C tous les canaux sont occupés et pour n = 0 , on trouve le système à vide.

## 2.6 *Dimension du vecteur d'état*

La dimension du vecteur d'état est définie par le nombre des composantes de ce vecteur. D'après la définition de $n_k(i,j)$, on voit aisément que le vecteur d'état a pour dimension:

$$D = P \times S \times S \tag{4}$$

Etant donné un nombre n, 0 < n < C, le nombre des vecteurs d'état dans lesquels on a n canaux occupés est égal au nombre obtenu par l'occupation de n objets identiques des D cases différentes. Ce nombre de vecteurs d'état est donc(1)

$$\binom{n + D-1}{D - 1} = \frac{(n+D-1)!}{n!\ (D-1)!} \tag{5}$$

Ainsi on voit aisément que le nombre total de vecteurs d'état est:

$$F = \sum_{n=0}^{C} \binom{n + D - 1}{D - 1} \tag{6}$$

## 2.7 *Caractéristiques d'un vecteur d'état*

Soit un vecteur d'état tel que défini par l'équation (3). De chaque vecteur d'état on peut déduire deux paramètres caractérisant des sous-populations associées avec cet état. D'abord la somme:

$$m_i' = \sum_{k=1}^{P} \sum_{j=1}^{S} n_k(i,j) + n_k(j,i) \tag{7}$$

donne le nombre total des appels, sans distinction de priorité, destinés à la station i ainsi que ceux générés par la station i. Donc $m_i'$ représente bien le nombre d'utilisateurs de la sous-population i qui sont actifs, étant donné que le système est dans l'état considéré. Ensuite la différence:

$$m_i'' = m_i - m_i' \tag{8}$$

définit le nombre des utilisateurs libres de la sous-population i, associé avec un état donné du système. On voit clairement que:

$$0 \leqslant m_i' \leqslant C$$

et

$$m_i - C \leqslant m_i'' \leqslant m_i$$

Dans les situations réelles, on peut supposer que le nombre des utilisateurs d'une sous-population est supérieur au nombre des canaux du système:

$$m_i - C > 0 \ ^*$$

Le produit $m_i'' \cdot m_j''$ représente le nombre de liaisons entre les sous-populations $S_i$ et $S_j$, quand il y a $m_i''$ ucilisateurs libres dans $S_i$ et $m_j''$ utilisateurs libres dans $S_j$. Chacune de ces liaisons est une source d'appel, car correspondant à une liaison, il faut deux utilisateurs de deux sous-populations distinctes, ou d'une même sous-population.

---

* Le cas où $m_i - C \leqslant 0$ n'est pas réaliste, et cette condition n'entraîne aucun changement dans la mise en équation du modèle et la solution des équations d'état dans ce cas n'est simplement qu'un cas particulier de celui que nous développons dans ce qui suit.

# CHAPITRE 3

# EQUATIONS D'ETAT

Dans ce chapitre, on s'intéresse à une méthode générale d'établir les équations d'état. On considère que le système sous considération peut être modelé par un processus multi-dimentionnel de mort et de naissance.

## 3.1 *Taux de génération des appels*

Selon l'assomption A-1, il est clair que pour $0 \leqslant n \leqslant C-1$, le taux de génération des appels de priorité k, d'origine i et destination j, est donné par:

$$\lambda_k(i,j) \quad m_i'' \times m_j''$$

On notera aussi que, quand tous les canaux du système sont occupés par des appels de priorité, la plus basse égale à $k_o$, tout appel de priorité $k < k_o$ ne sera plus admis dans le système. Posons:

$$n(k_o) = \sum_{k=1}^{k_o} \sum_{i,j} n_k(i,j) \tag{9}$$

on a:

$$\lambda_k \left[ N_{k,k_o;i,j}(C);i,j \right] = \begin{cases} 0 \text{ quand } n(k_o) = C, \text{ et } k < k_o \\ \lambda_k(i,j) \times m_i'' \times m_j'' \text{ pour } k > k_o \end{cases} \tag{10}$$

## 3.2 *Equations d'état en régime transitoire et sans rejection.*

Etant donné un vecteur d'état $N_{k;i,j}(n)$, nous appelons vecteurs d'état adjacents de ce vecteur $N_{k;i,j}(n)$ les deux vecteurs qui se déduisent de $N_{k;i,j}(n)$ par:

a- l'admission d'un nouvel appel de priorité k, d'origine i et destination j dans le système. On désigne ce nouveau vecteur par $N_{k;i,j}(n+1)$ , et

b- la terminaison du service d'un des appels en cours, supposé de priorité $\kappa$, d'origine i et destination j. Ce nouveau vecteur est désigné par $N_{k;i,j}(n-1)$.

On constate aisément que les seules composantes et paramètres caractérisant des vecteurs adjacents sont donnés par le tableau suivant. Tous les autres composantes et paramètres caractérisant ces vecteurs restent identiques.

Tableau 1: Les seules différences entre les composantes et paramètres des vecteurs d'état adjacents.

| $N_{k;i,j}(n)$ | $N_{k;i,j}(n-1)$ | $N_{k;i,j}(n+1)$ |
|---|---|---|
| $n_k(i,j)$ | $n_k(i,j) - 1$ | $n_k(i,j) + 1$ |
| $m''_i$ | $m''_i + 1$ | $m''_i - 1$ |
| $m''_j$ | $m''_j + 1$ | $m''_j - 1$ |

Soit $P\left[N_{k;i,j}(n);\ t\right]$ la probabilité de trouver le système à l'état $N_{k;i,j}(n)$, à l'instant t. Par suite des hypothèses A-3-1 et A-3-2, on trouve facilement la probabilité que le système soit au même état à l'instant t + dt;

$$P\left[N_{k;i,j}(n);\ t+dt\right] = P\left[N_{k;i,j}(n);\ t\right] +$$

$$\sum_{k,i,j}\Big\{-\Big(\lambda_k(N_{k;i,j}(n);i,j) + \mu_k(i,j)n_k(i,j)\Big)dt\ P\left[N_{k;i,j}(n);\ t\right]$$

$$+ \quad \lambda_k(N_{k;i,j}(n-1);i,j)\ dt\ P\left[N_{k;i,j}(n-1);\ t\right]$$

$$+ \quad (1+n_k(i,j))\ \mu_k(i,j)\ dt\ P\left[N_{k;i,j}(n+1);\ t\right]\Big\} \qquad (11)$$

avec

$$0 \leqslant n \leqslant C-1, \quad 1 \leqslant k \leqslant P, \quad i = j = 1, 2, \ldots S$$

La sommation dans le second membre se compose de D=P x S x S termes, chaque terme étant l'expression figurée dans les accolades définie par rapport à une composante des vecteurs d'état. Transposant $P\left[N_{k;i,j}(n);\ t\right]$ du second membre au premier, divisant ensuite par dt et faisant tendre dt à zéro, on obtient les équations d'état en régime transitoire sans réjection, définies pour tout vecteur d'état associé avec $0 \leqslant n \leqslant C-1$.

$$P'\left[N_{k;i,j}(n);\ t\right] =$$

$$\sum_{k,i,j}\Big\{-\Big(\lambda_k(N_{k;i,j}(n);i,j) + \mu_k(i,j)\ n_k(i,j)\Big)\ P\left[N_{k;i,j}(n);\ t\right]$$

$$+ \quad \lambda_k(N_{k;i,j}(n-1);i,j) \quad P\left[N_{k;i,j}(n-1)\ ;\ t\right]$$

$$+ \quad (1+n_k(i,j))\ \mu_k(i,j) \quad P\left[N_{k;i,j}(n+1)\ ;\ t\right]\Big\} \tag{12}$$

### 3.3 *Equations d'état en régime transitoire avec réjection*

Nous allons d'abord étudier l'effet de la réjection d'un appel en cours par un appel de plus haute priorité. Soit à l'instant t + dt le vecteur d'état

$$N_{k\cdot k_o;i,j}(C) = \left\{C;\ldots n_k(i,j) + 1,\ldots n_{k_o}(i',j'),\ 0..0\right\} \tag{13}$$

où l'indice $k_o$ est introduit pour indiquer la priorité la plus basse des appels présents. Supposons que $N_{k_o;i,j}$ (C) résulte de de la réjection d'un appel de priorité s, causée par l'arrivée d'un appel de priorité k, k < s. A l'instant t, c'est-à-dire avant la réjection, le système est à l'état

$$N_{k,s;i'',j''}(C) = \left\{C;\ldots n_k(i,j),..n_{k_o}(i',j')..n_s(i'',j''),\ 0..0\right\} \tag{14}$$

On voit bien que les composantes des vecteurs (13) et (14) doivent être identiques sauf pour les deux cas suivants:

Cas 1:

$$k_o < s$$

$n_s(i",j")$ dans le vecteur (14) doit être égale à 1.

Cas 2:

$$k_o = s$$

la composante $n_k(i,j)$ dans le vecteur (14) doit être égale à $n_k(i,j) + 1$. De plus, il est clair que l'appel qui vient d'être admis dans le système doit être de priorité la plus basse (égale à $k_o$); ainsi on a:

$$k \leqslant k_o$$

$$k_o \leqslant s$$

et $$k < s$$

Cette remarque nous permet de trouver tous les états $N_{k,s;i",j"}(C)$ étant donné un vecteur $N_{k,k_o;i,j}(C)$.

Supposons que le système soit à l'état $N_{k,s;i",j"}(C)$ au temps t, la transition de cet état à l'état $N_{k,k_o;i,j}(C)$, pendant l'intervalle de temps infiniment petit dt, est déterminée par les deux évènements suivants:

$E_1$ : l'appel de priorité s ne termine pas son service

$E_2$ : un appel de priorité k arrive

Les deux évènements sont indépendants; donc la proba-

bilité qu'ils se produisent simultanément pendant l'intervalle (t, t + dt) est donnée par:

$$\left[1 - n_s(i",j")\ \mu_s(i",j")\ \ dt\right]\ \ \lambda_k\left[N_{k,s;i",j"}(C);i,j\right] dt$$

et en négligeant les infiniment petits d'ordre supérieur de dt, on obtient:

$$\lambda_k\ \left[N_{k,s;i",j"}(C);i,j\right]\ dt \tag{15}$$

La probabilité de trouver le système dans l'état $N_{k,k_o;i,j}(C)$ à l'instant t + dt peut être déterminée sans difficulté comme suit:

$$P\left[N_{k,k_o,i,j}(C);\ t + dt\right] = P\left[N_{k,k_o,i,j}(C);\ t\right] +$$

$$\sum_{k,i,j}\Big\{-\left[\lambda_k\left[N_{k,k_o,i,j}(C);i,j\right] + \mu_k(i,j)\ n_k(i,j)\right] dt \times P\left[N_{k,k_o,i,j}(C)\right]$$

$$+ \quad \lambda_k\left[N_{k,k_o,i,j}(C-1);i,j\right]\ dt\ \ P\left[N_{k,k_o,i,j}(C-1);\ t\right] +$$

$$\sum_{i"=1}^{S}\ \sum_{j"=1}^{S}\ \sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\left[N_{k,s;i",j"}(C);i,j\right]\ dt \times P\ \left[N_{k,s;i",j"}(C);\ t\right]\Big\} \tag{16}$$

De l'équation (16) on obtient facilement l'équation différentielle en:

$$P\left[N_{k;i,j}(C);\ t\right] \text{ i.e. } P'\left[N_{k;i,j}(C);\ t\right] =$$

$$\sum_{k,i,j}\Big\{-\left[\lambda_k\left[N_{k,k_o;i,j}(C);i,j\right] + \mu_k(i,j)n_k(i,j)\right] P\left[N_{k,k_o;i,j}(C);\ t\right]$$

$$+ \lambda_k\left[N_{k,k_o;i,j}(C-1);i,j\right] P\left[N_{k,k_o;i,j}(C-1);\ t\right] +$$

$$\sum_{i''=1}^{S}\sum_{j''=1}^{S}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\left[N_{k,s;i'',j''}(C);i,j\right] P\left[N_{k,s;i'',j''}(C);\ t\right]\Big\} \qquad (17)$$

La triple sommation dans les accolades de l'équation (17) s'appelle le terme de réjection.

## 3.4 *Equations d'état en régime permanent*

On constate aisément que le système possède un régime permanent. Posons:

$$\lim_{t\to\infty} P\left[N_{k;i,j}(n);\ t\right] = P\left[N_{k;i,j}(n)\right]$$

Cette limite s'appelle la probabilité stationnaire de l'état $N_{k;i,j}(n)$.

Pour $t \to \infty$, les dérivées s'annulent dans les équations (12) et (17). On a alors:

$$\sum_{k,i,j}\Big\{-\Big[\lambda_k(n_{k;i,j}(n);i,j) + \mu_k(i,j)n_k(i,j)\Big]\, P\Big[N_{k;i,j}(n)\Big]$$

$$+ \quad \lambda_k(N_{k;i,j}(n-1);i,j) \quad P\Big[N_{k;i,j}(n-1)\Big]$$

$$+ \quad (1 + n_k(i,j))\ \mu_k(i,j)\ P\Big[N_{k;i,j}(n+1)\Big]\Big\} = 0$$

$$\text{pour} \qquad 0 \leqslant n \leqslant C-1. \tag{18}$$

et

$$\sum_{k,i,j}\Big\{-\Big[\lambda_k(N_{k,k_o;i,j}(C);i,j) + \mu_k(i,j)n_k(i,j)\Big]\, P\Big[N_{k,k_o;i,j}(C)\Big]$$

$$+ \quad \lambda_k(N_{k;k_o;i,j}(C-1);i,j) \quad P\Big[N_{k,k_o;i,j}(C-1)\Big] \quad +$$

$$\sum_{i''=1}^{S}\sum_{j''=1}^{S}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\Big[N_{k,s;i'',j''}(C);i,j\Big]\ P\Big[N_{k,s;i'',j''}(C)\Big]\Big\} = 0 \tag{19}$$

### 3.5 *Equations d'état du système sans réjection*

Il est intuitivement clair que pour les mêmes sous-populations générant les mêmes trafics et pour le même nombre de canaux $0 \leqslant n \leqslant C-1$, les fonctionnements des systèmes avec et sans réjection semblent analogues. La différence entre ces deux systèmes se manifeste seulement quand tous les canaux sont occupés. Cette remarque nous conduit directement aux équations d'état du système sans réjection comme ci-dessus, pour $n = C$.

$$\sum_{k,i,j} \Big\{ -\mu_k(i,j)n_k(i,j) \quad P\Big[N_{k;i,j}(C)\Big]$$

$$+ \quad \lambda_k\Big[N_{k;i,j}(C-1);i,j\Big] \; P\Big[N_{k;i,j}(C-1)\Big] \Big\} = 0 \qquad (20)$$

L'équation (20) s'obtient de l'équation (19) en:

- égalant le terme de réjection à zéro, et
- substituant $\lambda_k\Big[N_{k;i'',j''}(C);i,j\Big] = 0$ pour tout k,i,j

car quand le système est entièrement occupé aucun appel n'est admis dans le système. Pour $0 \leqslant n \leqslant C-1$, les équations d'état correspondantes sont données par l'équation (18).

On verra plus tard que l'étude du système sans réjection nous servira de guide dans la résolution du système avec réjection.

## 3.6 *Equations d'état du système sans réjection desservant une population infinie.*

Dans le cas particulier d'un système sans réjection desservant une population infinie, on peut obtenir les équations d'état simplement en substituant dans les équations (18) et (20), les taux de génération des appels par des constantes. Ce cas a été traité par Brandt [4]. Les détails des calculs seront donnés dans la section 4.3.

# CHAPITRE 4

# RESOLUTION DES EQUATIONS D'ETAT EN REGIME PERMANENT

Dans ce chapitre on s'intéressera tout d'abord à la résolution des équations d'état des systèmes sans réjection. Le résultat obtenu nous sera bien utile dans la résolution des équations d'état de notre modèle.

## 4.1 *Système sans réjection*

On trouvera sans peine que la solution des équations (18) et (20) est de la forme:

$$P\left[N_{k;i,j}(n)\right] = P(\vec{0}) \prod_{k,i,j} \frac{m_{i!}''}{n_k(i,j)!} \rho_k^{n_k(i,j)}(i,j) \qquad (21)$$

avec $k = 1,2,\ldots P, \qquad i,j = 1,2,\ldots P.$

où

$$\rho_k(i,j) = \frac{\lambda_k(i,j)}{\mu_k(i,j)} \qquad (22)$$

désigne l'intensité du trafic de priorité k, d'origine i et destination j.

$P(\vec{0})$ représente la probabilité que le système soit à l'état vide, en régime permanent.

$P(\vec{0})$ se détermine par la normalisation habituelle:

$$P(\vec{0})=\left[1+\sum_{\varepsilon}\prod_{i,j,k}\frac{m''_i!}{n_k(i,j)!}\;\rho_k^{n_k(i,j)}(i,j)\right]^{-1} \tag{23}$$

Le résultat énoncé ci-dessus se justifie simplement en utilisant l'équation (21) et en appliquant la définition des vecteurs d'état adjacents on a, pour

$$0 \leqslant n \leqslant C$$

$$P\left[N_{k;i,j}(n+1)\right] = \rho_k(i,j)\;\frac{m''_i\;m''_j}{n_k(i,j)+1}\;P\left[N_{k;i,j}(n)\right] \tag{24}$$

et

$$P\left[N_{k;i,j}(n-1)\right] = \frac{n_k(i,j)}{\rho_k(i,j)\,(m''_i+1)\,(m''_j+1)}\;P\left[N_{k;i,j}(n)\right] \tag{25}$$

Substituant les équations (24) et (25) dans les équations (18) et (20), on voit aisément que les dernières sont identiquement vérifiées.

Retournons maintenant au système avec réjection.

## 4.2 *Système avec réjection*

La résolution des équations d'état du système avec réjection est beaucoup plus laborieuse. Pour commencer, nous réécrirons l'équation (18) pour

$$n = 0, 1, \ldots 2, C-1$$

$$\sum_{k,i,j} \Big\{ -\lambda_k(N_{k;i,j}(0);i,j) \quad P\left[N_{k;i,j}(0)\right] + \left[1+n_k(i,j)\right] \mu_k(i,j) \quad P\left[N_{k;i,j}(1)\right] \Big\} = 0 \tag{26}$$

$$\sum_{k,i,j} \Big\{ -\left[\lambda_k(N_{k;i,j}(1);i,j) + \mu_k(i,j)n_k(i,j)\right] P\left[N_{k;i,j}(1)\right] + \lambda_k\left[N_{k;i,j}(0);i,j\right] P\left[N_{k;i,j}(0)\right] + (1+n_k(i,j))\, \mu_k(i,j)\left[P\ N_{k;i,j}(2)\right] \Big\} = 0 \tag{27}$$

$$\sum_{k,i,j} \Big\{ -\left[\lambda_k(N_{k;i,j}(C-2);i,j) + \mu_k(i,j)n_k(i,j)\right] P\left[N_{k;i,j}(C-2)\right] + \lambda_k(N_{k;i,j}(C-3);i,j) \quad P\left[N_{k;i,j}(C-3)\right] + (1+n_k(i,j))\, \mu_k(i,j) \quad P\left[N_{k;i,j}(C-1)\right] \Big\} = 0 \tag{28}$$

$$\sum_{k,i,j} \Big\{ -\left[\lambda_k(N_{k;i,j}(C-1);i,j) + \mu_k(i,j)n_k(i,j)\right] \quad P\left[N_{k;i,j}(C-1)\right] + \lambda_k\left[N_{k;i,j}(C-2);i,j\right] \quad P\left[N_{k;i,j}(C-2)\right] + (1+n_k(i,j))\, \mu_k(i,j) \quad P\left[N_{k;i,j}(C)\right] \Big\} = 0 \tag{29}$$

Et pour $n = C$

$$\sum_{k,i,j}\Big\{ -\lambda_k\left[N_{k,k_o;i,j}(C);i,j\right] + \mu_k(i,j)n_k(i,j)\Big] P\left[N_{k,k_o;i,j}(C)\right]$$

$$+ \qquad \lambda_k\left[N_{k,k_o;i,j}(C-1);i,j\right] P\left[N_{k,k_o;i,j}(C-1)\right] \qquad +$$

$$\sum_{i''}\sum_{j''}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\left[N_{k;s;i'',j'')}(C);i,j\right] P\left[N_{k,s;i'',j''}(C)\right]\Big\} = 0 \qquad (30)$$

En ajoutant ces équations on obtient:

$$\sum_{k,i,j}\Big\{-\lambda_k\left[N_{k,k_o;i,j}(C);i,j\right] P\left[N_{k,k_o;i,j}(C)\right] +$$

$$\sum_{i''}\sum_{j''}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\left[N_{k;s;i'',j''}(C);i,j\right] P\left[N_{k,s;i'',j''}(C)\right]\Big\} = 0 \qquad (31)$$

En substituant l'équation (31) dans l'équation (19), on trouve:

$$\sum_{k,i,j}\Big\{-\mu_k(i,j)n_k(i,j) \quad P\left[N_{k,k_o;i,j}(C)\right]$$

$$+ \quad \lambda_k(N_{k,k_o;i,j}(C-1);i,j \quad P\left[N_{k,k_o;i,j}(C-1)\right]\Big\} = 0 \qquad (32)$$

On voit nettement que l'équation (32) est identique avec l'équation (20), et que l'équation (31) relie le taux de génération des appels des sous-populations, à un état donné quand tous les canaux sont occupés, avec ceux des états qui peuvent transformer à cet état par une transaction avec réjection.

Il est maintenant évident que:

1- la distribution des probabilités de l'occupation des canaux du système avec réjection est identique avec celle du système sans réjection. Cette distribution est déterminée par les équations (21) et (23).

2- quand les canaux sont tous occupés dans le système avec réjection, les taux de génération des appels n'est plus proportionnel avec les liaisons mais doit satisfaire l'équation.

$$\sum_{k,i,j}\Bigg\{-\lambda_k\Big[N_{k,k_o;i,j}(C);i,j\Big]\frac{\rho_k(i,j)}{n_k(i,j)} \times \frac{1}{(m_i''+1)(m_j''+1)} +$$

$$\sum_{i''=1}^{S}\sum_{j''=1}^{S}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\Big[N_{k,s;i'',j''}(C);i,j\Big]\frac{\rho_s(i'',j'')}{m_i''\,m_j''\,(n_s(i'',j'')+1)}\Bigg\} = 0 \qquad (33)$$

L'équation (33) s'obtient en substituant l'équation (21) dans l'équation (31), et avec la convention:

$$\lambda_k\Big[N_{k;k_o;i,j}(C);i,j\Big] \equiv 0 \qquad (34)$$

pour $k \geqslant k_o$ en tenant compte du fait que tout appel de priorité égale ou plus basse que celle de priorité la plus basse présente dans le système, avec tous ses canaux occupés, n'est pas admis dans le système. L'équation (33) exprime clairement la condition d'équilibre du système avec réjection.

Nous allons terminer ce chapitre en considérant le cas intéressant d'une population infinie.

## 4.3 *Cas d'une population infinie*

Pour le cas où la population est infinie, on a le taux de génération d'appel, entre les sous-populations $S_i$ et $S_j$, et de priorité k, indépendant du nombre de liaison.

Précisément pour:

$$n \leqslant C-1$$

on obtient:

$$\lambda_k\left[N_{k;i,j}(n);i,j\right] = \lambda_k(i,j)$$

Les équations d'état dans ce cas sont, pour $0 \leqslant n \leqslant C-1$

$$\sum_{k,i,j}\Big\{-\left[\lambda_k(i,j) + \mu_k(i,j)\ n_k(i,j)\right]\ P\left[N_{k;i,j}(n)\right]$$

$$+ \qquad \lambda_k(i,j)\ P\left[N_{k,i,j}(n-1)\right]$$

$$+ \qquad (1+n_k(i,j))\ \mu_k(i,j)\ P\left[N_{k;i,j}(n+1)\right]\Big\} = 0 \qquad (35)$$

et pour n = C on n'a qu'à reprendre l'équation (19). Répétons l'opération faite à la section 4.3, on trouve que:

1- la distribution de la probabilité d'occupation des canaux, pour $0 \leqslant n \leqslant C$

$$P\left[N_{k;i,j}(n)\right] = P(\vec{0}) \prod_{i,j,k} \frac{\rho_k^{n_k(i,j)}}{n_k(i,j)!}(i,j) \tag{36}$$

2- l'équation reliant les taux de génération des appels, quand n = C, devient:

$$\sum_{k,i,j}\left\{-\lambda_k\left[N_{k,k_o;i,j}(C);i,j\right] \frac{\rho_k(i,j)}{n_k(i,j)}\right.$$

$$\left.+\sum_{i''}\sum_{j''}\sum_{\substack{s=k_o\\ s>k}}^{P}\lambda_k\left[N_{k,s;i'',j''}(C);i,j\right] \frac{\rho_s(i'',j'')}{n_s(i'',j'')+1}\right\} = 0 \tag{37}$$

et de même on a toujours la condition d'admission (35). Il est intéressant de noter que dans ce cas, on peut calculer explicitement la valeur de $P(\vec{0})$.

En effet, par le théorème des monomes multiples on a: (12, p. 92).

$$\sum_{\varepsilon(n)} P\left[N_{k;i,j}(n)\right] = \frac{1}{n!}\left(\sum_{i,j,k}\frac{\lambda_k(i,j)}{\mu_k(i,j)}\right)^n \tag{38}$$

Posons:

$$\rho = \sum_{i,j,k}\rho_k(i,j) \tag{39}$$

L'équation (38) devient:

$$\sum_{\varepsilon(n)} P\left[N_{k;i,j}(n)\right] = \frac{\rho^n}{n!}$$

Où $\varepsilon(n)$ désigne l'ensemble des états du système contenant n appels. Alors applicant la formule (23) on a:

$$P(0) = \frac{1}{\sum_{n=o}^{C} \frac{\rho^n}{n!}} = \frac{1}{E_n} \tag{40}$$

où

$$E_n = \sum_{n=o}^{C} \frac{\rho^n}{n!}$$

# CHAPITRE 5

# INDICES DE PERFORMANCE

Ce chapitre a pour objet la définition des principaux indices de performance du système.

## 5.1 *Trafic écoulé*

Le trafic écoulé est égal à la moyenne des canaux occupés en régime permanent.

$$Te = \sum_{n=o}^{C} n\, p\{n\} \tag{41}$$

où $p\{n\}$ est la probabilité d'avoir n appels dans le réseau sans distinction de priorité, d'origine et de destination. Ainsi pour une valeur de n

$$p\{n\} = \sum_{k,i,j} P\left[N_{k;i,j}(n)\right] \tag{42}$$

La sommation dans l'équation (42) s'étend sur les $\binom{n+D-1}{D-1}$ vecteurs d'état dans lesquels il y a n canaux occupés, (n = 0, 1, 2,...,C)

## 5.2 *Probabilité de congestion des canaux d'un appel de priorité $k_0$*

La probabilité de congestion des canaux pour un ap-

pel de priorité $k_o$ sans distinction de son origine et de sa destination est égale à la probabilité que le système soit complètement occupé par des messages de priorité la plus basse égale à $k_o$. Donc:

$$C_{k_o} = \sum_{\substack{k=1 \\ i,j}}^{k_o} P\left[N_{k,k_o;i,j}(C)\right] \tag{43}$$

La sommation s'étend sur le sous-espace des vecteurs d'état $N_{k;k_o;i,j}(C)$. Il est évident que la probabilité de congestion est commune pour toutes les sous-populations, car la priorité ne dépend pas des sous-populations.

## 5.3 *Probabilité de réjection d'un appel de priorité s et reliant les sous-populations i,j*

Prenons pour origine du temps le moment où le système est à l'état $N_{k,s;i'',j''}(C)$ dans lequel $n_s(i'',j'')$ est non nul. Soit t l'instant où la réjection d'un message de priorité s et reliant les sous-populations i'', j'' a lieu. Cette réjection se produit si les trois évènements suivants se produisent simultanément pendant (0,t):

E1 : le système se trouve à l'état $N_{k,s;i'',j''}(C)$

E2 : aucune terminaison de service des $n_s(i'',j'')$ appels en cours de priorité s et reliant les sous-populations i'', j''

E3 : exactement un appel de priorité $k < s-1$ arrive

On a:

$$P\left[E1\right] = P\left[N_{k,s;i'',j''}(C)\right] \tag{44}$$

$$P\left[E2\right] = \mu_s\left[N_{k,s;i'',j''}(C)\right] \exp\left[-\mu_s(N_{k,s;i'',j''}(C))\ t\right] dt \tag{45}$$

où

$$\mu_s\left[N_{k,s;i'',j''}(C)\right] = \mu_s(i'',j'').n_k(i'',j'') \tag{46}$$

et

$$P\left[E3\right] = \lambda_s(N_{k,s;i'',j''}(C))\ t \exp\left[-\lambda_s(N_{k,s;i'',j''}(C))\ t\right] \tag{47}$$

avec

$$\lambda_s(N_{k,s;i'',j''}) = \sum_{k=1}^{s-1} \sum_{i,j} \lambda_k\left[N_{k,s;i'',j''}(C),i,j\right]\ m_i''\ m_j'' \tag{48}$$

La probabilité de réjection d'un appel de priorité s et reliant les sous-populations (i",j") est donnée par:

$$Q_s(i'',j'') = \sum_{\varepsilon(s)} P\left[N_{k,s;i'',j''}(C)\right] \times \frac{1}{n_s(i'',j'')}$$

$$\int_0^\infty \mu_s\left[N_{k,s;i'',j''}(C)\right] \quad \lambda_s\left[N_{k,s;i'',j''}(C)\right] t \quad \times$$

$$\exp\left[-\ \mu_s\left[N_{k,s;i'',j''}(C)\right] - \lambda_s\left[N_{k,s;i'',j''}(C)\right]\right]\ t\ \ dt \tag{49}$$

En effectuant l'intégrale on a:

$$R_s(i'',j'') = \mu_s(i'',j'') \sum_{\varepsilon(s)} \frac{P\left[N_{k,s;i'',j''}(C)\right] \times \lambda_s\left[N_{k,s;i'',j''}(C)\right]}{\left\{\mu_s\left[N_{k,s;i'',j''}(C)\right] + \lambda_s\left[N_{k,s;i'',j''}(C)\right]\right\}^2} \quad (50)$$

La sommation dans l'équation (50) s'étend sur un sous-ensemble des vecteurs d'état caractérisés par le fait que la priorité s sont la priorité la plus basse présente dans le système et qu'il y a au moins un appel de priorité s reliant les sous-populations i",j".

Il est aussi intéressant de calculer la probabilité de réjection d'un appel de priorité s sans distinction de son origine et de sa destination. En répétant le raisonnement ci-dessus, on trouve:

$$R_s = \sum_{\varepsilon'(s)} P\left[N_{k,s;i,j}(C)\right] \frac{1}{n_s} \times$$

$$\frac{\mu_s\left[N_{k,s;i,j}(C)\right] \times \lambda_s\left[N_{k,s;i,j}(C)\right]}{\left[\mu_s\left[N_{k,s;i,j}(C)\right] + \lambda_s\left[N_{k,s;i,j}(C)\right]\right]^2} \quad (51)$$

où

$$n_s = \sum_{k=1}^{s-1} \sum_{i,j} n_k(i,j)$$

$$\mu_s = \sum_{k=1}^{S} \sum_{i,j} n_k(i,j)\ \mu_k(i,j)$$

$$\lambda_s = \sum_{k=1}^{s-1} \sum_{i,j} m_i''\ m_j''\ \lambda_k \left[N_{k,s;i,j}(C);i,j\right]$$

et $\varepsilon'(s)$ désigne le sous-ensemble des vecteurs d'état dont les appels sans distinction d'origine et destination, présent dans le système sont de priorité s.

## 5.4 *Probabilité de non admissibilité pour un appel* $(k_o, i^*j^*)$

La probabilité de non admissibilité pour un appel de priorité $k_o$ se définit de la même manière que la probabilité de réjection d'un appel de priorité s. Prenons pour origine du temps le moment où le système est à l'état $N_{k,s;i,j}(C)$. Supposons que l'appel $(k_o, i^*, j^*)$ arrive à l'instant t. Cet appel ne serait pas admis dans le système si les trois évènements suivants se produisent pendant le temps (o,t):

$E_1$ : le système est à l'état $N_{k,s;i,j}(C)$

$E_2$ : aucune terminaison de service des C appels en cours

$E_3$ : exactement un appel $(k_o; i^*, j^*)$ arrive

On a:

$$P\left[E_1\right] = P\left[N_{k,s;i,j}(C)\right]$$

$$P\left[E_2\right] = \mu_{k_o}\left[N_{k,s;i,j}(C)\right] \quad \exp\left[-\mu_{k_o}(N_{k,s;i,j}(C))\ t\right] dt$$

où

$$\mu_{k_o}(N_{k,s;i,j}(C)) = \sum_{k,i,j} \mu_k(i,j)\ n_k(i,j)$$

où la sommation est définie par rapport à l'état $N_{k,s;i,j}(C)$ et

$$P\left[E_3\right] = \lambda_{k_o}\left[N_{k_o,s;i,j}(C);i^*j^*\right] \exp\left[-\lambda_{k_o} N_{k_o,s;i,j}(C))\right] t$$

La probabilité de non admissibilité de l'appel $(k_{o;i^*j^*})$ est donnée par:

$$R_{k_o}(i^*j^*) = \sum_{\varepsilon(k_o)} P\left[N_{k,s;i,j}(C)\right] \times$$

$$\int_0^\infty \mu_{k_o}(N_{k,s;i,j}(C)) \quad \lambda_{k_o}(N_{k,s;i,j}(C);i^*j^*)$$

$$\exp\left[-\mu_{k_o}(N_{k,s;i,j}(C)) - \lambda_{k_o}\left[N_{k_o,s;i,j}(C);i^*j^*\right]\right] t.\ dt$$

où la sommation s'étend sur tout le sous-ensemble $\varepsilon(k_o)$ des états $N_{k,s;i,j}(C)$ tels que $s \geqslant k_o + 1$. En effectuant l'intégrale on a:

$$R_{k_o}(i^*j^*) = \sum_{\varepsilon(k_o)} P\left[N_{k_o,s;i,j}(C)\right] \times$$

$$\frac{\mu_{k_o}\left[N_{k,s;i,j}(C)\right] \quad \lambda_{k_o}\left[N_{k_o,s;i,j}(C);i^*j^*\right]}{\left\{\mu_{k_o}\left[N_{k,k_o;i,j}(C)\right] + \lambda_{k_o}\left[N_{k_o,s;i,j}(C);i^*j^*\right]\right\}^2} \tag{52}$$

avec

$\lambda_{k_o}\left[N_{k_{o,s;i,j}}(C)\right]$ définie par l'équation (19).

## 5.5 *Taux d'appels acceptés*

L'acceptation d'un appel (k;i,j) dépend de l'état du système. Aussi le taux d'appels acceptés est défini par:

$$\lambda = \sum_{k;i,j,n} P\left[N_{k;i,j}(n)\right] \lambda_k\left[N_{k;i,j}(n)\, i,j\right] \tag{53}$$

où le taux de génération des appels satisfait les équations de restriction (10) et la condition d'admission quand le système est saturé, définie par l'équation (33), et la sommation s'étend sur tout l'ensemble des états.

## 5.6 *Cas d'une population infinie*

Nous allons terminer ce chapitre en simplifiant les expressions des quelques indices de performance du système desservant une population infinie.

### 5.6.1 *Trafic écoulé. Efficacité*

En appliquant l'équation (41) et les équations (36) et (40), nous avons par la formule des monomes multiples:

$$T_{e\infty} = \frac{1}{E_C(\rho)} \sum_{n=o}^{C} n \sum_{k,i,j} \frac{\left[\rho_k(i,j)\right]^{n_k(i,j)}}{n_k(i,j)!}$$

$$T_{e\infty} = \frac{1}{E_C(\rho)} \sum_{n=o}^{C} n \frac{\rho^n}{n!} = \frac{1}{E_C(\rho)} \sum_{n=o}^{C-1} \frac{\rho^n}{m!} = \frac{E_{C-1}(\rho)}{E_C(\rho)}$$

où

$$E_C(\rho) = \sum_{i=o}^{C} \frac{\rho^i}{i!}$$

$$\rho = \sum_{k,i,j} \rho_k(i,j)$$

D'après la définition de $E_C(\rho)$ on peut écrire:

$$E(\rho) = \frac{T_{e\infty}}{\rho} = 1 - \frac{\rho^C/C!}{E_C(\rho)} = 1-E(C,s) \qquad (54)$$

Comme d'habitude $\rho$ est connu sous le nom de trafic offert. Ainsi le rapport $E(\rho)$, du trafic écoulé au trafic offert, est l'efficacité du système. Il est intéressant de noter que le deuxième membre de l'équation (54) n'est autre chose que la probabilité de non congestion des canaux du système. On en déduit donc que:

1- "L'efficacité du système est égale à la probabilité de non congestion des canaux".

2- "Le trafic écoulé est égal au produit du trafic offert par l'efficacité du système".

3- Le deuxième terme dans le deuxième membre de l'équation (54) est reconnu comme la formule classique d'Erlang [13], exprimant la perte du système.

### 5.6.2 *Probabilité de congestion des canaux pour un appel de priorité $k_o$*

Notons que pour un appel de priorité $k_o$, sans distinction de destination et d'origine, il n'y aurait de congestion de canaux que si tous les canaux du système étaient occupés par des appels de priorité la plus basse égale à $k_o$. Alors cette probabilité est donnée par:

$$C(k_o) = \frac{1}{E(\rho)} \cdot \prod_{k=1}^{k_o} \times \prod_{i,j=1}^{S} \frac{(\rho_k\ i,j)^{n_k(i,j)}}{n_k(i,j)!}$$

Posons:

$$\rho(k_o) = \sum_{k=1}^{k_o} \sum_{i,j=1}^{S} \rho_k(i,j)$$

et appliquons la formule des monomes multiples. On trouve aisément que:

$$C(k_o) = \frac{\rho(k_o)^C/C!}{E_C(\rho)}$$

et en exprimant en fonction de la formule de perte d'Erlang, on a:

$$C(k_o) = \left[\frac{\rho(k_o)}{\rho}\right]^C \cdot E(C,s) \qquad (55)$$

où

$$E(C,s) = \frac{\rho^C/C!}{E_i(\rho)}$$

désigne la perte d'Erglang.

L'équation (55) peut se réduire comme suit:

$$C(k_o) = \exp\left[-C \log \frac{\rho}{\rho(k_o)}\right] \cdot E(C,s)$$

On voit aisément que la perte des appels de priorité $k_o$ décroît exponentiellement en fonction du nombre de canaux C.

L'équation (55) est une généralisation de la formule d'Erlang au cas d'un système de communication desservant une population infinie avec une affectation prioritaire des lignes et avec réjection des appels de basse priorité à l'arrivée d'un appel de plus haute priorité.

# DEUXIEME PARTIE

# EXPLOITATION DES RESULTATS

## *INTRODUCTION*

Dans cette deuxième partie de ce rapport, l'exploitation des résultats obtenus dans la précédente sera traitée.

Elle comprend deux sections. Le chapitre 6 contient les techniques de base du calcul numérique des paramètres de performance du système. Le chapitre 7 illustre les résultats théoriques par un exemple numérique dont les calculs sont effectués suivant un programme développé dans le chapitre 6. Les résultats concernant un système avec une population infinie sont illustrées par des courbes représentatives.

# CHAPITRE 6

# TECHNIQUES DE CALCUL NUMERIQUE DES INDICES DE PERFORMANCE

Les résultats théoriques obtenus dans la première partie ne sont pas directement implémentables sur ordinateur.

On voit nettement que dans le calcul des indices de performance on a besoin des probabilités de trouver le système dans chacun de ses états. Il est évident que l'on doit d'abord générer l'ensemble des vecteurs d'état du système. Ceci est un problème très complexe si on considère le nombre des vecteurs d'état qui croît très rapidement en fonction des dimensions du système. Pour que l'exploitation des résultats théoriques soit réalisable dans le cas d'un système de dimension réelle, il faut prendre des précautions pour limiter le temps de calcul aussi bien que la longueur de la mémoire nécessaire du calculateur.

## 6.1 *Génération de l'ensemble des vecteurs d'état*

La génération de l'ensemble des vecteurs d'état s'effectue en deux opérations fondamentales. Rappelons en premier lieu la définition du vecteur d'état.

Chaque état du système est défini par un vecteur de dimension $D=PxS^2$ où P est le nombre des classes de priorité du

système et S le nombre de sous-populations desservies par le système. Chaque composante $n_k(i,j)$ avec $k=1,2,\ldots, P$ et $i,j = 1,2,\ldots, S$ est un nombre entier. La somme $n=\sum_{k,i,j} n_k(i,j)$ représente le nombre des appels en cours dans le système. Le nombre n est inférieur ou égal au nombre C des canaux du système.

Soit n un nombre entier tel que $1 \leq n \leq C$. Les deux opérations fondamentales de la génération de l'ensemble des vecteurs d'état sont:

a) la génération des partitions du nombre n et

b) la permutation des multi-ensembles.

L'opération (a) nous donne les valeurs des composantes $n_k(i,j)$ non nulles d'un vecteur d'état, étant donné que l'on trouve n appels présents dans cet état.

Etant donné le nombre n, il y a plusieurs états correspondant à un nombre n. L'opération (b) nous permet de trouver tous ces états en permutant les sous-ensembles des $n_k(i,j)$ non nulles obtenues par l'opération (a) parmi les D composantes du vecteur d'état.

Nous allons maintenant étudier en détail ces deux opérations.

## 6.2 *La génération des partitions d'un nombre entier n*

La génération des partitions d'un nombre entier n consiste à trouver toutes les collections de nombres entiers plus petits que n dont la somme est égale à n. Prenons pour exemple n = 4. Les partitions de n sont:

4, 3 + 1, 2 + 2, 2 + 1 + 1, 1 + 1 + 1 + 1.

On peut classer ces partitions comme dans le tableau ci-dessous:

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 4 |
| 0 | 0 | 3 | 1 |
| 0 | 2 | 0 | 2 |
| 0 | 2 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Tableau 2- Liste des partitions de n=4

Considérant chaque ligne comme un nombre, on voit clairement que ces partitions sont classées dans l'ordre croissant.

Le premier de ces nombres est égal àu nombre à partitionner et le dernier est formé par une suite de 1. Ceci nous conduit donc à représenter l'opération de partition par un ar-

bre comme l'indique la figure ci-dessous.

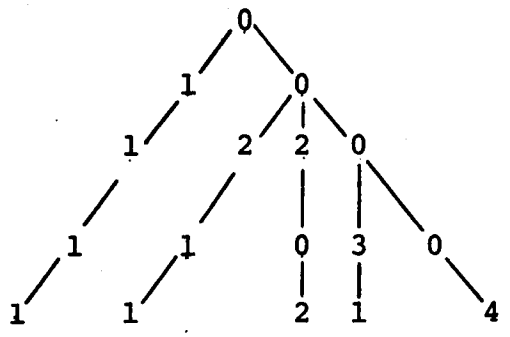


Figure 1- L'arbre de partition de 4

On vérifie sans peine que les observations ci-dessus sont applicables pour la partition d'un nombre n quelconque. L'implémentation de la génération des partitions sur l'ordinateur s'effectue suivant l'algorithme représenté par la figure 2. Sur la figure 2, on voit clairement que l'on n'a besoin, comme mémoire, que d'un seul régistre pour contenir la partition $B_i$ qui vient d'être engendrée.

## 6.3 *La permutation d'un multi-ensemble*

La permutation d'un multi-ensemble nous sert à engendrer le sous-ensemble de vecteurs d'état associés à une partition donnée $B_i$. Il est préférable d'illustrer cette technique par un exemple simple.

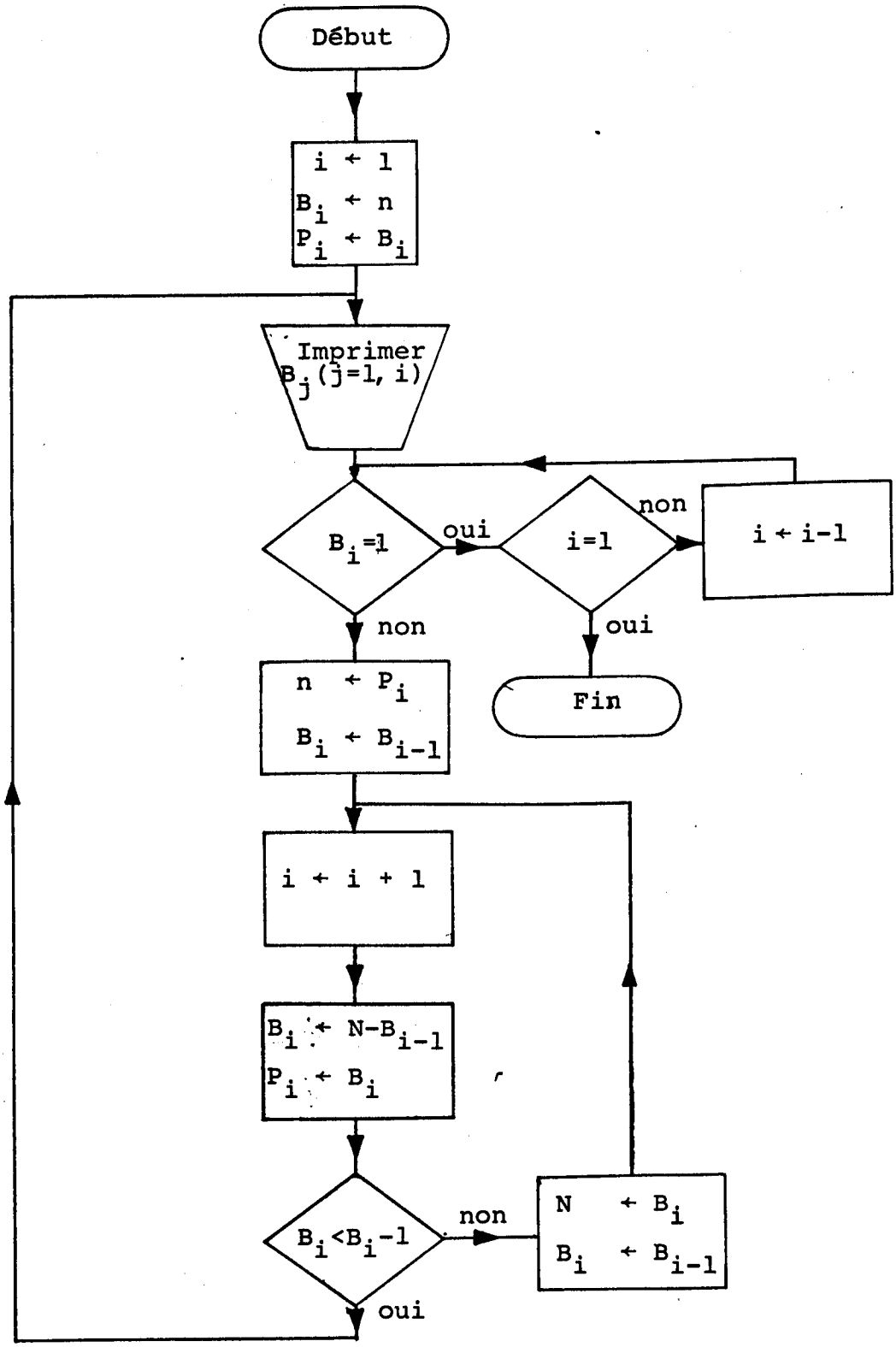


Figure 2- L'organigramme de la génération des partitions $B_i$ d'un nombre n

$P_i$ = régistre contenant la valeur de $B_i$ qui vient d'être engendrée

Considérons par exemple le sous-ensemble des vecteurs d'état, de dimension D=4, associés avec la partition

B= 1 1 3

du nombre n=5. Ces vecteurs sont donnés dans le tableau 3.

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 3 |
| 0 | 1 | 3 | 1 |
| 0 | 3 | 1 | 1 |
| 1 | 0 | 3 | 1 |
| 1 | 1 | 0 | 3 |
| 1 | 1 | 3 | 0 |
| 1 | 3 | 0 | 1 |
| 1 | 3 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 3 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

Tableau 3- Liste des vecteurs d'état de dimension D=4, associés avec la partition B= 1 1 3 du nombre n=5

Considérons chaque ligne dans ce tableau comme un nombre. On voit que ces nombres sont classés dans l'ordre croissant.

Le sous-ensemble de ces vecteurs est représentable par un arbre montré dans la figure 3. La première branche de l'arbre est associée avec le nombre le plus petit possible que l'on peut former avec les digits de la partition donnée.On passe d'une branche à l'autre en formant un nouveau nombre le plus petit possible mais plus grand que celui associé avec la branche que l'on vient de quitter tout en employant les digits de la partition donnée.

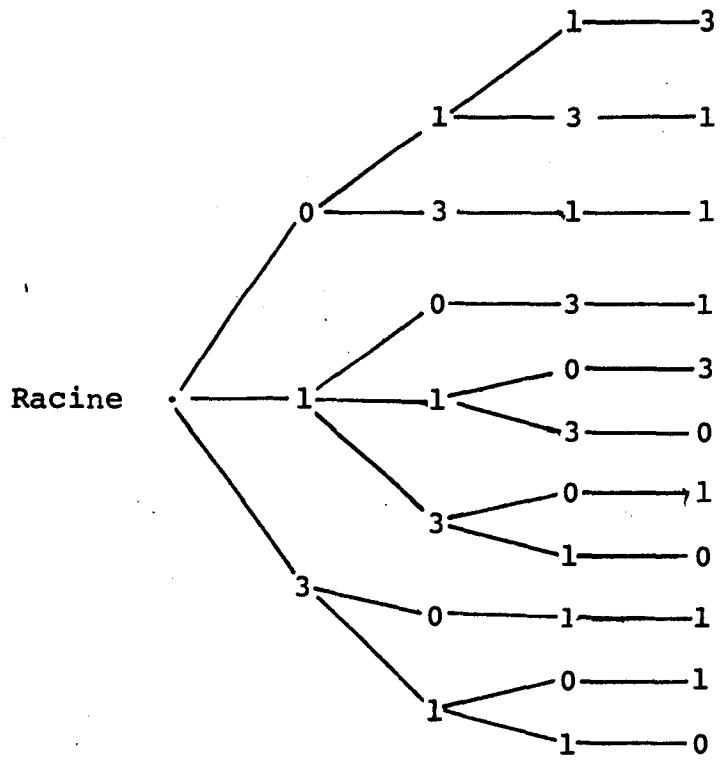


Figure 3- L'arbre représentant la permutation d'un multi-ensemble pour la génération des vecteurs d'état ayant D=4, et associés à la partition B= (1 1 3) du nombre n=5.

On continue ainsi jusqu'à la dernière branche qui correspond au plus grand nombre que l'on peut former avec les digits de la partition initiale. Le nombre de digits contenus dans le nombre maximum est égal à la dimension D des vecteurs d'état.

La technique de génération des vecteurs d'état associés avec une partition donnée décrite ci-dessus nous conduit au sous-programme de permutation d'un multi-ensemble illustré à la figure 4. Les notations employées dans ce sous-programme sont maintenant définies.

Considérons une partition donnée, par exemple:

$$B = 1\ 1\ 1\ 2\ 2\ \ 3$$

Admettons que la dimension du vecteur d'état est $D = 9$. Le plus petit nombre que l'on peut former avec les digits de B est donc:

$$A = 0\ 0\ 0\ \ 1\ 1\ 1\ \ 2\ 2\ \ \ 3$$

Notons que A est un vecteur d'état associé avec la partition B. Les éléments $A_i$ (i=1, D) se groupent en quatre groupes dans notre exemple:

$C_1$ : 000 avec $M_1$ = 3 éléments
$C_2$ : 111 avec $M_2$ = 3 éléments
$C_3$ : 22 avec $M_3$ = 2 éléments
$C_4$ : 3 avec $M_4$ = 1 élément

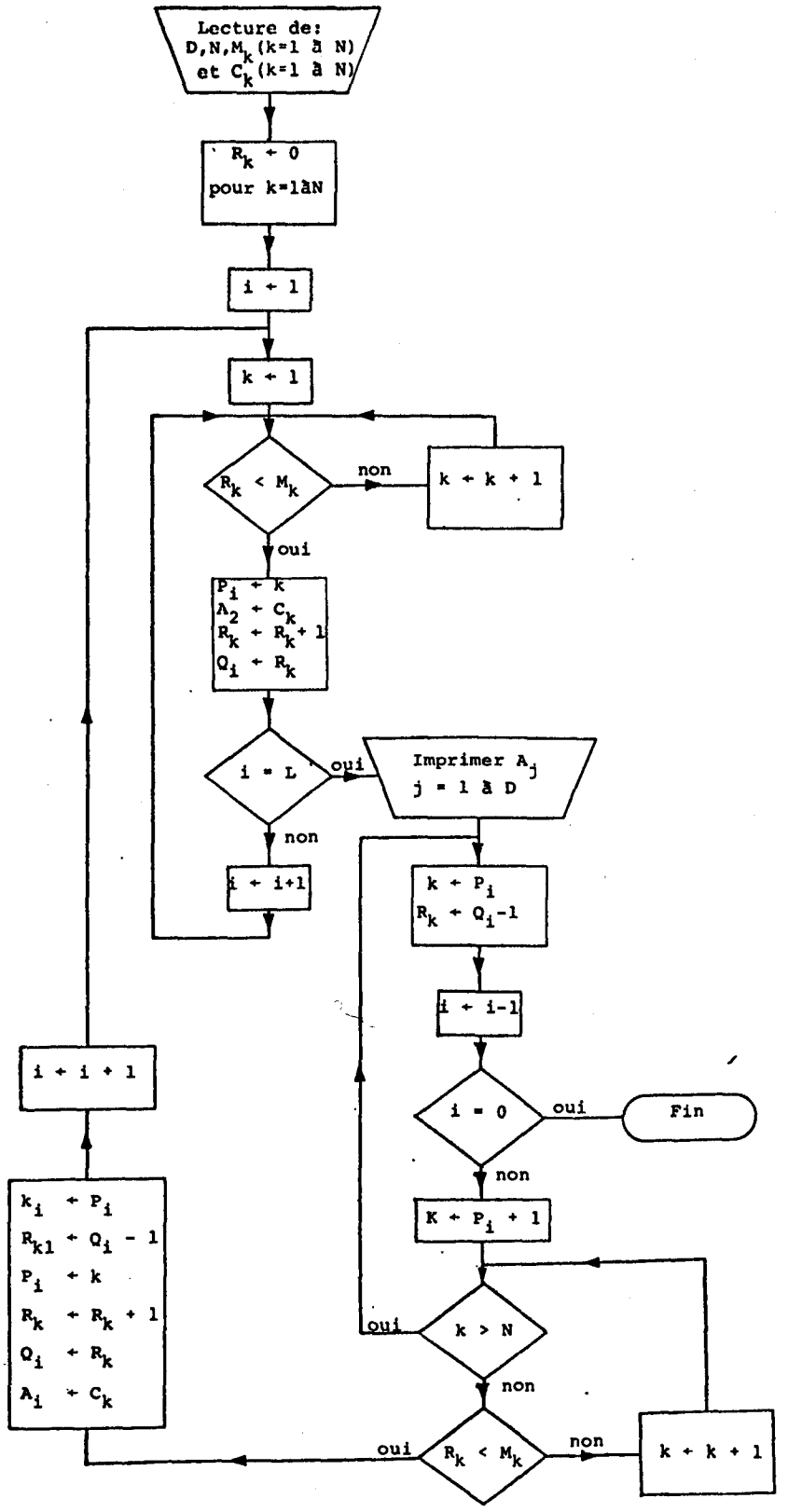


Fig. 4- L'organigramme de l'opération de génération de l'ensemble des vecteurs d'état associé avec une partition donnée $B_i$.

Nous désignons en plus:

$R_k$ : compteur des éléments de $C_k$ déjà utilisés ($1 \leq k \leq N$)

$P_i$ : régistre contenant la valeur de $A_i$ déjà engendrée

$Q_i$ : régistre contenant le nombre d'éléments de $C_k$ utilisés lorsque i éléments de A sont engendrés.

## 6.4 *Le programme principal*

Les techniques développées dans les sections 6.1 et 6.2 sont appliquées dans le programme principal pour le calcul des indices de performance du système défini dans le chapitre 5, précisément:

1- Le trafic écoulé

2- La probabilité de congestion des canaux d'un appel de priorité k

3- La probabilité de réjection d'un appel de priorité s d'origine i et destination j

4- La probabilité de non-admissibilité pour un appel d'origine i et destination j et de priorité k.

Il est évident que le programme a été conçu pour le cas le plus général d'un système de C canaux, P classe de priorité et S sous-populations avec C, P et S quelconques.

L'organigramme de ce programme est donné dans la figure 5. Le fait saillant est que chacun des paramètres caractéristiques du système est calculé partiellement au fur et à mesure qu'un vecteur d'état est engendré. Ceci nous permet de calcu-

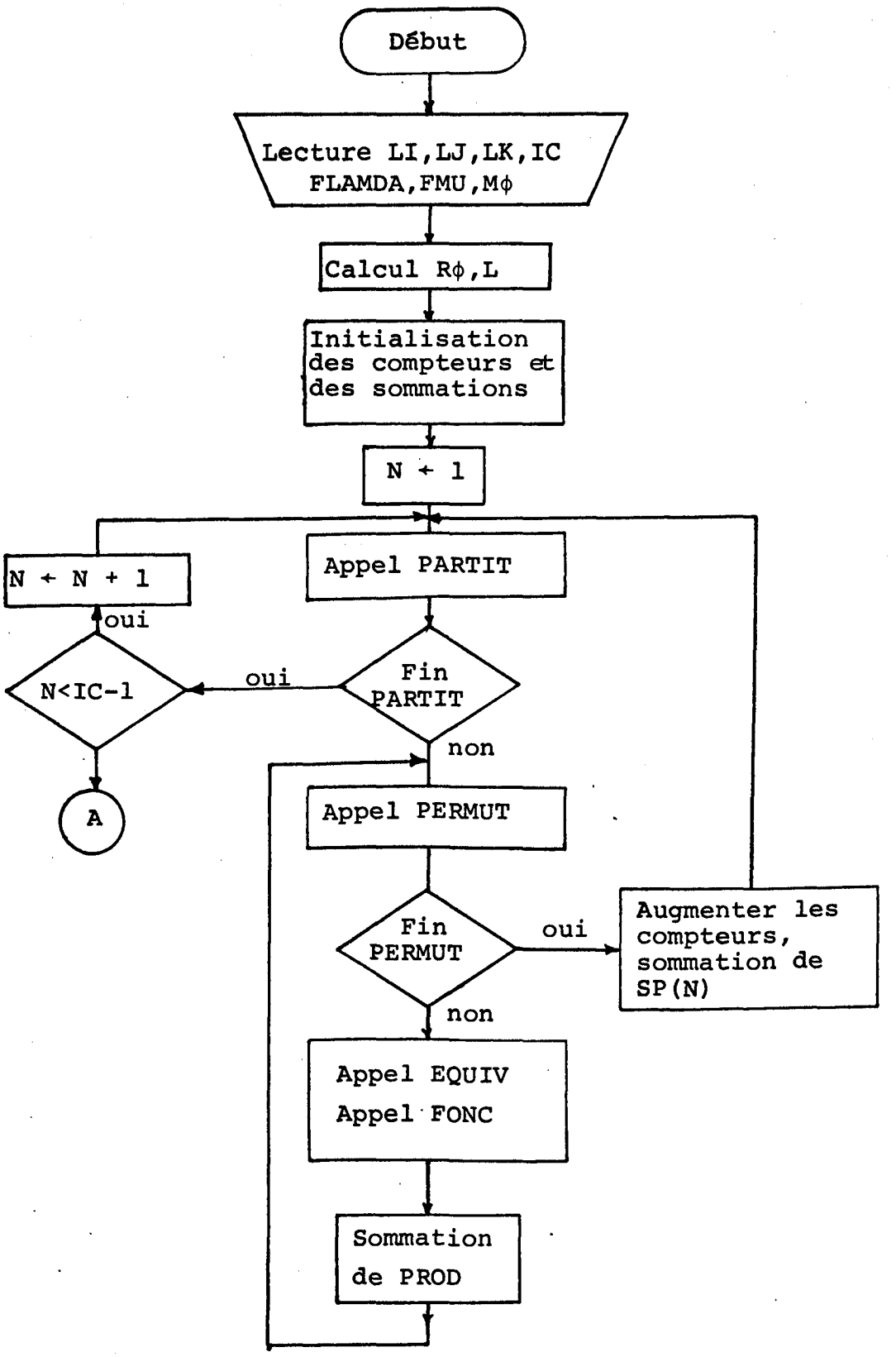


Fig. 5a- Organigramme du programme principal

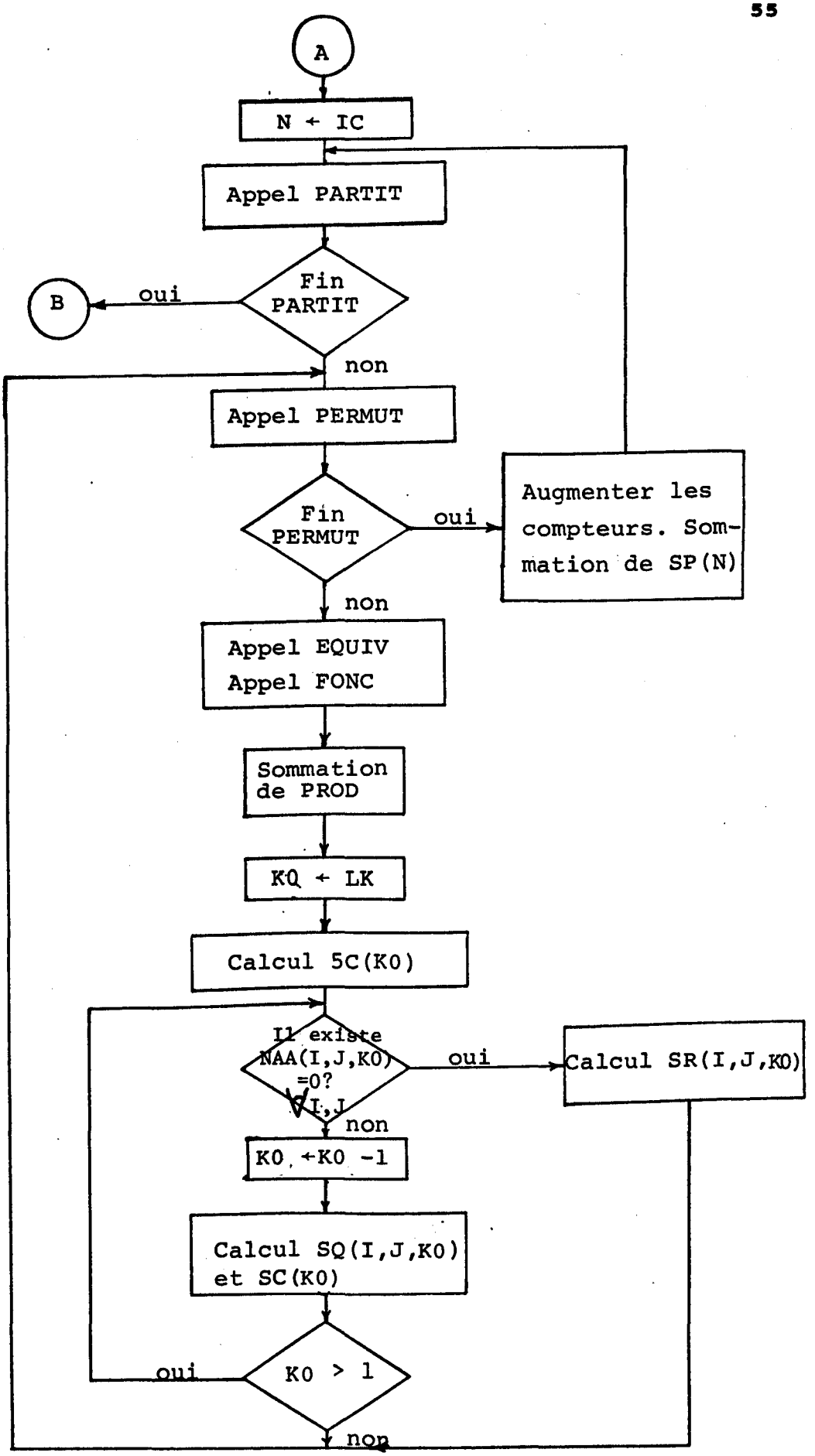


Fig. 5b- Organigramme du programme principal (suite)

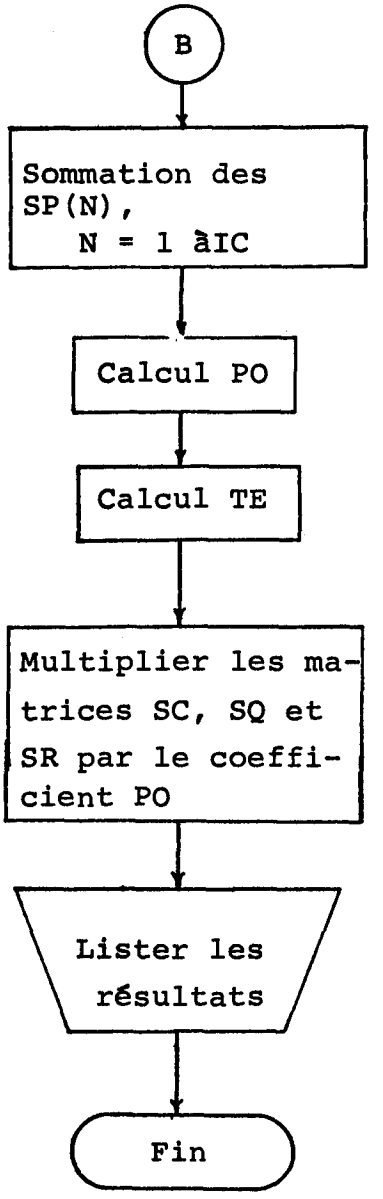


Fig. 5c- Organigramme du programme principal (suite et fin)

ler les indices de performance avec le minimum possible de temps de calcul et avec une longueur de mémoire minimale. Ainsi, en dépit de la dimension du problème, on peut obtenir les valeurs numériques des indices de performance d'un système réel dans un temps raisonnable. Les détails de calcul seront illustrés par un exemple numérique donné dans le chapitre suivant.

Les données sont:

1- C : nombre de canaux du système;

2- P : nombre de classes de priorité;

3- S : nombre de sous-populations

4- $||\lambda_k(i,j)||$ : matrice des taux de génération des appels entre la sous-population i et la sous-population j de priorité k; $(i,j = 1,2,\ldots,S,\ 1 \leqslant k \leqslant P)$;

5- $||\mu_k(i,j)||$ : matrice des taux de service des appels entre la sous-population i et la sous-population j et de priorité k, $(i,j = 1,2,\ldots,S,\ s \leqslant k \leqslant P)$

6- $[m_s]$ : vecteur des nombres des utilisateurs des sous-populations $(1 \leqslant s \leqslant S)$

*Calcul intermédiaire*

$||\rho_k(i,j)||$ : matrice des intensités de trafic de priorité k entre les sous-populations i et j $(i,j = 1,2,\ldots,S,\ 1 \leqslant k \leqslant P)$

Les résultats sont calculés partiellement au fur et à mesure qu'un vecteur d'état est engendré par les formules correspondantes comme l'indique le tableau 4 ci-dessous:

| Résultats du calcul | No. d'équation |
|---|---|
| Probabilité de trouver le système à vide | (23) |
| Trafic écoulé | (41) |
| Probabilité de congestion des canaux pour les appels de priorité k | (43) |
| Probabilité de réjection d'un appel de priorité $s_o$ entre les sous-populations i", j" | (50) |
| Probabilité de non-admissibilité d'un appel de priorité $1 \leqslant k_o \leqslant P-1$ entre les sous-populations i* et j* | (52) |

Tableau 4- Paramètres à calculer et leur équation correspondante.

*Sous-programmes*

Le programme principal englobe les sous-programmes ci-dessous:

1- Sous-programme PARTIT

2- Sous-programme PERMUT

3- Sous-programme EQUIV

4- Sous-programme FONCT

Les sous-programmes PARTIT et PERMUT sont écrits suivant les algorithmes discutés dans la section précédente. Le sous-programme PARTIT, chaque fois qu'il est appelé produit une nouvelle partition d'un nombre $1 \leqslant N \leqslant C$. Le sous-programme PERMUT, chaque fois qu'il est appelé engendre un nouveau vecteur d'état à partir de la nouvelle partition générée par le premier.

Le sous-programme EQUIV regroupe les composantes d'un vecteur d'état donné en P matrices de dimension S x S. Cette opération est nécessaire pour le calcul des probabilités de réjection et de non admissibilité d'un appel de priorité prédéterminée et d'origine et destination spécifiées.

Le sous-programme FONCT calcule pour chaque vecteur d'état:

a- le vecteur M2 des utilisateurs libres dans les sous-populations

b- la constante:

$$\text{FMU1} = \sum_{k,i,j} \mu_k(i,j)\, n_k(i,j)$$

dont on a besoin dans le calcul de la probabilité de non-admissibilité d'un appel

c- le produit:

$$\text{PROD} = \prod_{k,i,j} \frac{m_i^{\,n} \left[\rho_k(i,j)\right]^{n_k(i,j)}}{n_k(i,j)}$$

*Sommation partielle*

Les sommations dans le calcul du trafic écoulé et dans le calcul de la probabilité de trouver le système à l'état vide se calculent partiellement en trouvant:

a- la somme

$$P = \sum_{V} PROD$$

où V désigne le sous-ensemble des vecteurs d'état engendrés à partir d'une partition d'un nombre $1 \leqslant N \leqslant C$ et

b- la somme

$$SP(N) = \sum_{V'} P$$

où V' est le sous-ensemble des vecteurs d'état associés avec toutes les partitions d'un nombre $1 \leqslant N \leqslant C$.

# CHAPITRE 7

# RESULTATS NUMERIQUES

Ce chapitre illustre les résultats obtenus dans la première partie par un exemple numérique sur un système de dimension réduite qui satisfait les conditions spécifiées dans le contrat et par une série de courbes représentatives de la formule d'Erlang généralisée obtenue dans le cas d'un système desservant une population infinie avec priorité absolue.

## 7.1 *Exemple numérique*

L'exemple numérique ci-dessous représente un système de communication desservant une population de M = 24 utilisateurs groupés en S = 3 sous-populations comme l'indique le tableau 5.

| $m_1$ | $m_2$ | $m_3$ |
|---|---|---|
| 10 | 6 | 8 |

Tableau 5- Répartition des utilisateurs parmi les sous-populations

Le vecteur $[m_i] = [10 \quad 6 \quad 8]$

On suppose que le nombre des canaux utilisés dans le système est C = 5.

Le système peut être illustré par la figure 6. On suppose qu'il y a dans le système P = 2 classes de priorité. La dimension du vecteur d'état est alors:

$$D = P \times S \times S = 18.$$

La génération poissonnienne des appels suivant les priorités sont spécifiés dans les matrices suivantes:

$m_i' = \sum_{k=1}^{2} \sum_{j=1}^{3} m_k(i,j) + m_k(j,i)$

$m_i'' = m_i$

*Priorité k = 1*

$\lambda_1(i,j)$ = (calculer en etc)

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 18 | 16 | 20 |
| 2 | 15 | 15 | 21 |
| 3 | 15 | 18 | 26 |

$\rightarrow m_i'' \times m_j'$

units

*Priorité k = 2*

$\lambda_2(i,j)$ =

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 12 | 16 | 20 |
| 2 | 15 | 15 | 21 |
| 3 | 15 | 18 | 26 |

On a choisi les valeurs numériques de $\lambda_k(i,j)$ de telle manière que pour tout i,j

$$\lambda_1(i,j) < \lambda_2(i,j)$$

Ce qui implique que le temps moyen inter-arrivée des appels

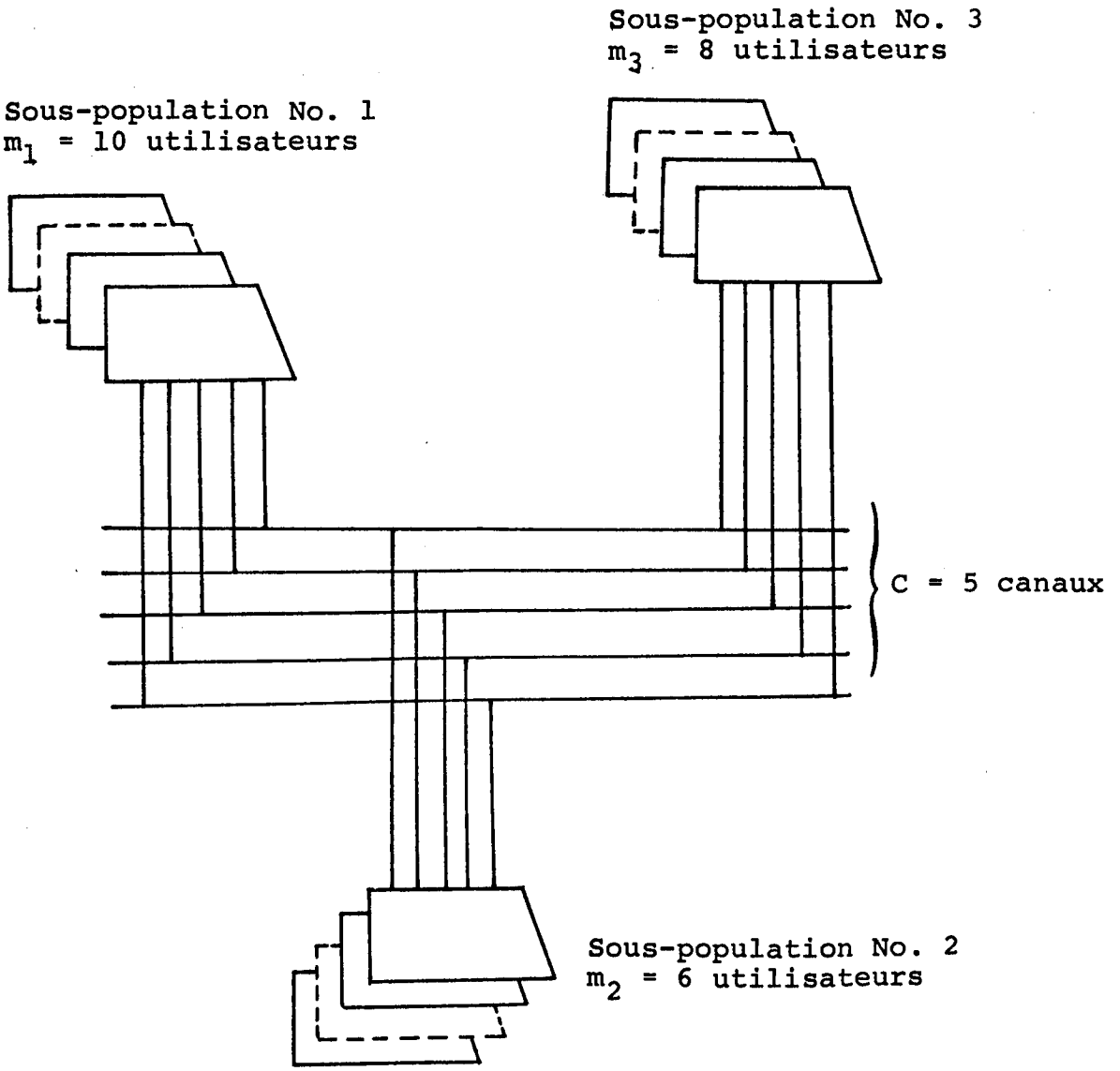


Fig. 6- Schéma du système hypothétique étudié dans l'exemple numérique

de priorité k = 1 est plus grand que celui des appels de priorité k = 2.

Les matrices des temps de services des appels selon les priorités sont données comme suit:

*Priorité k = 1*

$\mu_1(i,j)$ =

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 12 | 4 | 10 |
| 2 | 5 | 15 | 7 |
| 3 | 7.5 | 6 | 26 |

units

*Priorité k = 2*

$\mu_2(i,j)$ =

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 3 | 2 | 5 |
| 2 | 3 | 5 | 3 |
| 3 | 3 | 3 | 2.5 |

De ces deux matrices on voit nettement que le temps moyen des appels de priorité k = 2 est plus long que celui des appels de priorité k = 1.

On trouve facilement les matrices de trafic généré ci-dessous:

5 min computer time



*Priorité k = 1*

$\rho_1(i,j)$ =

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 1.5 | 4 | 2 |
| 2 | 3 | 1 | 3 |
| 3 | 2 | 3 | 1 |

*Priorité k = 2*

$\rho_2(i,j)$ =

| i,j | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 4 | 8 | 4 |
| 2 | 5 | 3 | 7 |
| 3 | 5 | 6 | 10 |

En utilisant un programme conçu à cet effet, on trouve:

1- nombre total de vecteur d'état: 33,648

2- probabilité de trouver le système à vide

$$P_o = 0.44737187 \times 10^{-17}$$

3- trafic écoulé

$$Te = 0.45007095$$

4- la probabilité de congestion des lignes pour les appels de priorité k = 1,2 sont trouvés dans le tableau 6.

| C(1) | C(2) |
|---|---|
| $.13137739 \times 10^{-2}$ | .64153010. |

Tableau 6- Probabilités de congestion des lignes pour les appels de priorité k = 1,2

4- les probabilités de réjection d'un appel de priorité k = 2 sont données dans le tableau 7.

| $SQ_2(i,j)$ | 1 | 2 | 3 |
|---|---|---|---|
| 1 | $.32487893 \times 10^{-4}$ | $.10564305 \times 10^{-3}$ | $.38868500 \times 10^{-4}$ |
| 2 | $.11115836 \times 10^{-3}$ | $.21940311 \times 10^{-3}$ | $.10702867 \times 10^{-3}$ |
| 3 | $.50606905 \times 10^{-4}$ | $.12066442 \times 10^{-3}$ | $.41916079 \times 10^{-4}$ |

Tableau 7- Probabilités de réjection d'un appel entre les sous-populations (i,j) et de priorité $k_Q$ = 2

5- Les probabilités de non admissibilité d'un appel k = 1 sont données dans le tableau 8.

| $SR_1(i,j)$ | 1 | 2 | 3 |
|---|---|---|---|
| 1 | $.28917869 \times 10^{-3}$ | $.19270895 \times 10^{-3}$ | $.48215478 \times 10^{-3}$ |
| 2 | $.28917869 \times 10^{-3}$ | $.48215478 \times 10^{-3}$ | $.28917869 \times 10^{-3}$ |
| 3 | $.28917869 \times 10^{-3}$ | $.28917869 \times 10^{-3}$ | $.24091822 \times 10^{-3}$ |

Tableau 8- Probabilités de non admissibilité pour les appels de priorité k = 1

## 7.2 *Courbes représentatives de la formule d'Erlang généralisée*

Les courbes représentatives de la formule de perte d'Erlang sont classiques. On peut les trouver par exemple à la page 254 de la référence [12].

Les courbes peuvent être utiles dans le calcul des pertes, sans distinction des priorités, d'un système desservant une population infinie avec affectation prioritaire des lignes et avec réjection d'un appel de basse priorité à l'arrivée d'un autre de plus haute priorité à condition qu'on remplace le trafic $\rho$ par le trafic total, précisément:

$$\rho = \sum_{k=1}^{P} \sum_{i,j=1}^{S} \rho_k(i,j)$$

Il est intéressant de calculer les pertes des appels d'une priorité donnée en appliquant la formule (55). Les courbes de la figure 7 représentant les pertes des appels de priorité $1 \leq k_o \leq P$ dont le trafic partiel des appels de priorité $1 \leq k < k_o$ est exprimé en fonction du trafic total par la relation:

$$\rho(k_o) = x(k_o) \cdot \rho$$

où $x(k_o)$ exprime le rapport du trafic partiel au trafic total.

Dans la figure 7 on fait varier le trafic total de 0.05 à 15 en supposant que pour toute valeur de $\rho$ on a $x(k_o) = .1, .3, .5$ et ce pour $C = 2, 10, 15$ .

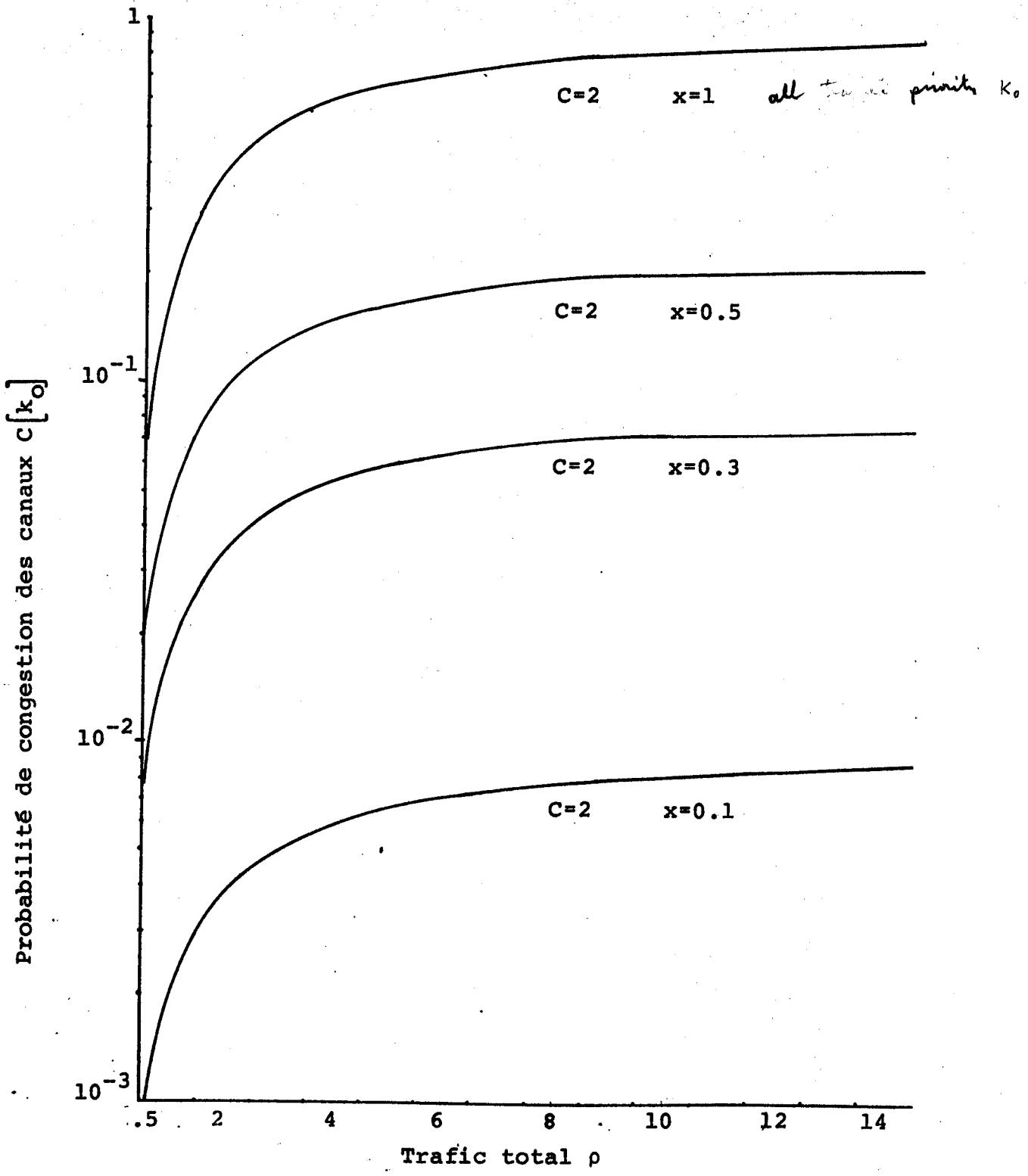


Fig. 7a- Probabilité de congestion des canaux pour C = 2 canaux et x = 1,0.5,0.3,0.1

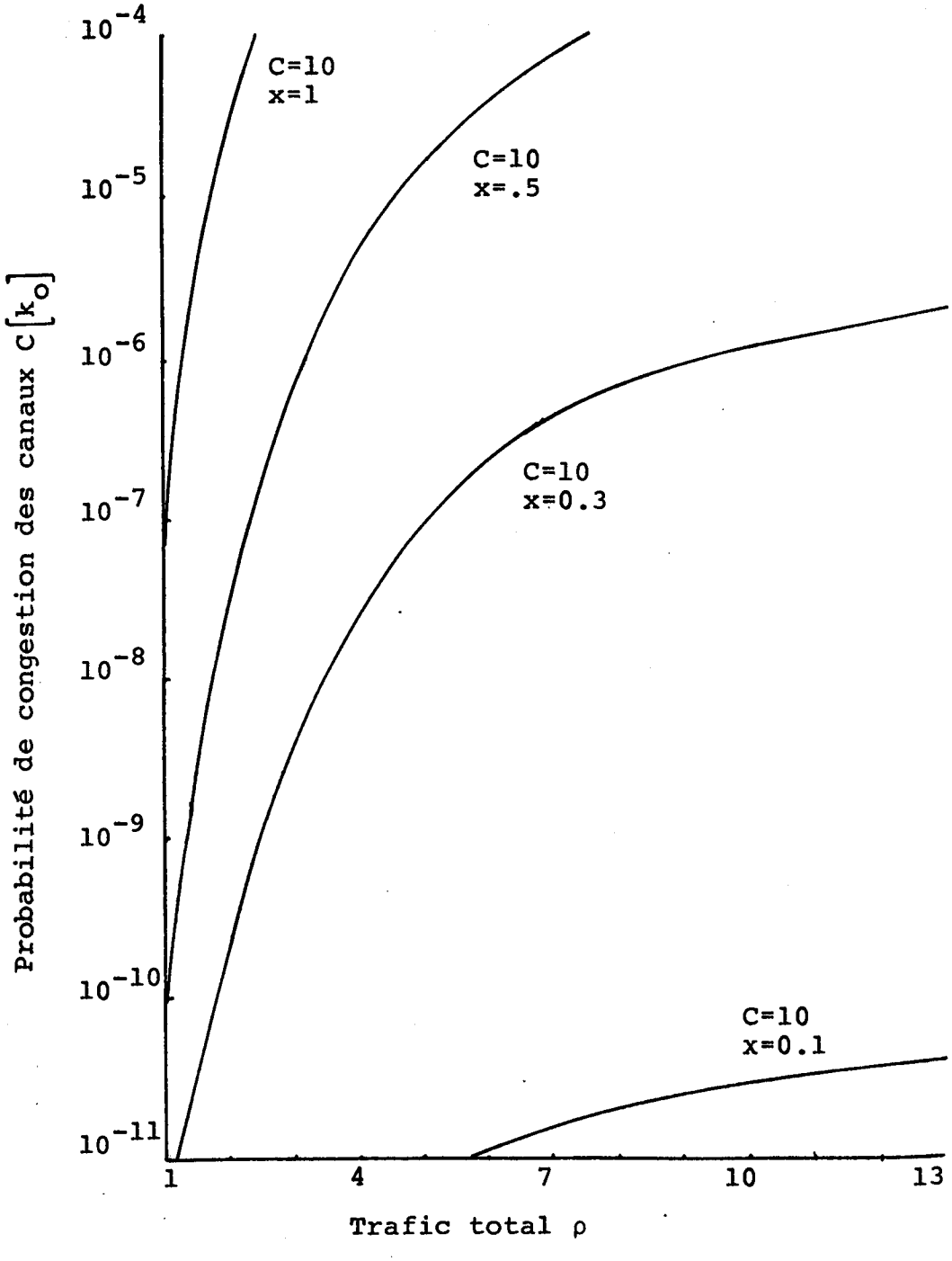


Fig. 7b- Probabilité de congestion des canaux pour C=10 canaux et x=1,0.5,0.3,0.1

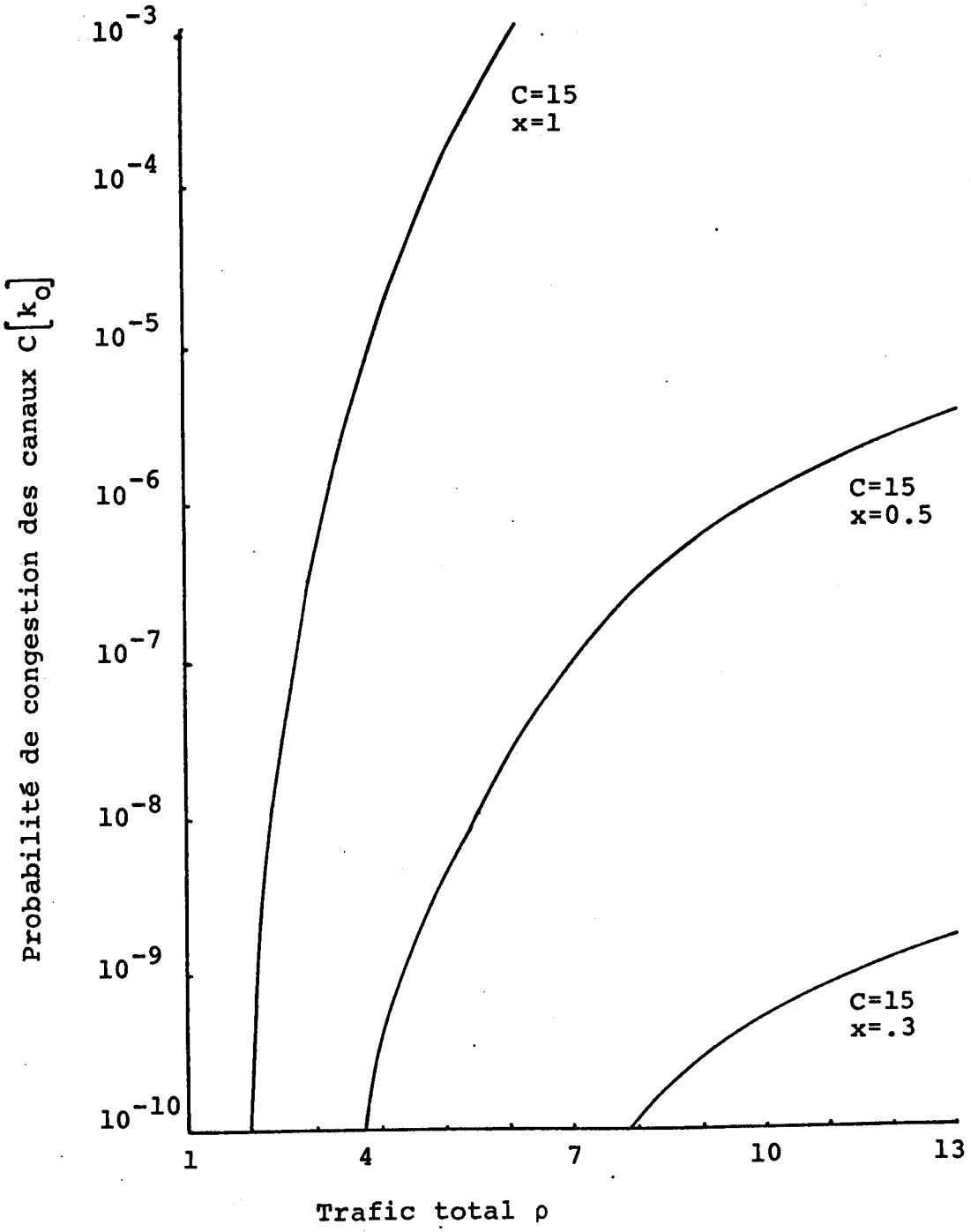


Fig. 7c- Probabilité de congestion vs trafic total pour C=15 canaux et x=1,0.5,0.3

# CONCLUSION

Un modèle a été construit pour représenter un système de communication ayant les traits caractéristiques spécifiés par le contrat, précisément:

1- desservant une population qui se compose de groupes de dimension finie

2- ayant un protocole de lignes prioritaire absolu, c'est-à-dire avec réjection d'un appel de basse priorité à l'arrivée d'un autre de plus haute priorité.

En décrivant l'état du système par un vecteur dont les composantes représentent le nombre d'appels d'une priorité donnée présent dans le système et reliant deux sous-populations prédéterminées, et en partant des hypothèses que la génération des appels entre deux sous-populations est poissionnienne et que la distribution des temps de service des appels est exponentielle, on a démontré que le système se comporte comme un processus multi-dimensionnel de mort et de naissance, dont les paramètres sont fonctions de l'état du système. Ce modèle englobe ceux des systèmes prioritaires sans réjection et des systèmes desservant une population infinie comme cas particuliers.

La résolution du système d'équations d'état en régime permanent a montré que le système prioritaire avec réjection se comporte identiquement comme un système sans réjection, excepté que la réjection implique que le taux d'admission des appels de hautes priorités aux moments où tous les canaux sont occupés doit satisfaire une équation d'équilibre.

Les indices de performances qui intéressent celui qui concoît ou celui qui exploite le système aussi bien que l'utilisateur sont définis et calculés en fonction de la distribution des états du système. Les résultats obtenus dans les cas particuliers sont identiques avec ceux donnés dans la littérature sous les mêmes hypothèses.

Les techniques de calcul numérique des indices de performance ont été introduites. Elles se basent sur deux opérations fondamentales:

1- la génération des partitions d'un nombre donné et
2- la permutation sans répétition des multi-ensembles

Des algorithmes pour programmer ces deux opérations ont été développés dans le but de minimiser le temps de calcul et la longueur de mémoire de l'ordinateur.

Un exemple d'exploitation des résultats a été donné en détail, en utilisant un réseau hypothétique. Les résultats essentiels concernant le cas d'un système desservant une population infinie, qui est considéré comme la limite de notre modèle, ont été représentés graphiquement pour illustrer l'effet de quelques paramètres, tels que le nombre des canaux et le trafic total, sur le comportement du système.

Le rapport se termine en suggérant la généralisation

de la technique développée ci-dessus à l'étude des deux problèmes intéressants suivants:

Problème I: Etude du cas d'un système dans lequel on impose, en plus des deux conditions précitées, une attribution particulière des canaux parmi les S sous-populations. Cette nouvelle hypothèse est introduite pour étudier le cas réaliste des systèmes où, soit par des contraintes d'ordre de sécurité de transmission, soit par des contraintes sur la capacité des appareils de transmission, une sous-population donnée ne peut utiliser qu'une sous-bande de fréquence prédéterminée parmi les C canaux du système.

Problème II: Etude d'un réseau de transmission des données dans lequel la circulation des messages d'origine et destination prédéterminés se fait suivant une route déterminée, selon un protocole de ligne prioritaire absolu.

La mise en équation du problème I demande une nouvelle définition du vecteur d'état, et sa résolution devrait être sujette aux conditions limites nouvelles.

Pour le problème II, on peut appliquer un modèle multidimensionnel pour représenter chaque noeud de transmission du réseau. Pour que le modèle soit utilisable dans l'étude analytique de l'ensemble du réseau, on prévoit que le signal sur chaque sortie d'un noeud doit obéir à un processus poissonnien.

Ceci est une généralisation du théorème de Burke [14]. Le calcul de la probabilité de débordement de la mémoire dans chaque noeud de transmission du réseau, sous ces conditions, peut se faire par l'application de la théorie des fonctions génératrices composées.

# REFERENCES


1- Feller, W.-"*An Introduction to Probability Theory and its Applications*", John Willy & Sons Inc., vol. 1, New York (1950).

2- Thériault, R.- "*Comportement statistique de deux systèmes de communication*", CRC Technical Memorandum, No. SA 8173, (Septembre 1973).

3- Saaty, T.- "*Elements of Queuing Theory*", McGraw-Hill Co. New York (1961), pp. 326-329.

4- Brandt, J.G.- "*A Multiserver Queuing System with Preemptive Priority*", Proceeding du 6ième Congrès International sur le Télétrafic, Munich, (1970), pp. 316/1-316/7.

5- Syski, R.- "*Introduction to Congestion Theory in Telephone Systems*", Oliver and Boyd, Edinbourgh and London, England (1960), chapitre 10.

6- Palm C.- "*Intensity Fluctuation in Telephone Traffic*", Ericson Technics, vol. 44, pp.1-185, (1943).

7- Cox, D.R. and Smith, W.L.- "*On the Superposition of Renewal Process*", Biometrika, vol. 41, pp. 91-99, (1974).

8- Newland, W.F.- "*A Method of Approach and Solution to some Fundamental Traffic Problems*", P.O. Electrical Engrs. Journal, vol. 25, pp. 119-131, (1932-1933).

9- O'Dell, G.F.- "*Theorical Principles of the Traffic Capacity of Automatic Switches*", P.O. Electrical Engrs. J, vol. 13, pp. 209, (1920).

10- Rubas, J.- "*Analysis of Congestion in Serial P.A.B.X'S*", Proceedings du 10ième Congrès International sur le Télétrafic, London, (1970), pp.211/1-211/7.

11- Jung, M.M.- "*Calculation of the Blocking Probability at Small Private Automatic Branch Exchanges Having Fully Available Connecting Circuit and Exchange Line Relay Set*", Philips Télecommunication Review, vol. 29, No. 3, (January 1971), pp. 103-113.

12- Riordan, J.- "*An Introduction to Combinatorial Analysis*" J. Wiley & Sons Inc., London, (1958), p. 92.

13- Cooper, R.B.- "*Introduction to Queuing Theory*", Macmillan Company, New York, (1972), p. 67.

14- Burke, P.J.- "*The Output of a Queuing System*", Operation Research, vol. 4, (1956), pp. 699-704.

# ANNEXE A

## PROGRAMME PRINCIPAL POUR LE CALCUL NUMERIQUE DES RESULTATS

On trouvera dans cette annexe le programme principal pour le calcul numérique des résultats.Dans l'application de ce programme à l'étude d'un système réel, on changera suivant les paramètres du système les dimensions des variables indiquées ci-dessous dans:

- le programme principal
- la sous-routine FONC
- la sous-routine EQUIV

### A1 *Variables spécifiant le système*

LI = LJ = S : nombre de sous-populations

LK = P : nombre de priorités

IC = C : nombre de canaux

L = S x P x C : dimension du vecteur d'état

### A2 *Dimension des variables à changer suivant les paramètres définissant le système*

RD(LI, LJ, LK)

M2(LI)

MΦ(LI)

FLAMDA(LI, LJ, LK)

FMU(LI, LJ, LK)

M(IC)

NAA(LI, LJ, LK)

NA(L)

NPS(L)

NQ(L)

NB(IC)

NC(IC)

NR(IC)

NP(IC)

SP(IC)

SC(LK)

SQ(LI, LJ, LK)

SR(LI, LJ, LK)

## A3 *Format des données numériques*

Les formats des données numériques du programme sont listés dans le tableau ci-dessous:

| Variable | Format |
|---|---|
| LI, LJ, LK, IC | (10I5) |
| $M\phi(I) = m_i$ | (10I.5) |
| FLAMDA(LI, LJ, LK) = $\lambda_k(i,j)$ | (10F5.3) |
| FMU(LI, LJ, LK) = $\mu_k(i,j)$ | (10F5.3) |

Tableau Format des données numériques

## A4 *Les sorties*

En plus des cinq principaux paramètres suivant

1- probabilité de trouver le système à l'état vide

2- le trafic écoulé

3- le vecteur des probabilités de congestion des lignes suivant les priorités

4- les matrices des probabilités de réjection d'un appel de priorité $1 \leqslant k \leqslant P$ entre les sous-populations

5- les matrices de probabilité de non admissibilité d'un appel de priorité $2 \leqslant k \leqslant P$ entre les sous-populations

le programme imprime aussi les paramètres intermédiaires suivants:

1- NB: liste des partitions d'un nombre $1 \leqslant n \leqslant C$

2- La sommation:

$$SP = \sum_{V} P\left[N(n;\ldots, n_k(i,j),\ldots)\right]$$

sur le sous-ensemble des vecteurs d'état associés avec une partition donnée d'un nombre n

3- La sommation:

$$SSP = \sum_{V'} P\left[N(n;\ldots, n_k(i,j),\ldots)\right]$$

sur le sous-ensemble des vecteurs d'état associés avec toutes les partitions obtenues des partitions d'un nombre n.

# ANNEXE A

## PROGRAMME PRINCIPAL

```
      DIMENSION RO(4,4,3),M2(4),MO(4),FLAMDA(4,4,3),FMU(4,4,3),M(5),
     1 NAA(4,4,3),NA(48),NPI(48),NQ(48),NB(5),NC(5),NR(5),NP(5),SP(5),
     2 SC(3),SQ(4,4,3),SR(4,4,3)
      COMMON /C10/LI,LJ,LK
      COMMON /C20/MO,RO,NAA,FMU
      DATA SC,SQ,SR/99*0./
      READ(5,1)LI,LJ,LK,IC
1     FORMAT(10I5)
      READ(5,2)(((FLAMDA(I,J,K),I=1,LI),J=1,LJ),K=1,LK)
      READ(5,3)(((FMU(I,J,K),I=1,LI),J=1,LJ),K=1,LK)
2     FORMAT(10F5.3)
3     FORMAT(10F5.1)
      READ(5,1)(MO(I),I=1,LI)
      DO 10 I=1,LI
      DO 10 J=1,LJ
      DO 10 K=1,LK
10    RO(I,J,K)=FLAMDA(I,J,K)/FMU(I,J,K)
      L=LI*LJ*LK
      WRITE(6,11)LI,LJ,LK,IC
11    FORMAT(' LI,LJ,LK,IC=',4I3)
      WRITE(6,12)(((FLAMDA(I,J,K),I=1,LI),J=1,LJ),K=1,LK)
12    FORMAT(' FLAMDA= '/(10F5.3))
      WRITE(6,13)(((FMU(I,J,K),I=1,LI),J=1,LJ),K=1,LK)
13    FORMAT(' FMU= '/(10F5.1))
      IC1=IC-1
      SSP=0.
      KON=0
      DO 100 N=1,IC1
      SP(N)=0.
      KON1=0
      CALL PARTIT(N,IFIN,NC,M,NG,L,NB,NI,NP)
20    IF(IFIN.EQ.1)GOTO 90
      WRITE(6,21)(NB(I),I=1,NI)
21    FORMAT(/' NB=',5I2)
      P=0.
      KON2=0
      CALL PERMUT(NC,M,NG,IFIN,NA,L,NPI,NQ,NR)
40    IF(IFIN.EQ.1)GOTO 60
      KON2=KON2+1
      CALL EQUIV(NA,NAA)
      CALL FONC(M2,PROD,FMU1)
      P=P+PROD
      CALL PERMU(IFIN,NA)
      GOTO 40
60    SP(N)=SP(N)+P
      KON1=KON1+KON2
      KON=KON+KON2
      WRITE(6,61)KON2,P
61    FORMAT(I6,' P=',E15.8)
      CALL PARTI(IFIN,NC,M,NG,NB,NI)
      GOTO 20
90    SSP=SSP+SP(N)
100   WRITE(6,62)KON1,SP(N)
```

```
62    FORMAT(I6,' SP= ',E15.8)                                    TES00540
      SP5=0.                                                      TES00550
      N=IC                                                        TES00560
      SP(N)=0.                                                    TES00570
      KON1=0                                                      TES00580
      CALL PARTIT(N,IFIN,NC,M,NG,L,NB,NI,NP)                      TES00590
120   IF(IFIN.EQ.1)GOTO 200                                       TES00600
      WRITE(6,21)(NB(I),I=1,NI)                                   TES00610
      P=0.                                                        TES00620
      P3=0.                                                       TES00630
      KON2=0                                                      TES00640
      CALL PERMUT(NC,M,NG,IFIN,NA,L,NPI,NQ,NR)                    TES00650
140   IF(IFIN.EQ.1)GOTO 190                                       TES00660
      KON2=KON2+1                                                 TES00670
      CALL EQUIV(NA,NAA)                                          TES00680
      CALL FONC(M2,PROD,FMU1)                                     TES00690
      P=P+PROD                                                    TES00700
      KO=LK                                                       TES00710
      SC(KO)=SC(KO)+PROD                                          TES00720
150   DO 151 J=1,LJ                                               TES00730
      DO 151 I=1,LI                                               TES00740
      IF(NAA(I,J,KO).NE.0)GOTO 160                                TES00750
151   CONTINUE                                                    TES00760
      KO=KO-1                                                     TES00770
      SC(KO)=SC(KO)+PROD                                          TES00780
      DO 155 J=1,LJ                                               TES00790
      DO 155 I=1,LI                                               TES00800
      T1=M2(I)*M2(J)*FLAMDA(I,J,KO)                               TES00810
      T2=T1+FMU1                                                  TES00820
      T2=T2*T2                                                    TES00830
155   SQ(I,J,KO)=SQ(I,J,KO)+PROD*T1*FMU1/T2                      TES00840
      IF(KO.GT.1)GOTO 150                                         TES00850
      GOTO 180                                                    TES00860
160   K1=KO-1                                                     TES00870
      FLMD=0.                                                     TES00880
      DO 161 K=1,K1                                               TES00890
      DO 161 J=1,LJ                                               TES00900
      DO 161 I=1,LI                                               TES00910
161   FLMD=FLMD+M2(I)*M2(J)*FLAMDA(I,J,K)                         TES00920
      DO 165 J=1,LJ                                               TES00930
      DO 165 I=1,LI                                               TES00940
      T1=FMU(I,J,KO)                                              TES00950
      T2=T1+FLMD                                                  TES00960
165   SR(I,J,KO)=SR(I,J,KO)+PROD*T1*FLMD/(T2*T2)                 TES00970
180   CALL PERMU(IFIN,NA)                                         TES00980
      GOTO 140                                                    TES00990
190    SP(N)=SP(N)+P                                              TES01000
      KON=KON+KON2                                                TES01010
      KON1=KON1+KON2                                              TES01020
      WRITE(6,61)KON2,P                                           TES01030
      CALL PARTI(IFIN,NC,M,NG,NB,NI)                              TES01040
      GOTO 120                                                    TES01050
200   WRITE(6,62)KON1,SP(N)                                       TES01060
```

```
      SSP=SSP+SP(N)                                                     TES01070
      PO=1./(1.+SSP)                                                    TES01080
      WRITE(6,201)KON,SSP                                               TES01090
201   FORMAT(/' NOMBRE TOTAL DE VECTEURS = ',I6,' SSP=',E15.8)          TES01100
      S=0.                                                              TES01110
      DO 210 N=1,IC                                                     TES01120
210   S=S+N*SP(N)                                                       TES01130
      TE=PO*S                                                           TES01140
      WRITE(6,211)PO,TE                                                 TES01150
211   FORMAT(/' PO=',E16.8,'  TE=',E16.8/)                              TES01160
      WRITE(6,219)                                                      TES01170
219   FORMAT(/' VECTEUR SC:'/)                                          TES01180
      DO 220 K=1,LK                                                     TES01190
220   SC(K)=SC(K)*PO                                                    TES01200
      WRITE(6,221)(SC(K),K=1,LK)                                        TES01210
221   FORMAT(10E16.8)                                                   TES01220
      WRITE(6,222)                                                      TES01230
222   FORMAT(/' MATRICE SQ:'/)                                          TES01240
      LK1=LK-1                                                          TES01250
      DO 230 K=1,LK1                                                    TES01260
      DO 230 J=1,LJ                                                     TES01270
      DO 225 I=1,LI                                                     TES01280
225   SQ(I,J,K)=SQ(I,J,K)*PO                                            TES01290
230   WRITE(6,231)(SQ(I,J,K),I=1,LI)                                    TES01300
231   FORMAT(5E16.8)                                                    TES01310
      WRITE(6,232)                                                      TES01320
232   FORMAT(/' MATRICE SR:'/)                                          TES01330
      DO 240 K=2,LK                                                     TES01340
      DO 240 J=1,LJ                                                     TES01350
      DO 235 I=1,LI                                                     TES01360
235   SR(I,J,K)=SR(I,J,K)*PO                                            TES01370
240   WRITE(6,231)(SR(I,J,K),I=1,LI)                                    TES01380
      STOP                                                              TES01390
      END                                                               TES01400
      SUBROUTINE FONC(M2,PROD,FMU1)                                     TES01410
      DIMENSION M2(4),MO(4),RO(4,4,3),NAA(4,4,3),FMU(4,4,3)             TES01420
      COMMON /C10/LI,LJ,LK                                              TES01430
      COMMON /C20/MO,RO,NAA,FMU                                         TES01440
      FMU1=0.                                                           TES01450
      PROD=1.                                                           TES01460
      DO 50 I=1,LI                                                      TES01470
      NS=0                                                              TES01480
      DO 30 J=1,LJ                                                      TES01490
      DO 30 K=1,LK                                                      TES01500
      NA1=NAA(I,J,K)                                                    TES01510
      IF(NA1.EQ.0)GOTO 30                                               TES01520
      FMU1=FMU1+FMU(I,J,K)*NA1                                          TES01530
      NS=NS+NA1                                                         TES01540
      PROD=PROD*RO(I,J,K)**NA1/FACT(NA1)                                TES01550
30    CONTINUE                                                          TES01560
      M1=MO(I)-NS                                                       TES01570
      M2(I)=M1                                                          TES01580
50    PROD=PROD*FACT(M1)                                                TES01590
```

```
      RETURN                                                            TES01600
      END                                                               TES01610
      SUBROUTINE EQUIV(NA,NAA)                                          TES01620
      DIMENSION NA(48),NAA(4,4,3)                                       TES01630
      COMMON /C10/LI,LJ,LK                                              TES01640
      N=0                                                               TES01650
      DO 10 K=1,LK                                                      TES01660
      DO 10 J=1,LJ                                                      TES01670
      DO 10 I=1,LI                                                      TES01680
      N=N+1                                                             TES01690
10    NAA(I,J,K)=NA(N)                                                  TES01700
      RETURN                                                            TES01710
      END                                                               TES01720
      FUNCTION FACT(N1)                                                 TES00010
      N=N1                                                              TES00020
      IF(N.EQ.0)GOTO 10                                                 TES00030
      FACT=N                                                            TES00040
5     IF(N.LE.2)GOTO 20                                                 TES00050
      N=N-1                                                             TES00060
      FACT=FACT*N                                                       TES00070
      GOTO 5                                                            TES00080
10    FACT=1                                                            TES00090
20    RETURN                                                            TES00100
      END                                                               TES00110
      SUBROUTINE PARTIT(NO,IFIN,NC,M,K,L,NB,I,NP)                       TES00010
      DIMENSION NB(1),NP(1),NC(1),M(1)                                  TES00020
      I=1                                                               TES00030
      NB(1)=NO                                                          TES00040
      NP(1)=NO                                                          TES00050
20    IFIN=0                                                            TES00060
      J=I                                                               TES00070
      K=2                                                               TES00080
21    NC(K)=NB(J)                                                       TES00090
      M(K)=1                                                            TES00100
22    J=J-1                                                             TES00110
      IF(J.LT.1)GOTO 25                                                 TES00120
      IF(NB(J).EQ.NC(K)) GOTO 23                                        TES00130
      K=K+1                                                             TES00140
      GOTO 21                                                           TES00150
23    M(K)=M(K)+1                                                       TES00160
      GOTO 22                                                           TES00170
25    J=0                                                               TES00180
      DO 26 J1=2,K                                                      TES00190
26    J=J+M(J1)                                                         TES00200
      M(1)=L-J                                                          TES00210
      NC(1)=0                                                           TES00220
      RETURN                                                            TES00230
      ENTRY PARTI(IFIN,NC,M,K,NB,I)                                     TES00240
50    IF(NB(I).NE.1)GOTO 60                                             TES00250
      IF(I.EQ.1)GOTO 100                                                TES00260
      I=I-1                                                             TES00270
      GOTO 50                                                           TES00280
60    N=NP(I)                                                           TES00290
```

```
      NB(I)=NB(I)-1                                                     TES00300
65    I=I+1                                                             TES00310
      NB(I)=N-NB(I-1)                                                   TES00320
      NP(I)=NB(I)                                                       TES00330
      IF(NB(I).LE.NB(I-1))GOTO 20                                       TES00340
      N=NB(I)                                                           TES00350
      NB(I)=NB(I-1)                                                     TES00360
      GOTO 65                                                           TES00370
100   IFIN=1                                                            TES00380
      RETURN                                                            TES00390
      END                                                               TES00400
      SUBROUTINE PERMUT(NC,M,N,IFIN,NA,L,NPI,NQ,NR)                     TES00010
      DIMENSION NC(1),M(1),NA(1),NPI(1),NQ(1),NR(1)                     TES00020
      DO 10 I=1,N                                                       TES00030
10    NR(I)=0                                                           TES00040
      I=1                                                               TES00050
50    K=1                                                               TES00060
60    IF(NR(K).LT.M(K))GOTO 70                                          TES00070
      K=K+1                                                             TES00080
      GOTO 60                                                           TES00090
70    NPI(I)=K                                                          TES00100
      NA(I)=NC(K)                                                       TES00110
      NR(K)=NR(K)+1                                                     TES00120
      NQ(I)=NR(K)                                                       TES00130
      IF(I.EQ.L) GOTO 100                                               TES00140
      I=I+1                                                             TES00150
      GOTO 60                                                           TES00160
100   IFIN=0                                                            TES00170
      RETURN                                                            TES00180
      ENTRY PERMU(IFIN,NA)                                              TES00190
200   K=NPI(I)                                                          TES00200
      NR(K)=NQ(I)-1                                                     TES00210
      I=I-1                                                             TES00220
      IF(I.EQ.0)GOTO 300                                                TES00230
      K=NPI(I)+1                                                        TES00240
210   IF(K.GT.N) GOTO 200                                               TES00250
      IF(NR(K).LT.M(K))GOTO 220                                         TES00260
      K=K+1                                                             TES00270
      GOTO 210                                                          TES00280
220   K1=NPI(I)                                                         TES00290
      NR(K1)=NQ(I)-1                                                    TES00300
      NPI(I)=K                                                          TES00310
      NR(K)=NR(K)+1                                                     TES00320
      NQ(I)=NR(K)                                                       TES00330
      NA(I)=NC(K)                                                       TES00340
      I=I+1                                                             TES00350
      GOTO 50                                                           TES00360
300   IFIN=1                                                            TES00370
      RETURN                                                            TES00380
      END                                                               TES00390
```

CACC / CCAC
82518

MORISSETTE, S.
--Etude statistique d'un système de...

LKC
P91 .C654 M67 1975
Etude statistique d'un
systeme de
telecommunication base sur
un mode d'affectation
prioritaire et servant un

DATE DUE
DATE DE RETOUR

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

LOWE-MARTIN No. 1137

INDUSTRY CANADA / INDUSTRIE CANADA
214625