

N^o d'ordre :02\2009-E\MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
 MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
 SCIENTIFIQUE
 UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE
 FACULTÉ DES MATHÉMATIQUES



THÈSE :

Présentée pour l'obtention du diplôme de **DOCTORAT d'ÉTAT**
EN MATHÉMATIQUES

Spécialité : **Probabilités et Statistique**

Par

Hafida SAGGOU

THÈME

FILES D'ATTENTE A DEUX NIVEAUX DE PRIORITÉ
ET
PARADOXE DANS LES RÉSEAUX A.T.M

Soutenue publiquement, le 27/05/09 ; devant le jury composé de :

Mr M. DJEDOUR	Professeur	USTHB	Président
Mr J. PELLAUMAIL	Professeur	Univ. Rennes	Directeur de Thèse
Mr A. AISSANI	Professeur	USTHB	Co-Promoteur de Thèse
Mr D. AISSANI	Professeur	Univ. Bejaia	Examineur
Mr F. CHARLOT	Professeur	Univ. Rouen	Examineur
Mr B. CHOUAF	Professeur	Univ. Sidi Belabbes	Examineur

Remerciements

Je remercie vivement le Professeur Emérite Jean Pellaumail qui a bien voulu prendre en charge la Direction de cette Thèse. Sa confiance en moi m'a fait énormément plaisir. Son attitude d'écoute, son aide, ses encouragements et ses conseils qu'il n'a cessé de me prodiguer, en particulier lors de mes déplacements à l'I.N.S.A de RENNES. Je lui exprime toute ma gratitude.

J'exprime ma reconnaissance et mes plus vifs remerciements au Professeur Amar Aissani Co-Promoteur pour son aide multiple.

Je remercie le Professeur Mohamed Djedour pour l'honneur et le privilège qu'il me fait pour avoir accepté de présider le jury de thèse et pour ses encouragements.

Je remercie les Professeurs Djamil Aissani et Benamar Chouaf pour l'insigne honneur qu'ils m'ont fait pour avoir accepté d'examiner ce travail.

Je remercie particulièrement mon professeur François Charlot pour son enseignement en graduation, en post -graduation et pour m'avoir fait l'honneur d'accepter d'être examinateur dans ce jury, et Madame le Professeur Noufissa Mikou pour son aide.

Je remercie tous mes collègues de la Faculté de Mathématiques pour leurs multiples encouragements et conseils.

Table des matières

Introduction	5
0.1 La Circulation de l'Information	6
0.1.1 La Circulation des Personnes	6
0.1.2 Le Mode de Transfert Asynchrone ou A.T.M	6
1 Les Principes de l'A.T.M	11
1.1 Les Fondements de l'A.T.M	11
1.1.1 La Cellule	12
1.2 Protocoles A.T.M.	14
1.2.1 Couche Physique	15
1.2.2 Couche A.T.M.	15
1.2.3 Couche AAL	15
1.2.4 Commutation des Cellules dans A.T.M.	16
1.3 Multiplexage et Commutation	19
1.3.1 Technique Spaciale	19
1.3.2 Techniques Temporelles	20
1.4 Comparaison Entre la Commutation de Circuits et Par Paquets	21
1.5 Les Caractéristiques Principales de l'A.T.M.	22
1.6 Contrôle de Congestion	25
1.6.1 Contrôle Par Boucle Ouverte	25

	3
1.6.2	Contrôle Par Boucle Fermée 26
1.6.3	Algorithme du Seau Percé 26
1.6.4	Contrôleur Espaceur 27
1.6.5	Contrôle de Débit 27
1.7	Données du Problème 28
2	Un Paradoxe dans la Modélisation des Réseaux A.T.M. 31
2.0	Préliminaire 31
2.1	Introduction 32
2.2	Modèle Mathématique 41
2.3	Pseudo-Simulation 42
2.4	Retard d'une Cellule Non Prioritaire 44
2.5	Chaîne Incluse 45
2.6	l'évolution de Y Entre $v(n)$ et $w(n)$ 47
2.7	Méthodes des Convexes 50
2.8	Régime Stationnaire de la Chaîne Incluse 51
2.9	Loi P-Géométrique 53
3	Autre Chaîne Incluse 56
3.0	Introduction 56
3.1	Modèle Mathématique 56
3.2	Chaîne Incluse 57
3.3	Etude Entre u' et u 57
4	Modèle de File d'Attente avec Rappels, Serveur Non Fiable et Deux Types de Clients 60
4.0	Introduction 60
4.1	Notation et Modèle Mathématique 61

	4
4.2 Condition de stabilité	63
4.3 Distribution stationnaire des états du système	66
4.4 Distribution du nombre de clients transitoires servis durant la période d'occupation	72
Conclusion	74
Annexe	81
Glossaire	81
Listing du Programme	85
Execution du Programme	137

Introduction

L'une des caractéristiques du monde actuel est l'intensification et la diversification de la transmission des informations. Les améliorations technologiques sont impressionnantes : au début de l'année 1999, le CNET de Lannion a réalisé une transmission, sur une seule fibre optique, d'un débit de un gigabit par seconde.

Cette amélioration constante du « hardware » doit être accompagnée de la mise au point d'un « software » approprié. Dans cette perspective, le Centre National d'Etudes des Télécommunications, à Lannion (centre de recherche et de développement du groupe France Télécom), a proposé, dès le début des années 80, une nouvelle technique de transfert de l'information baptisé désormais A.T.M, Asynchronous Transfer Mode.

L'A.T.M est un ensemble de protocoles dont l'objectif est de spécifier, de permettre et de construire des réseaux capables de transporter, sans en modifier l'infrastructure, tous les types possibles d'informations. Le terme « asynchrone » exprime que chaque source émet à son rythme propre, en respectant des contraintes de débit préétablies, sans tenir compte des autres sources émettrices.

Lorsqu'une artère transporte un grand nombre de communications à faible débit, la faisabilité du mode A.T.M repose sur la « cohérence » du hasard. L'étude détaillée du comportement d'une telle artère est donc nécessairement, en partie, de nature probabiliste.

Le premier chapitre de ce travail donne un bref aperçu sur les réseaux A.T.M. L'objectif en est de donner suffisamment d'informations pour comprendre la validité des modèles mathématiques considérés dans les autres parties.

Le deuxième chapitre constitue le cœur de ce travail. On y effectue l'étude détaillée du régime transitoire d'une file d'attente comportant deux niveaux de priorités. Ceci permet de modéliser les phénomènes de congestion dus aux artères qui peuvent se produire à la sortie des « carrefours A.T.M » (brasseurs ou commutateurs). On utilise la technique de la chaîne

incluse.

L'objectif de cette partie est plus de proposer des outils mathématiques que de résoudre des problèmes industriels précis alors que la technologie et la demande du public sont encore en pleine évolution. On a, notamment, mis en évidence un paradoxe sur l'utilisation de l'approximation par une loi de Gauss ou, au contraire, par une loi « presque » géométrique.

Par ailleurs, on montre dans le chapitre trois comment adapter ces mêmes outils à l'étude d'une autre chaîne incluse.

Le chapitre quatre est réservé à l'étude d'une file d'attente avec rappel, serveur non fiable et deux types de clients.

0.1 La Circulation de l'Information

0.1.1 La Circulation des Personnes

Sans la circulation des soldats, des commerçants et des pèlerins, notre planète serait une mosaïque de tribus aux niveaux de civilisation disparates. Cette circulation des personnes a été étroitement associée aux avancées technologiques : le cheval, la roue, la pirogue, la marine à voile, l'avion. Or, si rapide que soit le concorde, il lui faut tout de même 24 heures pour faire le tour du monde alors que deux ou trois dixièmes de seconde suffisent à une information circulant sur une fibre optique pour faire le même trajet.

0.1.2 Le Mode de Transfert Asynchrone ou A.T.M

Les réseaux longue distance classiques (par exemple le plus ancien le télégraphe) ne s'adaptent pas bien à l'offre de nouveaux services, ce qui explique le travail qui se réalise de nos jours tendant à la mise en œuvre des réseaux R.N.I.S.L.B. On espère de ces réseaux qu'ils soient capables de s'adapter à l'évolution des services attendus. Le manque de flexibilité des réseaux existants est dû au mode de transfert utilisé.

Un mode de transfert est une technique, ou un ensemble de techniques, utilisée dans un réseau de télécommunication, couvrant des aspects liés à la transmission, la commutation et le multiplexage ; les modes de transfert les plus utilisés sont la commutation de circuits utilisée par exemple dans les réseaux téléphoniques, et la commutation de paquets utilisée

dans les réseaux internet.

Le mode de transfert qui a été retenu pour la mise en œuvre des réseaux R.N.I.S.L.B est le mode de transfert asynchrone (A.T.M Asynchronous Transfer Mode).

A.T.M, C'est quoi au juste ?

Chaque nouveauté économique, culturelle ou technologique crée ses codes, ses sigles, ses références ; aujourd'hui pour être branché, il faut connaître l'A.T.M.

Le sigle A.T.M, Asynchronous Transfer Mode, exprime que l'A.T.M est un mode de transfert de l'information c'est à dire un ensemble de normes et de protocoles que doivent respecter tous les constructeurs A.T.M, mode dans lequel la synchronisation est relativement plus souple que dans les systèmes de transmissions classiques.

L'A.T.M, c'est un TGV de l'information. Les données regroupées « en wagons », circulent à grande vitesse dans les câbles de fibre optique.

Nous appelons réseau A.T.M tout réseau qui utilise l'A.T.M comme mode de transfert.

Histoire de l'A.T.M.

La technique A.T.M est née au C.N.E.T-LANNION au début des années 80.

La télévision par câble se développe, il faut trouver des solutions pour créer un réseau très haut débit capable de transporter données, sons, images. . . .

Par ailleurs, dans la même année, d'autres centres de recherche dans le monde travaillaient sur des concepts similaires et concouraient à ceux du C.N.E.T. En 1987 le C.C.I.T.T, dans le souci de rationalisme économique, a choisi la technique A.T.M reconnaissant ainsi les résultats de la recherche effectuée au C.N.E.T et obligeant les autres centres de recherche à converger vers cette technique.

Les caractéristiques principales de l'A.T.M.

Depuis l'origine de l'A.T.M, les principes physiques de fonctionnement n'ont pas varié ; la technique A.T.M combine commutation de circuits et commutation de paquets et définit des chemins virtuels sur lesquels sont transmis des paquets de taille fixe, appelés cellules (cells) dans la terminologie du C.C.I.T.T.

Une connexion est établie pour chaque nouvelle communication, identifiée par un numéro binaire représentant une adresse logique qui identifie le circuit virtuel auquel est affecté la

cellule. Le chemin est marqué dans les mémoires des commutateurs. Cette gestion simple des liaisons est exécutée en hardware et favorise la rapidité de la commutation des cellules

La technique A.T.M est donc orientée connexion.

L'utilisateur peut ensuite envoyer un flux de cellules à son rythme propre, avec un débit éventuellement variable au cours de la connexion. Ce mode asynchrone permet un partage simple et souple des capacités du réseau. En effet, comme les espaces de temps ne sont pas affectés aux sources de trafic de manière constante, une source peut insérer de l'information à son rythme dans les espaces libres laissés par les autres sources. De ce fait un tel réseau permet d'intégrer toutes sortes de débits. Etant transparent à l'information présente dans les cellules, il peut supporter tous les types de service. Ceci est rendu possible en particulier par l'utilisation de liens de transmission à hauts débits, par exemple 150 Mbts/s ; plus tard on peut penser que le débit sera encore plus grand.

La technique A.T.M permet d'atteindre des vitesses très élevées de commutation (plusieurs Gbit/s avec la technologie actuelle), et d'assurer un délai très faible d'attente pour la transmission des cellules à travers le réseau.

Pourquoi des cellules de 53 octets ?

Lorsque l'A.T.M est rentré dans le processus de normalisation au C.C .I.T, la position française s'est trouvée remise en question par les américains et les japonais qui préféraient une cellule de taille plus longue pour la transmission des données.

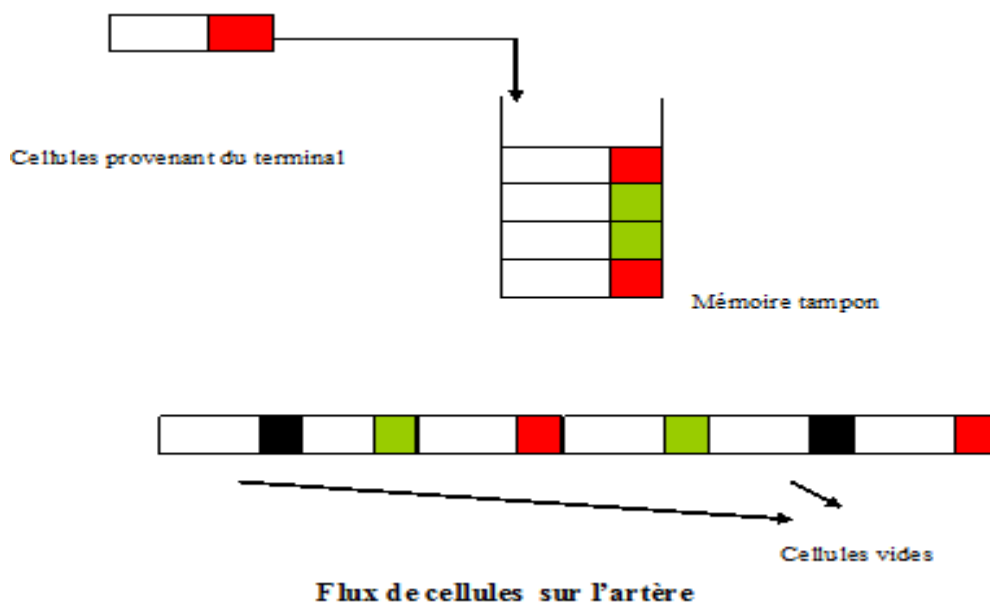
Ce conflit, démarré en juillet 1987 à Hambourg, a duré deux années, jusqu'à la réunion de Melbourne, en juin 1989, avec l'adoption de la première recommandation I.121. A ce moment, les français proposaient 32 octets de champ d'information et 4 octets d'en-tête, les américains $64 + 5$ et les japonais $66 + 6$. Enfin, après une opposition finale entre les européens et les américains, la taille retenue a été de $(32 + 64)/2 = 48$ octets pour le champ d'information, et de $(4 + 5 + 6)/3 = 5$ octets pour l'en-tête.

Multiplexage de cellules

Selon le principe de la technique A.T.M, les sources de trafic envoient leur flux d'information sous forme de cellules, à leur rythme propre : il n'y a plus d'allocation temporelle périodique comme pour la technique temporelle synchrone. Ce débit peut être éventuellement variable au cours de la connexion. Le mode asynchrone permet un partage simple et souple des capacités du réseau, mais pose des problèmes de congestions lorsque plusieurs sources sont autorisées

à émettre en même temps. Il donne naissance à des conflits entre cellules à chaque nœud de communication : c'est pourquoi des files d'attente sont placées à la sortie des commutateurs. Nous effectuerons, dans la deuxième et la troisième partie, une étude mathématique qui aide à gérer ces problèmes de trafic, lorsqu'il y a deux classes de priorités.

Le principe du multiplexage est très simple (figure ci-dessous) ; lorsque l'on veut multiplexer un flux provenant d'un lien partiellement chargé, on insère simplement les cellules provenant de la source à la place des cellules vides. Les cellules étant disposées de manière non déterministe, on introduit une mémoire tampon qui reçoit les cellules de la source, en attendant le passage de cellules vides.



Remarquons simplement que pour assurer une certaine fluidité dans l'écoulement des cellules d'une source et ne pas faire déborder la mémoire tampon, il faut limiter la charge totale du flux de cellules : il a été démontré, au C.N.E.T-LANNION, qu'une limitation de la charge à 0.85 est suffisante pour assurer un taux de perte de $1E-10$ avec une mémoire tampon de 64 places. Lorsqu'il y a un très grand nombre de sources, par exemple plusieurs centaines, qui émettent par cellules et non par rafales, on peut supposer que le flux d'arrivée est poissonnien.

Le propos de cette thèse est d'apporter une contribution à la gestion du trafic en utilisant le critère de priorité.

Il est à noter que, quelle que soit la technologie adoptée lors de la traversée des carrefours A.T.M, l'asynchronisation du mode A.T.M provoque des files d'attente. Leur étude est l'un des objets essentiels de [PBL97]. Le chapitre 11 de cet ouvrage aborde l'utilisation de priorités.

Le problème est le suivant. Du seul fait de l'existence de files d'attente, il semble difficile de dépasser des charges élevées pour les cellules associées à des communications ne tolérant que de faibles taux de perte ou de retard, ce qui est le cas, par exemple, des communications téléphoniques. Il est naturel de considérer deux classes de communications, la première classe étant prioritaire par rapport à la deuxième classe. Pour la classe prioritaire, on se limite à des taux de charge souvent compris entre 0.8 et 0.85.

Il est facile de pressentir intuitivement que l'un des inconvénients majeurs de cette utilisation de priorités est le risque d'avoir des délais de transmissions très longs pour les communications non prioritaires. Ce point est analysé au chapitre 11 précité mais l'étude qui y est effectuée est essentiellement descriptive. L'objectif de cette thèse est de définir et d'étudier un modèle mathématique adapté aux réseaux A.T.M dont les cellules peuvent avoir deux niveaux de priorité et un modèle de files d'attente avec rappel comportant deux classes de clients (transitaires et récurrents).

Chapitre 1

Les Principes de l'A.T.M

L'objet de ce chapitre est de décrire les principes généraux de l'A.T.M.

1.1 Les Fondements de l'A.T.M

L'A.T.M a été défini pour répondre à une sorte de défi posé par l'évolution des besoins en télécommunication et des techniques de transfert d'information. Il est le fruit tout à la fois d'une intuition technique et d'une analyse de la situation en matière de services à haut débit ; il découle de cette analyse qu'un mode de transfert intégré est bien la seule manière fiable de relever le défi.

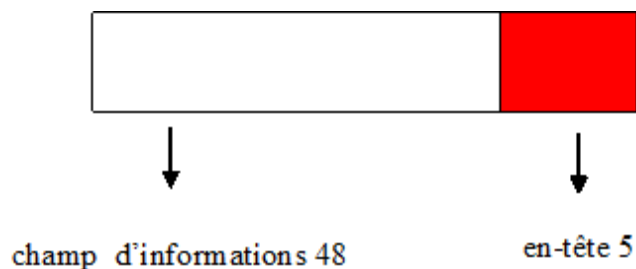
L'A.T.M (Asynchronous Transfert Mode) est une technique de commutation, de multiplexage, voire de transmission, qui est une variante de la commutation par paquets en ce qu'elle fait appel à des paquets courts et de taille fixe appelés cellules.

Dans les commutateurs, le traitement de ces cellules est limité à l'analyse de l'en-tête pour permettre l'acheminement vers les files d'attente appropriées.

1.1.1 La Cellule

La cellule est l'unité de base du transport. Elle comporte deux parties (figure ci-dessous) .

- Un en-tête, ou étiquette.
- Un champ d'information, ou charge utile.



Une première remarque est que l'en-tête diffère légèrement suivant que l'on se place à une interface réseau/utilisateur (User to Network Interface : UNI) ou à une interface réseau /réseau (Network to Network Interface : NNI).

Le format normalisé par le C.C.I.T prévoit 5 octets pour l'en-tête et 48 pour la charge utile.

L'en-tête figuré ci-dessous contient essentiellement un numéro binaire représentant une adresse logique qui identifie le circuit virtuel auquel est affectée la cellule. En réalité l'adresse comprend deux numéros indépendants et complémentaires : l'un identifie une voie virtuelle (VCI), l'autre un faisceau virtuel, c'est-à-dire un groupe de circuits virtuels (VPI) ; son rôle principal est d'identifier les cellules appartenant à une même connexion et de permettre l'acheminement.

Détail des 5 octets d'en-tête de cellule à l'UNI (Interface usager).

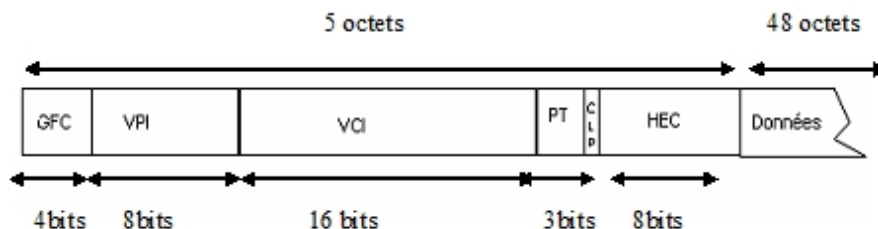


FIG. 1.1 – Détail des 5 octets d'en-tête de cellule à l'UNI (Interface usager)

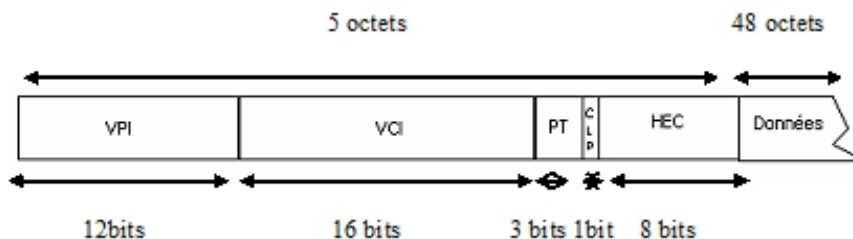


FIG. 1.2 – Détail des 5 octets d'en-tête de cellule au NNI (interface nœud du réseau)

VPI /VCI : Champ adresse (identificateur de : Conduit Virtuel / Circuit Virtuel)

GFC : Generic Flow Control (contrôle de flux)

PT : Payload Type (type d'information transporté dans la cellule)

CLP : Cell Loss Priority (priorité à la perte utilisée en contrôle de flux)

HEC : Header Error Control (protection de l'en-tête).

Le champ GFC est composé de 4 bits. Il est logiquement prévu pour les fonctions locales de contrôle de flux, mais n'est pas utilisé à ce jour. Selon les spécialistes de l'A.T.M Forum, l'ensemble des bits doivent être positionnés à 1.

Le champ adresse (VPI/ VCI) est constitué de 24 bits à l'interface UNI (8 bits pour le VPI, et 16 bits pour le VCI) et de 28 bits pour l'interface NNI (champ VPI étendu à 12 bits).

Un canal virtuel, identifié par le couple VPI / VCI, est une route permettant le transport de cellules A.T.M.

Un conduit virtuel, identifié par un VPI particulier, est un regroupement de canaux virtuels. A l'issue de la phase d'établissement de connexion, les en-têtes des cellules comprennent, dans

le champ VPI /VCI, des informations permettant l'identification du circuit virtuel sur lequel transitent les informations. L'identificateur du circuit virtuel (couple VPI/VCI) est défini localement entre deux commutateurs. En d'autres termes, une connexion entre deux usagers sera définie par une succession de couples VPI/VCI entre chaque commutateur participant à la connexion .

Le champ Payload Type (PT) est constitué de 3 bits. Il est utilisé pour distinguer les cellules usagers des autres cellules, par exemple les cellules de gestion de trafic.

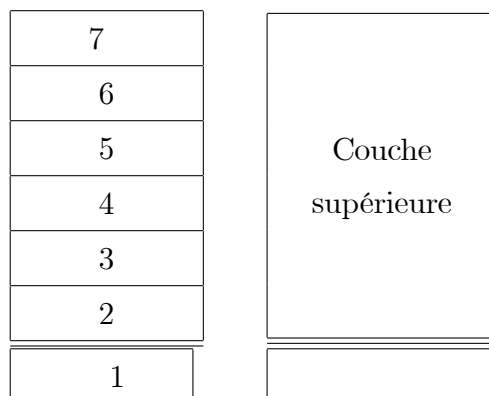
Le champ Cell Loss Priority (CLP) permet de séparer les cellules en deux niveaux de priorité. Les cellules de priorité basse possèdent un bit CLP positionné à 1, les cellules de priorité haute possèdent un bit CLP positionné à 0. Dans le cas où la saturation d'un lien entre commutateurs est atteinte, ce sont une dégradation de la qualité de l'image, ce qui est moins pénalisant qu'une interruption de la les cellules de priorité basse qui sont détruites en premier ; ceci entraîne, par exemple, transmission.

Le champ Header Error Control (H.E.C) est un Checksum permettant de protéger l'en-tête de la cellule contre les erreurs de transmissions.

1.2 Protocoles A.T.M.

Par rapport au modèle O.S.I, A.T.M assure les fonctions de la couche liaison ainsi qu'une partie de la couche réseau.

Il s'appuie sur la couche physique et fournit le service de transport pour les besoins de la couche AAL, Adaptation to A.T.M Layer, qui adapte les données des couches supérieure.



1.2.1 Couche Physique

La couche physique concerne l'adaptation d'A.T.M sur un support physique. Elle est constituée de deux sous couches :

P.M, "Physical Medium ", qui s'occupe en particulier du codage et du décodage du signal.

T.C, "Transmission Convergence", s'occupe de l'adaptation du débit d'A.T.M au débit physique, la protection de l'en-tête des cellules A.T.M.

1.2.2 Couche A.T.M.

La couche A.T.M est responsable de l'acheminement des données, sous formes de cellules sur le réseau, de la traduction des indicateurs VPI/VCI. Dans le cas de l'interface UNI, elle y apporte le contrôle de flux grâce au champ GFC.

A.T.M ne se préoccupe pas du contenu de la charge utile dont la gestion est laissée aux couches supérieures.

1.2.3 Couche AAL

Cette couche se situe entre la couche ATM et les couches supérieures. Elle offre aux applications des services adaptés à chaque type de trafic susceptible d'être transporté par A.T.M : son, données ou image.

La couche AAL est découpée en deux sous couches .

CS gestion des erreurs, gigue des cellules.

SAR Segmentation et réassemblage.

La sous couche CS (convergence Sublayer) sous couche de convergence, gère également un autre, celui de la gigue des cellules par l'utilisation d'un tampon de réception.

1.2.4 Commutation des Cellules dans A.T.M.

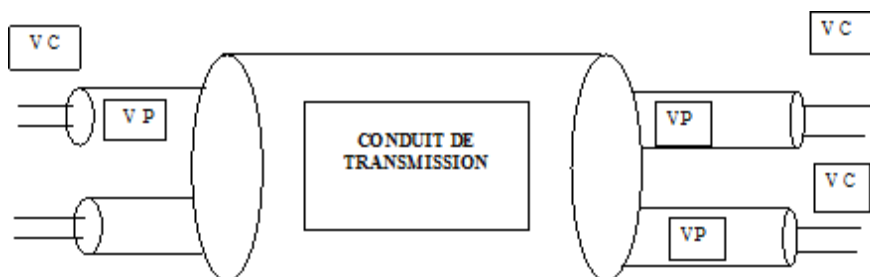
A l'issue de la phase de connexion, seul le chemin est enregistré dans les commutateurs A.T.M.

Les liaisons entre les commutateurs sont alors partagées avec d'autres commutateurs, ce qui confère au chemin entre commutateurs, l'appellation de circuit (VC).

A l'issue de la phase d'établissement de connexion, l'en-tête de chaque cellule émise contient des informations permettant l'identification du circuit virtuel à emprunter. Plus précisément, un circuit est réalisé à partir de deux identifiants :

- . Le numéro de canal virtuel (Virtual Chanel Identifier : VCI)
- . Le numéro de conduit virtuel (Virtual Path Identifier : VPI) permettant d'acheminer un ensemble de canaux virtuels, ou faisceau virtuel.

Ainsi, l'en-tête de chaque cellule comporte deux informations indissociables : VPI /VCI. On peut donc voir le support de transmission (circuit physique) comme une juxtaposition de plusieurs conduits virtuels, eux-mêmes composés de plusieurs circuits virtuels.



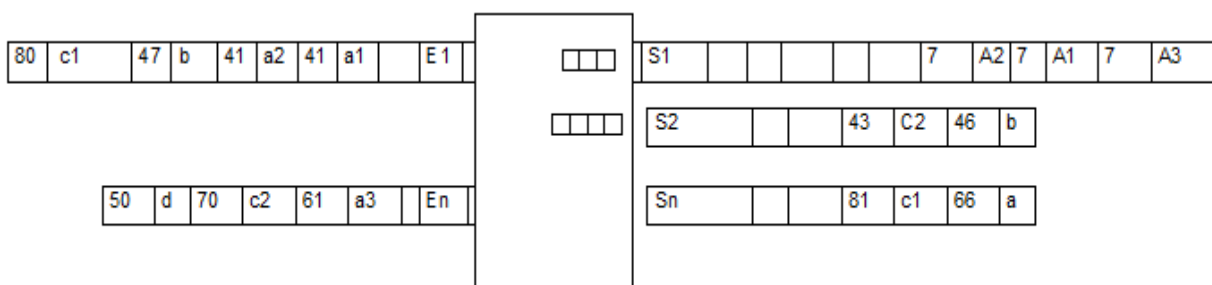
Les identificateurs (VPI,VCI) permettent de développer deux types de commutateurs :

- Les commutateurs A.T.M de voies virtuelles, qui prennent en compte les deux identificateurs (VPI et VCI).
- Les commutateurs A.T.M de faisceaux virtuels, généralement appelés brasseurs A.T.M (A.T.M Digital Gross Connects ou A.T.M DGC), qui n'utilisent que l'identificateur de faisceau virtuel (VPI), pour acheminer l'information le long d'une route. Ils sont contrôlés par les appareils de gestion du réseau. De tels brasseurs peuvent être utilisés notamment pour fournir des routes de secours en cas de congestion.

Pour illustrer ce qui vient d'être expliqué et pour le rendre plus compréhensible, nous allons prendre un exemple qui examine tous les cas possibles d'affectations d'en-tête des cellules à l'entrée du commutateur.

Du point de vue extérieur, les cellules A.T.M sont physiquement commutées d'une artère d'entrée E_i en une artère de sortie S_j et, parallèlement, leur valeur d'en-tête est traduite par une table de description des circuits virtuels. Celui-ci décrit, pour chaque circuit virtuel entrant, la voie de sortie et la nouvelle valeur de (VPI /VCI) sur cette voie.

La figure ci-dessous montre comment s'effectue le routage dans un commutateur.



Port entrée	VPI/VCI	Port Sortie	VPI/VCI
$E1$	(0, 41)	$S1$	(0, 7)
	(0, 41)	$S2$	(0, 7)
	(0, 47)	$S3$	(0, 46)
	(0, 80)	S_N	(0, 81)
E_N	(0, 61)	$S1$	(0, 7)
	(0, 70)	$S2$	(0, 43)
	(0, 47)	$S2$	(0, 46)
	(0, 80)	S_N	(0, 66)

Pour différencier l'ordre des cellules d'une même connexion, nous notons a_1, a_2, a_3 les cellules de la communication a . De même c_1, c_2 représentent les cellules de la communication c .

La cellule a_1 qui arrive sur l'artère $E1$ avec l'en-tête VPI /VCI (0,41) est transmise sur la voie de sortie $S1$ avec VPI /VCI (0,7).

La cellule a_2 est retransmise de la même façon que a_1 car elle se trouve sur la même artère d'entrée.

La cellule b qui arrive sur l'artère E1 avec l'en-tête VPI/VCI (0,47) est retransmise sur la voie de sortie S2 avec VPI/VCI (0,46).

La cellule c1 arrive sur l'artère E1 avec l'en-tête VPI/VCI (0,80) et est retransmise sur l'artère Sn avec VPI/VCI (0,81).

La cellule a3 arrive sur l'artère En avec l'en-tête VPI/VCI (0,61) et est retransmise sur la voie de sortie S1 avec l'en-tête (0,7).

La cellule c2 qui arrive sur l'artère En avec l'en-tête VPI/VCI (0,70) est retransmise sur la voie de sortie S2 VPI/VCI (0,43).

Enfin la cellule d qui arrive sur l'artère En avec l'en-tête VPI/VCI (0,50) est retransmise sur la voie de sortie Sn avec l'en-tête VPI/VCI (0,66) .

Nous remarquons dans cet exemple que deux cellules d'entrée différentes (par exemple E1 et En) arrivent simultanément dans le commutateur et sont dirigées vers la même sortie S1.

Pour pallier à ce problème il faut installer une file d'attente à la sortie du commutateur pour stocker les cellules qui ne peuvent pas être desservies et pour empêcher la perte des cellules . Cette solution permet de résoudre le problème de conflit possible.

Plusieurs files d'attente sont présentées à la sortie du commutateur. Nous n'en étudierons qu'une seule les autres sont analogues.

Il est à noter que nous ne nous intéresserons pas à la technologie du commutateur. On suppose qu'il fonctionne à une vitesse supérieure à celle des artères entrée sortie.

En effet sur un temps cellule, il fait écouler au moins 2000 communications par seconde. Il n'est donc pas intéressant d'étudier les retards des cellules à l'intérieur du commutateur, d'autant plus que leur étude sera rapidement obsolète vue l'évolution rapide de la technologie.

Nous pouvons donc supposer, d'après ce qui a été dit plus haut, que le commutateur fonctionne point-à-point puisque aucun stockage à l'intérieur du commutateur n'est nécessaire . Il ne peut donc survenir aucun blocage.

En conclusion, un commutateur A.T.M exécute les trois fonctions de base suivantes :

- Routage.
- Mise en file d'attente.
- Traduction de l'en-tête.

Le rôle du commutateur est donc d'établir une connexion entre une artère d'entrée et une artère de sortie en fonction d'une information de routage.

Le commutateur doit également gérer plusieurs flux de cellules différenciées par des niveaux de priorités et fournir un traitement préférentiel aux cellules de haute priorité (files d'attente par niveau de priorité). Si la souplesse a été très vite mise en exergue lorsqu'il s'est agi de choisir les techniques des réseaux à haut débit, c'est en partie pour pallier à une incapacité avérée de la prospective à imaginer l'évolution en matière de demande de services. Non pas que cette prospective soit mal conduite, mais le marché semble réellement imprévisible.

Avant de décrire les techniques qui ont amené à la définition du mode de transfert A.T.M, il est utile de préciser les caractéristiques des modes de commutation .

1.3 Multiplexage et Commutation

Trois grandes familles de mécanismes de multiplexage et de commutation diffèrent à la fois par la manière dont les supports de transmission sont partagés entre les voies de communication et le degré de compatibilité entre le multiplexage et la commutation.

1.3.1 Technique Spaciale

Dans une première famille, les commutateurs établissent des itinéraires physiques permanents pour chaque voie de communication. Quel que soit le mode de multiplexage retenu, il est indispensable de démultiplexer préalablement à toute opération de commutation.

La ressource mise à la disposition de l'utilisateur peut être considérée comme un lien physique établi pour la durée de la communication et présentant une largeur de bande déterminée. Il revient au terminal de fixer le mode et le degré d'utilisation de cette bande passante. On parle donc d'un mode de transfert en bande de base. On parle également d'un mode de transfert en mode circuit ou commutation de circuit, tel que le téléphone.

Lorsque nous, ou notre ordinateur, passons un appel téléphonique, le commutateur téléphonique local cherche à établir dans le réseau un chemin que l'on nomme voie téléphone ou circuit entre votre téléphone et celui de votre correspondant. Ce circuit physique emprunte divers types de lignes : des câbles électriques, des fibres optiques ... La technique d'établissement d'un circuit dans le réseau téléphonique porte le nom de commutation de

circuit.

Lorsqu'un signal d'appel transite dans le commutateur, une connexion physique est établie entre la ligne sur laquelle parvient l'appel et l'une des lignes de sortie en direction du destinataire.

Dès qu'un appel a donné lieu à l'établissement d'un circuit, un chemin est établi entre l'appelant et l'appelé jusqu'à ce que la commutation soit interrompue.

La commutation de circuits possède une propriété importante : l'obligation d'établir un circuit de bout-en-bout avant que toute information soit transmise. Le temps écoulé entre l'émission de l'appel et la sonnerie chez le destinataire correspond au temps d'établissement du circuit. Il peut atteindre facilement 10 secondes, voire beaucoup plus sur de longues distances ou lors d'appels internationaux. Pendant tout ce temps le système téléphonique cherche un chemin dans le réseau pour établir un circuit entre les correspondants .

Notons qu'avant même que la transmission d'information ne commence, le signal d'appel s'est propagé dans le réseau sur le circuit en cours d'établissement jusqu'au destinataire, puis un accusé de réception a été délivré à l'appelant (le signal de la sonnerie et/ou le signal de décrochement du téléphone).

Pour un bon nombre d'applications, notamment la vérification des cartes de crédits à partir d'un terminal point de vente, ces longs temps d'établissement de circuits sont plutôt indésirables.

En contrepartie, lorsque le circuit est établi entre les deux parties en communication, le délai de transmission des informations est très faible ; il est de 5ms pour 1000 km .

De plus, lorsque le circuit est établi, il n'y a pas de risque de congestion. Lorsque l'appel a été mené à bien, il ne risque plus d'y avoir d'occupation de la ligne. En revanche il peut très bien y avoir une congestion avant que le circuit ne soit établi, par exemple en cas de surcharge des commutateurs ou des artères de communications vers le destinataire.

1.3.2 Techniques Temporelles

Dans les deux autres familles, les ressources (le débit disponible sur un support de transmission ou un point de connexion) sont partagées dans le temps.

Mode Temporel Synchrones

Dans le mode de transfert temporel synchrone, les ressources sont organisées en trames récurrentes, elles-mêmes découpées en intervalles de temps. Ce sont ces intervalles de temps qui transportent les unités de données. Chaque voie de communication se voit réserver dans chaque trame un intervalle de temps de position déterminée.

Mode Temporel Asynchrone

Dans le mode temporel asynchrone, les ressources, toujours partagées dans le temps, sont organisées en unités de données appelées paquets ou cellules qui se suivent sans référence absolue de position. Ceci oblige à faire porter l'identification de la voie de communication par l'unité de données (identificateur par étiquette) .

Il n'y a plus d'affectation fixe d'intervalles de temps aux unités de données d'une voie et la source peut émettre des unités de données à son rythme propre, sans référence directe avec le réseau, sinon évidemment du point de vue du rythme de transmission instantanée.

On parle ici d'un transfert en mode paquet .

Le multiplexage provoque alors des files d'attente avec une part plus ou moins importante d'indéterminisme.

1.4 Comparaison Entre la Commutation de Circuits et Par Paquets

La commutation de circuits et la commutation par paquets présentent certaines différences. La différence majeure est que la commutation de circuits réserve statistiquement et à l'avance un circuit et l'intégralité de sa bande passante pour une communication. La commutation par paquets ne procède pas de la sorte. En effet, un circuit n'est utilisé qu'en cas de besoin ; il est libéré dès qu'il n'est plus nécessaire. En commutation de circuits, la bande passante initialisée pendant l'allocation d'un circuit peut être partiellement gaspillée (par exemple pendant des périodes de silence). Il en est tout autrement pour la commutation par paquets.

En effet plutôt que de gaspiller inutilement la bande passante, elle est immédiatement attribuée à une autre communication particulière. Toutefois, cette façon de faire ne présente pas

que des avantages. En effet, lorsqu'un brusque à coup de trafic survient, cela peut submerger les files d'attente du commutateur de paquets et sa capacité de mémorisation locale étant saturée, cela risque d'entraîner des pertes de paquets. Contrairement à ce qui se passe avec la commutation de circuits, en commutation par paquets, il est clair que chaque commutateur doit effectuer diverses conversions, notamment des vitesses de transmission et de codage des signaux, sur les artères de communications internes au réseau. Il met également en œuvre tout un ensemble de mécanismes de contrôle de flux pour éviter la saturation et la congestion du réseau.

Sur certains réseaux à commutation par paquets, la remise en ordre des paquets n'est pas garantie. En revanche, en commutation de circuits, il n'est jamais nécessaire de remettre en ordre les informations transmises.

Une autre différence est à l'avantage de la commutation de circuits, c'est la transparence totale du réseau. Les deux correspondants peuvent utiliser le protocole de dialogue, la vitesse de transmission, le format des données qu'ils souhaitent (dans la limite du possible quant à la vitesse de transmission). En revanche en commutation par paquets, c'est l'opérateur qui détermine et impose tous les paramètres de base du réseau. L'utilisateur doit impérativement les respecter.

Enfin une différence importante entre les deux techniques concerne la facturation des communications. En commutation par paquets les coûts sont étroitement liés au volume d'informations (nombres d'octets ou de paquets) transmises dans le réseau. La distance entre les correspondants et la durée d'une communication n'ont pratiquement pas d'influence sur les coûts, à l'exception peut être du trafic international. En revanche, en communication de circuits les coûts sont principalement basés sur la durée et la distance et non sur le trafic. Quelles que soient leurs différences, il est à noter que ces deux techniques sont très utilisées dans les réseaux.

1.5 Les Caractéristiques Principales de l'A.T.M.

L'A.T.M a hérité, à travers l'en-tête des cellules, de la flexibilité de la commutation de paquets et, à travers le champ d'information de taille constante transmis de façon transparente, de la simplicité de la commutation de circuits.

Les réseaux A.T.M sont orientés connexions. Pour faire un appel, il faut envoyer un message

d'établissement de la connexion, après quoi toutes les cellules emprunteront le même chemin jusqu'à destination. La remise en ordre n'est pas garantie mais leur ordonnancement l'est. Si par exemple les cellules 1,2 sont envoyées dans cet ordre et si elles arrivent toutes les deux, ce sera toujours dans ce même ordre, jamais 2 avant 1 .

Pour un certain nombre d'artères, le débit prévu est de 155Mbits/s. Ce débit permet notamment de transmettre des images de télévision à très haute définition.

Dans cette thèse, nous utiliserons souvent le mot "artère " pour désigner un canal sur lequel circulent des informations les unes derrière les autres sous forme d'une suite ininterrompue de cellules.

Dans les réseaux A.T.M et sur les longues distances, le support physique de ces artères sera de plus en plus souvent la fibre optique.

Il est probable que, au moins pendant quelques années, la plupart des artères A.T.M fonctionnant sur des longues ou assez longues distances auront des débits de 150 Mbits/s ou plus. Pour la simplicité nous choisissons ce débit 150Mbits /s. Cela donne une vitesse de traitement du commutateur légèrement supérieure à 360000 cellules par seconde (une cellule fait 53 octets). C'est bien plus que la capacité d'une file d'attente et même bien plus que la capacité totale d'absorption du réseau .Cela veut dire que entre le départ et l'arrivée d'une cellule le nombre de cellules dans l'artère est plus élevé que dans la file d'attente. Cela est dû au débit de 150Mbits /s

Le fait que les cellules soient petites et de taille fixe (53 octets) permet de concevoir des commutateurs susceptibles de fonctionner sous les contraintes temporelles présentés ci-dessus.

Si les cellules avaient été plus longues et de taille variable, la potentialité d'atteindre de telles vitesses de commutation aurait conduit à des systèmes très complexes. C'est pour cette raison que la technologie A.T.M est basée sur des petites tailles fixes.

Les réseaux A.T.M sont des réseaux à circuits virtuels. En effet lorsqu'un circuit virtuel est établi cela signifie qu'une route a été balisée dans le réseau, allant d'une source vers un destinataire, c'est à dire que chaque commutation a marqué cette route par une entrée dans sa table de routage interne lui permettant de faire circuler sur le circuit virtuel les paquets qui y sont destinés.

Tout commutateur réserve un certain nombre de ressources à l'établissement de nouveaux circuits virtuels.

Deux principes communs régissent le fonctionnement de tous les commutateurs A.T.M

- 1- commuter les cellules avec un taux de perte de cellules aussi faible que possible
- 2- ne jamais réordonner les cellules sur un circuit virtuel.

Grâce au principe 1, on permet la perte de cellules en cas d'urgence avec un taux le plus faible possible. Un taux de perte de l'ordre $10.E-12$ (une cellule perdue toutes les $10.E.12$ transmissions) est tout à fait acceptable.

Le principe 2 assure que les cellules provenant d'un circuit virtuel doivent sortir du commutateur dans l'ordre d'arrivée, sans exception.

Une bonne question se pose à propos de tous les commutateurs A.T.M : que peut-on faire lorsque deux ou plusieurs cellules arrivent sur des entrées différentes et sont à destination d'une même sortie pendant le même cycle ?

Devant une telle situation, les réseaux A.T.M sont sujets aux congestions.

Le contrôle de trafic et de congestion est un des problèmes clés du développement des nouveaux réseaux A.T.M.

La conséquence du partage de ressources est que des phénomènes de congestion peuvent apparaître si plusieurs utilisateurs requièrent les mêmes ressources au même instant. La congestion est définie comme un état du réseau (ou plus précisément de certains éléments du réseau) dans lequel le réseau n'est plus capable de répondre aux objectifs de performances fixés.

Dans les réseaux A.T.M, des paramètres concernant les performances du transfert des paquets dans les connexions déjà établies, tels que la probabilité qu'un paquet soit perdu, la probabilité qu'un paquet soit erroné, le délai moyen de la traversée d'un paquet... doivent être considérés. C'est le but de cette thèse.

Rappelons quelques caractéristiques du mode A.T.M. Dans un réseau à commutation de paquets en mode connecté (exemple A.T.M) un circuit virtuel est établi avant la transmission des informations de l'utilisateur. Ces informations sont transmises dans des paquets. Les différents circuits virtuels partagent la capacité des liens de transmission. Différents paquets de différents circuits virtuels peuvent requérir un même lien de transmission en même temps. Pour éviter les pertes, des buffers sont prévus dans les commutateurs et les multiplexeurs. Nous nous intéressons au buffer qui se trouve à la sortie du commutateur. Notons que, en fait ce buffer est souvent fictif.

Il serait intéressant de rerouter une partie des connexions passant sur ce commutateur sur des commutateurs moins utilisés.

Afin que le réseau puisse répondre aux objectifs de performances, des mécanismes de contrôle doivent être prévus.

1.6 Contrôle de Congestion

Il faut d'abord se convaincre que même si les artères de l'A.TM auront des débits énormes et si on pourra mettre à chaque carrefour des mémoires considérables, on ne pourra pas accorder un accès libre à tous les usagers.

L'A.T.M détruit sélectivement des cellules en cas de saturation d'un commutateur ; l'A.T.M peut également refuser de nouvelles connexions si le réseau supporte trop de trafic.

Il existe plusieurs méthodes de contrôle de congestion dont on cite quelques unes.

Les méthodes employées pour éliminer ou éviter la congestion dans un réseau peuvent être de deux sortes : réactives ou préventives. Une méthode réactive consiste à prendre des décisions pour éliminer la congestion une fois que celle-ci s'est produite.

Dans ce cas une congestion peut être résolue en envoyant un message en retour (feed back) aux sources pour leur faire réduire leur débit. Si l'artère A.T.M a un débit de l'ordre de 150 Mbit /s, le signal met un temps important pour remonter jusqu'aux dites sources avant que la réduction des débits devienne effective. De ce fait un grand nombre de cellules seront perdues avant que la congestion ne soit éliminée.

Une technique préventive agit avant qu'une congestion ne se produise.

1.6.1 Contrôle Par Boucle Ouverte

Dans les systèmes en boucle ouverte, on essaye de résoudre un problème au mieux lors de la phase de conception, essentiellement en s'assurant qu'un problème n'arrivera pas. Dès que le système est en cours de fonctionnement, aucune correction même mineure n'est possible.

Les moyens de contrôle d'un système en boucle ouverte peuvent être, par exemple, de décider à quel moment on choisit : d'augmenter le trafic, de recevoir de nouveaux paquets, de détruire des paquets et de décider lesquels.

1.6.2 Contrôle Par Boucle Fermée

Les moyens de contrôle du réseau par boucle fermée sont basés sur la rétroaction (la réalisation d'une boucle de retour).

- Surveiller le système afin de détecter quand et où la congestion apparaît.
- Envoyer cette information aux endroits où les actions doivent être prises.

1.6.3 Algorithme du Seau Percé

Imaginez un seau d'eau dont le fond est percé d'un trou par lequel l'eau s'évacue goutte à goutte. Peu importe à quel débit l'eau arrive dans le seau, son écoulement par le trou se fait à une vitesse constante c , tant qu'il y a de l'eau dans le seau ; l'écoulement s'interrompt dès qu'il n'y a plus d'eau. Lorsque le seau est plein le surplus d'eau arrivant est perdu, le seau déborde sans que cela influe sur l'écoulement d'eau dans le trou.

Le principe du seau peut être transposé sans problème à la transmission de paquets dans un réseau. Un seau percé est représenté d'une façon concrète par une file d'attente de taille fixe.

C'est JONATHAN TURNER (1986) qui, le premier a eu l'idée de réguler le débit en utilisant l'algorithme du seau percé (LEAKY BUCKET ALGORITHM). En fait, ce mécanisme n'est rien d'autre qu'un serveur gérant une file d'attente avec un temps de service constant.

Il est à noter que ce mécanisme admet plusieurs variantes technologiques. Nous allons en expliquer une facile à comprendre.

Soit S une source à réguler. Appelons m la taille du leaky bucket (m est un entier) et d le débit alloué à la source S . On se réfère à une artère A et d est la proportion de A attribuée à S . Soit t le produit de $1/d$ et du temps cellule où le temps-cellule est le temps qu'une cellule met à traverser un point A' de A .

Au début la source S dispose de m jetons. Le nombre total de jetons dont dispose S ne peut jamais dépasser m . Lorsqu'une cellule est transmise, un jeton est détruit. Quand S n'a plus de jeton, elle ne peut plus envoyer de cellules. Enfin, S reçoit de nouveaux jetons avec la période t tant que le capital de S est inférieur à m ; par contre, tant que ce capital vaut m , S ne peut pas recevoir de jeton.

1.6.4 Contrôleur Espaceur

Le principe du contrôleur espaceur est de réguler séparément chaque connexion. Un débit maximum peut être exprimé par un délai d'inter-arrivée des cellules, soit un délai T . On utilise T pour contrôler les instants d'émission des cellules.

Si dans un intervalle T une cellule arrive, elle peut être transmise sans délai ; si une deuxième cellule arrive, elle sera conservée jusqu'au début du prochain intervalle de temps T .

Ce mécanisme suppose des capacités de mémorisation avant transmission. Il est évident que, si la taille de la file est dépassée, les cellules en excès seront perdues.

Le principe de ce mécanisme est le suivant :

Soit une connexion B dont le délai ne doit pas dépasser T .

A la cellule numéro k de la connexion B est associé une heure théorique d'arrivée (Theoretical Arrival Time) soit $TAT(k)$; soit t l'heure réelle de l'arrivée de la cellule numéro k .

TAT est calculée par récurrence de la façon suivante :

$$TAT(k+1) = \sup (t , TAT(k)) + T$$

1.6.5 Contrôle de Débit

Un contrôle de débit est mis en place à l'accès du réseau. Il sort du cadre de cette thèse de détailler tous les mécanismes de contrôle qui sont complexes. Pour chaque connexion, on a :

- Une mesure du débit offert qui utilise habituellement des algorithmes de type "seau percé" (LEAKY BUCKET).
- Un contrôle qui permet d'éliminer les cellules en excès par rapport au contrat, où les déclasser d'un niveau de priorité. Cette méthode porte le nom de violation tagging.
- Un espacement (spacing) effectué sur les cellules, avant leur injection dans le réseau, qui dépend de la bande passante réservée pour la connexion considérée.

1.7 Données du Problème

Le but de ce travail est d'étudier les retards pris par les cellules dans les éléments actifs du réseau en fonction du trafic accepté. La connaissance de la loi de ces retards permet au concepteur de choisir des mémoires rapides en fonction d'un taux de perte toléré de cellules. Le cœur de notre démarche consiste à mettre en évidence certaines variables aléatoires associées aux retards d'une connexion donnée. Le modèle mathématique sous-jacent est alors un processus markovien homogène qui évolue à temps discret et dont on considère essentiellement le régime stationnaire et le régime transitoire. Ce modèle sera étudié dans le chapitre suivant (on introduira la notion de priorité).

Notre objectif dans cette thèse n'est pas d'étudier la gestion du réseau global, mais de donner un ordre de grandeur des taux de charge acceptables sur l'artère considérée en fonction de la taille tolérée de la file d'attente afin d'avoir des probabilités de perte de l'ordre de $10.E-9$.

Pour les besoins de notre modélisation mathématique il n'est pas nécessaire de connaître exactement tous les débits possibles. Ceux-ci vont d'ailleurs profondément évoluer avec la technologie "comme l'exprime par exemple la célèbre loi de MOORE sur le doublement de la puissance des processeurs, tous les deux ans ". Enfin, il est hasardeux de prédire les caractéristiques des services futurs.

Par contre les ordres de grandeurs jouent un rôle essentiel, plus précisément nous utiliserons très souvent l'hypothèse " poissonnienne " :

Celle-ci est valable si on mélange un très grand nombre de connexions, par exemple 1000 est un très grand nombre qui justifie pleinement l'hypothèse poissonnienne.

C'est pourquoi il est important de savoir qu'une artère A.T.M de 150 Mbits/s peut transporter simultanément au moins 2000 communications téléphoniques, voire beaucoup plus.

Nous allons maintenant donner quelques règles importantes pour la modélisation mathématique. Du point de vue technologique, un réseau A.T.M est bi-directionnel.

Pour la modélisation mathématique il est préférable de faire "comme si " le réseau était orienté, du moins quand on effectue une étude fine du comportement des cellules non prioritaires comme c'est le cas dans cette thèse. Cela signifie que l'on étudie les cellules qui vont dans un sens sans tenir compte de celles qui vont en sens inverse.

Une autre caractéristique fondamentale pour notre étude est que toutes les cellules d'une

même connexion restent toujours dans le même ordre et passent exactement par les mêmes circuits virtuels, car si on leur permettait de prendre des routes différentes, et relativement indépendantes les unes des autres, du seul fait des files d'attente, elles ne seraient pas dans le même ordre à l'arrivée qu'au départ. Pour pouvoir les lire, il faudrait alors les stocker à l'arrivée et les remettre dans l'ordre ce qui poserait des problèmes technologiques considérables : notamment, cela obligerait à augmenter la taille de l'en-tête pour contenir le numéro d'ordre de chaque cellule.

La discipline de service est partout " premier arrivée premier servi" pour toutes les cellules d'une même connexion empruntant exactement les mêmes circuits virtuels.

Nous supposons que juste avant l'arrivée au multiplexeur chaque artère dispose d'un contrôleur de congestion.

Le contexte technologique dans lequel nous nous plaçons est le suivant. Nous considérons un carrefour (commutateur). Nous supposons que le nombre d'artères entrantes est élevé, par exemple 60, le nombre d'artères de sorties importe peu dans le modèle car elles peuvent être étudiées de la même façon. Soit A une telle artère de sortie. Précisons bien que A est une artère unique, par exemple un canal d'une fibre optique ou la fibre optique elle-même si celle-ci n'a qu'un canal. Appelons $C(A)$ l'ensemble des connexions qui sortent du carrefour par A . On suppose, et ceci est le point essentiel, que le nombre des éléments de $C(A)$, c'est à dire le nombre de connexions qui utilisent A , est très élevé, par exemple supérieur à 1000. Nous supposons que les cellules de chaque connexion arrivent une à une.

Puisqu'il y a plusieurs artères à l'entrée du commutateur, il peut y avoir plusieurs cellules qui arrivent durant un même temps-cellule et qui veulent sortir par la même artère de sortie, ce qui provoque un engorgement au niveau de la file d'attente considérée.

Il est à noter que la saturation de la file d'attente est due uniquement au hasard et non pas à un dépassement des débits alloués par rapport au débit de sortie.

Pour limiter les inconvénients d'une telle congestion, des méthodes de rejet sélectif de cellules ont été proposées pour soulager le réseau.

En effet si l'on détruit une cellule toutes les cellules appartenant au même message seront détruites à l'arrivée, donc autant les détruire directement dans le réseau. Du point de vue de la modélisation mathématique, ceci va provoquer une modification des taux d'arrivée des cellules prioritaires ou des cellules non prioritaires. De ce fait étudier le régime stationnaire seul ne suffit pas. Il est donc nécessaire d'introduire le régime transitoire.

En utilisant la technique basée sur une chaîne incluse liée à certains temps d'arrêt, on peut décomposer l'étude au voisinage de la congestion en deux phases. C'est une technique classique et efficace pour expliquer plus en détail le passage de l'état transitoire à l'état stationnaire. Ce mécanisme nous permet alors d'étudier la loi des retards des cellules non prioritaires.

Nous établissons des formules explicites et exploitables afin de pouvoir étudier les fluctuations des cellules des communications non prioritaires.

Chapitre 2

Un Paradoxe dans la Modélisation des Réseaux A.T.M.

2.0 Préliminaires

Dans la première partie, on a donné un bref aperçu des réseaux ATM et leurs problèmes de congestion. Dans la suite de ce travail, on se propose d'effectuer l'étude mathématique fine d'un cas particulier rencontré par les constructeurs. Le modèle adapté à cette situation est une file d'attente avec deux niveaux de priorités.

L'objectif de cette deuxième partie est double. D'une part, elle se propose de donner des outils pour étudier le régime transitoire d'un modèle qui prend en compte des priorités. D'autre part, elle met l'accent sur un paradoxe crucial dans la modélisation des réseaux ATM. Notons que cette partie reprend et complète les travaux effectués en (cf[SaP 97] et[SaP 99]).

Quelle que soit la technologie adoptée lors de la traversée des carrefours ATM, l'asynchronisme du mode ATM provoque des files d'attente. Leur étude est l'un des objets essentiels de [PBL 97]. Le chapitre 11 de cet ouvrage aborde l'utilisation de priorités. Le problème est le suivant. Du seul fait de l'existence de files d'attente, il semble difficile de dépasser des charges élevées pour les cellules associées à des communications ne tolérant que de faibles taux de perte ou de retard, ce qui est le cas, par exemple, des communications téléphoniques. Il est alors naturel de considérer deux classes de communications, la première classe étant prioritaire par rapport à la deuxième classe.

Il est facile de pressentir intuitivement que l'un des inconvénients majeurs de cette utilisation de priorités est le risque d'avoir des délais de transmission très longs pour les commu-

nications non prioritaires. Ce point est analysé au chapitre 11 précité mais l'étude qui y est effectuée est descriptive et peu détaillée sur le plan mathématique. Notre objectif, dans ce chapitre, est de définir et d'étudier un modèle mathématique adapté aux réseaux ATM dont les cellules peuvent avoir deux niveaux de priorité.

2.1 Introduction

Le modèle considéré correspond aux cellules en attente dans un carrefour (brasseur ou commutateur) ATM en se restreignant aux cellules qui doivent quitter ce carrefour par une sortie fixée S . L'unité de temps est le temps de passage d'une cellule ATM en un point de l'artère de sortie S . On se place dans le cas où il y a deux niveaux de priorité : X est le nombre de cellules prioritaires et Y est le nombre de cellules non prioritaires, toutes ces cellules étant en attente dans le carrefour étudié. On pose $Z = (X, Y)$ et l'objectif est d'étudier le régime transitoire du processus Z .

Pour chaque classe, les arrivées sont poissonniennes avec des taux respectifs α et β pour la première classe (classe prioritaire) et la deuxième classe (classe non prioritaire). Durant chaque unité de temps, c'est -à-dire chaque temps-cellule, une cellule est servie (si la file n'est pas vide). On sert d'abord les cellules de la première classe. A l'intérieur de chaque classe, la discipline est " premier arrivé, premier servi".

Par ailleurs, on suppose que les blocages dûs au carrefour sont négligeables, ce qui sera probablement de plus en plus vrai avec l'évolution de la technologie. Les blocages pris en compte sont donc uniquement ceux qui sont dûs à la congestion sur l'artère S .

Quand on se limite à un modèle aussi simple et sous réserve de choisir X et Y "pas trop grands", il est possible de mettre en mémoire, à chaque instant, toutes les probabilités d'état et donc d'avoir une description exhaustive de l'évolution transitoire du processus Z . Ceci est rappelé à la section 2.3 et est appelé pseudo-simulation pour alléger l'expression et parce que l'étude à effectuer est, d'un certain point de vue, très proche d'une simulation. Par ailleurs, le processus Z ou des processus analogues ont été précédemment étudiés. Par exemple, on connaît explicitement la loi stationnaire du retard d'une cellule non prioritaire (cf. la section 2.4). Appelons U ce retard.

On constate graphiquement que la loi de U est " presque géométrique " en ce sens que le rapport

$$P[U = k]/P[U = k + 1]$$

devient très vite constant. La figure 1 illustre ce fait. La courbe associée aux rapports est en trait plein et la courbe associée aux probabilités d'état est en pointillé. Evidemment, on n'a pas choisi le même vecteur unitaire pour les deux courbes. On a très souvent, pour ne pas dire presque toujours, cette situation dans les files d'attente modélisant des réseaux ATM (cf. [PBL 97]). Pour alléger l'expression, dans un tel cas, on dira que la loi considérée est P.géométrique.

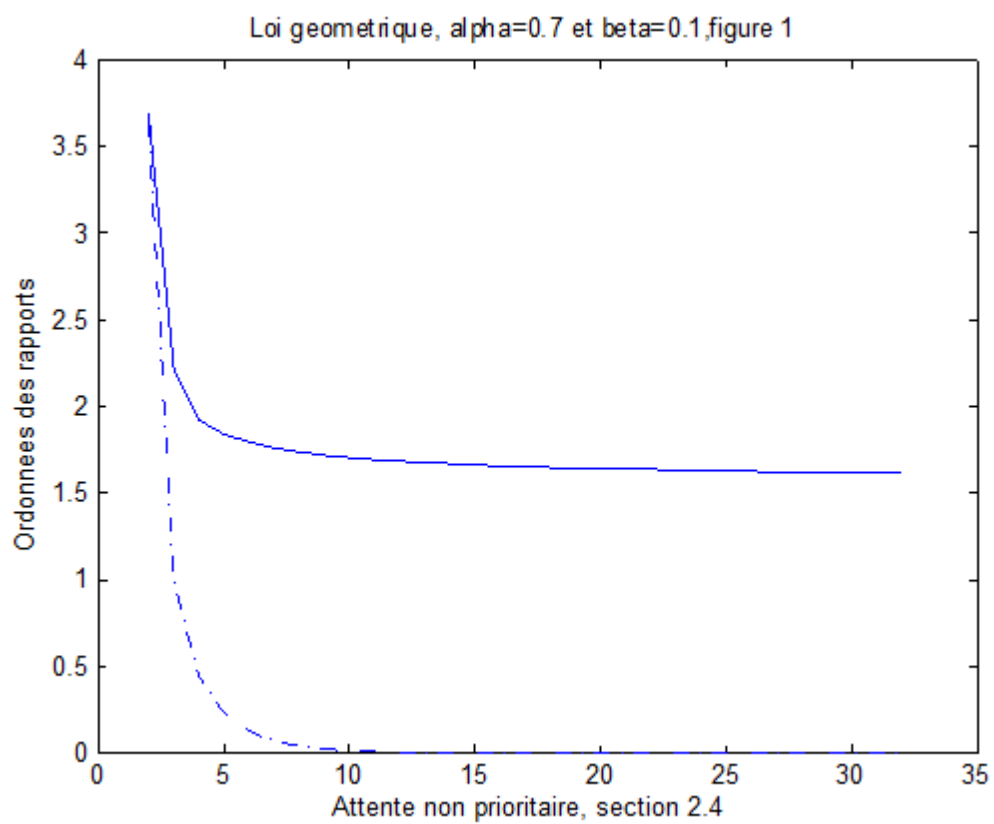
L'étude du régime stationnaire de Z ne suffit pas. Que va-t-il se passer si on détruit plusieurs cellules ou plusieurs rafales de cellules? Si l'on modifie certains taux d'accès en cas de risque de congestion? etc. L'outil de base pour une étude plus fine du processus Z est connu depuis un demi-siècle : c'est la chaîne incluse.

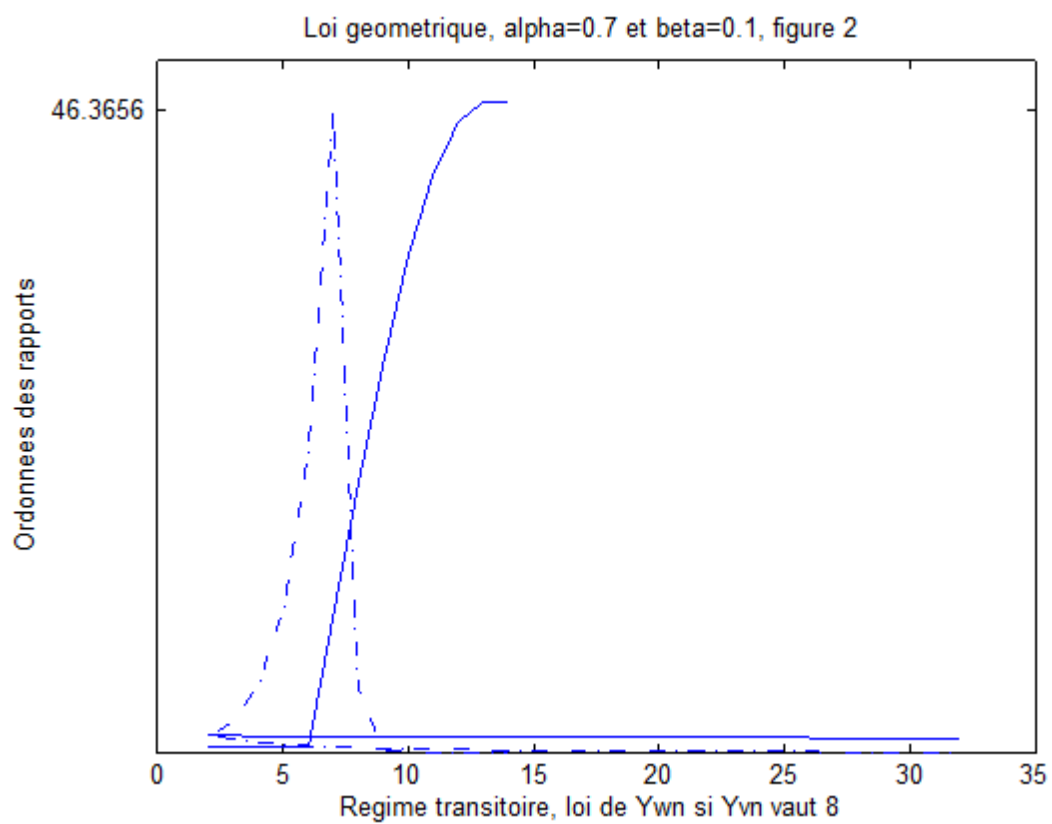
L'essentiel de cette partie est d'étudier la chaîne incluse définie à la section 2.5 car il nous semble que c'est la plus significative. Il est toutefois clair que les techniques proposées sont adaptables à d'autres exemples de chaînes incluses. Notamment, une autre chaîne incluse sera introduite et étudiée dans la troisième partie. La première étape, techniquement cruciale est difficile, est d'obtenir la fonction génératrice associée à cette chaîne incluse sous forme d'une fraction rationnelle. C'est la relation (2, 1) de la section 2.6. La plus part des résultats établis dans les sections suivantes reposent sur cette relation (2, 1).

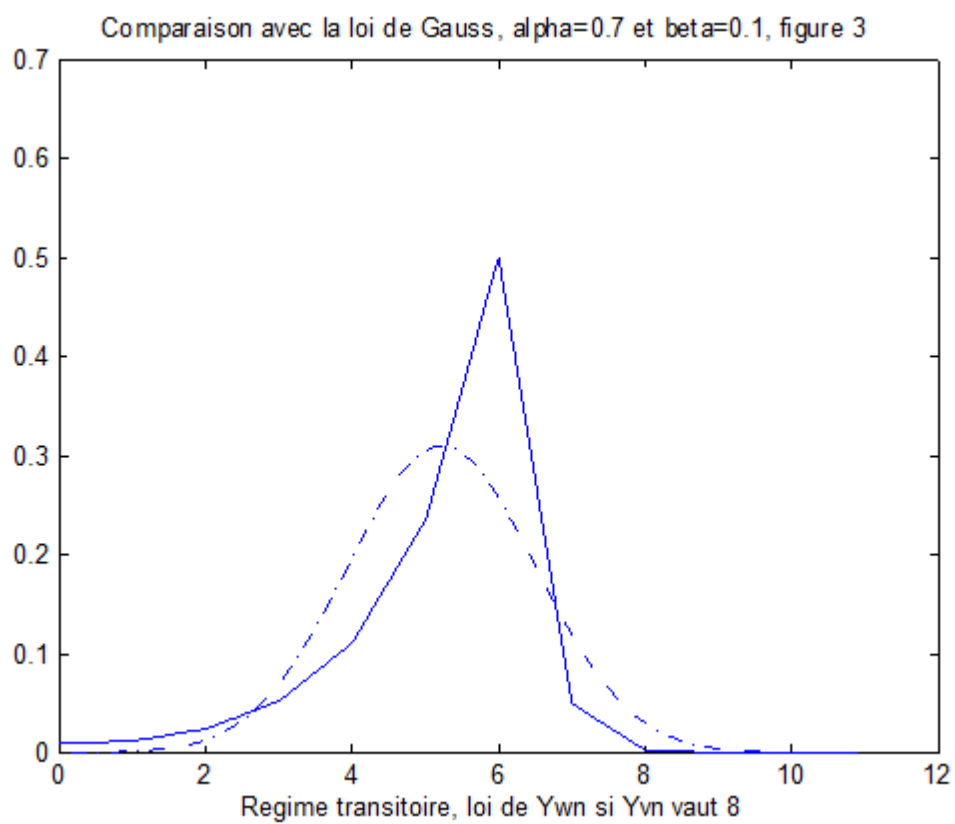
Pour illustrer l'évolution du processus Z , on a représenté graphiquement la loi de Y à l'instant $w(n)$ sachant que Y vaut 8 à l'instant $v(n)$. La définition des temps d'arrêt $v(n)$ et $w(n)$ est donnée la section 2.5.

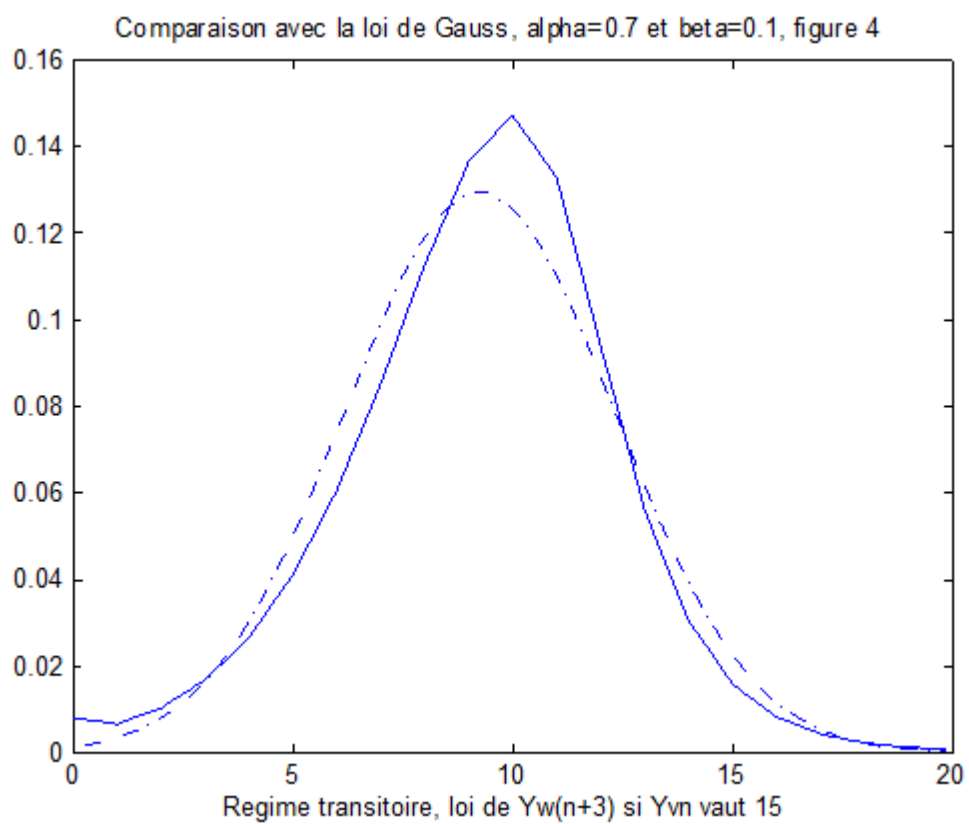
Et c'est là qu'est le paradoxe!

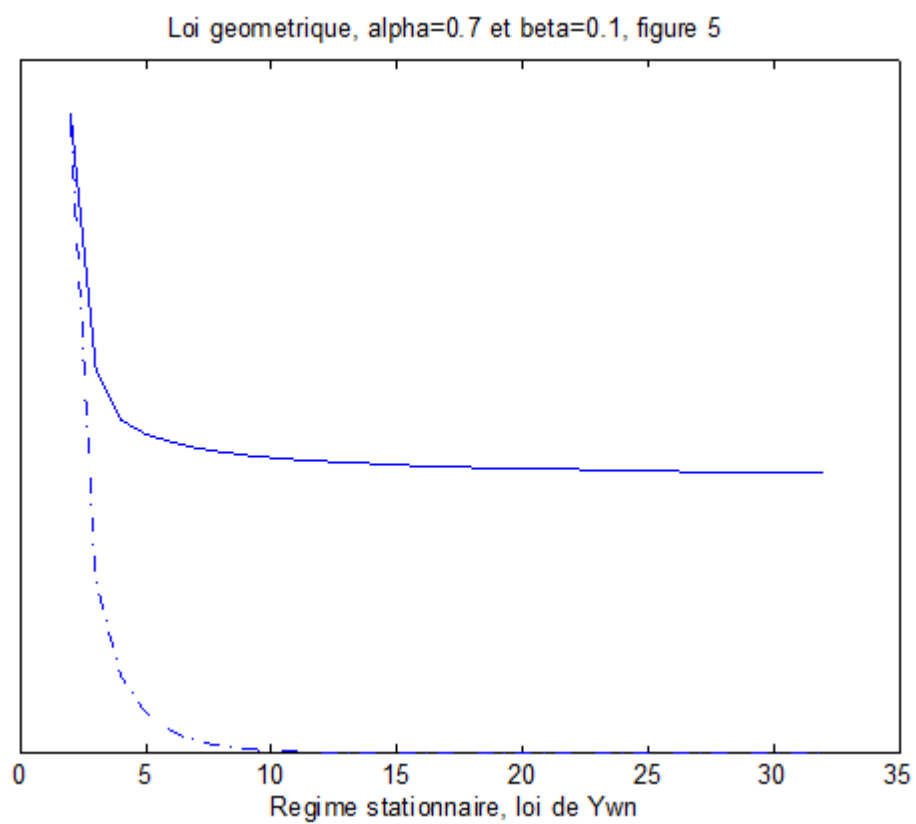
D'une part, cette loi est P-géométrique (figure2). D'autre part, cette loi est proche d'une loi de Gauss (figure 3). De plus, si la valeur initiale de Y en $v(n)$ est grande (par exemple supérieure à $(2k + 8)$ avec k entier positif), la variation de Y entre $v(n)$ et $w(n + k)$ peut être considérée comme la somme de plusieurs variables aléatoires (par exemple k ou $k + 1$) qui sont " presque indépendantes" puisque la probabilité de revenir dans l'état zéro (file vide) est négligeable. Cette variation est donc très voisine d'une loi de Gauss (figure 4).



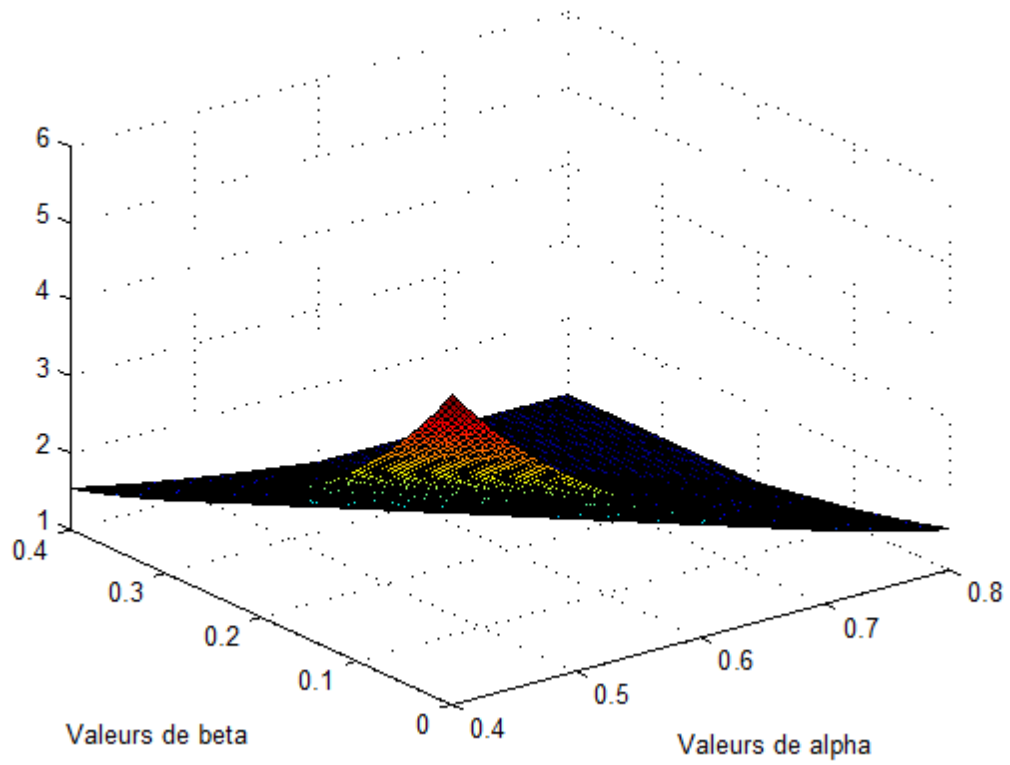








Valeurs de gamma en regime stationnaire, figure 6



Or, la loi de Gauss n'est pas P-géométrique !

Plus précisément, pour une variable aléatoire D qui suit une loi gaussienne, le rapport

$$\text{Proba}[t \leq D \leq t + dt] / \text{Proba}[t + dt \leq D < t + 2dt]$$

est " presque " une fonction linéaire de t si le pas de temps dt est assez fin et ce n'est pas du tout une fonction constante. Ceci est facile à vérifier, soit à l'aide d'un graphique, soit en dérivant la densité de la loi gaussienne.

Autrement dit, si on se limite aux petites valeurs de D , l'approximation gaussienne et donc, plus généralement, l'approximation par un processus de diffusion, peut s'avérer excellente. Cette approximation, judicieusement utilisée, peut donner des résultats significativement convaincants si on s'intéresse aux premiers moments.

Par contre, cette même approximation, si astucieuse soit-elle, est totalement erronée quand on considère des valeurs élevées des variables aléatoires. Or, l'essentiel, dans les réseaux ATM, est de déterminer, avec le maximum possible de précision, les seuils correspondant à des probabilités très faibles, par exemple de l'ordre de $1.E-9$. Les valeurs de ces seuils sont donc, en général, assez élevées.

La démarche globale que l'on propose est alors la suivante. Pour les "petites" valeurs des variables aléatoires, on procède par "pseudo-simulation" comme expliqué à la section 2.3. Si on choisit des contraintes de bords raisonnables, celles-ci influent très peu sur les probabilités des états associés à de " petites " valeurs des variables aléatoires ; quand les valeurs des variables aléatoires deviennent plus élevées, on procède par extrapolation en utilisant leur caractère **P-géométrique**.

Dans une telle démarche, à quoi peut servir une étude théorique mathématique ? Réponse : essentiellement en permettant de calculer quelques valeurs exactes.

D'une part, il est très souvent possible de calculer les premiers moments (cf. la fin des sections 2.6 et 2.8). Ceci peut aider à détecter des erreurs de programmation et à donner une première estimation de la fiabilité des calculs effectués. D' autre part, on peut parfois donner la valeur exacte de la limite des rapports des probabilités successives pour certaines des variables aléatoires considérées. Ces valeurs limites sont reportées sur les figures 2 et 5. Elles constituent un auxiliaire extrêmement précieux car leur calcul par pseudo-simulation est assez long et instable ce qui constitue un excellent test de la fiabilité des calculs effectués.

Par exemple, avec un pentium de base, le calcul d'une seule valeur de ngammap (cf. la section 2.9) par pseudo-simulation avec une erreur inférieure à 3 % demande plus d'une heure de temps calcul. Il a semblé intéressant de tracer la surface associée (figure 6) à cette même variable ngammap, en fonction de alpha et de bêta. Ce tracé, avec plus de cinq mille points distincts rigoureusement exacts, demande – avec le même pentium de base – dix minutes de temps calcul par résolution de la formule implicite (R1) (cf, la fin de la section 2.9).

2.2 Modèle Mathématique

Le processus Z qui fait l'objet de cette étude est un processus markovien homogène qui évolue à temps discret. Ce processus $Z = (X, Y)$ est un couple de processus X et Y à valeurs entières .Le processus X évolue indépendamment du processus Y . Au contraire, l'évolution de Y dépende fortement, de l'état de X .L'évolution de Z dépend de deux paramètres positifs α et β avec $\alpha + \beta < 1$. Plus précisément, pour chaque instant t on a :

$$X(t + 1) = A(t) + Sup.[0, X(t) - 1]$$

où $A(t)$ est une variable aléatoire indépendante de $Z(t)$ qui suit la loi de Poisson de paramètre α . et qui représente le nombre de cellules qui arrivent durant le temps cellu. Ce processus X est appelé ‘premier processus de base ‘ dans [PBL 97] et il y est étudié avec beaucoup de détails et sous plusieurs aspects. Voir aussi [PBL 97] . Voir aussi [Pra 65] pour les travaux et références antérieurs.

Si $X(t)$ est nul, on a :

$$Y(t + 1) = B(t) + Sup.\{0, Y(t) - 1\}.$$

Si $X(t)$ est strictement positif, on a :

$$Y(t + 1) = B(t) + Y(t).$$

Dans tous les cas, $B(t)$ est une variable aléatoire indépendante de $Z(t)$ qui suit la loi de Poisson de paramètre β et qui re présente le nombre de cellules qui arrivent durant le temps cellule . Ce qui précède détermine parfaitement et complètement l'évolution du processus Z .

Notons que, si on pose $Z' = X + Y$, le processus Z' est un processus markovien homogène dont l'évolution du processus est caractérisée par, quel que soit l'instant t :

$$Z'(t + 1) = C(t) + \sup\{0, Z'(t) - 1\}.$$

où $C(t)$ est une variable aléatoire indépendante de $Z'(t)$ qui suit la loi de Poisson de paramètre $(\alpha + \beta)$. Le processus Z' est donc du type : Premier processus de base $(\alpha + \beta)$ au sens de [PBL97] mais avec un taux $(\alpha + \beta)$.

premier processus de base :est un processus markovien a espace d'état discret et à temps discret.

On peut interpréter le processus Z de la façon suivante (cf .4 B1) page 71 de [PBL 97]. On considère un carrefour d'inforoutes, multiplexeur, brasseur ou commutateur. On suppose que le nombre d'artères entrantes est élevé, par exemple 60. Soit $C(S)$ l'ensemble des connexions qui sortent du carrefour par S . On suppose que $C(S)$ est très grand, par exemple supérieur à 1000, et que chaque connexion sortant par S a un débit faible par exemple inférieur au millième du débit de S . On suppose aussi que chacune de ces connexions a un débit bien régulé, par exemple par un contrôleur – espaceur, et émet cellule par cellule. On suppose que l'on peut négliger les blocages dus au carrefour lui-même .

Le processus Z correspond au nombre de cellules en attente dans le carrefour considéré et devant ressortir par S . L 'unité de temps est le "temps –cellule" c'est -à- dire le temps que met une cellule à parcourir un point de l'artère de sortie S . Notons que l'on peut généraliser cette interprétation de plusieurs façons (cf . 4 . B .4 , 4 . B . 5 , 4 . D .3 etc ... de [PBL 97].

La différence essentielle entre le processus Z et le " premier processus de base " introduit en 4.B.1 de [PBL 97] est que l'on suppose qu'il y a deux classes de connexions. Les processus X et Y correspondent aux cellules en attente de chacune de ces deux classes. On suppose que les cellules associées à X , c'est -à - dire à la première classe, passent systématiquement avant les cellules associées à Y , c'est -à - dire à la deuxième classe.

2.3 Pseudo-Simulation

Il est facile d'étudier le processus Z par simulation puisque son évolution est relativement simple. Toutefois, une étude par simulation ne permet pas de mettre aisément en évidence des évènements dont la probabilité est très faible. Or ,dans l'étude des réseaux ATM, on

souhaite pouvoir prendre en compte des événements dont la probabilité est de l'ordre de 10^{-9} . Il faut alors calculer le système complet des probabilités conjointes (au sens de Kolmogorov) et non pas simuler quelques " réalisations du hasard". Autrement dit, il faut calculer des probabilités et non pas des variables de comptage. Pour faciliter l'expression, cette façon d'aborder l'étude d'un processus markovien a été appelée P- simulation dans [PBL 94] et [PBL 97]. En effet, du point de vue d'un ingénieur, cette démarche reste très proche de la simulation.

Le point de départ de cette démarche est le suivant. On pose :

$$p[j, k] = P[X(t) = j \text{ et } Y(t) = k]$$

$$q[j, k] = P[X(t+1) = j \text{ et } Y(t+1) = k]$$

Procéder par P- simulation c'est tout simplement calculer le tableau q si on suppose que le tableau p est connu. On peut noter que l'application f qui associe à p la famille $q = f(p)$ est une application linéaire (qui dépend des paramètres α et β). Quelle que soit la condition initiale que l'on se donne à l'instant $t = 0$, en utilisant k fois l'application f , on obtient les probabilités du processus Z aux instants $t = 1, 2, \dots, k$. On dispose ainsi du système des probabilités conjointes. On peut donc, au moins en théorie, étudier les lois de probabilité de tous les événements possibles et imaginables qui ne dépendent que de l'évolution du processus Z .

Par exemple, dans la suite de cete étude, on introduit des temps d'arrêt et la chaîne incluse associée. Il est facile de mettre au point des procédures qui calculent, par P-simulation, les lois des délais entre ces temps d'arrêt ou les lois de l'état du processus Z à ces temps d'arrêt. La technique est tout à fait analogue à celle expliquée en 10.D.5 page 211 de [PBL 97].

On peut aussi, en appliquant k fois la fonction f à partir d'une condition initiale raisonnable, obtenir très rapidement le régime stationnaire de Z c'est - à -dire un tableau $p[.,.]$ tel que $f(p)$ soit très peu différent de p . Ce tableau est l'unique solution du système linéaire $p = f(p)$ telle que

$$\sum_{j \geq 0} \sum_{k \geq 0} p[j, k] = 1$$

Résoudre ce système linéaire par une méthode " aveugle " comme la méthode de Gauss est assez peu satisfaisant. La méthode itérative ci-dessus converge très rapidement. Elle est donc en fait plus intéressante compte tenu de ce qui a été dit auparavant et de sa facilité de mise en oeuvre . Cependant , si on veut obtenir la loi du régime stationnaire de Z avec des temps de calculs les plus faibles possibles, par exemple pour construire des abaques associées, la

méthode de très loin la plus efficace est la méthode des convexes. En effet, on est dans un cas bien particulier de [BeP 89], c'est-à-dire que l'on sait calculer la loi stationnaire de Z suivant un schéma presque explicite qui ne fait intervenir presque que des sommes et des produits de termes positifs.

2.4 Retard d'une Cellule Non Prioritaire

On se propose de montrer comment calculer le retard d'une cellule non prioritaire (donc associée à Y) choisie au hasard, retard pris lors de la traversée d'un carrefour. Lorsqu'une telle cellule arrive, le processus Z est en régime stationnaire. La loi du nombre total de cellules en attente est donc la loi stationnaire du processus Z' qui est du type "premier processus de base de taux $(\alpha + \beta)$ ". Appelons $\gamma(k)$ la probabilité stationnaire d'avoir Z' égal à k . On donne une expression explicite de $\gamma(k)$ en 8.A.3 page 151 de [PBL 97]. Toutefois, il est aussi rapide, sinon plus, de procéder par itérations en remarquant que la suite $\gamma(k)$, $k > 0$, est presque géométrique à partir d'un rang assez peu élevé (cf. la section 9. A de [PBL 97]). Attention, dans le calcul de la famille γ , le taux d'arrivée est $(\alpha + \beta)$ et non pas α .

Pour une nouvelle cellule N non prioritaire qui arrive alors qu'il y a k cellules en attente, ces k cellules passeront avant la cellule N . Le délai avant le passage de N sera donc le délai avant le premier trou d'un processus X du type "premier processus de base de taux α , ce processus X étant placé dans la condition initiale :

$$\text{Proba}[X(0) = i] = \gamma(i).$$

Compte tenu de la relation [5D3] page 103 de [PBL 97], (voir aussi [PBL 65]) la probabilité qu'une cellule non prioritaire, choisie au hasard, attende le délai $k > 0$ avant de passer vaut :

$$\sum_{i=1}^k \gamma(i) \frac{i}{k} e^{-\alpha k} \alpha^{k-i} \frac{k^{k-i}}{(k-i)!}$$

La probabilité de ne pas attendre vaut $\gamma(0)$. La loi de probabilité ainsi définie donne déjà de bonnes indications sur l'attente d'une cellule non prioritaire. Toutefois, si on veut mettre en place un système de régulation qui tient compte de l'état de Z , il est intéressant d'effectuer une étude du processus Z qui ne se limite pas à son régime stationnaire. On peut alors utiliser la technique de la chaîne incluse.

2.5 Chaîne Incluse

Cette technique de la ‘‘chaîne incluse’’ est très ancienne. En général étant donné un processus markovien, on peut lui associer plusieurs chaînes incluses intéressantes. Pour étudier le processus Z , on va introduire une chaîne incluse qui permet de séparer l’étude de Z pour X strictement positif de l’étude de Z pour X égal à zéro. Pour simplifier la compréhension des notations, on suppose que X est strictement positif a l’instant initial $t = 0$, on pose $w(0) = 0$. Puis, pour $n \geq 0$, on pose :

$$\begin{aligned} v(n+1) &:= \inf.\{t, t > w(n), X(t) = 0\} \\ w(n+1) &:= \inf.\{t, t > v(n+1), X(t) > 0\}. \end{aligned}$$

Les suites de variables aléatoires $v(n)$ et $w(n)$, $n > 0$, sont ainsi parfaitement déterminées. Ce sont des temps d’arrêts (cf , par exemple [MeP 80]). On dit aussi temps optionnels. L’évolution d’un processus markovien à partir d’un temps d’arrêt ne dépend que de l’état de ce processus) à ce temps d’arrêt (propriété de Markov forte). L’intérêt des temps d’arrêt $v(n)$ et $w(n)$ est évidemment d’avoir, quel que soit $n > 0$:

$$\begin{aligned} X(t) &= 0 \quad \text{pour } v(n) \leq t < w(n), \\ X(t) &> 0 \quad \text{pour } w(n) \leq t < v(n+1). \end{aligned}$$

De ce fait , entre $v(n)$ et $w(n)$, Y évolue comme un ‘‘premier processus de base’’ ; au contraire, entre $w(n)$ et $v(n+1)$, Y ne peut qu’augmenter. Etudier la chaîne incluse associée aux temps d’arrêt $v(n)$ et $w(n)$, $n > 0$, c’est étudier les délais et les évolutions entre deux tels temps d’arrêt successifs. On peut étudier ces délais et ces évolutions par P- simulation ,ne serait-ce qu’à titre de vérification. On peut aussi obtenir des formules presque explicites pour les principales probabilités concernées : c’est ce que nous allons étudier maintenant .

Pour n et k strictement positifs ,on a :

$$P[w(n) \geq v(n) + k + 1 \text{ sachant } w(n) \geq v(n) + k] = e^{-\alpha}.$$

Le délai entre $v(n)$ et $w(n)$ suit donc une loi géométrique issue de 1 et de paramètre $\exp(-\alpha)$ c’est -à -dire que l’on a, pour $k \geq 1$:

$$P[w(n) - v(n) = k] = e^{-\alpha(k-1)}[1 - e^{-\alpha}].$$

L’évolution de Y entre $v(n)$ et $w(n)$ sera étudiée à la section suivante. Par contre , pour $n > 0$, entre $w(n)$ et $v(n+1)$, Y ne peut qu’augmenter. Cette augmentation suit une loi de Poisson de paramètre βk si $v(n+1) - w(n)$ est égal à k .

L'évolution de X a été abondamment étudiée en [PBL 97] (voir aussi [Pra 65]. Notamment, le délai $v(n+1) - w(n)$ est le délai " avant le premier trou " comme cela a été étudié en 7.A.6 de [PBL 97] sauf que la "condition initiale " n'est pas exactement la même puisque :

$$P[X_{w(n)-1} = 0 \text{ et } X_{w(n)} > 0] = 1$$

Ceci revient simplement à tout multiplier (ou diviser) par $[1 - e^{-\alpha}]$ (cf .7. E.3 page 143 de [PBL 97]. Les relations [7A2] et [7.A.5] de [PBL 97] donne, pour $j > 0$:

$$P[v(n+1) - w(n) = j] = \frac{1}{1 - e^{-\alpha}} \alpha^j \frac{(j+1)^{j-1}}{j!} \exp[-(j+1)\alpha]$$

La fonction génératrice de l'augmentation de Y entre $w(n)$ et $v(n+1)$ est donc égale à :

$$\begin{aligned} \sum_{k=0}^{\infty} z^k P[Y_{v(n+1)} - Y_{w(n)} = k] &= \sum_{j=1}^{\infty} \frac{1}{1 - e^{-\alpha}} \alpha^j \frac{(j+1)^{j-1}}{j!} e^{-(j+1)\alpha} e^{j\beta(z-1)} \\ &= \sum_{j=1}^{\infty} \frac{1}{e^{\alpha} - 1} \alpha^j \frac{(j+1)^{j-1}}{j!} e^{-j(\alpha+\beta)} e^{j\beta z}. \end{aligned}$$

L'identification des termes en z^k donne :

$$P[Y_{v(n+1)} - Y_{w(n)} = k] = \frac{1}{e^{\alpha} - 1} \sum_{j=1}^{\infty} \alpha^j \frac{(j+1)^{j-1}}{j!} e^{-j(\alpha+\beta)} \frac{(j\beta)^k}{k!}$$

On pose :

$$r(z) := \sum_{j=0}^{\infty} \frac{(j+1)^{j-1}}{j!} [\alpha e^{-\alpha} e^{\beta(z-1)}]^j.$$

La fonction génératrice de l'augmentation de Y entre $w(n)$ et $v(n+1)$ devient :

$$\sum_{k=0}^{\infty} z^k P[Y_{v(n+1)} - Y_{w(n)} = k] = \frac{r - 1}{e^{\alpha} - 1}.$$

Notons que : $r(1) = e^{\alpha}$.

Par ailleurs, la valeur en 1 de la dérivée de r se calcule immédiatement à partir des relations [7A7] et [7A5] de [PBL 97] :

$$r'(1) = \sum_{j=1}^{\infty} j\beta \frac{(j+1)^{j-1}}{j!} [\alpha e^{-\alpha}]^j = \frac{\alpha\beta e^{\alpha}}{(1 - \alpha)}.$$

On a donc aussi :

$$E[Y_{v(n+1)} - Y_{w(n)}] = \frac{\alpha\beta e^{\alpha}}{(e^{\alpha} - 1)(1 - \alpha)}.$$

2.6 l'évolution de Y Entre $v(n)$ et $w(n)$

On se propose maintenant d'étudier l'évolution de Y entre $v(n)$ et $w(n)$. On a déjà noté que le délai $w(n) - v(n)$ suit la loi géométrique de paramètre $\exp(-\alpha)$. De plus, sur cet intervalle "stochastique", Y évolue comme un "premier processus de base" de paramètre β . Tout ce qui suit dépend de l'état de Y à l'instant initial $v(n)$. On pose :

$$g_j(z) = \sum_{k=0}^{\infty} z^k P[Y_{w(n)} = k \text{ et } w(n) = v(n) + j].$$

On vérifie immédiatement que l'on a pour $j \geq 1$ (cf . , par exemple, la section 7A3 et page 127 de [PBL 97]) :

$$z g_{j+1}(z) = [g_j(z) + (z - 1)g_j(0)] c(z) \quad \text{où } c(z) = e^{-\alpha} e^{\beta(z-1)}.$$

Cette relation vaut même pour $j = 0$ avec la convention suivante :

$$g_0(z) = (e^\alpha - 1) \sum_{k=0}^{\infty} z^k P[Y_{v(n)} = k].$$

On pose aussi :

$$\widehat{g}_0(k) = (e^\alpha - 1) P[Y_{v(n)} = k].$$

En raisonnant par récurrence croissante sur j , ceci donne :

$$z^j g_j(z) = c^j g_0(z) + (z - 1) \sum_{i=0}^{j-1} z^i c^{j-i} g_i(0).$$

Dans cette relation, l'identification des coefficients de z à la puissance j donne :

$$g_j(0) = e^{-j(\alpha+\beta)} \left\{ \sum_{i=0}^j \frac{[j\beta]^i}{i!} \widehat{g}_0(j-i) \right\} + \sum_{i=0}^{j-1} e^{-(j-i)(\alpha+\beta)} \left\{ \frac{[(j-i)\beta]^{j-i-1}}{(j-i-1)!} - \frac{[(j-i)\beta]^{j-i}}{(j-i)!} \right\} g_i(0).$$

Si on pose :

$$f_j = e^{j(\alpha+\beta)} g_j(0),$$

la relation précédente devient :

$$\begin{cases} f_j = \sum_{i=0}^j \frac{[j\beta]^i}{i!} \widehat{g}_0(j-i) + \left(\frac{1}{\beta} - 1\right) \sum_{i=1}^j \frac{[i\beta]^i}{i!} f_{j-i}, \\ f_0 = g_0(0) = \widehat{g}_0(0). \end{cases}$$

Ceci permet de calculer très rapidement par récurrence la famille f_j , pour $j > 0$, et donc la famille $g_j(0)$. Il faut noter que ces calculs ne font intervenir que des sommes et des produits de termes positifs. Ces relations sont les généralisations naturelles de l'étude effectuée en 7.A. 3 page 127 de [PBL 97] .

On pose :

$$\widehat{s}(k) = \Pr oba[Y_{w(n)} = k]$$

et

$$s(z) = \sum_{k=0}^{\infty} z^k P[Y_{w(n)} = k] = \sum_{k=0}^{\infty} z^k \widehat{s}(k).$$

ce qui implique

$$s(z) = \sum_{j=1}^{\infty} g_j(z).$$

Soit D le domaine complexe défini par :

$$e^{-\alpha} < |z| < 1.$$

Pour z élément de D on a :

$$\left| \frac{c(z)}{z} \right| < 1.$$

Dans la couronne D , la série de terme général $(c/z)^j$ à la puissance j est donc convergente et on a :

$$\sum_{j=1}^{\infty} \left(\frac{c}{z} \right)^j = \frac{c(z)}{z - c(z)}.$$

Or on a vu que :

$$g_j(z) = \left(\frac{c}{z} \right)^j g_0(z) + (z - 1) \sum_{i=0}^{j-1} \left(\frac{c}{z} \right)^{j-i} g_i(0)$$

En additionnant ces relations, on a pour tout z élément de D :

$$s(z) = \frac{c(z)}{z - c(z)} \left\{ g_0(z) + (z - 1) \sum_{i=0}^{\infty} g_i(0) \right\}$$

ce qui peut aussi s'écrire :

$$[z - c(z)] s(z) = c(z) \{g_0(z) + (z - 1) [g_0(0) + s(0)]\}. \quad (2.1)$$

Pour $|z| < 1$, toutes les fonctions apparaissant dans la relation ci-dessus sont holomorphes. Puisqu'il y a égalité sur le domaine D , il y a égalité sur tout le domaine $|z| < 1$.

L'identification des coefficients de z^k dans la relation (*) ci-dessus donne, pour $k > 0$:

$$\widehat{s}(k) = e^{\alpha+\beta}\widehat{s}(k-1) - \sum_{j=0}^{k-1} \frac{\beta^{k-j}}{(k-j)!} [\widehat{s}(j) + \widehat{g}_0(j)] - \widehat{g}_0(k) - \left(\frac{k}{\beta} - 1\right) \frac{\beta^k}{(k)!} [s(0) + \widehat{g}_0(0)]. \quad (2.2)$$

Cette relation permet de calculer la suite $\widehat{s}(k)$, $k > 0$, par récurrence croissante sur k en fonction de la famille $\widehat{g}_0(k)$, $k > 0$. Ceci nécessite d'avoir calculé, auparavant, le réel $s(0)$ ce qui a été fait précédemment puisque l'on a :

$$\widehat{s}(0) = s(0) = \sum_{j=1}^{\infty} g_j(0) = \sum_{j=1}^{\infty} e^{-j(\alpha+\beta)} f_j$$

Rappelons que , pour $k \geq 0$:

$$\widehat{s}(k) = P[Y_{w(n)} = k]$$

Par ailleurs, si on dérive la relation (*) par rapport à z et si on choisit $z = 1$, on obtient :

$$E[Y_{w(n)}] = E[Y_{v(n)}] + P[Y_{v(n)} = 0] + \frac{\beta - 1 + e^{-\alpha} P[Y_{w(n)} = 0]}{1 - e^{-\alpha}}.$$

Si on a calculé $s(0)$, cette relation donne l'espérance de Y en $w(n)$. On peut itérer ce procédé et obtenir ainsi des relations entre $s(0)$ et les moments successifs de Y en $w(n)$. Ce calcul peut d'ailleurs être conduit formellement (c f le chapitre 8 de [PBL 97]).

Si on suppose que les évènements $[Y_{v(n)} = 0]$ et $[Y_{w(n)} = 0]$ ont des probabilités négligeables, la relation (*) devient :

$$(z - c(z)) s(z) = c(z) g_0(z)$$

En dérivant deux fois par rapport à z cette relation puis en choisissant z égal à 1 dans cette relation, on obtient :

$$Var(Y_{w(n)}) = \frac{1}{1 - e^{-\alpha}} \left\{ \beta^2 + \beta - 1 + \frac{1 - \beta^2 + 2\beta e^{-\alpha(\beta-1)}}{1 - e^{-\alpha}} \right\}$$

et

$$E(Y_{w(n)}) = \frac{\beta - 1}{1 - e^{-\alpha}} + E(Y_{v(n)}).$$

2.7 Méthodes des Convexes

Lorsqu'on exécute les calculs de la section 2.6 qui précède, on constate que, si la valeur obtenue pour $s(0)$ n'est pas parfaitement exacte, les valeurs $\hat{s}(k)$ deviennent négatives extrêmement rapidement. Ce phénomène est extrêmement fréquent dans les calculs de systèmes en équilibre et est dû à l'instabilité intrinsèque de la méthode utilisée. Or, plus cette instabilité est grande, plus la méthode des convexes est efficace (cf .[Sag 89], [POS 90] et [Pel 92a]). Nous allons donc expliquer et adapter cette méthode dans le cas particulier considéré ici.

On suppose que l'on ne connaît pas $s(0)$ (on ne va pas utiliser les quantités f_j). Si on part d'une valeur initiale $s(0)$ quelconque et si on calcule itérativement les valeurs $\hat{s}(k)$ à l'aide des relations (R) à la fin de la section 2.6, par linéarité ces valeurs sont de la forme :

$$\hat{s}(k) = \lambda(k) s(0) + \mu(k)$$

Si on choisit comme valeur initiale $s(0) = 0$, le programme associé aux relations (R) rend les valeurs suivantes :

$$t'(k) = \mu(k).$$

De même, si on choisit comme valeur initiale $s(0) = 1$, le programme rend les valeurs suivantes :

$$t''(k) = \lambda(k) + \mu(k).$$

Les valeurs $t'(k)$ et $t''(k)$ sont données par le programme. On en déduit :

$$\mu(k) = t'(k) \text{ et } \lambda(k) = t''(k) - t'(k).$$

Or, quel que soit k , $\hat{s}(k)$ est compris entre 0 et 1. Ceci implique :

$$0 \leq \lambda(k) s(0) + \mu(k) \leq 1.$$

Dans le cas $\lambda(k) > 0$, on a :

$$-\frac{\mu(k)}{\lambda(k)} \leq s(0) \leq \frac{1 - \mu(k)}{\lambda(k)}.$$

Dans le cas $\lambda(k) < 0$, on a :

$$-\frac{\mu(k)}{\lambda(k)} \geq s(0) \geq \frac{1 - \mu(k)}{\lambda(k)}.$$

On détermine ainsi une suite de convexes auxquels $s(0)$ appartient. Dans la situation considérée ici, le diamètre de ces convexes ne décroît pas très vite vers zéro. Il faut raffiner la méthode de la façon suivante. Pour tout entier k , on pose :

$$S(k) = \sum_{j=0}^k \hat{s}(j).$$

Par linéarité, on a :

$$S(k) = \lambda'(k)s(0) + \mu'(k).$$

D'autre part :

$$0 \leq S(k) \leq 1.$$

On peut donc reprendre exactement le raisonnement effectué ci-dessus en remplaçant $\lambda(k)$ et $\mu(k)$ par $\lambda'(k)$ et $\mu'(k)$. Ceci donne une autre suite de convexes. On constate, cette fois-ci, que le diamètre de l'intersection de tous ces convexes tend extrêmement rapidement vers zéro. En fait, la borne inférieure de cette intersection est associée à $\hat{s}(k)$ et la borne supérieure à $S(k)$, ce qui n'est pas surprenant. On constate aussi, par le calcul informatique, que $\lambda(k)$ est toujours positif mais on n'a pas cherché à prouver ce résultat anecdotique.

A la fin, on peut choisir $s(0)$ égal au milieu du convexe intersection. On peut aussi le choisir égal au bord du convexe atteint le plus rapidement. Une fois que l'on a ainsi choisi $s(0)$, on calcule $\hat{s}(k)$ à l'aide des relations de récurrence (R). On constate (et ceci a un caractère assez général dans l'utilisation de la méthode des convexes) que si le calcul des convexes a été poussé jusqu'à la valeur de récurrence n , alors les valeurs $\hat{s}(k)$ sont fiables pour k inférieur à $(n - 3)$ environ. On vérifie informatiquement ce point en comparant les résultats obtenus avec ceux que donnent la P-simulation. On note aussi que la méthode des convexes que l'on vient d'expliquer est infiniment plus rapide que la P-simulation.

2.8 Régime Stationnaire de la Chaîne Incluse

En ce qui concerne le processus Z , on a déjà noté que son évolution en régime stationnaire était un point crucial de l'étude. Ceci est un peu moins déterminant pour la chaîne incluse aux temps d'arrêt $v(n)$ et $w(n)$. Notamment s'il y a une rupture des conditions d'utilisation, par exemple si le dépassement de la capacité du buffer provoque des procédures diverses de sauvegarde, il ne suffit pas d'étudier le régime stationnaire mais il faut aussi analyser les

comportements transitoires. La chaîne incluse introduite précédemment est alors l'auxiliaire technique idéal. Ceci dit, il est tout de même intéressant de connaître le régime stationnaire de la chaîne incluse. Dans ce cas la loi de Y en $v(n)$ est la même que celle en $v(n+1)$ et la loi de Y en $w(n)$ est la même que celle en $w(n+1)$.

Dans cette section, on suppose désormais qu'il en est ainsi.

On a vu, à la fin de la section 2.5, que la fonction génératrice de l'augmentation de Y entre $w(n)$ et $v(n+1)$ était égale

$$\frac{r(z) - 1}{e^\alpha - 1}.$$

Or cette augmentation est indépendante de la valeur de Y en $w(n)$. La fonction génératrice de la valeur de Y en $v(n+1)$ est donc égale à

$$s(z) \frac{r(z) - 1}{e^\alpha - 1}.$$

Puisqu'on est en régime stationnaire, cette fonction est aussi la fonction génératrice de la valeur de Y en $v(n)$ ce qui donne

$$g_0(z) = s(z) [r(z) - 1].$$

Ce qui implique

$$g_0(0) = s(0) [r(0) - 1].$$

En reportant ces deux relations dans la relation (2.1) on obtient :

$$(z - c(z)) s(z) = (z - 1)c(z)s(0)r(0).$$

Ce qui équivaut à

$$(z - cr(z)) s(z) = (z - 1)c(z)s(0)r(0).$$

Pour $z = 0$ et $z = 1$ cette relation est une identité. Dérivons cette relation et choisissons $z = 1$; on obtient :

$$[1 - c'(1)r(1) - c(1)r'(1)]s(1) + [1 - c(1)r(1)]s'(1) = c(1)s(0)r(0).$$

Or on connaît c d'une part et, d'autre part, $r(1)$ et $r'(1)$ (cf. la fin de la section 2.5).

On en déduit :

$$s(0)r(0) = e^\alpha \left[1 - \frac{\beta}{1 - \alpha} \right].$$

La relation donnée plus haut s'écrit donc finalement :

$$(z - cr(z)) s(z) = (z - 1)c(z)e^\alpha \left[1 - \frac{\beta}{1 - \alpha}\right]. \quad (2.3)$$

Puisque les fonctions c et r sont connues explicitement, cette relation (2.3) donne complètement la fonction s et donc la loi stationnaire de Y en $w(n)$. On a donc aussi la loi stationnaire de Y en $v(n)$ puisque

$$g_0(z) = s(z) [r(z) - 1].$$

En dérivant deux fois par rapport à z la relation (2.3) puis en choisissant z égal à 1 dans cette relation, on obtient :

$$E[Y_{w(n)}] = \frac{\beta}{2(1 - \alpha - \beta)} \left[2(1 - \alpha) - \beta + \alpha\beta \left[\frac{1}{1 - \alpha} + \frac{1}{(1 - \alpha)^2} \right] \right].$$

Rappelons que l'espérance de l'accroissement de Y entre $w(n)$ et $v(n+1)$ a été donnée explicitement à la fin de la section 2.5.

2.9 Loi P-Géométrique

Le calcul informatique de la loi de Y en $w(n)$ montre que, en général, cette loi est "presque géométrique à partir d'un certain rang". Pour alléger l'expression, on dit alors que Y suit une loi P-géométrique (pseudo-géométrique). Si on voulait effectuer une étude théorique de cette propriété, il faudrait reprendre et adapter la démarche sophistiquée proposée à la section 9. A de [PBL 97]. Nous allons seulement expliquer heuristiquement cette démarche dans le cas où la valeur de Y en $v(n)$ est bornée. Posons :

$$\gamma = \lim_{k \rightarrow +\infty} \left\{ \frac{P[Y_{w(n)} = k + 1]}{P[Y_{w(n)} = k]} \right\}$$

Dire que Y en $w(n)$ suit une loi P-géométrique signifie que cette limite existe et est atteinte extrêmement rapidement. La formule (2, 1) de la section 2.6 qui précède permet de déterminer γ . Le raisonnement heuristique est le suivant :

Pour k grand, dans la relation (2, 1) le coefficient de z^k est peu différent de zéro dans le membre de droite (les probabilités associées à la loi de Poisson de paramètre β décroissent

très vite vers zéro) et, dans le membre de gauche, du terme associé de $[z - c(z)] s(z)$ ce qui implique

$$\hat{s}(k-1) \text{ peut être différent de } \sum_{j=0}^k \hat{s}(j) \hat{c}(k-j).$$

Or, pour k "grand", $\hat{s}(k)$ est peu différent de $A\gamma^k$ où A est une constante. On en déduit, à la limite

$$\frac{1}{\gamma} = c\left(\frac{1}{\gamma}\right) = \exp\left\{-\alpha + \beta\left[\frac{1}{\gamma} - 1\right]\right\}.$$

Il est facile de vérifier que l'équation ci-dessus a une seule racine ($1/\gamma$) strictement supérieure à 1. Le calcul informatique confirme le raisonnement qui précède. Par exemple, pour $\alpha = 0.8$ et $\beta = 0.1$ on a

$1/\gamma = 47.65$ et, pour $\alpha = 0.7$ et $\beta = 0.1$, on a $1/\gamma = 46.3656$ et, pour $Y_{v(n)} \leq 3$ et $k \geq 10$,

$$\left| \frac{1}{\gamma} - \frac{P[Y_{w(n)} = k]}{P[Y_{w(n)} = k + 1]} \right| < 10^{-2}$$

Redisons que le raisonnement qui précède ne vaut que si la loi de Y en $v(n)$ correspond à une variable aléatoire bornée. On note que dans ce cas, le rapport γ ne dépend pas de cette loi de Y en $v(n)$. Par contre, la constante A dépend de cette loi. Quand Y en $v(n)$ est "grand", entre $v(n)$ et $w(n)$ le processus Y connaît une décroissance de taux moyen $1 - \beta$ par unité de temps. Ceci et la section 2.5 donnent une idée très précise de l'évolution de Y entre les temps d'arrêt $v(n)$ et $w(n)$.

Quand la loi initiale est quelconque, il faut adapter le raisonnement donné précédemment. Considérons le cas où on est en régime stationnaire. On a vu, à la section 2.8 qui précède, que la fonction génératrice de la loi de Y en $w(n)$ (en régime stationnaire) satisfait à la relation (" $< \mathcal{L} * *$ "). En raisonnant exactement comme ci-dessus, on constate que l'on a :

$$\frac{1}{\gamma} = (cr)\left(\frac{1}{\gamma}\right) \tag{2.4}$$

Pour les mêmes exemples que ci-dessus, soit $\alpha = 0.8$ et $\beta = 0.1$, on obtient :

$$1/\gamma = 1.23, \text{ pour } \alpha = 0.7 \text{ et } \beta = 0.1, \text{ on a } 1/\gamma = 1.5386.$$

Là encore, le calcul informatique confirme ce résultat. On note aussi que la "décroissance géométrique" est beaucoup plus lente que dans le cas non stationnaire. On note de plus que

le rapport de deux probabilités successives tend vers γ beaucoup plus lentement que dans le cas où la loi de Y en $v(n)$ est une mesure de Dirac. Par ailleurs, soit z une valeur réelle supérieure à 1 telle que

$$z = \exp[(\alpha + \beta)(z - 1)] \quad (2.5)$$

Une telle valeur existe, est unique et est facile à calculer numériquement. On a alors :

$$\exp[-\alpha + \beta(z - 1)] = z \exp(-\alpha z) \quad (2.6)$$

Par ailleurs, rappelons que (pour un tel z) :

$$c(z) = \exp[-\alpha + \beta(z - 1)]$$

et que

$$\begin{aligned} r(z) &:= \sum_{j=0}^{\infty} \frac{(j+1)^{j-1}}{j!} [\alpha e^{-\alpha} e^{\beta(z-1)}]^j \\ &= \sum_{j=0}^{\infty} \frac{(j+1)^{j-1}}{j!} [\alpha z e^{-\alpha z}]^j. \end{aligned}$$

soit, compte tenu de la relation [7A4] de [PBL97] :

$$r(z) = e^{\alpha z}$$

ces valeurs reportées dans (2.6) donnent

$$c(z) = \frac{z}{r(z)}.$$

donc $\gamma = 1/z$ est une solution de (2.4). En fait c'est la seule solution réelle de (R') comprise entre 0 et 1. On peut le vérifier en procédant par équivalences ou en montrant que (2.4) n'a qu'une seule telle solution. Il est numériquement plus facile de calculer la racine de (2.5) que celle de (2.4).

Chapitre 3

Autre Chaîne Incluse

3.0 Introduction

La chaîne incluse introduite à la section 2.5 du chapitre 2 n'est pas la seule qui soit intéressante à étudier. Nous allons, dans ce chapitre, en introduire une autre peut être moins importante du point de vue expérimental mais probablement plus intéressante du point de vue strictement mathématique car elle correspond à des points de régénération.

Remarque :

Une grande partie des notations sont différentes de celles du chapitre 2.

3.1 Modèle Mathématique

Plus précisément, les processus X, Y et $Z=(X, Y)$ sont définis exactement comme précédemment, les paramètres α et β ont la même signification.

On pose encore :

$$c(z) = e^{-\alpha} \cdot e^{\beta(z-1)}.$$

et on définit $r(z)$ comme à la section 2.5 du chapitre 2 .

3.2 Chaîne Incluse

Les temps d'arrêt $v(n)$ et $w(n)$ et tout ce qu'y rattache n sont plus définis de la même façon que précédemment . Plus précisément, on pose :

$$\begin{aligned} u'' &= \inf\{t : t > 0 \text{ et } (X + Y)(t) = 0\} \\ u' &= \inf\{t : t \geq u'' \text{ et } (X + Y)(t) > 0\} \\ u &= \inf\{t : t > u' \text{ et } (X + Y)(t) = 0\} \end{aligned}$$

Les temps d'arrêt u'' et u sont intéressants parce qu'ils sont associés à des "points de régénération". L'étude de Z entre u'' et u est exhaustive relativement à l'étude complète de Z . L'étude de Z entre u'' et u' est facile (délai géométrique avec un "taux d'arrêt" $1 - e^{-(\alpha+\beta)}$). L'essentiel ce qui suit porte donc sur l'étude entre u' et u .

3.3 Etude Entre u' et u

On pose :

$$\begin{aligned} w(0) &= u' \text{ puis, pour } n > 0 : \\ v(n) &= \inf\{t : t \geq w(n-1) \text{ et } (X(t) = 0 \text{ ou } t = u)\} \\ w(n) &= \inf\{t : t \geq v(n) \text{ et } (X(t) > 0 \text{ ou } t = u)\} \\ g_{n,0}(z) &= \sum_{k=0}^{\infty} z^k P\{Y_{v(n)} = k\}, \end{aligned}$$

et pour $j \geq 1$:

$$\begin{aligned} g_{n,j}(z) &= \sum_{k=0}^{\infty} z^k P\{Y_{v(n)+j} = k \text{ et } w(n) > v(n) + j, X_{v(n)+j} = 0\} \\ s_n(z) &= \sum_{j=1}^{\infty} g_{n,j}(z) \end{aligned}$$

En raisonnant comme à la section 2.6 du chapitre 2 on vérifie facilement que l'on a pour $n > 0$:

$$g_{n,j+1}(z) = [g_{n,j}(z) - g_{n,j}(0)] \frac{c(z)}{z}.$$

d'où

$$(z - c(z))s_n(z) = c(z)[g_{n,0}(z) - g_{n,0}(0) - s_n(0)]. \quad (3.1)$$

On a aussi

$$\sum_{k \geq 0} z^k P\{Y_{w(n)} = k \text{ et } X_{w(n)} > 0\} = (e^\alpha - 1) s_n(z).$$

Par ailleurs, la fonction génératrice de l'augmentation de Y entre $w(n)$ et $v(n+1)$ est la même qu'à la section 2.5 du chapitre 2, rappelons qu'elle est égale à $(r(z) - 1)/(e^\alpha - 1)$. Les trois relations que l'on vient d'indiquer impliquent, pour $n > 0$:

$$g_{n+1,0}(z) = \frac{r(z) - 1}{e^\alpha - 1} (e^\alpha - 1) \frac{c(z)}{z - c(z)} [g_{n,0}(z) - g_{n,0}(0) - s_n(0)] + s_n(0) + g_{n,0}(0)$$

ou encore pour $n > 0$:

$$g_{n+1,0}(z) = \varphi(z) [g_{n,0}(z) - g_{n,0}(0) - s_n(0)] + s_n(0) + g_{n,0}(0) \quad (3.2)$$

avec $\varphi(z) = \frac{c(z)(r(z)-1)}{z-c(z)}$.

Il reste à déterminer $g_{1,0}$. Si X est nul en u' on a $v(1) = u'$. Par contre, si X est strictement positif en u' on a $v(1) > u'$ et il faut tenir compte de l'augmentation entre u' et $v(1)$ comme à la section 2.5 du chapitre 2. Les deux éventualités que l'on vient d'évoquer ont les probabilités respectives

$$\frac{e^{-\alpha}}{1 - e^{-(\alpha+\beta)}} \quad \text{et} \quad \frac{1 - e^{-\alpha}}{1 - e^{-(\alpha+\beta)}}.$$

Puisque, en u' , $X + Y$ est strictement positive ceci implique

$$g_{1,0}(z) = \frac{e^{-\alpha}}{1 - e^{-(\alpha+\beta)}} \{e^{\beta(z-1)} - e^{-\beta} + b(z)(r(z) - 1)\}. \quad (3.3)$$

Les relations qui précèdent donnent une description complète explicite de l'évolution de X entre u' et u .

On pose

$$\sharp_n(z) = g_{n,0}(z) - g_{n,0}(0) = \sum_{k \geq 1} z^k \Pr oba[Y_{v(n)} = k].$$

La relation (3.2) donne

$$g_{n+1,0}(0) = \varphi(0)s_n(0) + s_n(0) + g_{n,0}(0)$$

$$\sharp_{n+1}(z) = g_{n+1,0}(z) - g_{n+1,0}(0)$$

soit en utilisant (3.2) :

$$\sharp_{n+1}(z) = \varphi(z) [g_{n,0}(z) - g_{n,0}(0)] - \varphi(z)s_n(0) + \varphi(0)s_n(0)$$

$$\sharp_{n+1}(z) = \varphi(z)\sharp_n(z) - [\varphi(z) - \varphi(0)]s_n(0). (**)$$

Remarque :

(2.3) ne permet pas de retrouver (3.2), contient moins d'information que(3.2).

En dérivant (**) et en faisant $z = 0$ on peut calculer $s_n(0)$ en fonction de $\#_{n+1}'(0)$ et $\#_n'(0)$.

On a aussi

$$s_n(0) = \frac{g_{n+1,0}(0) - g_{n,0}(0)}{1 - \varphi(0)}.$$

Ce qui est plus directement utilisable.

Evidemment on peut prouver(**) directement et cette preuve est plus simple que la preuve (***)).

On a

$$-s_n(0) = \frac{1}{\varphi'(0)} [\#_{n+1}'(0) - \varphi(0)\#_n'(0) - \varphi'(0)\#_n(0)]$$

Notons que (cf.[SaP99])

$$-\varphi(0) = -1 + r(0) = (e^\alpha - 1)P[Y_{v(n+1)} - Y_{w(n)} = 0]$$

$$\begin{aligned} \#_1(z) &= g_{1,0}(z) - g_{1,0}(0) \\ &= \frac{e^{-\alpha}}{1 - e^{-(\alpha+\beta)}} \{e^{\beta(z-1)} - e^{-\beta} + b(z)(r(z) - 1)\} - g_{1,0}(0). \end{aligned}$$

En dérivant la relation (3.4) et choisissant $z = 1$ on obtient

$$E[Y_{v(1)}] = \frac{\beta}{1 - e^{-(\alpha+\beta)}}.$$

En dérivant (3.2) terme à terme on obtient la relation de récurrence suivante :

$$g_{n+1,0}^{(k)}(z) - \varphi(z)g_{n,0}^{(k)}(z) = \varphi^{(k)}(z) (g_{n,0}(z) - g_{n,0}(0) - s_n(0)) + \sum_{j=1}^{k-1} C_k^j \varphi^{(j)}(z) g_{n,0}^{(k-j)}(z).$$

Pour $k = 1$ et en choisissant $z = 1$ on a :

Pour $n \geq 1$:

$$E[Y_{v(n+1)}] = E[Y_{v(n)}] + \varphi'(1) (1 - g_{n,0}(0) - s_n(0)).$$

Chapitre 4

Modèle de File d'Attente avec Rappels, Serveur Non Fiable et Deux Types de Clients

4.0 Introduction

Il existe un grand nombre de travaux ayant traité les systèmes d'attente avec rappels et une croissance rapide de littérature dans ce domaine, caractérisés par l'aspect suivant : lorsqu'un client arrive et trouve tous les serveurs occupés et aucune position d'attente disponible, il rejoint un pool virtuel de clients bloqués appelé orbite et répète sa tentative après un temps aléatoire jusqu'à ce qu'il parvienne à la zone de service. La capacité de l'orbite est infinie. Ces modèles ont une application relative à la performance du trafic des systèmes de communication téléphonique, des réseaux de télécommunications, des réseaux d'ordinateurs et des systèmes de communication. Les travaux détaillés des références en rapport avec les files d'attente avec rappels peuvent être trouvés dans la monographie de Falin and Templeton [FAT97], Kulkarni and Liang [KUL97], Choi et Chang [CHC99]. Choi et all [CPP93]. Voir également Artalejo [ART99a], [ART99b] pour la bibliographie.

Depuis les travaux de Aissani [AIS88] et Kulkarni et Choi [KUC90] une attention considérable a été accordée à l'analyse des systèmes avec rappels sujets à des pannes et réparations. Les interruptions de service mènent à la croissance des temps d'attente pour les clients abonnés au système, et il en résulte l'impatience des clients. Pour la littérature en rapport,

voir Aissani [AIS93], Aissani et Artalejo [AIA98], Artalejo [ART94], Yang et Li [YAL94], Wang et al [WCL01], Krishna Kumar et al [KPV02].

Les tentatives de rappels du client en tête de file d'attente commencent seulement lorsque le serveur se vide tel que défini par Gomez-Corral [COR99].

D'autre part, Boxma and Cohen ont étudié une file d'attente M/G/1 avec rappels où il y a un nombre fixe de clients permanents présents qui rejoignent la file à l'achèvement de leur service. Ce système avec clients permanents dans le contexte d'essais répétés a été analysé par Farahmand du point de vue d'essais répétés classiques et constants. P.Moreno a généralisé le deuxième cas en considérant une distribution de service différente pour les deux types de clients et loi de rappel générale pour les clients transitoires. Notre objectif principal est de considérer ce modèle où le client s'impatiente à un instant particulier et où le serveur est sujet à des pannes actives de loi de Poisson avec réparation. Une fois la réparation achevée, le serveur reprend le service immédiatement. Nous supposons que la discipline en orbite est F.C.F.S.

4.1 Notation et Modèle Mathématique

Nous considérons une seule file d'attente avec rappels, pannes et réparations. Les hypothèses suivantes décrivent le modèle mathématique.

1-La capacité de l'orbite est infinie .

2-La seule file d'attente avec rappels a deux classes de clients :

- Transitoires (appelés également clients ordinaires, transit).
- Un nombre fixe K ($K \geq 1$) de récurrents (appelés également clients permanents).

3-Processus des clients récurrents

*Il y a un nombre fixe K de clients récurrents dans le système. Après avoir reçu le service, les clients récurrents retournent immédiatement à l'orbite conformément à une discipline FCFS.

*Nous supposons également que seul le client récurrent en tête de file de l'orbite est autorisé à accéder au service. Les temps successifs entre les tentatives de rappels de tout client récurrent sont de loi exponentielle de moyenne fixe γ^{-1} (rappel constant).

4-Processus des arrivées des clients transitaires

- Les clients transitaires arrivent suivant un processus de Poisson de taux λ ; si le client transitaire trouve le serveur libre à son arrivée, il occupe le serveur et quitte le système après achèvement du service, autrement il décidera soit de quitter le système sans être servi avec une probabilité $1 - p$ ou d'entrer dans le groupe d'essais répétés (appelé orbite) avec une probabilité p suivant la discipline F.C.F.S. Nous supposons que seul le client transitaire en tête de l'orbite est autorisé à accéder au service.

- Les temps successifs d'inter rappels de tous les clients transitaires sont régis par une fonction de répartition $A(x)$, de densité correspondante $a(x)$ et de transformée de Laplace $L_A(x)$.

5 -Processus de service

- Le serveur est sujet à des pannes actives, c'est-à-dire que le serveur tombe en panne si et seulement si il est en service. Il tombe en panne après un temps aléatoire de loi exponentielle de moyenne $\frac{1}{\mu}$. lorsque le serveur tombe en panne, il est immédiatement réparé et le temps requis pour sa réparation est une variable aléatoire de distribution $C(x)$, de densité $c(x)$, de transformée de Laplace Stieltjes transform $L_C(s)$ et possédant les deux premiers moments γ_1, γ_2 .

- Le client dont le service est interrompu reste au service. Une fois la réparation achevée, le serveur reprend le service du client. Le serveur n'est pas autorisé à accepter de nouveaux clients jusqu'à ce que le client en service quitte le système. Le serveur est dit bloqué s'il est occupé ou en réparation.

- La distribution du service pour les deux types de clients est la même, et les durées de service successifs sont indépendantes de fonction de répartition commune $B(x)$, de densité $b(x)$, de transformée de Laplace $L_B(s)$ et possédant les deux premiers moments β_1, β_2 .

- Après chaque service, le serveur devient libre; le client suivant à servir est déterminé par une concurrence entre la loi générale du rappel pour les clients transitaires et deux lois exponentielles de taux γ et λ (le client transite en tête d'orbite -le cas échéant- le client récurrent en tête d'orbite et une nouvelle arrivée éventuelle se concurrencent pour le service).

6- Les durées des inter rappels, les durées de service pour les deux types de clients, les durées de réparation et les inter- arrivées sont supposées être mutuellement indépendantes. Le temps écoulé jusqu'à panne est indépendant des autres temps sauf le temps de service.

7-Les taux conditionnels de complétion (d'achèvement) (conditional completion rates) pour les rappels ,pour le service, et pour la réparation sont respectivement :

$\alpha(x)$, $\beta(x)$ et $\gamma(x)$ i.e

$$\alpha(x) = \frac{a(x)}{1 - A(x)}, \quad \beta(x) = \frac{b(x)}{1 - B(x)}, \quad \gamma(x) = \frac{c(x)}{1 - C(x)}.$$

L'état du système à l'instant t peut être décrit par le processus de Markov

$$\{X(t), t \geq 0\} = \{(J(t), J^*(t), Q(t), \xi_0(t), \xi_1(t)); t \geq 0\}.$$

- $J(t)$ représente l'état du serveur (0, 1, 2 ou 3) suivant que le serveur est libre, occupé par un client transitaire, occupé par un client récurrent ou en réparation respectivement.
- $J^*(t)$ représente le type de client en service après la panne (1 : si le client en service est transitaire après la panne, 2 : si le client en service est récurrent après la panne).
- $Q(t)$ correspond au nombre de clients en orbite à l'instant t .
- si $J(t) = 0$ et $Q(t) > K$, alors $\xi_0(t)$ représente le temps de rappel écoulé (the elapsed retrial time)du client transitaire.
- If $J(t) = 1$ ou $J(t) = 2$ alors $\xi_1(t)$ est le temps de service écoulé (the elapsed service time).
- If $J^*(t) = 1$ ou $J^*(t) = 2$ nous définissons $\xi_2(t)$ le temps de réparation écoulé (the elapsed repair time).

4.2 Condition de stabilité

Dans cette section, nous étudions la condition nécessaire et suffisante pour la stabilité du système. Avant d'étudier cela, nous devons définir la durée de service généralisé S_n^* du $n^{\text{ème}}$ client c'est-à-dire le laps de temps depuis que le client $n^{\text{ème}}$ a commencé à être servi jusqu'à achèvement du service, où S_n^* comprend la durée de service et un éventuel temps de réparation. D'après les hypothèses de la section 1, S_n^* is independent de n , $n \geq 1$. Nous posons alors S^* la durée de service généralisé pour tout client en service et $B^*(x)$ sa distribution correspondante.

Lemme 1 *La fonction de répartition, la transformée de Laplace et la moyenne de S^* sont :*

$$B^*(x) = \sum_{n \geq 0} \int_0^x e^{-\mu y} \frac{(\mu y)^n}{n!} C_n(x-y) dB(y) \quad (4.1)$$

$$L_{B^*}(s) = L_B(s + \mu - \mu L_C(s)) \quad (4.2)$$

$$E(S^*) = \beta_1 \{1 + \mu\gamma_1\} \quad (4.3)$$

respectivement où $C_n(x)$ représente la convolée d'ordre n de $C(x)$.

Preuve. Soit $N(S)$, $N_0(S)$ le nombre de pannes durant l'intervalle depuis que le client en service commence à être servi jusqu'à achèvement du service, le nombre de pannes durant la même période où le client en service demeure en position de service après la panne, respectivement, où S est la durée de service.

$$N(S) = N_0(S), \quad S^* = S + \sum_{i=1}^{N(S)} \omega_i$$

où ω_i représente la durée de la i -ème réparation. Nous savons que $(N(S)/S)$, est un processus de Poisson de taux μ , alors :

$$\begin{aligned} B^*(x) &= P(S^* \leq x) = \sum_{n=0}^{\infty} P(S^* \leq x, N(S) = n) \\ &= \sum_{n=0}^{\infty} \int_0^{\infty} P(S^* \leq x, N(S) = n / S = y) dB(y) \\ &= \sum_{n=0}^{\infty} \int_0^x P(N(y) = n) P\left(\sum_{i=1}^n \omega_i \leq x - y\right) dB(y) \\ &= \sum_{n \geq 0} \int_0^x e^{-\mu y} \frac{(\mu y)^n}{n!} C_n(x - y) dB(y) \end{aligned}$$

□

Sa transformée de Laplace stieltjes est donnée (après integration par parties)

$$\begin{aligned} L_{B^*}(s) &= \int_0^{\infty} e^{-st} dB^*(t) = \int_0^{\infty} e^{-st} \sum_{i \geq 0} \frac{(\mu t)^i}{i!} e^{-\mu t} (L_C(s))^i dB(t) \\ &= \int_0^{\infty} e^{-st} e^{-\mu t} \sum_{i \geq 0} \frac{(\mu t)^i}{i!} (L_C(s))^i dB(t) \\ &= L_B(s + \mu - \mu L_C(s)) \end{aligned}$$

et sa moyenne est donnée par :

$$E(B^*) = -\frac{dL_{B^*}(0)}{ds} = \beta_1 \{1 + \mu\gamma_1\}.$$

Nous calculons la moyenne de W_1 et de W_2 correspondant aux temps moyens où W_1, W_2 sont la durée de service, et la durée totale de réparation durant laquelle le client en service demeure en position de service après panne, respectivement. Dans ce qui suit, nous donnons une méthode directe pour calculer ces moyennes .

On a : $W_2 = \sum_{i=1}^{N_0(W_1)} \omega_i$, en utilisant l'identité de Wald

$$E(W_2) = E(N_0(W_1)) E(\omega_1) = \mu\beta_1\gamma_1,$$

ce qui donne $E(B^*) = E(W_1) + E(W_2)$.

Soit t_n l'instant du n -ième départ et Q_n le nombre de clients en orbite juste après l'instant t_n , alors

$$J(t_n^+) = 0, \quad Q(t_n^+) = Q_n \quad \forall n \in N.$$

Nous obtenons l'équation récurrente fondamentale suivante :

$$Q_{n+1} = Q_n - \beta^{n+1} + \nu^{n+1}.$$

où ν^{n+1} représente le nombre de clients qui sont arrivés durant la durée de service généralisé du $(n+1)$ ième client et qui choisissent de rentrer en orbite.

$$\beta^{n+1} = \begin{cases} 1 & \text{si le client quitte l'orbite après le } n\text{-ième départ} \\ 0 & \text{sin on} \end{cases}.$$

Notons que le client quitte l'orbite si le nombre de clients en orbite est supérieur à $K+1$.

La séquence de variables aléatoires $Q_n, n \geq 1$ forme une chaîne de Markov incluse aux instants de départ pour notre file qui est irréductible et apériodique sur l'espace des états N . Dans le théorème suivant nous établissons l'ergodicité de la chaîne incluse aux instants de départ.

Théorème 2 Soit Q_n le nombre de clients en orbite à l'instant du n -ième départ, $n \geq 1$. Alors $\{Q_n, n \geq 1\}$ est ergodique si et seulement si

$$\lambda p \beta_1 \{1 + \mu \gamma_1\} < L_A (\lambda + \gamma)$$

Preuve. Pour prouver l'ergodicité, nous utilisons le critère de Foster :

Une chaîne de Markov $\{Y_n, n \in N\}$ avec espace d'état S irréductible et apériodique est ergodique s'il existe une fonction positive $f(s), s \in S$, et $\epsilon > 0$ tel que l'accroissement moyen

$\chi_s = E(f(Y_{n+1}) - f(Y_n) / Y_n = s)$ est fini pour tous $s \in S$ et $\chi_s \leq -\epsilon$ pour tous $s \in S$ sauf peut-être d'un nombre fini.

Dans notre cas, nous considérons la fonction $f(j) = j$ pour tous $j \in N$. Alors l'accroissement moyen est donné par :

$$\chi_j = \begin{cases} \lambda p \beta_1 \{1 + \mu \gamma_1\} & \text{si } j = K \\ \lambda p \beta_1 \{1 + \mu \gamma_1\} - L_A(\lambda + \gamma) & \text{si } j \geq K + 1 \end{cases}$$

D'où si $\lambda p \beta_1 \{1 + \mu \gamma_1\} < L_A(\lambda + \gamma)$, le critère de Foster est vérifié et donc la chaîne de Markov $\{Q_n, n \geq 1\}$ est ergodique.

La même inégalité est aussi nécessaire pour l'ergodicité. D'après les travaux de Sennott, Humblet et Tweedie [SHT83], nous pouvons garantir la non ergodicité si la chaîne de Markov $\{Q_n, n \geq 1\}$ satisfait la condition de Kaplan :

$\chi_j < +\infty$ pour tous $j \geq 0$, et il existe $j_0 \in N$ tel que $\chi_j \geq 0$ pour $j \geq j_0$.

dans notre cas , $\lambda p \beta_1 \{1 + \mu \gamma_1\} \geq L_A(\lambda + \gamma)$ implique la non ergodicité de la chaîne de Markov.

;Si $p = 1, \mu = 0, \gamma = 0$ et $\gamma_1 = 0$ nous retrouvons la condition de stabilité donnée par Gomez-Corral [GOM99]. \square

Si $p = 1, \mu = 0$ et $\gamma_1 = 0$ nous retrouvons la condition de stabilité trouvée par P.Moreno[MOR04].

4.3 Distribution stationnaire des états du système

Dans cette section nous étudions la distribution stationnaire des états du système.

Pour le processus $\{X(t), t \geq 0\}$, nous définissons :

$$\cdot P_{(0,K)}(t) = P(J(t) = 0, Q(t) = K)$$

$$\cdot P_{(0,i)}(t, w) dw = P(J(t) = 0, Q(t) = i, w < \xi_0(t) < w + dw), \text{ for } i \geq K + 1$$

$$\cdot P_{(1,i)}(t, x) dx = P(J(t) = 1, Q(t) = i, x < \xi_1(t) < x + dx), \text{ for } i \geq K$$

$$\cdot P_{(2,i)}(t, x) dx = P(J(t) = 2, Q(t) = i, x < \xi_1(t) < x + dx), \text{ for } i \geq K - 1$$

$$\cdot P_{(3,1,i)}(t, x, y) dx dy = P(J(t) = 3, J^*(t) = 1, Q(t) = i, x < \xi_1(t) < x + dx, y < \xi_2(t) < y + dy), \text{ for } i \geq K$$

$$\cdot P_{(3,2,i)}(t, x, y) dx dy = P(J(t) = 3, J^* = 2, Q(t) = i, x < \xi_1(t) < x + dx, y < \xi_2(t) < y + dy), \text{ for } i \geq K$$

En utilisant la méthode de la variable supplémentaire, nous obtenons les probabilités limites suivantes :

$$\begin{aligned} \lim_{t \rightarrow \infty} P_{(0,K)}(t) &= P_{(0,K)}, & \lim_{t \rightarrow \infty} P_{(1,i)}(t, x) &= P_{(1,i)}(x), & \lim_{t \rightarrow \infty} P_{(3,1,i)}(t, x, y) &= P_{(3,1,i)}(x, y), \\ \lim_{t \rightarrow \infty} P_{(0,i)}(t, w) &= P_{(0,i)}(w), & \lim_{t \rightarrow \infty} P_{(2,i)}(t, x) &= P_{(2,i)}(x), & \lim_{t \rightarrow \infty} P_{(3,2,i)}(t, x, y) &= P_{(3,2,i)}(x, y). \end{aligned}$$

Les équations d'état à l'équilibre s'écrivent :

$$(\lambda + \gamma) P_{(0,K)} = \int_0^\infty P_{1,K}(x) \beta(x) dx + \int_0^\infty P_{(2,K-1)}(x) \beta(x) dx \quad (4.4)$$

$$\left(\frac{\partial}{\partial w} + \lambda + \gamma + \alpha(w) \right) P_{(0,i)}(w) = 0, \quad i \geq K + 1, \quad (4.5)$$

$$\left(\frac{\partial}{\partial x} + p\lambda + \mu + \beta(x) \right) P_{(1,i)}(x) = \int_0^\infty P_{(3,1,i)}(x, y) \gamma(y) dy + \lambda p (1 - \delta_{i,K}) P_{(1,i-1)}(x), \quad i \geq K \quad (4.6)$$

$$\left(\frac{\partial}{\partial x} + p\lambda + \mu + \beta(x) \right) P_{(2,i)}(x) = \int_0^\infty P_{(3,2,i)}(x, y) \gamma(y) dy + \lambda p (1 - \delta_{i,K-1}) P_{(2,i-1)}(x) \quad i \geq K - 1 \quad (4.7)$$

$$\left(\frac{\partial}{\partial y} + p\lambda + \gamma(y) \right) P_{(3,1,i)}(x, y) = \lambda p (1 - \delta_{i,K}) P_{(3,1,i-1)}(x, y), \quad i \geq K \quad (4.8)$$

$$\left(\frac{\partial}{\partial y} + p\lambda + \gamma(y) \right) P_{(3,2,i)}(x, y) = \lambda p (1 - \delta_{i,K-1}) P_{(3,2,i-1)}(x, y), \quad i \geq K - 1 \quad (4.9)$$

$$P_{(0,i)}(0) = \int_0^\infty P_{(1,i)}(x) \beta(x) dx + \int_0^\infty P_{(2,i-1)}(x) \beta(x) dx \quad i \geq K, \quad (4.10)$$

$$P_{(1,i)}(0) = \int_0^\infty P_{(0,i+1)}(x) \alpha(x) dx + \lambda \delta_{i,K} P_{(0,i)} + \lambda (1 - \delta_{i,K}) \int_0^\infty P_{(0,i)}(x) dx, \quad i \geq K \quad (4.11)$$

$$P_{(2,i)}(0) = \gamma \delta_{i,K-1} P_{(0,K)} + \gamma (1 - \delta_{i,K-1}) \int_0^\infty P_{(0,i+1)}(x) dx, \quad i \geq K - 1 \quad (4.12)$$

$$P_{(3,1,i)}(x, 0) = \mu P_{(1,i)}(x) \quad i \geq K, \quad (4.13)$$

$$P_{(3,2,i)}(x, 0) = \mu P_{(2,i)}(x) \quad i \geq K - 1, \quad (4.14)$$

et l'équation de normalisation

$$\begin{aligned} & P_{(0,K)} + \sum_{i=K+1}^{\infty} \int_0^\infty P_{(0,i)}(w) dw + \sum_{i=K}^{\infty} \int_0^\infty P_{(1,i)}(x) dx \\ & + \sum_{i=K-1}^{\infty} \int_0^\infty P_{(2,i)}(w) dw + \sum_{i=K}^{\infty} \int_0^\infty \int_0^\infty P_{(3,1,i)}(x, y) dx dy \\ & + \sum_{i=K-1}^{\infty} \int_0^\infty \int_0^\infty P_{(3,2,i)}(x, y) dx dy = 1 \end{aligned} \quad (4.15)$$

Pour résoudre le système d'équations 44 – 415, nous définissons les fonctions génératrices suivantes :

$$P_0(z, w) = \sum_{i=K+1}^{\infty} P_{(0,i)}(w) z^i, P_1(z, x) = \sum_{i=K}^{\infty} P_{(1,i)}(x) z^i, P_2(z, x) = \sum_{i=K-1}^{\infty} P_{(2,i)}(x) z^i,$$

$$P_{(3,1)}(z, x, y) = \sum_{i=K}^{\infty} P_{(3,1,i)}(x, y) z^i, P_{(3,2)}(z, x, y) = \sum_{i=K-1}^{\infty} P_{(3,2,i)}(x, y) z^i$$

qui sont convergentes pour tout $x \geq 0$, $y \geq 0$ et dans le domaine $|z| \leq 1$.

En utilisant les fonctions génératrices ci-dessus, nous obtenons les équations différentielles suivantes :

$$\left(\frac{\partial}{\partial x} + \lambda + \gamma + \alpha(x) \right) P_0(z, x) = 0 \quad (4.16)$$

$$\left(\frac{\partial}{\partial x} + p\lambda + \mu + \beta(x) \right) P_1(z, x) = \int_0^{\infty} P_{(3,1)}(z, x, y) \gamma(y) dy + \lambda p z P_1(z, x) \quad (4.17)$$

$$\left(\frac{\partial}{\partial x} + p\lambda + \mu + \beta(x) \right) P_2(z, x) = \int_0^{\infty} P_{(3,2)}(z, x, y) \gamma(y) dy + \lambda p z P_2(z, x) \quad (4.18)$$

$$P_0(z, 0) = \int_0^{\infty} \beta(x) (P_1(z, x) + z P_2(z, x)) dx - z^K \int_0^{\infty} (P_{(1,K)}(x) + P_{(2,K-1)}(x)) \beta(x) dx \quad (4.19)$$

$$P_1(z, 0) = \lambda z^K P_{0,K} + \lambda \int_0^{\infty} P_0(z, x) dx + z^{-1} \int_0^{\infty} P_0(z, x) \alpha(x) dx \quad (4.20)$$

$$P_2(z, 0) = \gamma z^{K-1} P_{0,K} + \gamma z^{-1} \int_0^{\infty} P_0(z, x) dx \quad (4.21)$$

$$P_{(3,1)}(z, x, 0) = \mu P_{3,1}(z, x) \quad (4.22)$$

$$P_{(3,2)}(z, x, 0) = \mu P_{3,2}(z, x) \quad (4.23)$$

$$\left(\frac{\partial}{\partial y} + p\lambda + \gamma(y) \right) P_{(3,1)}(z, x, y) = \lambda p z P_{(3,1)}(z, x, y) \quad (4.24)$$

$$\left(\frac{\partial}{\partial y} + p\lambda + \gamma(y) \right) P_{(3,2)}(z, x, y) = \lambda p z P_{(3,2)}(z, x, y) \quad (4.25)$$

dont les solutions sont données par :

$$P_0(z, x) = P_0(z, 0) (1 - A(x)) \exp(-(\lambda + \gamma)x) \quad (4.26)$$

$$P_1(z, x) = P_1(z, 0) (1 - B(x)) \exp -xG(\lambda p(1 - z)) \quad (4.27)$$

avec $G(x) = x + \mu - \mu L_C(x)$

$$P_2(z, x) = P_2(z, 0) (1 - B(x)) \exp -xG(\lambda p(1 - z)) \quad (4.28)$$

$$P_0(z, 0) = \int_0^{\infty} \beta(x) (P_1(z, x) + z P_2(z, x)) dx - z^K \int_0^{\infty} (P_{(1,K)}(x) + P_{(2,K-1)}(x)) \beta(x) dx \quad (4.29)$$

$$P_1(z, 0) = \lambda z^K P_{(0,K)} + \gamma P_0(z, 0) \frac{(1 - L_A(\lambda + \gamma))}{(\lambda + \gamma)} \quad (4.30)$$

$$zP_2(z, 0) = \gamma z^K P_{0,K} + \gamma P_0(z, 0) \left\{ \frac{\lambda(1 - L_A(\lambda + \gamma))}{(\lambda + \gamma)} + z^{-1} L_A(\lambda + \gamma) \right\} \quad (4.31)$$

$$P_{(3,1)}(z, x, y) = \mu P_1(z, x) (1 - C(y)) \exp(-\lambda p(1 - z)y) \quad (4.32)$$

$$P_{(3,2)}(z, x, y) = \mu P_2(z, x) (1 - C(y)) \exp(-\lambda p(1 - z)y) \quad (4.33)$$

Le théorème suivant décrit la distribution des états du système en équilibre en fonction des fonctions génératrices.

Théorème 3 Si $\lambda p \beta_1 \{1 + \mu \gamma_1\} < L_A(\lambda + \gamma)$ et $K(z) = L_B(G(\lambda p(1 - z)))$ alors

$$P_{(0,K)} = \frac{L_A(\lambda + \gamma) - p\lambda\beta_1(1 + \mu\gamma_1)}{pL_A(\lambda + \gamma) - (1 - p)L_A(\lambda + \gamma)(1 + (\lambda + \gamma)\beta_1(1 + \mu\gamma_1))}, \quad (4.34)$$

$$P_0(z, w) = \frac{z^{K+1} [\lambda + \gamma] (1 - K(z)) \exp(-(\lambda + \gamma)w) (1 - A(w)) P_{(0,K)}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))}, \quad (4.35)$$

$$P_1(z, x) = \frac{z^K L_A(\lambda + \gamma) \{\lambda(1 - z) + \gamma(1 - K(z))\} \exp(-G(p\lambda(1 - z))x) (1 - B(x)) P_{(0,K)}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))}, \quad (4.36)$$

$$P_2(z, x) = \frac{\gamma z^{K-1} L_A(\lambda + \gamma) (K(z) - z) \exp(-G(p\lambda(1 - z))x) (1 - B(x)) P_{(0,K)}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))}, \quad (4.37)$$

$$P_{(3,1)}(z, x, y) = \frac{\mu z^K L_A(\lambda + \gamma) \{\lambda(1 - z) + \gamma(1 - K(z))\}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))} \quad (4.38)$$

$$\times \exp(-G(p\lambda(1 - z))x) (1 - B(x)) (1 - C(y)) \quad (4.39)$$

$$\times \exp(-\lambda p(1 - z)y) P_{(0,K)} \quad (4.40)$$

$$P_{(3,2)}(z, x, y) = \frac{\mu \gamma z^{K-1} L_A(\lambda + \gamma) (K(z) - z)}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))} \quad (4.41)$$

$$\times \exp(-G(p\lambda(1 - z))x) (1 - B(x)) (1 - C(y)) \quad (4.42)$$

$$\exp(-\lambda p(1 - z)y) P_{(0,K)} \quad (4.43)$$

Si $p = 1$, $\mu = 0$, $K = 0$ et $\gamma = 0$ nous retrouvons la distribution du système donnée par Gomez-Corral [GOC99].

Remarque 4 Si $p = 1$, $\mu = 0$ et $\gamma_1 = 0$ nous retrouvons la distribution du système donnée par P.Moreno[MOR04].

Remarque 5 Si $p = 1$, $r = 1$, $K = 0$ et $\gamma = 0$ nous retrouvons la distribution du système donnée par X.Wu.et all..[WBH05] et [TSA10].

Théorème 6

$$P_0(1) = \frac{\lambda p P_{0,K} (1 - L_A(\lambda + \gamma)) \beta_1 (1 + \mu \gamma_1)}{\lambda p \beta_1 (1 + \mu \gamma_1) - L_A(\lambda + \gamma)}, \quad (4.44)$$

$$P_1(1) = \frac{-P_{0,K} L_A(\lambda + \gamma) \{\lambda + \gamma \lambda p \beta_1 (1 + \mu \gamma_1)\}}{\lambda p \beta_1 (1 + \mu \gamma_1) - L_A(\lambda + \gamma)}, \quad (4.45)$$

$$P_2(1) = \frac{\beta_1 \gamma P_{0,K} L_A(\lambda + \gamma) \{\beta_1 \lambda p (1 + \mu \gamma_1) - 1\}}{\lambda p \beta_1 (1 + \mu \gamma_1) - L_A(\lambda + \gamma)}, \quad (4.46)$$

$$P_{(3,1)}(1) = \frac{-\mu P_{0,K} L_A(\lambda + \gamma) \{\lambda + \gamma \lambda p \beta_1 (1 + \mu \gamma_1)\}}{\lambda p \beta_1 (1 + \mu \gamma_1) - L_A(\lambda + \gamma)}, \quad (4.47)$$

$$P_{(3,2)}(1) = \frac{\mu \beta_1 \gamma P_{0,K} L_A(\lambda + \gamma) \{\beta_1 \lambda p (1 + \mu \gamma_1) - 1\}}{\lambda p \beta_1 (1 + \mu \gamma_1) - L_A(\lambda + \gamma)}, \quad (4.48)$$

Dans le corollaire suivant, nous donnons quelques mesures de performance du système en régime stationnaire en utilisant le théorème ci-dessus. .

Corollaire 7 Si $\lambda p \beta_1 \{1 + \mu \gamma_1\} < L_A(\lambda + \gamma)$ alors :

- 1) L'orbite est vide de clients transitoires et le serveur est libre avec la probabilité $P_{(0,K)}$.
- 2) Le serveur est libre et le système n'est pas vide avec la probabilité $P_{(0,K)} + P_0(1)$.
- 3) Le serveur est occupé par un client transitaire avec la probabilité $P_1(1)$.
- 4) Le serveur est occupé par un client récurrent avec la probabilité $P_2(1)$.
- 5) Le serveur est en réparation et le client en service après panne est un client transitaire avec la probabilité $P_{(3,1)}(1)$.
- 6) Le serveur est en réparation et le client en service après panne est un client récurrent avec la probabilité $P_{(3,2)}(1)$.

Proposition 8 Soit T_q , T et N_q le nombre de clients transitaires **en orbite**, le nombre de clients dans le système et le nombre de clients en orbite en régime stationnaire respectivement, alors :

Les fonctions génératrices de T_q , T et N_q sont données par :

$$T_q(z) = \frac{P_{0,K}}{\lambda p} \frac{L_A(\lambda + \gamma) \{K(z) [\lambda p - (\lambda + \gamma)] - \lambda p z + \lambda + \gamma\}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))} \quad (4.49)$$

$$T(z) = \frac{P_{0,K}}{\lambda p} \frac{L_A(\lambda + \gamma) \{\lambda p K(z) (1 - z) + (1 - K(z))(-\lambda p z + \lambda z + K(z))\}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))} \quad (4.50)$$

$$Q(z) = \frac{z^{K-1} P_{0,K}}{\lambda p} \frac{L_A(\lambda + \gamma) \{\lambda p z K(z) (1 - z) + (1 - K(z))(-\lambda p z^2 + \lambda z + \gamma K(z))\}}{L_A(\lambda + \gamma) (1 - z) K(z) - z(1 - K(z))}. \quad (4.51)$$

Remarque 9 Pour $p = 1$, $\mu = 0$ $\gamma_1 = 0$ nous retrouvons les résultats de P.Moreno [MOR04] avec $b_1(x) = b_2(x) = b(x)$.

Proposition 10 Le nombre moyen de clients transitaires en orbite et dans le système est donné par :

$$N_{T_q} = \Delta + \Omega; \quad (4.52)$$

avec :

$$\Delta = \frac{L_A(\lambda + \gamma) \{(\lambda + \gamma - \lambda p) [K^{(2)}(1) - 2(K^{(1)}(1))^2] - 2\lambda K^{(1)}(1)\}}{2(-L_A(\lambda + \gamma) + K^{(1)}(1))^2}$$

$$\Omega = \frac{2(K^{(2)}(1) (-\lambda p + \lambda + \gamma) + 2\lambda p K^{(1)}(1) + 2\lambda p K^{(2)}(1))}{2(-L_A(\lambda + \gamma) + K^{(1)}(1))^2}.$$

et

$$N_T = \Gamma - \Lambda + \Psi; \quad (4.53)$$

$$\Gamma = \frac{L_A(\lambda + \gamma) K^{(1)}(1) [\lambda(1 - p) + K^{(1)}(1) (-2\lambda p - 2\lambda - 1)]}{2(-L_A(\lambda + \gamma) + K^{(1)}(1))^2}$$

$$\Lambda = \frac{L_A(\lambda + \gamma) (K^{(2)}(1)) - 2(K^{(2)}(1) (-\lambda - 1 + \lambda p))}{2(-L_A(\lambda + \gamma) + K^{(1)}(1))^2}$$

$$\Psi = \frac{K^{(1)}(1)(\lambda + 2 - 3\lambda p)}{2(-L_A(\lambda + \gamma) + K^{(1)}(1))^2}$$

$$K^{(1)}(1) = \lambda p \beta_1 (1 + \mu \gamma_1)$$

$$K^{(2)}(1) = (\lambda p)^2 \mu \gamma_2 \beta_1 + (\lambda p)^2 \beta_2 (1 + \mu \gamma_1)^2$$

$K^{(1)}(x)$ et $K^{(2)}(x)$ sont les deux premières dérivées de $K(x)$.

4.4 Distribution du nombre de clients transitaires servis durant la période d'occupation

Dans cette section nous donnons les trois premières probabilités du nombre de clients transitaires servis durant la période d'occupation, en fonction des transformées de Laplace du service généralisé et du rapel.

Proposition 11 *Soit T le nombre de clients transitaires servis durant la période d'occupation. Alors*

$$P(T = 1) = L_{B^*}(\lambda p), \quad (4.54)$$

$$P(T = 2) = -\lambda p L'_{B^*}(\lambda p) L_A(\lambda + \gamma) L_{B^*}(\lambda p), \quad (4.55)$$

$$P(T = 3) = A + B + C \quad (4.56)$$

avec

$$A = \frac{1}{2} \lambda^2 p^2 L''_{B^*}(\lambda p) (L_A(\lambda + \gamma))^2 L_{B^*}^2(\lambda p)$$

$$B = \lambda^2 p^2 (L'_{B^*}(\lambda p))^2 L_{B^*}(\lambda p) (L_A(\lambda + \gamma))^2$$

$$C = (-\lambda p L'_{B^*}(\lambda p)) (1 - L_A(\lambda + \gamma)) L_{B^*}(\lambda p) L_A(\lambda + \gamma) L_{B^*}(\lambda p)$$

$L'_{B^*}(s)$ et $L''_{B^*}(s)$ sont la première et la seconde dérivée respectives de $L_{B^*}(s)$.

Preuve. L'événement $\{T = 1\}$ ne peut se produire que si et seulement si

- Aucun client transitaire entre en orbite durant le premier service généralisé. □

L'événement $\{T = 2\}$ ne peut se produire que si et seulement si

- Un client transitaire entre en orbite durant le premier service généralisé, le client suivant qui entre en service vient de l'orbite est un client transitaire et aucun client n'entre en orbite durant le deuxième service généralisé.

L'événement $\{T = 3\}$ ne peut se produire que si et seulement si

- Un client transitaire entre en orbite durant le premier service généralisé, le client transitaire entre en service, un client transitaire entre en orbite durant le second service généralisé, le second client en service qui vient de l'orbite est un client transitaire et aucun client n'entre en orbite durant le troisième service généralisé.

- Un client transitaire entre en orbite durant le premier service généralisé, le nouveau client en service est le client transitaire, le client en orbite reste en orbite, aucun client n'entre en

orbite durant le second service généralisé, le client suivant en service qui vient de l'orbite est un client transitaire et aucun client n'entre en orbite durant le troisième service généralisé.

- Deux clients transitaires entrent en orbite durant le premier service généralisé, le client suivant en service qui vient de l'orbite est un transitaire, aucun client n'entre en orbite durant le second service généralisé, le client suivant en service qui vient de l'orbite est un transitaire, aucun client n'entre en orbite durant le troisième service généralisé.

Conclusion

L'asynchronisme du mode ATM provoque des files d'attente. Si on veut que ces files d'attente ne soient pas trop longues, il faut que le taux de charge soit inférieur à 0,8 environ. Toutefois, la charge restante peut être utilisée pour transférer des données "relativement" non prioritaires pour lesquelles on peut accepter des retards de quelques secondes (lorsque des cellules sont supprimées et qu'il faut relancer une partie du message) : fax, communication téléphonique qui dépasse trenteminutes, minitel parfois, etc.

Dans une telle éventualité, il faut mettre en place des mécanismes de régulation. Notamment, il faut décider à partir de quel seuil on commence à « supprimer » les cellules non prioritaires.

L'objectif de ce travail n'est pas d'étudier de tels mécanismes de régulation, ce qui serait prématuré, mais de donner des outils mathématiques pour se faire. La méthode de la chaîne incluse et alors quasi -incontournable. On a étudié la chaîne incluse qui dissocie le temps durant lequel il n'y a pas de cellules prioritaires en attente - ce qui permet l'écoulement des cellules non prioritaires - du temps durant lequel il y a des cellules prioritaires - ce qui provoque l'engorgement des cellules non prioritaires. Il est essentiel de noter qu'il est technologiquement facile de voir dans laquelle des deux situations ci-dessus on se trouve. On peut donc imaginer des mécanismes de régulation qui s'appuient sur les "

temps d'arrêt" associés au passage d'une situation à l'autre.

Un point essentiel est d'estimer la probabilité des événements rares. Il se trouve que les longueurs des files d'attente des cellules non prioritaires aux temps d'arrêt précités suivent des lois presque géométriques, en un sens précisé précédemment.

L'approximation par des processus de diffusions est donc nécessairement aberrante pour le calcul des probabilités des états rares.

Le rapport limite lié à de telles lois est alors un paramètre crucial. Dans deux cas distincts,

on montre que ce rapport est exactement une racine d'une équation simple.

Le premier cas est celui où on peut observer le nombre de cellules non prioritaires à l'instant où arrive une cellule prioritaire, alors qu'il n'en y avait pas avant. Le deuxième cas est celui où une telle observation est trop coûteuse est on se contente de considérer le régime stationnaire de la chaîne incluse, ce qui conduira évidemment à une régulation moins fine.

Le dernier chapitre est réservé aux systèmes de files d'attente avec rappel, serveur non fiable et deux types de clients. Nous avons donné la condition de stabilité du système, les performances du système. Nous avons également trouvé les trois premières probabilités du nombre de clients servis durant une période d'occupation.

L'objet d'un ouvrage futur est d'examiner une extension de ce modèle.

Les programmes informatiques associés aux vérifications évoquées précédemment (chapitre 2) sont rédigés en PASCAL ou en MATTLAB.

Bibliographie

- [ABM08] Atencia I, Bouza, and ,Moreno P 2008 An $M^{[X]}/G/1$ retrial queue with the server breakdowns and constant rate repeated attempts', *Revue Annals of Operations Research*, Vol 157, pp 225-243
- [Ais88] A.Aissani, On the $M/G/1$ queueing system with repeated orders and unreliable server, *J.Technology* 6, 98-123,1988.
- [Ais93] A.Aissani, Unreliable queueing with repeated orders, *Microelectronics and Reliability* 33,209-2106, 1993.
- [Ais94] A.Aissani, A retrial queue with redundancy and unreliable server, *Queueing systems* 17, 413-449, 1994.
- [Ais03] A.Aissani, An $M^X/G/1$ retrial queue with unreliable server and vacations, *Proceedings 17th European simulation Multiconference*, 175-180, 2003.
- [AiA98] A.Aissani and J.R.Artalejo, On the single server retrial queue subject to breakdown, *Queueing Systems* 30-309-321,1998.
- [Art94] J.A.R.Artalejo, New results in retrial queueing systems with breakdowns of the servers, *Statistica Neerlandica* 48-23-36, 1994.
- [Art99a] J.R.Artalejo, Accessible bibliography on retrial queues, *Mathematical and Computer Modelling*, vol 30, 1-6, 1999a.
- [Art99b] A classified bibliography of research on retrial queues progress in 1990-1999, *Top*, vol 7, 187-211.
- [AtM06] Atencia I,Moreno P, 'A discrete- time $Geo/G/1$ retrial queue with the server subject to starting failures,' *Annal of Operations Research*, Vol 141, pp 85-107.
- [BeP89] R.Bellamine et J.Pellaumail, *Opérateurs positifs et probabilités stationnaires*.
- [BGL90] B.Bensaou, J.Guibert and J.W.Roberts, Fluid queueing models for a superposition of bursty sources, *ITC Broadband Seminar*, May 1990.

- [BDM94] M.Boisseau, M.Demange et J.M.Munier. Réseaux A.T.M, Eyrolles 1994.
- [BoC91] O.J.Boxma, J.W.Cohen, The M/G/1 queue with permanent customers, IEEE Journal on S ELECTED Areas in Communications 9(2) (1991)179-184.
- [BGL87] P.E.Boyer, A.Gravey et J.R.Louvion, Modèles de files d'attente pour un RNIS large bande. Note Technique LAA/SLC/228, CNET Lannion,1987.
- [BGSC92] P.E.Boyer, F.M.Guillemin, M.J.Serval and J.P.Coudreuse, Spacing cells protects and enhances utilization of A.T.M networks links ; IEE Networks, 1992.
- [BoG92] P.E.Boyer and F.M.Guillemin, An upper-bound for the clumping tolerance parameter at the TB Interface. A.T.M Forum, 1992.
- [BoT92] P.E.Boyer and D.P.Tranchier, A reservation principale with applications to the A.T.M trafic control. Computer Networks and ISDN Systems 24, 321-334-1992.
- [Boy90] P.E.Boyer, Définition d'un contrôle de congestion pour un réseau temporel asynchrone.Note technique LAA/RSM/165, CNET Lannion, 1990.
- [BSG92] P.E.Boyer, M.J.Serval and F.M.Guillemin, The spacer-controller an efficient UPC/NPC for A.T.M networks, ISS'92, Yokohama, 1992. Appl.Prob.30, pp.237-246-1993,9.B.7.
- [CPP93] B.D.Choi. K.K.Park,C.E.M.Pearce. An M/M/1 retrial queue with control policy and general retrial times, Queueing Systems 14, 275-292, 1993.
- [CoP91] J.W.Cohen and C.D.Pack, Queueing performance and control in A.T.M North Holland, 1991.
- [Coo81] R.B.Cooper, Introduction to queueing theorie, Edward Arnold 1981.
- [Cou91] J.P.Coudreuse, L'écho des recherches. Spécial ATM, vol.1 et 2, N^o144 et 145, CNET France 1991.
- [Dje06] Djellab,N.V 2006 'On the single -server retrial queue', Yugoslav Journal of Operations Research, Vol 16,pp 45-53
- [Dug97] O.Dugeon, Machine d'admission pour le réseau d'infrastructure. Thèse Rennes, 1997.
- [Eve53] J.L.Everett, State probabilities in congestion problems characterized by constant holding times. J.Operations Research Soc.Am, vol.1, pp. 279-285-1953.
- [Far96] K.Farahmad, Single line queue with recurrent repeated demands, Queueing Systems 22-425-435, 1996.

- [FaT97] G.I.Falin and J.G.C.Templeton Retrial queues, Chapman and Hall, NewYork.
- [GBDR92] F.Guillemain, P.E.Boyer, A.Dupuis and L.Romoeuf, Peak rate enforcement in ATM networks. Infocom'92, Firenze, 1992.
- [GLB87] A.Gravey, J.R.Louvion et P.Boyer, Modèles de files d'attente pour un RNIS large bande 1987.
- [Gom99] A.Gomez-Corral, Stochastic analysis of single server retrial queue with the general retrial times, Naval Research Logistics 46-561-581, 1999.
- [GrB93] A.Gravey and P.E.Boyer, Cell delay variation spécification in ATM networks C-15. Modelling and Performance evaluation of ATM Technology, Elsevier, 1993.
- [GuD92] F.M.Guillemain and A.Dupuis, A basic requirement for the policing function in ATM networks. Computer Networks and ISDN Systems, 311-320-1992.
- [JaP00] Z.Jadda, J.Pellaumail, Basic processes in ATM networks. Journal of pure and Applied Algebra, 146, 185-196, 2000.
- [koG96] D.Kofman et M.Gagnaire, Réseaux haut débit. Intéréditions, 1996.
- [KKA02] B.Krishma Kumar and D.Arivudainambi, The M/G/1 retrial queue with Bernoulli schedules and general retrial times, Computers and Mathematics with applications 43, 15-30, 2002.
- [KPV02] B.Krishma Kumar, S.Pavai Madheswari and A.Vijayakumar, The M/G/1 retrial queue with feedback and starting failures, Applied Mathematical Modelling 26, 1057-1075, 2002.
- [KuC90] V.G.Kulkarni and B.D.Choi, Retrial queue with server subject to breakdown and repairs, Queueing Systems 7, 191-208, 1990.
- [KuL97] V.G.Kulkarni and H.M.Liang, Retrial queues revisited, Frontiers in queueing. Models and applications in Science and engineering. J.H.Dshalalow(Editor), CRC Press, Boca Raton, FL,19-34 ,1997.
- [KuL00] V.G.Kulkarni and H.M.Liang Retrial queues revisited, 2000.
- [Leb91] J.Y.LE Boudec, The ATM tutorial. Computer networks and ISDN systems 24, 279-309, 1992.
- [Lop02] M.J.Lopez-Herrero, On the number of customers served in M/G/1 Retrial queue : first moments and maximum entropy approach, Computer & Operations Research 29, 1739-1757, 2002.

- [Mai97] M.Maimant, Telecoms réseaux, communications d'entreprise. Interéditions, 1997.
- [Met80] M.Métivier and J.Pellaumail, Stochastic Intégration. Academic Press, 1980.
- [Mor04] P.Moreno, An M/G/1 retrial queue with recurrent customers and general retrial times, Applied Mathematics and Computation 159, 651-666, 2004.
- [Mor] Moreno P'A discrete- timeGeo/G/1 retrial queue with unreliable server and general server lifetime,'
'Journal of Mathematics Sciences, Vol 132pp, 643- 655.
- [Neu89] M.F.Neuts, Structured stochastic matrices of M/G/1 type and their applications.Marcel Dekker, 1989.
- [PBL94] J.Pellaumail, P.Boyer et P.Leguesdron, Autoroutes ATM. Hermès,1997.
- [Pel92a] J.Pellaumail, Graphes, Simulation, L-matrices. Hermès, 1992.
- [Pel96] J.Pellaumail, Majoration des retards dans les réseaux ATM. RAIRO, Recherche Opérationnelle, vol.30, N^o1,p.51-64,1996.
- [POS90] J.Pellaumail, B.Oumahdi et H.Saggou, Monotone ill-conditioned matrices and stationary probabilities.22ème Congrès National d'Analyse Numérique, Loctudy, France,pp. 296-298, 1990.
- [Pra65] N.U.Prabhu, Queues and inventories. John Wily, 1965.
- [Pry94] M.DE.Prycher, ATM.Mode de Transfert Asynchrone, 1994.
- [Sag89] H.Saggou, Etude d'une file avec priorité et loi non exponentielles.Thèse de Magister, Alger, 1989.
- [SaP97] H.Saggou et J. Pellaumail, Chaîne incluse pour les communications ATM non prioritaires. LANS N^o = 72/1997.
- [SaP98] H.Saggou et J. Pellaumail, Modélisation des communications ATM non prioritaires. LANS N^o = 82/1998.
- [SaP99] H.Saggou et J.Pellaumail, Un paradoxe dans la modélisation des réseaux ATM L'IRMAR N^o34/99, 1999.
- [SaP03] H.Saggou et J.Pellaumail.The diffusion processes approximation often proves unsuitable for the studying rare events. Maghreb Mathematical Review Rf :130/03 MMR N^o37/02, 2003.
- [SaP05] H.Saggou et J.Pellaumail Imbedded process and applied priority or A.T.M networks. Maghreb Mathematical Review Rf : 15/05 MMR, N^o10 / 04 2005.

- [SaM08] H-Saggou and N-Mikkou queueing Model with retrial ,breakdown,repair,loss,two classes of customers and the distribution of the number of customers served during the busy period preprint 2008 revue R.0
- [ShK06] Sherman,N.P and Kharoufeh,J.P 2006' An M/M/1 retrial queue unreliable server Operations Research Letters, Vol 34, pp 697-705
- [SHT83] L.I.Sennott, P.A.Humblet and R.I.Tweedie, Mean drift and the non-ergodicity of Markov chains, Queueing Systems 31, 783-788, 1983.
- [Tan97] A.Tanenbaum, Réseaux troisième édition. Interditions, 1997.
- [TSA10] S-Taleb,H-Saggou and A-Aissani, M/G/1 Retrial queue with geometric loss and random reserved timeV7N1-2 2010 I-J-O-R International Journal of Operational Research.
- [WCL01] J.Wang,J.Cao and Q.Li, analysis of the retrial queue with server breakdowns and repairs, Queueing Systems 38, 363-380, 2001.
- [WBHW05] XY.WU.P.Brill. M.Hynka and J.Wang, An M/G/1 retrial queue with balking and retrial during service , J.Operational Research 1(1 – 2) ,30-51 2005
- [YaG94] T.Yang and H.Li, The M/G/1 retrial queue with the server subject to starting failures, Queueing Systems 16, 83-96, 1994.
- [YPT90] T.Yang, M.J.M.Posner and J.G.C.Templeton, The M/G/1 retrial queue with non-persistent customers, Queueing Systems 7, 1990.

Annexe

Glossaire

A

Adresse

Destination ou provenance d'un message envoyé à travers un réseau. Une adresse peut désigner une entité unique ou un groupe d'entités.

ou

Code ayant pour but d'identifier chaque ordinateur connecté à l'internet.

B

Bande passante

Quantité maximale d'informations en bits par seconde que peut véhiculer un canal de communication

Bit

Terme né de la contraction de binary digit qui désigne un chiffre binaire prenant les valeurs 0 et 1, ou une unité binaire de quantité d'information.

Bits/s

Unité utilisée pour désigner le débit des transmissions de données circulant sur les réseaux de communication.

Buffer

Voir tampon.

C

Canal

Support de communication. Généralement utilisé pour le niveau physique.

Exemple pour A.T.M

« chemin de transfert de l'information ». Un canal est identifié par les numéros VPI/VCI.

CCIT

Comité Consultatif International pour le Téléphone et le Télégraphe : association internationale assurant la normalisation des communications à l'échelon mondial et éditant des recommandations appliquées par de très nombreux pays. Le CCITT, dépendant de l'U.I.T (International Télécommunication Union), ou I.T.U (International Télécommunication Union) est maintenant remplacé par l'ITU_T.S (Télécom sector) et l'I.T.T.U_R.S

(Radiotélécom Sector).

Circuit

Ensemble de deux voies de transmission associées entre deux points pour assurer la transmission bidirectionnelle des signaux appartenant à une communication.

CNET

Centre National d'Etude des Télécommunications

Connexion

Ensemble de moyens qui permettent de transmettre les données en garantissant certaines propriétés à l'utilisateur.

Une connexion ressemble à une communication téléphonique ; elle a trois phases : établissement, transmission, libération.

D*Débit*

Nombre de bits écoulés par seconde ; notation : bit /s : b/s, bits par secondes, kb /s pour kilobits/s (10 .E.3) , Mb/s pour Mega bits/s (10.E.6) et Gb/s pour Giga bits/s (10.E.9).

F*Flux* (d'information)

L'ensemble des informations en transit sur un réseau.

Format

Description de la structure d'une entité de données (paquet, message)

I*I.121 Broadband Aspects of ISDN*

Aspects large bande du RNIS

L*Large bande*

Liaison ou réseau capable de véhiculer un très grand nombre de signaux à des fréquences élevées

M*Multiplexage*

Opération qui consiste à écouler sur une même voie plusieurs communications d'utilisateurs différents.

N*Nœud*

Dans un réseau à commutation de paquets un nœud est un ordinateur où est effectué le routage des paquets. Les paquets arrivants sont mémorisés et renvoyés sur la voie de sortie qui leur convient.

R*RNIS*

Réseau Numérique à Intégration de service : réseau mettant à profit les techniques de numération pour transporter sur la même infrastructure plusieurs services concernant la voix, les données ou les images.

Routage

Action de rechercher le chemin pour permettre à des données de transiter vers leur destination.

T*Tampon*

Zone mémoire où sont stockées temporairement les données d'E/S . Homonyme de buffer.

Trafic

Densité d'information sur un canal de transmission

V*VCI*

Virtuel Channel Identifier : identifiant de voie virtuelle. Une voie virtuelle permet la transmission d'informations de bout en bout.

VPI

Virtuel Path Identifier : identifiant de conduit virtuel. Un conduit virtuel est un ensemble de voies virtuelles.

Listing du Programme

program priorite ;

{ $N+$ }

{ L'objectif de ce programme est d'étudier une file d'attente avec priorités. Le processus X correspond aux cellules prioritaires et le processus Y correspond aux cellules non-prioritaires. Le taux d'arrivées de X est alpha et celui de Y est beta. Dans une première partie, on procède par "pseudo-simulation", c'est-à-dire que l'on étudie le système fini des probabilités conjointes et le régime stationnaire par itérations successives.

Ensuite, en comparant aux résultats obtenus dans la première partie, on vérifie par le calcul informatique les formules établies dans le Chapitre 2. }

```

{ DECLARATIONS }

const

{ Tailles des tableaux : }

max_Poisson=15; max_Poisson_long=100;

max_X=39; max_Y=39; max_delai_occupation=400;

{ nbi = Nombre d'itérations dans les calculs par itérations successives }

nbi_X=2000; nbi_X_Y=200;

{ Le paramètre qui suit est notamment utilisé dans le calcul, par P-simulation,
de la matrice de transition de v(n) à w(n) }

max_wn_moins_vn=400;

{ Le paramètre qui suit est utilisé dans le calcul de r }

max_r=150;

igfich=0; { si igfich = 0 on met sur un fichier,
           si igfich = 1 on sort a l'écran,
           si igfich = 2 on sort sur imprimante }

type

t_Poisson      = array[0..max_Poisson] of real;
t_Poisson_long = array[0..max_Poisson_long] of real;
t_processus_X  = array[0..max_X] of real;
t_processus_Y  = array[0..max_Y] of real;
t_processus_X_Y = array[0..max_X,0..max_Y] of real;
t_transition_Y_Y = array[0..max_Y,0..max_Y] of real;
t_delai_occupation_X = array[0..max_delai_occupation] of real;

var bg1details, bg2details, bgcumulee, bg_convexe_detaille,
    bg_transition_alpha,
    bg_stationnaire_alpha, bg_af_r : boolean;

```

```

ig_gamma_stationnaire,i_initial,
max_delai                : integer ;
alpha, beta              : extended ;
z, ngamma, ngammap      : real ;
vn,wn                    : t_processus_Y ;
tgf                      : text ;
nomfich                  : string[40] ;

```

{ Commentaires concernant les variables globales.

alpha est le taux d'arrivées du processus prioritaire X et beta est le taux d'arrivées du processus non prioritaire Y.

z est utilisé, dans la vérification des formules (2.1) et (2.3); on se limite à z réel.

bg_transition_alpha est un booléen de gestion. Il est égal à vrai tant que la matrice de transition de $v(n)$ à $w(n)$ n'a pas été, calculée.

Cette matrice de transition de $v(n)$ à $w(n)$ est, en général, appelée tr1; toutefois, elle n'est pas mise en variable globale car elle peut prendre beaucoup de place mémoire.

bg_stationnaire_alpha est un booléen de gestion. Il est égal à vrai tant que les probabilités stationnaires de la chaîne incluse n'a pas été, calculée.

Ces probabilités stationnaires de la chaîne incluse aux temps d'arrêt $v(n)$ et $w(n)$ respectivement sont mises dans les tableaux vn et wn; ces tableaux ,tant cruciaux et prenant peu de place mémoire, ils sont mis en variables globales.

ngamma et ngammap sont associées à la loi Pseudo-géométrique de la chaîne incluse. Si ig_gamma_stationnaire vaut 1, le calcul de gamma est effectué pour une loi initiale de Dirac. Si ig_gamma_stationnaire vaut 2 ou 3, le calcul de gamma

est associé au régime stationnaire ; s'il vaut 2 ce calcul n'utilise pas r.

Suivant les valeurs des booléens de gestion bg1details et bg2details, le programme rend plus ou moins de résultats intermédiaires. En général, bg2details est mis égal à faux. Par contre, bg1details est choisi égal à vrai pour le premier jeu de paramètres puis remis égal à faux ensuite. De même, la valeur de r n'est affichée que si bg_af_r est égal à vrai.

Quand bg_convexe_detaille est vrai, le programme utilise la méthode des convexes avec un grand nombre de détails superflus quant au fond, ce qui permet d'effectuer de nombreuses vérifications. Sinon, le programme utilise la méthode des convexes sous une forme simplifiée.

A quelques rares exceptions près, les variables locales se terminent par un chiffre.}

```
procedure lignes(i1 : integer) ;
```

```
{ Cette procédure permet de sauter quelques lignes à l'exécution. }
```

```
var j1 : integer ;
```

```
begin for j1 :=1 to i1 do writeln(tgf,j1 :3) ; end ;
```

```
procedure arret ; { Cette procédure arrête le programme quand on travaille à l'écran. }
```

```
begin
```

```
if igfich=1 then
```

```
begin
```

```
writeln(tgf) ; writeln(tgf,' * * * ATTENTION : Effectuez un envoi pour continuer') ;
```

```
readln ;
```

```
end ;
```

```
end ;
```

```

{ MISES A ZERO }

procedure zero_Poisson( var pr1 :t_Poisson);

var p1 : integer;

begin for p1 :=0 to max_Poisson do pr1[p1] :=0; end;

procedure zero_Poisson_long( var pr1 :t_Poisson_long);

var p1 : integer;

begin for p1 :=0 to max_Poisson_long do pr1[p1] :=0; end;

procedure zero_processus_X(var a1 :t_processus_X);

var i1 : integer;

begin for i1 :=0 to max_X do a1[i1] :=0; end;

procedure zero_processus_Y(var a1 :t_processus_Y);

var i1 : integer;

begin for i1 :=0 to max_Y do a1[i1] :=0; end;

procedure zero_processus_X_Y(var a1 :t_processus_X_Y);

var i1,j1 : integer;

begin for i1 :=0 to max_X do
for j1 :=0 to max_Y do a1[i1,j1] :=0; end;

procedure zero_transition_Y_Y(var a1 :t_transition_Y_Y);

var i1,j1 : integer;

begin for i1 :=0 to max_Y do
for j1 :=0 to max_Y do a1[i1,j1] :=0; end;

procedure zero_delai_occupation(var a1 :t_delai_occupation_X);

var i1 : integer;

begin for i1 :=0 to max_delai_occupation do a1[i1] :=0; end;

```

```
{ AFFICHAGES DE BASE et NORMALISATION }
```

```

procedure af_debut ;
begin
writeln(tgf) ; writeln(tgf, ' MODELISATION ATM '); writeln(tgf) ;
writeln(tgf, ' Etude d"une file d"attente avec deux niveaux de priorités. ');
writeln(tgf, ' L"objectif essentiel est de vérifier des formules. ');
writeln(tgf, ' La modélisation repose sur la technique de la chaîne incluse. ');
writeln(tgf) ;
writeln(tgf, ' max_Poisson = ', max_Poisson :4, ' et max_Poisson_long = ', max_Poisson_long :4) ;
writeln(tgf, ' max_r = ', max_r :4, ', max_X = ', max_X :4, ' et max_Y = ', max_Y :4) ;
writeln(tgf, ' max_delai_occupation = ', max_delai_occupation :4, ' et nbi_X = ', nbi_X :4) ;
writeln(tgf, ' nbi_X_Y = ', nbi_X_Y :4, ' et max_wn_moins_vn = ', max_wn_moins_vn :
4) ;
arret ;
end ;

procedure af_processus_X(a1 :t_processus_X) ;
var il,j1, n1 : integer ;
begin
writeln(tgf) ; j1 :=0 ; n1 :=max_X ; if n1>19 then n1 :=19 ;
for il :=0 to n1 do
begin
write(tgf,a1[il] :14, ' '); j1 :=j1+1 ;
if j1=5 then begin writeln(tgf) ; j1 :=0 ; end ;
end ;
writeln(tgf) ; arret ;
end ;

```

```

procedure af_processus_Y(a1 :t_processus_Y);
var i1,j1, n1 : integer;
begin
writeln(tgf); j1 :=0; n1 :=max_Y; if n1>19 then n1 :=19;
for i1 :=0 to n1 do
begin write(tgf,a1[i1] :14,' ');j1 :=j1+1;
if j1=5 then begin writeln(tgf); j1 :=0; end;
end;
writeln(tgf); arret;
end;

procedure af_processus_X_Y(p1 :t_processus_X_Y);
var i1,j1,k1,k2 : integer;
begin
writeln(tgf,' Voici une partie');
writeln(tgf,' de la probabilité stationnaire du couple (X,Y) : ');
k1 :=max_X; k2 :=max_Y; if k1>9 then k1 :=9; if k2>9 then k2 :=9;
for i1 :=0 to k1 do
begin writeln(tgf,' Pour X = ',i1 :4,' les probabilités valent : ');
for j1 :=0 to k2 do write(tgf,p1[i1,j1]); writeln(tgf);
end;
writeln(tgf);arret;
end;

```

```

procedure af_delai_occupation(a1 :t_delai_occupation_X);
var i1,j1,k1 : integer;
begin
k1 :=max_delai_occupation; if k1>19 then k1 :=19; writeln(tgf); j1 :=0;
for i1 :=0 to k1 do
begin
write(tgf,i1 :4,' ',a1[i1] :12,' *');j1 :=j1+1;
if j1=4 then begin writeln(tgf); j1 :=0; end;
end;
writeln(tgf); arret;
end;

procedure suite_geo(v1 :t_processus_Y);
var i1,i2,n1 : integer;
r1,r2 : real;
begin
writeln(tgf,' A-t-on une suite géométrique? '); n1 :=max_Y-1; if n1>19 then n1 :=19;
writeln(tgf,' Les rapports successifs valent : '); i2 :=0;
for i1 :=0 to n1 do
begin i2 :=1+i2; r1 :=v1[i1];r2 :=v1[1+i1]; if r2=0 then r2 :=1; r1 :=r1/r2;
write(tgf,r1 :14,' '); if i2=5 then begin i2 :=0; writeln(tgf); end;
end;
writeln(tgf); arret;
end;

```

```

procedure af_transition_Y_Y(tr1 :t_transition_Y_Y);
var i1,j1,k1,k2 : integer;

begin
writeln(tgf,' la matrice de transition est : ');

k1 :=max_Y; if k1>10 then k1 :=10; k2 :=max_Y; if k2>20 then k2 :=20;

for i1 :=0 to k1 do

begin writeln(tgf,' i1 = ',i1 :4);

for j1 :=0 to k2 do write(tgf,tr1[i1,j1] :12, ' '); writeln(tgf);

end;

writeln(tgf);arret;

end;

```

```

procedure normal_YY(var tr1 :t_transition_Y_Y);
{ Cette procédure dit si une matrice est bien une matrice de transition et
effectue le réajustement associé; ceci est nécessaire quand on effectue le
produit d'un grand nombre de telles matrices }

var i1,j1,k1 : integer;

s1,s2 : real;

begin

s2 :=0;

for i1 :=0 to max_Y do

begin

s1 :=0;

for j1 :=0 to max_Y do s1 :=s1+tr1[i1,j1];

s2 :=s2+abs(1-s1); if s1=0 then s1 :=1;

for j1 :=0 to max_Y do tr1[i1,j1] :=tr1[i1,j1]/s1;

end;

```

```

writeln(tgf,' le défaut de transition vaut : ',s2);

if bg2details then

begin

writeln(tgf,' la matrice de transition normalisée est : ');

af_transition_Y_Y(tr1); arret;

end;

end;

{ CONSTRUCTIONS ET OPERATIONS DE BASE }

procedure cons_Poisson(alpha1 :real; var pr1 : t_Poisson);
{ Cette procédure construit la loi de Poisson de paramètre alpha1
et la met dans le tableau pr1 }
var j2 : integer;
x1,x2,s1 : real;
begin
zero_Poisson(pr1); x1 :=alpha1; x2 :=exp(-x1); s1 :=x2; pr1[0] :=x2;
for j2 :=1 to (max_Poisson-1) do begin x2 :=x2*x1/j2; pr1[j2] :=x2; s1 :=s1+x2; end;
x1 :=1-s1; if x1<0 then x1 :=0; pr1[max_Poisson] :=x1;
end;

procedure cons_Poisson_long(alpha1 :real; var pr1 : t_Poisson_long);
{ Cette procédure construit la loi de Poisson de paramètre alpha1 et la met dans
le tableau pr1; il faut utiliser cette procédure quand alpha est " grand " }
var j2 : integer;
x1,x2,s1 : real;
begin
zero_Poisson_long(pr1); x1 :=alpha1; x2 :=exp(-x1); s1 :=x2; pr1[0] :=x2;

```

```

for j2 :=1 to (max_Poisson_long-1) do begin x2 :=x2*x1/j2; pr1[j2] :=x2; s1 :=s1+x2;
end;

```

```

x1 :=1-s1; if x1<0 then x1 :=0; pr1[max_Poisson_long] :=x1;

```

```

end;

```

```

procedure moyenne_theorique_X(alpha1,x2 : real);

```

```

{ Cette procédure calcule et affiche la moyenne théorique du régime
stationnaire de X et la compare au nombre réel x2 }

```

```

var x1 : real;

```

```

begin

```

```

x1 :=alpha1+((alpha1*alpha1)/(2*(1-alpha1)));

```

```

writeln(tgf,' La moyenne théorique est : ',x1);

```

```

x1 :=x1-x2; writeln(tgf,' L"ecart entre les deux moyennes vaut : ',x1);

```

```

end;

```

```

{ PSEUDO-SIMULATION }

```

```

procedure stationnaire_X(alpha1 :real;pr1 :t_Poisson;var r1 :t_processus_X);

```

```

{ Cette procédure calcule la loi stationnaire de X et la met dans le tableau r1.

```

```

ATTENTION : pr1 est la loi d'arrivée; en général on prend pour pr1 la loi de

```

```

Poisson, mais on pourrait choisir une autre loi d'arrivée;

```

```

le tableau pr1 doit être construit avant la procédure }

```

```

var i1,i2,j1,n1,n2,p1 : integer;

```

```

x1,x2 : real;

```

```

r2 : t_processus_X;

```

```

begin

```

```

zero_processus_X(r1);r1[0] :=1; j1 :=0;

```

```

for n1 :=1 to nbi_X do

```

```

begin

```

```

if igfich=1 then begin j1 :=n1 mod 100; if j1=0 then write(n1 :5); end;
zero_processus_X(r2);
for i1 :=0 to max_X do
begin x1 :=r1[i1];
for p1 :=0 to max_Poisson do
begin if i1>0 then i2 :=i1-1 else i2 :=0; i2 :=i2+p1;
if i2>max_X then i2 :=max_X; r2[i2] :=r2[i2]+pr1[p1]*x1;
end;
end;
r1 :=r2;
end;
if igfich=1 then writeln;
end;

procedure busy_stationnaire_X(var d1 :t_delai_occupation_X);
var i1,i2,n1,n2,p1 : integer;
x1,x2 : real;
r1,r2 : t_processus_X;
pr1 : t_Poisson;
begin
{ Dans cette procédure, on calcule, par itérations successives, la loi du
délai d'occupation du processus de base X quand ce processus est nul avant
0 et est strictement positif en 0 : ce n'est pas exactement la variable R
du chapitre 4 du livre " Autoroutes ATM " }
zero_delai_occupation(d1); x1 :=1-exp(-alpha);
cons_Poisson(alpha,pr1); zero_processus_X(r1);
n1 :=max_Poisson; if n1>max_X then n1 :=max_X;

```

```

for i1 :=1 to n1 do r1[i1] :=pr1[i1]/x1;
for n1 :=1 to max_delai_occupation do
begin
if igfich=1 then write(n1 :5);
zero_processus_X(r2);
for i1 :=1 to max_X do
begin
x1 :=r1[i1];
for p1 :=0 to max_Poisson do
begin
if i1>0 then i2 :=i1-1 else i2 :=0;
i2 :=i2+p1; if i2>max_X then i2 :=max_X;
r2[i2] :=r2[i2]+pr1[p1]*x1;
end;
end;
r1 :=r2;
d1[n1] :=r2[0];
end;
if igfich=1 then writeln;
end;

procedure attente_stationnaire(var d1 :t_delai_occupation_X);
var i1,i2,p1,n1 : integer;
x1 : real;
r1,r2,gamma1 : t_processus_X;
pr1 : t_Poisson;
begin

```

```

{ Dans cette procédure, on calcule, par itérations successives, la loi du
délai d'attente d'une cellule non prioritaire en régime stationnaire ;
cette procédure est très proche de busy_stationnaire sauf que la loi
initiale est la loi stationnaire du processus de base X soumis à un flux
entrant de taux alpha+beta }
x1 :=alpha+beta; cons_Poisson(x1,pr1); stationnaire_X(x1,pr1,gamma1);
if bg2details then writeln(tgf,' La loi stationnaire de X est calculée ');
writeln(tgf); arret; zero_delai_occupation(d1); cons_Poisson(alpha,pr1);
for i1 :=0 to max_X do r1[i1] :=gamma1[i1];
for n1 :=1 to max_delai_occupation do
begin
{ L'instruction ci-dessous permet de voir à quelle vitesse s'exécute le programme }
if igfich=1 then write(n1 :4);
zero_processus_X(r2);
for i1 :=1 to max_X do
begin
x1 :=r1[i1];
for p1 :=0 to max_Poisson do
begin
if i1>0 then i2 :=i1-1 else i2 :=0;
i2 :=i2+p1; if i2>max_X then i2 :=max_X;
r2[i2] :=r2[i2]+pr1[p1]*x1;
end;
end;
r1 :=r2; d1[n1] :=r2[0];
end;

```

```

if igfich=1 then writeln;
if bg1details then
begin
writeln(tgf,' La loi du délai d"attente d"une cellule non prioritaire,');
writeln(tgf,' calcule par P-simulation, est : ');
af_delai_occupation(d1);
writeln(tgf,' Pour vérification, la loi résiduelle est : ');
af_processus_X(r1);
end;
end;

procedure stationnaire_X_Y;
{ Cette procédure calcule, par itérations successives, et affiche la loi
stationnaire du processus Z = (X,Y) }
var i1,i2,j1,j2,h1,k1,t1 : integer;
x1,x2,x3 : real;
a1,b1 : t_Poisson;
r1 : t_processus_X;
p1,p2 : t_processus_X_Y;
begin
{ On prend la loi stationnaire de X comme initialisation }
cons_Poisson(alpha,a1); stationnaire_X(alpha,a1,r1); zero_processus_X_Y(p1);
for i1 :=0 to max_X do p1[i1,0] :=r1[i1]; { Fin de l'initialisation de p1 }
cons_Poisson(beta,b1);
for t1 :=1 to nbi_X_Y do
begin
{ l'instruction ci-dessous sert à observer la vitesse d'exécution }

```

```

if igfich=1 then write(t1 :4); zero_processus_X_Y(p2);
for i1 :=1 to max_X do
for j1 :=0 to max_Y do
begin
x1 :=p1[i1,j1];
for h1 :=0 to max_Poisson do
begin
x2 :=x1*a1[h1];
for k1 :=0 to max_Poisson do
begin
i2 :=i1-1+h1; if i2>max_X then i2 :=max_X; j2 :=j1+k1;
if j2>max_X then j2 :=max_X; p2[i2,j2] :=p2[i2,j2]+x2*b1[k1];
end;
end;
end;
i1 :=0; { cas i1 = 0 }
for j1 :=0 to max_Y do
begin
x1 :=p1[i1,j1];
for h1 :=0 to max_Poisson do
begin x2 :=x1*a1[h1];
for k1 :=0 to max_Poisson do
begin
i2 :=i1+h1; if i2>max_X then i2 :=max_X; if j1>0 then j2 :=j1-1 else j2 :=0;
j2 :=j2+k1; if j2>max_X then j2 :=max_X; p2[i2,j2] :=p2[i2,j2]+x2*b1[k1];
end;
end;
end;
end;

```

```

end;

end;

p1 :=p2;

end;

writeln(tgf,' Fin de la procédure stationnaire _X_Y'); af_processus_X_Y(p2);

end;

procEDURE transition_vn_wn(var tr1 : t_transition_Y_Y);
{ Cette procédure calcule la matrice de transition qui donne la loi de Y
en w(n) connaissant la valeur de Y en v(n) }
var i1,i2,j1,j2,h1,k1,t1 : integer;
x1,x2,x3 : real;
b1 : t_Poisson;
p1,p2 : t_processus_Y;
begin
bg_transition_alphabeta :=false;
cons_Poisson(beta,b1); x2 :=exp(-alpha); x3 :=1-x2; zero_transition_Y_Y(tr1);
for i1 :=0 to max_Y do
begin
if igfich=1 then write(' i1 = ',i1 :5); { i1 est la valeur de Y en v(n) }
{ L'instruction ci-dessus permet de voir la vitesse d'exécution }
{ initialisation de p1 : }
zero_processus_Y(p1); p1[i1] :=1;
for t1 :=1 to max_wn_moins_vn do
begin
zero_processus_Y(p2);
for j1 :=0 to max_Y do

```

```

begin
x1 :=p1[j1];
for k1 :=0 to max_Poisson do
begin
if j1>0 then j2 :=j1-1 else j2 :=0;
j2 :=j2+k1; if j2>max_Y then j2 :=max_Y;
p2[j2] :=p2[j2]+x1*b1[k1];
end;
end;
for j1 :=0 to max_Y do
begin
tr1[i1,j1] :=tr1[i1,j1]+p2[j1]*x3;
p1[j1] :=p2[j1]*x2;
end;
end;
end;
writeln(tgf);
writeln(tgf,' Fin de la procédure transition_v(n)_w(n) ');
if bg2details then af_transition_Y_Y(tr1); arret;
end;

procedure transition_wn_vn_plus_un(var tr1 : t_transition_Y_Y);
{ Cette procédure calcule la matrice de transition qui donne la loi de Y
en v(n+1) connaissant la valeur de Y en w(n) }
var i1,i2,j1,j2,
h1,k1,t1 : integer;
x1,x2,x3 : real;

```

```

poi1 : t_Poisson_long;
d1 : t_delai_occupation_X;
begin
zero_transition_Y_Y(tr1); busy_stationnaire_X(d1);
{ d1 est la loi du délai entre w(n) et v(n+1); le nombre d'arrivées entre w(n)
et v(n+1) suit une loi de Poisson dont le paramètre est proportionnel au délai }
for i1 :=1 to max_delai_occupation do
begin x1 :=d1[i1]; x2 :=beta*i1; cons_Poisson_long(x2,poi1);
for j2 :=0 to max_Poisson_long do
begin h1 :=j2; if h1>max_Y then h1 :=max_Y;
tr1[0,h1] :=tr1[0,h1]+poi1[j2]*x1;
end;
end;
for j1 :=1 to max_Y do
for j2 :=0 to max_Y do
begin h1 :=j1+j2; if h1>max_Y then h1 :=max_Y;
tr1[j1,h1] :=tr1[j1,h1]+tr1[0,j2];
end;
writeln(tgf); writeln(tgf,' fin de la procédure transition_wn_vn_plus_un ');
if bg2details then af_transition_Y_Y(tr1); arret;
end;

procedure transitions(var tr1,tr3 :t_transition_Y_Y);
{ Cette procédure calcule deux matrices de transition; d'une part la matrice de transition
qui donne la loi de Y en w(n) connaissant la loi de Y en v(n) : ceci se fait par simple
appel de la procédure transition_vn_wn; d'autre part, la matrice de transition qui donne
la loi de Y en v(n+1) connaissant la valeur de Y en v(n); pour cela, on effectue le produit

```

de la matrice de transition de $v(n)$ à $w(n)$ par la matrice de transition de $w(n)$ à $v(n+1)$ }

```
var i1,j1,k1,t1 : integer ;
```

```
x1,s1 : real ;
```

```
tr2 : t_transition_Y_Y ;
```

```
begin
```

```
transition_vn_wn(tr1) ; transition_wn_vn_plus_un(tr2) ;
```

```
if bg1details then begin normal_YY(tr1) ; normal_YY(tr2) ; end ;
```

```
for i1 :=0 to max_Y do
```

```
for j1 :=0 to max_Y do
```

```
begin s1 :=0 ;
```

```
for k1 :=0 to max_Y do s1 :=s1+tr1[i1,k1]*tr2[k1,j1] ;
```

```
tr3[i1,j1] :=s1 ;
```

```
end ;
```

```
normal_YY(tr3) ;
```

```
end ;
```

```
procedure Y_vn_wn_stationnaire(var tr1 : t_transition_Y_Y) ;
```

{ Cette procédure calcule la loi stationnaire de Y en $v(n)$ et en $w(n)$ par itérations successives et en utilisant la matrice de transition de $v(n)$ à $v(n+1)$ }

```
var i1,j1,j2,k1,t1 : integer ;
```

```
x1,s1 : real ;
```

```
p1,p2 : t_processus_Y ;
```

```
tr2 : t_transition_Y_Y ;
```

```
begin
```

```
bg_stationnaire_alpha :=false ;
```

```
transitions(tr1,tr2) ; zero_processus_Y(p1) ; p1[0] :=1 ; j2 :=0 ;
```

```
for t1 :=0 to 1000 do
```

```

begin
if igfich=1 then begin j2 :=t1 mod 50; if j2=0 then write(t1 :5); end;
for i1 :=0 to max_Y do
begin s1 :=0; for j1 :=0 to max_Y do s1 :=s1+p1[j1]*tr2[j1,i1];
p2[i1] :=s1;
end;
p1 :=p2;
if t1=500 then
begin writeln(tgf,' pour t1 = 500 on a : '); af_processus_Y(p2); end;
end;
vn :=p2; writeln(tgf);
writeln(tgf,' Fin du calcul de la loi stationnaire en v(n)'); arret;
if bg1details then
begin
writeln(tgf); writeln(tgf,' pour t1 = 1 000 on a : '); writeln(tgf);
writeln(tgf,' La loi stationnaire de Y en v(n) est : '); af_processus_Y(p2);
end;
zero_processus_Y(p2);
for i1 :=0 to max_Y do
begin s1 :=0; for j1 :=0 to max_Y do s1 :=s1+p1[j1]*tr1[j1,i1];
p2[i1] :=s1;
end;
if bg1details then
begin writeln(tgf,' La loi stationnaire de Y en w(n) est : ');
af_processus_Y(p2);
end;
end;

```

```
wn :=p2;
```

```
end;
```

```
{ FIN de la PSEUDO-SIMULATION }
```

{ Tout ce qui précède relève de la "Pseudo-simulation" c'est-à-dire n'utilise que des propriétés

élémentaires basées sur l'évolution markovienne, notamment des calculs par itérations successives.

Au contraire, les procédures qui suivent ont essentiellement pour but de tester et de vérifier les formules établies dans le chapitre }

```
{ SECTION 2.9 : LOI P-GEOMETRIQUE }
```

{ Dans cette partie, on se focalise sur le caractère P-géométrique de certaines lois.

Cette partie est mise assez tôt pour des raisons techniques mais elle peut être lue après ce qui correspond aux sections 2.4, 2.5, 2.6, 2.7 et 2.8 }

```
procedure calcule_r(var r1 :extended);
```

```
{ Cette procedure calcule la valeur en z de 1 plus [ la fonction génératrice de ywn - yv(n+1) ] * (exp(alpha)-1) }
```

```
var i1,j1 : integer;
```

```
x1,x2,s1,s2 : extended;
```

```
begin
```

```
x1 :=-(alpha+beta)+beta*z; x1 := alpha*exp(x1); s1 :=1;
```

```
for j1 :=1 to max_r do
```

```
begin x2 :=1; for i1 :=1 to j1 do x2 :=x2*x1*(j1+1)/i1;
```

```
x2 :=x2/(j1+1); s1 :=s1+x2;
```

```
end;
```

```
r1 :=s1; if bg_af_r then writeln(tgf,' r1 = ',r1);
```

```

end;

function nfg(x1 :real) :real; { ATTENTION : cette procédure modifie z }
{ RAPPEL : on est en régime stationnaire si ig_gamma_stationnaire vaut 2 ou 3 }
var x2,x3,r1 : extended;

begin

z :=x1 ; x2 :=-alpha+beta*(x1-1) ; x2 :=exp(x2) ; x3 :=(alpha+beta)*(x1-1) ; x3 :=exp(x3) ;

case ig_gamma_stationnaire of

1 : nfg :=x2 ; 2 : nfg :=x3 ; 3 : begin calcule_r(r1) ; nfg :=x2*r1 ; end ; end ;

end ;

procedure calcul_ngammap ;

{ ATTENTION : cette procédure modifie z }

var il : integer ;

x1,x2,x3,y1 : real ;

begin

x1 :=1 ; x2 :=1 ; y1 :=-1 ;

while y1<0 do begin x2 :=x2*2 ; y1 :=nfg(x2)-x2 ; end ; x2 :=x2*3 ;

for il :=1 to 35 do

begin

x3 :=(x1+x2)/2 ; y1 :=nfg(x3)-x3 ; if y1>0 then x2 :=x3 else x1 :=x3 ;

end ;

writeln(tgf,' Calcul de ngammap : ');

if bg2details then

begin x3 :=nfg(x1)-x1 ; y1 :=nfg(x2)-x2 ; writeln(tgf,' x1 = ',x1 :20,' * ',x3 :20) ;

writeln(tgf,' x2 = ',x2 :20,' * ',y1 :20) ;

end ;

ngammap :=(x1+x2)/2 ; ngamma :=1/ngammap ;

```

```
writeln(tgf,' ngamma = ',ngamma,' et ngammap = ',ngammap); arret;
end;
```

```
{ SECTION 2.4 : délai d'attente d'une cellule non prioritaire }
```

```
function special(x1 :real;k1 :integer) :real;
```

```
{ Cette procédure calcule  $(x1^{*k1})$  à la puissance k1 et divise par factorielle(k1) }
```

```
var i1 : integer;
```

```
y1 : real;
```

```
begin
```

```
y1 :=1;
```

```
for i1 :=1 to k1 do y1 :=y1*x1*k1/i1;
```

```
special :=y1;
```

```
end;
```

```
procedure attente_non_prioritaire (var delai1 :t_processus_X);
```

```
{ Cette procédure calcule le délai d'attente d'une cellule non prioritaire
```

```
choisie au hasard suivant la formule de la section 2.4; cette formule
```

```
utilise gamma qui est calculé par P-simulation }
```

```
var i1,k1 : integer;
```

```
s1,x1 : real;
```

```
pr1 : t_Poisson;
```

```
gamma1 : t_processus_X;
```

```
begin
```

```
x1 :=alpha+beta;
```

```
cons_Poisson(x1,pr1);
```

```
stationnaire_X(alpha,pr1,gamma1);
```

```
zero_processus_X(delai1);
```

```

for k1 :=1 to max_X do { A modifier pour max_X très grand }
begin
if igfich=1 then write(k1 :5);
x1 :=exp(-(alpha*k1))*special(alpha,k1)/(k1*alpha);
s1 :=x1*gamma1[1];
for i1 :=2 to k1 do
begin
x1 :=x1*(k1+1-i1)/(alpha*k1);
s1 :=s1+x1*i1*gamma1[i1];
end;
delai1[k1] :=s1;
end;
if igfich=1 then writeln;
if bg1details then
begin
writeln(tgf,' La loi du délai d"attente calculée par les formules est :');
af_processus_X(delai1);
end;
end;

procedure verif_attente_non_prioritaire;
var
i1,n1 : integer;
s1 : real;
d1 : t_delai_occupation_X;
delai1 : t_processus_X;
begin

```

```

writeln(tgf);
writeln(tgf,' Ce qui suit est associé à la section 2.4 '); writeln(tgf);
writeln(tgf,' On vérifie que le délai d"attente d"une cellule non prioritaire ');
writeln(tgf,' a la même loi, que l"on calcule celle-ci par P-simulation pure ');
writeln(tgf,' ou par les formules de la section 2.4 '); arret;
n1 :=max_delai_occupation; if max_delai_occupation>max_X then n1 :=max_X;
attente_stationnaire(d1);
attente_non_prioritaire(delai1);
writeln(tgf,' Les premières valeurs associées aux deux lois sont : '); writeln(tgf);
for i1 :=0 to 3 do
writeln(tgf,' d1['',i1 :2,']= ',d1[i1],', delai1['',i1,']= ',delai1[i1] );
s1 :=0;
for i1 :=0 to n1 do s1 :=s1+abs(d1[i1]-delai1[i1]);
writeln(tgf,' La différence entre les deux lois vaut : ',s1);
lignes(10); arret;
end;

{ SECTION 2.5 : entre w(n) et v(n+1) }

procedure delai_vn_plus_un_moins_wn(var delai2 :t_delai_occupation_X);
{ Calcul de la loi du délai entre w(n) et v(n+1) par la formule
donnée au début de la deuxième partie de la section 2.5 }
var i1,j1 : integer;
x1,x2 : extended;
begin
x1 :=exp(-alpha); x1 :=x1/(1-x1);
delai2[0] :=0;
for j1 :=1 to max_delai_occupation do

```

```

begin
x2 :=x1;
for i1 :=j1 downto 1 do x2 :=x2*alpha*(1+j1)*exp(-alpha)/i1;
delai2[j1] :=x2/(1+j1);
end;
end;

procedure verif_vn_plus_un_moins_wn;
var
i1,n1 : integer;
s1 : real;
d1,d2 : t_delai_occupation_X;
begin
writeln(tgf); writeln(tgf,' Ce qui suit est associé à la section 2.5 '); writeln(tgf);
writeln(tgf,' On vérifie que le délai entre w(n) et v(n+1) ');
writeln(tgf,' a la même loi, que l'on calcule celle-ci par P-simulation pure ');
writeln(tgf,' ou par la formule du milieu de la section 2.5 '); busy_stationnaire_X(d1);
if bg1details then
begin writeln(tgf,' La loi du délai d'occupation du processus X,');
writeln(tgf,' calcule par P-simulation, est : '); af_delai_occupation(d1);
end;
delai_vn_plus_un_moins_wn(d2);
writeln(tgf,' Les premières valeurs associées aux deux lois sont : '); writeln(tgf);
for i1 :=0 to 3 do
writeln(tgf,' d1[',i1 :2,']= ',d1[i1],', d2[',i1,']= ',d2[i1] ); writeln(tgf);
n1 :=max_delai_occupation; if max_delai_occupation>max_X then n1 :=max_X;
s1 :=0; for i1 :=0 to n1 do s1 :=s1+abs(d1[i1]-d2[i1]); writeln(tgf);

```

```

writeln(tgf,' La différence entre les deux lois vaut : ',s1); arret;

end;

procedure augmentation_de_Y(var augmentation3 :t_processus_X);
{ Calcul de la loi de la variation de Y entre w(n) et v(n+1) par la formule de la fin
de la section 2.5; ce calcul n'utilise pas explicitement la procédure précédente qui
donne la loi du délai entre w(n) et v(n+1); il n'utilise pas du tout la P-simulation }
var i1,j1,k1,m1 : integer;
x1,x2,x3,x4,s1 : extended;
begin
x1 :=exp(alpha); x2 :=1/(x1-1); x1 :=alpha*exp(-(alpha+beta)); x4 :=1; m1 :=20*max_X;
for k1 :=0 to max_X do
begin
s1 :=0;
for j1 :=1 to m1 do
begin x3 :=x2*x4;
for i1 :=j1 downto 1 do begin x3 :=x3*(1+j1)*x1/i1; end;
for i1 :=1 to k1 do x3 :=x3*j1;
x3 :=x3/(1+j1); s1 :=s1+x3;
end;
augmentation3[k1] :=s1; x4 :=x4*beta/(k1+1);
end;
if bg1details then
begin
writeln(tgf,' La loi de l'augmentation de Y calculée par les formules est :');
writeln(tgf); af_processus_X(augmentation3);
end;

```

```

end;

procedure verif_moyenne_augmentation_de_Y;
{ Cette procédure effectue plusieurs vérifications plus ou moins complémentaires;
notamment elle vérifie la formule donnant la moyenne de l'augmentation de Y entre
w(n) et v(n+1); elle vérifie aussi la matrice de transition de w(n) ... v(n+1) }
var i1,i2,i3,n1 :integer;
x1,x2,x3,s1 :real;
augmentation1 :t_processus_X;
tr1 :t_transition_Y_Y;
begin
writeln(tgf); writeln(tgf,' Ce qui suit est encore associé à la section 2.5 ');
writeln(tgf);
augmentation_de_Y(augmentation1); transition_wn_vn_plus_un(tr1);
x1 :=0;
for i1 :=0 to max_X do x1 :=x1+i1*augmentation1[i1];
writeln(tgf,'La moyenne de l'augmentation de Y entre w(n) et v(n+1) ');
writeln(tgf,'calculée par les formules de la section 2.5 est : ',x1);
x2 :=(1-exp(-alpha))*(1-alpha);
x3 :=alpha*beta/x2;
writeln(tgf,'La moyenne de l'augmentation de Y entre w(n) et v(n+1) ');
writeln(tgf,'calculée par la formule explicite');
writeln(tgf,'donnée la fin de la section 2.5 est : ',x3); writeln(tgf);
writeln(tgf,' La différence entre ces deux moyennes est : ',(x1-x3));
s1 :=abs(x1-x3);
if s1>1.E-7 then
begin

```

```

writeln(tgf,' Cet écart est grand car max_X est petit ');
writeln(tgf,' Cela permet de contrôler si max_X suffit ou non ');
end;
writeln(tgf);
writeln(tgf,' Voici les premières valeurs de la loi de l'augmentation de Y ');
writeln(tgf,' calculées par les formules et par P-simulation '); writeln(tgf);
for i1 :=0 to 3 do
begin
writeln(tgf,' k = ',i1 :3,' * ',augmentation1[i1],' * * ',tr1[i1,(2*i1)]);
end;
writeln(tgf); s1 :=0; n1 :=round(max_X/3);
for i1 :=0 to n1 do
for i2 :=0 to n1 do
begin
i3 :=i1+i2; if i3>max_X then i3 :=max_X;
x1 :=augmentation1[i1]-tr1[i2,i3];
s1 :=s1+abs(x1);
end;
writeln(tgf,' L'ecart entre la matrice de transition donnée par les formules');
writeln(tgf,' et celle donnée par P-simulation est, en un certain sens : ',s1);
arret; lignes(8); end;

{ SECTION 2.6 : les fonctions f et g puis la formule * }

procedure calcule_f(var f0 :t_delai_occupation_X;var g0chapeau :t_processus_Y);
{ Cette procédure calcule la famille f, qui est liée aux délais, en fonction de la
famille g0chapeau qui est liée à la condition initiale en v(n); ceci utilise
les formules du milieu de la section 2.6 }

```

```

var i1,j1,j2 : integer;

x1,x2,x3,x4,s1 : real;

begin

zero_delai_occupation(f0); f0[0] :=g0chapeau[0];

for j1 :=1 to max_delai do

begin

s1 :=0;x1 :=1;

j2 :=j1; if j2>max_Y then j2 :=max_Y;

for i1 :=0 to j2 do

begin

if i1>0 then begin x1 :=x1/i1; x1 :=x1*j1*beta; end;

s1 :=s1+x1*g0chapeau[j1-i1];;

end;

x1 :=((1/beta)-1);

for i1 :=1 to j1 do s1 :=s1+x1*f0[j1-i1]*special(beta,i1);

f0[j1] :=s1;

end;

if bg1details then

begin

writeln(tgf,' La famille f0 calculée par les formules est : ');

af_delai_occupation(f0);

end;

end;

procedure cons_faux_Poisson(beta1 :real; var pr1 : t_processus_Y);

{ Cette procédure calcule beta1 à la puissance k divisé par factorielle(k)

et met cette famille dans le tableau pr1 }

```

```

var j2 : integer ;

x1,x2 : real ;

begin

zero_processus_Y(pr1) ; x1 :=beta1 ;

x2 :=1 ; pr1[0] :=x2 ;

for j2 :=1 to max_Y do

begin x2 :=x2*x1/j2 ; pr1[j2] :=x2 ; end ;

end ;

procedure verif_etoile(var tr1 : t_transition_Y_Y) ;

{ Cette procédure sert à vérifier la relation (2.1) portant sur s qui est donnée au début
du troisième quart de la section 2.6 ; elle permet aussi de comparer les lois de Y en
w(n) calculées d'une part par les formules et, d'autre part, par P-simulation }

var i1,j1,j2,k1 : integer ;

x1,x2,s0,s1,c1,z1,g0z,e1 : real ;

v0,g0chapeau,w0,s2,faup1 : t_processus_Y ;

g0 : t_delai_occupation_X ;

begin

z1 :=z ; writeln(tgf) ;

writeln(tgf,' i_initial = ',i_initial,' et z = ',z1) ; writeln(tgf) ;

zero_processus_Y(v0) ; v0[i_initial] :=1 ; { v0 est la loi initiale en v(n) }

if bg_transition_alphabeta then transition_vn_wn(tr1) ;

{ La boucle ci-dessous permet d'avoir une procédure qu'il serait facile
d'adapter si la loi initiale n'était pas une mesure de Dirac }

for i1 :=0 to max_Y do

begin

s1 :=0 ; for j1 :=0 to max_Y do s1 :=s1+v0[j1]*tr1[j1,i1] ; w0[i1] :=s1 ;

```

```

end;

{ ici, w0 est le loi de Y en w(n) calcul,e par P-simulation }

s1 :=0; x1 :=1;

for i1 :=0 to max_Y do begin s1 :=s1+x1*w0[i1]; x1 :=x1*z1; end;

c1 :=beta*z1-beta-alpha; c1 :=exp(c1); e1 :=s1*(z1-c1);

{ ci-dessous, on utilise d'abord la définition de g0z=g0(z) donnée au début de la section
2.6 }

x1 :=-1+exp(alpha); g0z :=0; x2 :=1;

for i1 :=0 to max_Y do

begin g0chapeau[i1] :=v0[i1]*x1; g0z :=g0z+x2*g0chapeau[i1]; x2 :=x2*z1; end;

if bg1details then

begin

writeln(tgf,' la famille g0chapeau est : '); af_processus_Y(g0chapeau);

end;

calculer_f(g0,g0chapeau); { ici g0 est égal à la famille f }

s0 :=0;

for j1 :=1 to max_delai do

begin x1 :=j1*(alpha+beta); g0[j1] :=g0[j1]*exp(-x1); s0 :=s0+g0[j1]; end;

if bg1details then

begin

writeln(tgf,' la famille g0 calculée par les formules est : ');

af_delai_occupation(g0);

end;

x1 :=s0+g0chapeau[0]; x1 :=(g0z+(z1-1)*x1)*c1;

writeln(tgf,' Le premier membre de la relation (2.1) vaut : ',e1);

writeln(tgf,' Le deuxième membre de la relation (2.1) vaut : ',x1);

x1 :=abs(e1-x1); x2 :=1.E-7;

```

```

writeln(tgf,' L"ecart entre les deux membres ');
writeln(tgf,' de la relation (2.1) vaut : ',x1);
if x2<x1 then
begin
writeln(tgf,' On remarque que l"ecart entre les deux membres est assez élevé;');
writeln(tgf,' ceci est normal; cet écart diminue quand on augmente les tailles');
writeln(tgf,' des tableaux car on améliore alors l"approximation. ');
end;
arret;
cons_faux_Poisson(beta,faup1);
s2[0] :=s0;
for k1 :=1 to max_Y do
begin
x1 :=s2[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s2[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1];
x2 :=faup1[k1]*(-1+k1/beta)*(g0chapeau[0]+s0);
s2[k1] :=x1-x2;
end;
writeln(tgf,' Les premières valeurs associées à la loi de Y en w(n) calculée');
writeln(tgf,' par les formules et par P-simulation sont respectivement :');
writeln(tgf);k1 :=10; if k1>max_Y then k1 :=max_Y;
for i1 :=0 to k1 do
writeln(tgf,' s2[' ,i1 :2,']= ',s2[i1], ' w0[' ,i1,']= ',w0[i1] ); writeln(tgf);
writeln(tgf,' On constate que les valeurs calculées par les formules deviennent ');
writeln(tgf,' rapidement négatives quand i augmente. Ceci est normal et ');

```

```

writeln(tgf,' est dû à l'instabilite des calculs effectués. De ce fait, ');
writeln(tgf,' la méthode des convexes est extrêmement efficace. ');
writeln(tgf); arret;
suite_geo(w0);
calcul_ngammap; lignes(2);
end;

procedure test_etoile(var tr1 : t_transition_Y_Y);
{ Cette procédure teste verif_etoile pour plusieurs jeux de paramètresCAD 2.1 }
var il : integer;

begin
writeln(tgf);writeln(tgf,' On vérifie maintenant la formule (2.3 *) ');
i_initial :=1; z :=0; verif_etoile(tr1); bg1details :=false;
{ i_initial :=3; z :=1/2; verif_etoile(tr1);
entre accolades pour alléger l'exécution }
i_initial :=5; z :=1/3; verif_etoile(tr1); bg1details :=true;
end;

procedure convexe_etoile(var tr1 : t_transition_Y_Y);
{ Cette procédure permet de calculer, par la méthode des convexes, la loi de Y en w(n)
en partant de la relation (*) (2.1)mais sans utiliser s(0); les explications associées
sont données dans la section 2.7; elle permet aussi de comparer les lois de Y en
w(n) calculées d'une part par ces formules et, d'autre part, par P-simulation.
ATTENTION : cette procédure est longue car elle donne tous les détails ce qui
permet d'effectuer des vérifications. Il y a, juste après cette procédure, une
autre procédure simplifiée beaucoup plus courte. }
var bg1,bg2,bg3,bg4,bg5 : boolean;
i1,j1,j3,k1,k3,n1,n2 : integer;

```

```

x1,x2,x3,x4,x5,x6,
e1,e2,s0,s1,ss2,ss3 : real;
v0,g0chapeau,w0,s2,s3,
faup1,c1,c2,c3,c4 : t_processus_Y;
g0 : t_delai_occupation_X;
begin
e1 :=1.E-11; e2 :=1.E-18; n1 :=1+i_initial; if n1>max_Y then n1 :=max_Y;
writeln(tgf); writeln(tgf,' i_initial = ',i_initial); writeln(tgf);
zero_processus_Y(v0); v0[i_initial] :=1; { v0 est la loi initiale en v(n) }
if bg_transition_alphabeta then transition_vn_wn(tr1);
{ La boucle ci-dessous permet d'avoir une procédure qu'il serait facile
d'adapter si la loi initiale n'était pas une mesure de Dirac }
for i1 :=0 to max_Y do
begin s1 :=0; for j1 :=0 to max_Y do s1 :=s1+v0[j1]*tr1[j1,i1]; w0[i1] :=s1; end;
{ ici, w0 est le loi de Y en w(n) calcul,e par P-simulation }
x1 :=-1+exp(alpha);
for i1 :=0 to max_Y do begin g0chapeau[i1] :=v0[i1]*x1; end;
if bg2details then
begin writeln(tgf,' la famille g0chapeau est : '); af_processus_Y(g0chapeau);
end;
cons_faux_Poisson(beta,faup1);
for i1 :=0 to 1 do
begin
s0 :=i1; s2[0] :=s0; s3[0] :=0;
for k1 :=1 to max_Y do
begin

```

```

x1 :=s2[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s2[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1]; x2 :=faup1[k1]*(-1+k1/beta)*(g0chapeau[0]+s0);
s2[k1] :=x1-x2; if i1=0 then s3[k1] :=s2[k1];
end;
end;
if bg1details then
begin
writeln(tgf,' Les pseudo-probabilités calculées à partir de s0 = 0 sont :');
af_processus_Y(s3);
writeln(tgf,' Les pseudo-probabilités calculées à partir de s0 = 1 sont :');
af_processus_Y(s2); arret;
end;
{ Méthode des convexes : dans une première étape, on utilise s(k) }

zero_processus_Y(c1); zero_processus_Y(c2); bg4 :=true; bg3 :=true; k1 :=0;
while bg3 do
begin
k1 :=1+k1; if k1>=max_Y then bg3 :=false;
x1 :=s3[k1]; x2 :=s2[k1]-s3[k1]; if abs(x2)<e2 then bg3 :=false;
x3 :=-x1/x2; x4 :=(1-x1)/x2;
if x2>=0 then begin c1[k1] :=x3; c2[k1] :=x4; end;
if x2<0 then begin c1[k1] :=x4; c2[k1] :=x3; bg4 :=false; end;
end;
if bg1details then
begin
writeln(tgf,' Les s(k) donnent comme bornes du convexe : ');

```

```

for k1 :=1 to max_Y do
writeln(tgf,k1 :4,' * ',c1[k1],' * * ',c2[k1]);
arret ;
end ;
if bg4 then writeln(tgf,' Les dénominateurs sont toujours positifs ')
else writeln(tgf,' Les dénominateurs ne sont pas toujours positifs ');
bg1 :=true ; bg2 :=true ; bg3 :=true ;
j1 :=n1 ; x1 :=0 ; { x1 est la borne inférieure globale du convexe }
while bg3 do
begin
j1 :=1+j1 ; x3 :=c1[j1] ; x4 :=abs(x3-x1) ; if x4<e1 then bg3 :=false ;
if ((x3<x1) and bg3) then bg1 :=false ;
if ((x3>x1) and bg3) then x1 :=x3 ;
if j1>=max_Y then bg3 :=false ;
end ;
bg3 :=true ; k1 :=n1 ; x2 :=1 ; { x2 est la borne supérieure globale du convexe } ;
while bg3 do
begin
k1 :=1+k1 ; x3 :=c2[k1] ; x4 :=abs(x2-x3) ; if x4<e1 then bg3 :=false ;
if ((x3>x2) and bg3) then bg2 :=false ;
if ((x3<x2) and bg3) then x2 :=x3 ;
if k1>=max_Y then bg3 :=false ;
end ;
if bg1 then writeln(tgf,' La borne inférieure du convexe croît toujours ')
else writeln(tgf,' La borne inférieure du convexe ne croît pas toujours ');
if bg2 then writeln(tgf,' La borne supérieure du convexe décroît toujours ')

```

```

else writeln(tgf, ' La borne supérieure du convexe ne décroît pas toujours ');
if (bg1 and bg2 ) then writeln(tgf, ' Sur le domaine observe, le convexe décroît régulièrement
');
writeln(tgf, ' On s"est arrête aux valeurs : ', j1 :3, ' et ', k1 :3);
x5 :=x1 ; x6 :=x2 ; j3 :=j1 ; k3 :=k1 ; x3 :=abs(x2-x1);
writeln(tgf, ' et le diamètre du convexe partiel vaut alors : ', x3);
arret ;

{ Dans une deuxième étape on utilise les sommes partielles des s(k) }

ss2 :=s2[0] ; ss3 :=s3[0];
for k1 :=1 to max_Y do
begin ss2 :=ss2+s2[k1] ; s2[k1] :=ss2 ; ss3 :=ss3+s3[k1] ; s3[k1] :=ss3 ; end;
if bg2details then
begin
writeln(tgf, ' Les sommes des pseudo-probabilités calculées à partir de s0 = 0 sont :');
af_processus_Y(s3);
writeln(tgf, ' Les sommes des pseudo-probabilités calculées à partir de s0 = 1 sont :');
af_processus_Y(s2);
arret ;
end;
zero_processus_Y(c3) ; zero_processus_Y(c4) ; bg4 :=true ; bg3 :=true ; k1 :=0 ;
while bg3 do
begin
k1 :=1+k1 ; if k1>=max_Y then bg3 :=false;
x1 :=s3[k1] ; x2 :=s2[k1]-s3[k1] ; if abs(x2)<e2 then bg3 :=false;
x3 :=-x1/x2 ; x4 :=(1-x1)/x2 ;
if x2>=0 then begin c3[k1] :=x3 ; c4[k1] :=x4 ; end ;

```

```

if x2<0 then begin c3[k1] :=x4; c4[k1] :=x3; bg4 :=false; end;

end;

if bg1details then

begin

writeln(tgf,' Les sommes des s(k) donnent comme bornes du convexe : ');

for k1 :=1 to max_Y do

writeln(tgf,k1 :4,' * ',c3[k1],' * * ',c4[k1]);

if bg4 then writeln(tgf,' Les dénominateurs sont toujours positifs ')

else writeln(tgf,' Les dénominateurs ne sont pas toujours positifs ');

arret;

end;

bg1 :=true; bg2 :=true; bg3 :=true;

j1 :=n1; x1 :=0; { x1 est la borne inférieure globale du convexe }

while bg3 do

begin

j1 :=1+j1; x3 :=c3[j1]; x4 :=abs(x3-x1); if x4<e1 then bg3 :=false;

if ((x3<x1) and bg3) then bg1 :=false;

if ((x3>x1) and bg3) then x1 :=x3;

if j1>=max_Y then bg3 :=false;

end;

bg3 :=true; k1 :=n1; x2 :=1; { x2 est la borne supérieure globale du convexe };

while bg3 do

begin

k1 :=1+k1;

x3 :=c4[k1]; x4 :=abs(x2-x3); if x4<e1 then bg3 :=false;

if ((x3>x2) and bg3) then bg2 :=false;

```

```

if ((x3<x2) and bg3) then x2 :=x3;
if k1>=max_Y then bg3 :=false;
end;
if bg1 then writeln(tgf,' La borne inférieure du convexe croît toujours ');
else writeln(tgf,' La borne inférieure du convexe ne croît pas toujours ');
if bg2 then writeln(tgf,' La borne supérieure du convexe décroît toujours ');
else writeln(tgf,' La borne supérieure du convexe ne décroît pas toujours ');
if (bg1 and bg2 ) then
writeln(tgf,' Sur le domaine observe, le convexe décroît régulièrement ');
writeln(tgf,' On s'est arrêté aux valeurs : ', j1 :3,' et ',k1 :3);
x3 :=abs(x2-x1);
writeln(tgf,' et le diamètre du convexe partiel vaut alors : ',x3);
x3 :=c4[k1]-c1[j3];
writeln(tgf,' Le diamètre du convexe final vaut : ',x3);
writeln(tgf,' Si cette quantité est négative, il faut augmenter e1 ');
writeln(tgf,' ou modifier le programme ');
arret;
if j3<k1 then begin n2 :=j3-3; s0 :=c1[j3]; end;
if k1<j3 then begin n2 :=k1-3; s0 :=c4[k1]; end;
if k1=j3 then begin n2 :=j3-3; s0 :=(c1[j3]+c4[k1])/2; end;
s2[0] :=s0;
for k1 :=1 to n2 do
begin
x1 :=s2[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s2[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1]; x2 :=faup1[k1]*(-1+k1/beta)*(g0chapeau[0]+s0);

```

```

s2[k1] :=x1-x2;

end;

writeln(tgf,' Après utilisation de la méthode des convexes, ');
writeln(tgf,' les premières valeurs associées à la loi de Y en w(n)');
writeln(tgf,' calculées par les formules et celles associées à la loi de ');
writeln(tgf,' Y en w(n) calculées par P-simulation sont respectivement : ');
writeln(tgf);

for i1 :=0 to n2 do

writeln(tgf,' s2[,i1 :2,']= ',s2[i1],', w0[,i1,']= ',w0[i1] ); arret;

end;

procedure convexe_etoile_simplifiee(var tr1 : t_transition_Y_Y);
{ Cette procédure permet de calculer, par la méthode des convexes, la loi de Y en w(n)
en partant de la relation (*) (2.1) mais sans utiliser s(0); les explications associées
sont données dans la section 2.7; elle permet aussi de comparer les lois de Y en
w(n) calculées d'une part par ces formules et, d'autre part, par P-simulation }
var bg1,bg2,bg3,bg4,bg5 : boolean;
i1,j1,j3,k1,k3,n1,n2 : integer;
x1,x2,x3,x4,x5,x6,
e1,e2,s0,s1,ss2,ss3 : real;
v0,g0chapeau,w0,s2,s3,
faup1,c1,c2 : t_processus_Y;
g0 : t_delai_occupation_X;
begin

writeln(tgf); writeln(tgf,' On va maintenant utiliser la méthode des convexes ');
writeln(tgf,' sous une forme simplifiée ');
e1 :=1.E-12; e2 :=1.E-18;

```

```

writeln(tgf); writeln(tgf,' i_initial = ',i_initial); writeln(tgf);
zero_processus_Y(v0); v0[i_initial] :=1; { v0 est la loi initiale en v(n) }
if bg_transition_alphabeta then transition_vn_wn(tr1);
{ La boucle ci-dessous permet d'avoir une procédure qu'il serait facile
d'adapter si la loi initiale n'était pas une mesure de Dirac }
for i1 :=0 to max_Y do
begin s1 :=0; for j1 :=0 to max_Y do s1 :=s1+v0[j1]*tr1[j1,i1]; w0[i1] :=s1; end;
{ ici, w0 est le loi de Y en w(n) calculée par P-simulation }
x1 :=-1+exp(alpha);
for i1 :=0 to max_Y do begin g0chapeau[i1] :=v0[i1]*x1; end;
cons_faux_Poisson(beta,faup1);
s2[0] :=1; s3[0] :=0; ss2 :=s2[0]; ss3 :=s3[0];
zero_processus_Y(c1); zero_processus_Y(c2); bg3 :=true; bg4 :=true;
k1 :=0;
while bg3 do
begin
k1 :=1+k1; if k1>=max_Y then bg3 :=false;
{ cas s0=1 }
x1 :=s2[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s2[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1]; x2 :=faup1[k1]*(-1+k1/beta)*(g0chapeau[0]+1);
s2[k1] :=x1-x2;
{ cas s0=0 }
x1 :=s3[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s3[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1]; x2 :=faup1[k1]*(-1+k1/beta)*g0chapeau[0];

```

```

s3[k1] :=x1-x2;
ss2 :=ss2+s2[k1]; ss3 :=ss3+s3[k1];

{ Méthode des convexes : dans une première étape, on utilise s(k) }

x1 :=s3[k1]; x2 :=s2[k1]-s3[k1]; if abs(x2)<e2 then bg3 :=false;
x3 :=-x1/x2; x4 :=(1-x1)/x2; if x2>=0 then c1[k1] :=x3;
if x2<0 then begin c1[k1] :=x4; bg4 :=false; end;

{ Dans une deuxième étape, on utilise les sommes des s(k) }

x1 :=ss3; x2 :=ss2-ss3; if abs(x2)<e2 then bg3 :=false;
x3 :=-x1/x2; x4 :=(1-x1)/x2;
if x2>=0 then c2[k1] :=x4;
if x2<0 then begin c2[k1] :=x3; bg4 :=false; end;
x1 :=c2[k1]-c1[k1]; if x1<e1 then bg3 :=false;
if x1<0 then writeln(tgf,' ATTENTION : le convexe est vide ');
end;

n2 :=k1; { fin de boucle sur k1 }

if bg4 then writeln(tgf,' Les denominateurs sont toujours positifs ')
else writeln(tgf,' Les dénominateurs ne sont pas toujours positifs ');
if bg1details then
begin
writeln(tgf,' Les bornes successives du convexe sont : ');
for j1 :=0 to k1 do
writeln(tgf,j1 :3,' * ',c1[j1],' * * * ',c2[j1]);
end;
bg1 :=true; bg2 :=true; x1 :=0; x2 :=1;

{ x1 est la borne inférieure globale du convexe }
{ x2 est la borne supérieure globale du convexe }

```

```

n1 :=1+i_initial; if n1>max_Y then n1 :=max_Y;
for j1 :=n1 to n2 do
begin
if c1[j1]<x1 then bg1 :=false else x1 :=c1[j1];
if c2[j1]>x2 then bg2 :=false else x2 :=c2[j1];
end;
if bg1 then writeln(tgf,' La borne inférieure du convexe croît toujours ');
else writeln(tgf,' La borne inférieure du convexe ne croît pas toujours ');
if bg2 then writeln(tgf,' La borne supérieure du convexe décroît toujours ');
else writeln(tgf,' La borne supérieure du convexe ne décroît pas toujours ');
if (bg1 and bg2 ) then
writeln(tgf,' Sur le domaine observe, le convexe décroît régulièrement ');
writeln(tgf,' Le convexe final est : ',x1,' * * ',x2);
x3 :=x2-x1; writeln(tgf,' et son diamètre vaut : ',x3); arret;
{ Fin de la méthode des convexes }
{ Calcul de la loi de Y en wn à partir de la valeur s0 que l'on vient de déterminer }
s0 :=x2; s2[0] :=s0;
for k1 :=1 to n2 do
begin
x1 :=s2[k1-1]*exp(alpha+beta);
for j1 :=0 to (k1-1) do x1 :=x1-((s2[j1]+g0chapeau[j1])*faup1[k1-j1]);
x1 :=x1-g0chapeau[k1];
x2 :=faup1[k1]*(-1+k1/beta)*(g0chapeau[0]+s0);
s2[k1] :=x1-x2;
end;
writeln(tgf,' Après utilisation de la méthode des convexes simplifiée, ');

```

```

writeln(tgf,' les premières valeurs associées à la loi de Y en w(n)');
writeln(tgf,' calculées par les formules et celles associées à la loi de ');
writeln(tgf,' Y en w(n) calculées par P-simulation sont respectivement : ');
writeln(tgf);
for i1 :=0 to n2 do
begin writeln(tgf,' s2['i1 :2,']= ',s2[i1], ' w0['i1,']= ',w0[i1] );
if k1=(n2-3) then
writeln(tgf,' A partir d"ici les résultats ne sont plus fiables ');
end;
arret;
end;

procedure test_convexe_etoile(var tr1 : t_transition_Y_Y);
{ Cette procédure teste convexe_etoile pour plusieurs jeux de paramètres }
var i1 : integer;
begin
writeln(tgf);writeln(tgf,' On utilise maintenant la méthode des convexes ');
writeln(tgf,' On commence par une procédure très détaillée ');
writeln(tgf);
i_initial :=1; convexe_etoile(tr1); bgl1details :=false;
i_initial :=3; convexe_etoile(tr1);
i_initial :=5; convexe_etoile(tr1); bgl1details :=true;
end;

procedure test_convexe_simplifie(var tr1 : t_transition_Y_Y);
{ Cette procédure teste convexe_etoile_simplifiee pour plusieurs jeux de paramètres }
var i1 : integer;
begin

```

```
writeln(tgf);writeln(tgf,' On utilise maintenant la méthode des convexes '); writeln(tgf);
i_initial :=1; convexe_etoile_simplifiee(tr1); bg1details :=false;
{ i_initial :=3; convexe_etoile_simplifiee(tr1);
entre accolades pour all,ger l'ex,cution }
i_initial :=5; convexe_etoile_simplifiee(tr1); bg1details :=true;
end;
```

```
{ CHAINE INCLUSE EN REGIME STATIONNAIRE :
```

```
La formule ** et le calcul de moyenne associé }
```

```
procedure calcule_s(var s1 :real;var tr1 : t_transition_Y_Y);
```

```
{ Ce calcul de s repose sur la P-simulation }
```

```
var i1,j1,j2 : integer;
```

```
x1,x2,s0,c1,r1 : real;
```

```
begin
```

```
if bg_stationnaire_alpha then Y_vn_wn_stationnaire(tr1); s0 :=0; x1 :=1;
```

```
for i1 :=0 to max_Y do begin s0 :=s0+x1*wn[i1]; x1 :=x1*z; end;
```

```
s1 :=s0; writeln(tgf,' s1 = ',s1);
```

```
end;
```

```
procedure verif_etoile_etoile(var tr1 : t_transition_Y_Y);
```

```
{ Cette procédure sert à vérifier la relation (**) (2.3) portant sur s qui est
```

```
donnée au début du troisième quart de la section 2.6; on est en régime
```

```
stationnaire et on calcule s par pseudo-simulation }
```

```
var s0,s1,c1,x1,x2 : real;
```

```
r1 : extended;
```

```
begin
```

```
writeln(tgf,' z = ',z); writeln(tgf);
```

```

calculer(r1);
{ Cette procédure calculer est placée immédiatement après la P-simulation }
calculer(s1,tr1); writeln(tgf);
c1 :=beta*z-beta-alpha; c1 :=exp(c1); x1 :=c1*r1; x1 :=s1*(z-x1);
writeln(tgf,'le premier membre de la relation (**) (2.3)vaut :',x1); writeln(tgf);
s0 :=(1-beta/(1-alpha))*exp(alpha); x2 :=c1*(z-1)*s0; x1 :=x1-x2;
writeln(tgf,'le second membre de la relation (**)(2.3)vaut :',x2); writeln(tgf);
writeln(tgf,' La différence entre les deux membres vaut : ',x1); writeln(tgf);
writeln(tgf,' Cette différence est assez grande si la charge est forte;');
writeln(tgf,' ceci est du aux erreurs de calcul. '); arret;
end;

procedure test_etoile_etoile(var tr1 : t_transition_Y_Y);
{ Cette procédure teste verif_etoile_etoile
pour plusieurs jeux de paramètres}
begin
writeln(tgf);writeln(tgf,' On vérifie maintenant la formule (**)(2.3) ');
writeln(tgf);
z :=0; verif_etoile_etoile(tr1); bgldetails :=false;
{ z :=1/2; verif_etoile_etoile(tr1); entre accolades pour alléger l'exécution }
z :=1/3; verif_etoile_etoile(tr1); bgldetails :=true;
end;

procedure verif_chaine_incluse_stationnaire(var tr1 : t_transition_Y_Y);
{ Cette procédure permet de vérifier les formules explicites donnant s0 et les moyennes
de Y en w(n) puis en v(n). Ces vérifications se font en utilisant la P_simulation }
var il :integer;
p1,e1,s0,s1,s2,

```

```

x1,x2,x3,x4,e2,e3 :real;

begin

writeln(tgf); writeln(tgf,' Ce qui suit porte sur le régime stationnaire ');
writeln(tgf,' de la chaine incluse. '); writeln(tgf);

if bg_stationnaire_alpha then Y_vn_wn_stationnaire(tr1);

s1 :=0;

for i1 :=1 to max_Y do s1 :=s1+i1*wn[i1];

p1 :=s1;

writeln(tgf,'La moyenne de Y_en_wn calculée par Pseudo_simulation est :',p1);

s1 :=1-(beta+alpha); x1 :=1-alpha;

x2 :=1/x1; x3 :=x2*x2;

e1 :=alpha*beta*(x2+x3)-beta+2*x1;

e2 :=beta*e1/(2*s1);

writeln(tgf,'La moyenne de Y en w(n) en régime stationnaire calculée ');
writeln(tgf,' par la formule globale explicite est : ',e2);

x4 :=e2-p1;

writeln(tgf,'L"écart entre les deux moyennes données ci-dessus vaut :',x4);

x1 :=(1-exp(-alpha))*(1-alpha);

x1 :=alpha*beta/x1; e3 :=e2+x1;

writeln(tgf,'La moyenne de Y en v(n) en régime stationnaire calculée');
writeln(tgf,' par la formule globale explicite est : ',e3);

s1 :=0;

for i1 :=0 to max_Y do s1 :=s1+i1*vn[i1];

writeln(tgf,'La moyenne de Y en v(n) en régime stationnaire calculée');
writeln(tgf,' par P-simulation est : ',s1);

writeln(tgf,' La différence entre ces deux moyennes est : ',(e3-s1));

```

```

s1 :=abs(e3-s1);
if s1>1.E-7 then
begin
writeln(tgf,' Cet ecart est grand car max_X est petit ');
writeln(tgf,' Cela permet de contrôler si max_X suffit ou non ');
end;
arret;

writeln(tgf,' Etude de wn '); suite_geo(wn); arret;
writeln(tgf,' Etude de vn '); suite_geo(vn); arret;
calcul_ngammap;
end;

procedure verifications;
{ Cette procédure regroupe les diverses procédures de vérification où
seuls alpha et beta sont fixés }
var tr1 : t_transition_Y_Y;
begin
writeln(tgf); writeln(tgf,' ATTENTION : on modifie alpha et beta ');
writeln(tgf); writeln(tgf,' alpha = ',alpha,' et beta = ',beta);
bg_transition_alphabeta :=true; bg_stationnaire_alphabeta :=true;
ig_gamma_stationnaire :=1; verif_attente_non_prioritaire; { section 2.4 }
verif_vn_plus_un_moins_wn; { section 2.5 }
verif_moyenne_augmentation_de_Y; { section 2.5 }
transition_vn_wn(tr1); test_etoile(tr1); { section 2.6 }
if bg_convexe_detaille then test_convexe_etoile(tr1)
else test_convexe_simplifie(tr1);
ig_gamma_stationnaire :=2;

```

```

verif_chaine_incluse_stationnaire(tr1); { section 2.7 }
test_etoile_etoile(tr1); { section 2.7 }
end;

procedure go_on;
begin
bg1details :=true; bg2details :=false; bg_af_r :=false;
max_delai :=max_delai_occupation; if max_delai>max_Y then max_delai :=max_Y;
bg_convexe_detaille :=true; alpha :=0.2; beta :=0.3; verifications;
bg_convexe_detaille :=false; bg1details :=false;
{ Les deux lignes qui suivent sont mises entre accolades pour raccourcir l'exécution.
alpha :=0.5; beta :=0.2; verifications;
alpha :=0.75; beta :=0.1; verifications; }
alpha :=0.8; beta :=0.1; verifications;
end;

procedure nomdefichier;
begin
nomfich := 'hafich'; end;

procedure debutprog;
begin case igfich of 0 : begin nomdefichier; assign(tgf,nomfich); end;
1 : assign(tgf,'con');
2 : assign(tgf,'lpt1');
end; { fin du case }
rewrite(tgf); { rewrite : creation d'un fichier;
reset : ouverture d'un fichier existant. }
end;

```

```
procedure finprog ;  
begin writeln(tgf) ; writeln(tgf, ' FIN DU FICHER-PROGRAMME ');  
writeln(tgf) ; close(tgf) ;  
end ;  
  
begin debutprog ; af_debut ; go_on ; finprog ; end.
```

Execution du Programme

MODELISATION ATM

Etude d'une file d'attente avec deux niveaux de priorités.

L'objectif essentiel est de vérifier des formules.

La modélisation repose sur la technique de la chaîne incluse.

$\max_Poisson = 15$ et $\max_Poisson_long = 100$

$\max_r = 150$, $\max_X = 39$ et $\max_Y = 39$

$\max_delai_occupation = 400$ et $nbi_X = 2000$

$nbi_X_Y = 200$ et $\max_wn_moins_vn = 400$

$\alpha = 2.000000000000000E-0001$ et $\beta = 3.000000000000000E-0001$

Ce qui suit est associé à la section 2.4

On vérifie que le délai d'attente d'une cellule non prioritaire a la même loi, que l'on calcule celle-ci par P-simulation pure ou par les formules de la section 2.4

La loi du délai d'attente d'une cellule non prioritaire, calculé par P-simulation, est :

0 0.000E+0000 * 1 2.656E-0001 * 2 1.257E-0001 * 3 5.833E-0002 *
 4 2.701E-0002 * 5 1.253E-0002 * 6 5.819E-0003 * 7 2.709E-0003 *
 8 1.263E-0003 * 9 5.897E-0004 * 10 2.757E-0004 * 11 1.290E-0004 *
 12 6.040E-0005 * 13 2.830E-0005 * 14 1.327E-0005 * 15 6.225E-0006 *
 16 2.921E-0006 * 17 1.371E-0006 * 18 6.438E-0007 * 19 3.024E-0007 *

Pour vérification, la loi résiduelle est :

0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000

0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000
 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000
 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000

La loi du délai d'attente calculée par les formules est :

0.00000E+0000 2.65564E-0001 1.25666E-0001 5.83330E-0002 2.70125E-0002
 1.25250E-0002 5.81917E-0003 2.70880E-0003 1.26304E-0003 5.89738E-0004
 2.75678E-0004 1.28992E-0004 6.04038E-0005 2.83044E-0005 1.32704E-0005
 6.22470E-0006 2.92095E-0006 1.37113E-0006 6.43807E-0007 3.02373E-0007

Les premières valeurs associées aux deux lois sont :

d1[0]= 0.000000000000000E+0000 delai1[0]= 0.000000000000000E+0000
 d1[1]= 2.65564027262826E-0001 delai1[1]= 2.65564027262826E-0001
 d1[2]= 1.25666298849183E-0001 delai1[2]= 1.25666298849183E-0001
 d1[3]= 5.83330386522789E-0002 delai1[3]= 5.83330386522221E-0002

La différence entre les deux lois vaut : 8.27305227428114E-0014

Ce qui suit est associé à la section 2.5

On vérifie que le délai entre $w(n)$ et $v(n+1)$ a la même loi, que l'on calcule celle-ci par P-simulation pure

ou par la formule du milieu de la section 2.5

La loi du délai d'occupation du processus X, calculé par P-simulation, est :

0 0.000E+0000 * 1 7.396E-0001 * 2 1.817E-0001 * 3 5.288E-0002 *
 4 1.691E-0002 * 5 5.742E-0003 * 6 2.032E-0003 * 7 7.415E-0004 *
 8 2.769E-0004 * 9 1.053E-0004 * 10 4.067E-0005 * 11 1.590E-0005 *
 12 6.279E-0006 * 13 2.502E-0006 * 14 1.005E-0006 * 15 4.060E-0007 *
 16 1.651E-0007 * 17 6.746E-0008 * 18 2.769E-0008 * 19 1.142E-0008 *

Les premières valeurs associées aux deux lois sont :

$$d1[0] = 0.0000000000000000E+0000 \quad d2[0] = 0.0000000000000000E+0000$$

$$d1[1] = 7.39584962610024E-0001 \quad d2[1] = 7.39584962610024E-0001$$

$$d1[2] = 1.81656286020825E-0001 \quad d2[2] = 1.81656286020825E-0001$$

$$d1[3] = 5.28809201263130E-0002 \quad d2[3] = 5.28809201262561E-0002$$

La différence entre les deux lois vaut : 7.89531298931776E-0014

Ce qui suit est encore associé à la section 2.5

La loi de l'augmentation de Y calculée par les formules est :

$$6.75930E-0001 \quad 2.52455E-0001 \quad 5.72736E-0002 \quad 1.14171E-0002 \quad 2.30408E-0003$$

$$4.84681E-0004 \quad 1.05520E-0004 \quad 2.35749E-0005 \quad 5.37601E-0006 \quad 1.24658E-0006$$

$$2.93083E-0007 \quad 6.97118E-0008 \quad 1.67461E-0008 \quad 4.05704E-0009 \quad 9.90173E-0010$$

$$2.43232E-0010 \quad 6.00909E-0011 \quad 1.49211E-0011 \quad 3.72191E-0012 \quad 9.32191E-0013$$

La moyenne de l'augmentation de Y entre $w(n)$ et $v(n+1)$

calculée par les formules de la section 5 est : 4.13749167459173E-0001

La moyenne de l'augmentation de Y entre $w(n)$ et $v(n+1)$

calculée par la formule explicite

donnée la fin de la section 5 est : 4.13749167459173E-0001

La différence entre ces deux moyennes est : 0.0000000000000000E+0000

Voici les premières valeurs de la loi de l'augmentation de Y

calculées par les formules et par P-simulation

$$k = 0 * 6.75929853407069E-0001 * * 6.75929853408888E-0001$$

$$k = 1 * 2.52454569489146E-0001 * * 2.52454569490055E-0001$$

$$k = 2 * 5.72736060603347E-0002 * * 5.72736060605052E-0002$$

L'écart entre la matrice de transition donnée par les formules

et celle donnée par P-simulation est, en un certain sens : 4.08946996952779E-0011

On vérifie maintenant la formule (*) (2.1)

$i_initial = 1$ et $z = 0.000000000000000E+0000$

la famille g_0 chapeau est :

0.00000E+0000 2.21403E-0001 0.00000E+0000 0.00000E+0000 0.00000E+0000
 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000
 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000
 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000 0.00000E+0000

La famille f_0 calculée par les formules est :

0 0.000E+0000 * 1 2.214E-0001 * 2 2.878E-0001 * 3 3.841E-0001 *
 4 5.163E-0001 * 5 6.957E-0001 * 6 9.383E-0001 * 7 1.266E+0000 *
 8 1.709E+0000 * 9 2.306E+0000 * 10 3.113E+0000 * 11 4.202E+0000 *
 12 5.672E+0000 * 13 7.657E+0000 * 14 1.034E+0001 * 15 1.395E+0001 *
 16 1.883E+0001 * 17 2.542E+0001 * 18 3.431E+0001 * 19 4.632E+0001 *

la famille g_0 calculée par les formules est :

0 0.000E+0000 * 1 1.343E-0001 * 2 1.059E-0001 * 3 8.571E-0002 *
 4 6.988E-0002 * 5 5.711E-0002 * 6 4.672E-0002 * 7 3.823E-0002 *
 8 3.130E-0002 * 9 2.562E-0002 * 10 2.098E-0002 * 11 1.717E-0002 *
 12 1.406E-0002 * 13 1.151E-0002 * 14 9.425E-0003 * 15 7.716E-0003 *
 16 6.317E-0003 * 17 5.172E-0003 * 18 4.235E-0003 * 19 3.467E-0003 *

Le premier membre de la relation (*) (2.1) vaut : -4.30905009458002E-0001

Le deuxième membre de la relation (*) (2.1) vaut : -4.30731047680183E-0001

L'écart entre les deux membres

de la relation (*) vaut : 1.73961777818477E-0004

On remarque que l'écart entre les deux membres est assez élevé; ceci est normal; cet écart diminue quand on augmente les tailles

des tableaux car on améliore alors l'approximation.

Les premières valeurs associées à la loi de Y en $w(n)$ calculée par les formules et par P-simulation sont respectivement :

$s2[0]= 7.10155440260678E-0001$ $w0[0]= 7.10442254744521E-0001$
 $s2[1]= 2.39290181439173E-0001$ $w0[1]= 2.39476244097659E-0001$
 $s2[2]= 4.32682980497248E-0002$ $w0[2]= 4.34332003699751E-0002$
 $s2[3]= 5.66869683294158E-0003$ $w0[3]= 5.86982462854024E-0003$
 $s2[4]= 4.29600874124247E-0004$ $w0[4]= 6.91317658605861E-0004$
 $s2[5]= -2.65548190067744E-0004$ $w0[5]= 7.74823967808480E-0005$
 $s2[6]= -4.41304397708731E-0004$ $w0[6]= 8.60069059865909E-0006$
 $s2[7]= -5.89155015624954E-0004$ $w0[7]= 9.55746141884789E-0007$
 $s2[8]= -7.73906633575017E-0004$ $w0[8]= 1.06346538652959E-0007$
 $s2[9]= -1.01521509198221E-0003$ $w0[9]= 1.18367634717958E-0008$
 $s2[10]= -1.33161169009455E-0003$ $w0[10]= 1.31741397025067E-0009$

On constate que les valeurs calculées par les formules deviennent rapidement négatives quand i augmente. Ceci est normal et

est dû à l'instabilité des calculs effectués.

De ce fait, la méthode des convexes est extrêmement efficace.

A-t-on une suite géométrique ?

Les rapports successifs valent :

2.96665E+0000 5.51367E+0000 7.39940E+0000 8.49078E+0000 8.92225E+0000
 9.00886E+0000 8.99893E+0000 8.98709E+0000 8.98443E+0000 8.98485E+0000

8.98526E+0000 8.98533E+0000 8.98531E+0000 8.98530E+0000 8.98530E+0000
 8.98530E+0000 8.98530E+0000 8.98530E+0000 8.98530E+0000 8.98530E+0000

Calcul de ngammap :

ngamma = 1.11292895498764E-0001 et ngammap = 8.98529951546516E+0000

i_initial = 5 et z = 3.33333333333485E-0001

Le premier membre de la relation (*) (2.1)vaut : -1.06654772465163E-0001

Le deuxième membre de la relation (*) (2.1)vaut : -1.06526600799498E-0001

L'écart entre les deux membres

de la relation (*) (2.1)vaut : 1.28171665664922E-0004

On remarque que l'écart entre les deux membres est assez élevé ;

ceci est normal ; cet écart diminue quand on augmente les tailles

des tableaux car on améliore alors l'approximation.

Les premières valeurs associées à la loi de Y en w(n) calculée

par les formules et par P-simulation sont respectivement :

s2[0]= 2.39745204735300E-0001 w0[0]= 2.40032019219825E-0001

s2[1]= 1.55527813860317E-0001 w0[1]= 1.55713876518348E-0001

s2[2]= 1.37840109317949E-0001 w0[2]= 1.38005011635641E-0001

s2[3]= 1.68120601555756E-0001 w0[3]= 1.68321729348690E-0001

s2[4]= 2.18766297858565E-0001 w0[4]= 2.19028014638297E-0001

s2[5]= 6.53328888680562E-0002 w0[5]= 6.56759194491769E-0002

s2[6]= 1.10394780183753E-0002 w0[6]= 1.14893830999279E-0002

s2[7]= 9.41694184987796E-0004 w0[7]= 1.53180493785143E-0003

s2[8]= -5.94398727517209E-0004 w0[8]= 1.79614240924719E-0004

s2[9]= -9.95112241474416E-0004 w0[9]= 2.01146719605305E-0005

s2[10]= -1.32937999043570E-0003 w0[10]= 2.23299698917309E-0006

On constate que les valeurs calculées par les formules deviennent rapidement négatives quand i augmente. Ceci est normal et

est dû à l'instabilité des calculs effectués.

De ce fait, la méthode des convexes est extrêmement efficace.

A-t-on une suite géométrique ?

Les rapports successifs valent :

1.54149E+0000 1.12832E+0000 8.19888E-0001 7.68494E-0001 3.33498E+0000

5.71623E+0000 7.50055E+0000 8.52830E+0000 8.92951E+0000 9.00793E+0000

8.99790E+0000 8.98687E+0000 8.98447E+0000 8.98488E+0000 8.98526E+0000

8.98533E+0000 8.98531E+0000 8.98530E+0000 8.98530E+0000 8.98530E+0000

Calcul de ngammap :

$\text{ngamma} = 1.11292895498764E-0001$ et $\text{ngammap} = 8.98529951546516E+0000$

On utilise maintenant la méthode des convexes

On commence par une procédure très détaillée

$i_{\text{initial}} = 1$

Les pseudo-probabilités calculées à partir de $s_0 = 0$ sont :

0.00000E+0000 -2.21403E-0001 -3.65031E-0001 -4.92326E-0001 -6.47584E-0001

-8.49613E-0001 -1.11441E+0000 -1.46171E+0000 -1.91724E+0000 -2.51473E+0000

-3.29842E+0000 -4.32634E+0000 -5.67460E+0000 -7.44304E+0000 -9.76260E+0000

-1.28050E+0001 -1.67956E+0001 -2.20298E+0001 -2.88951E+0001 -3.79001E+0001

Les pseudo-probabilités calculées à partir de $s_0 = 1$ sont :

1.00000E+0000 4.27319E-0001 2.09913E-0001 2.08921E-0001 2.64911E-0001

3.46389E-0001 4.54217E-0001 5.95756E-0001 7.81416E-0001 1.02494E+0000

1.34435E+0000 1.76330E+0000 2.31282E+0000 3.03359E+0000 3.97898E+0000
 5.21899E+0000 6.84544E+0000 8.97876E+0000 1.17769E+0001 1.54471E+0001

Les $s(k)$ donnent comme bornes du convexe :

1 * 3.41291041560908E-0001 * * 1.88278512409670E+0000
 2 * 6.34898920033265E-0001 * * 2.37419821340882E+0000
 3 * 7.02071702331523E-0001 * * 2.12810279261248E+0000
 4 * 7.09684642192769E-0001 * * 1.80558107177239E+0000
 5 * 7.10377470221829E-0001 * * 1.54649675716792E+0000
 6 * 7.10436771807508E-0001 * * 1.34793676265508E+0000
 7 * 7.10441790221921E-0001 * * 1.19647679968330E+0000
 8 * 7.10442215340663E-0001 * * 1.08099735724136E+0000
 9 * 7.10442251403947E-0001 * * 9.92954940496020E-0001
 10 * 7.10442254464397E-0001 * * 9.25831025873777E-0001
 11 * 7.10442254724512E-0001 * * 8.74655490110854E-0001
 12 * 7.10442254747250E-0001 * * 8.35639062793234E-0001
 13 * 7.10442254748159E-0001 * * 8.05892787147968E-0001
 14 * 7.10442254749069E-0001 * * 7.83214111136658E-0001
 15 * 7.10442254748159E-0001 * * 7.65923800518067E-0001
 16 * 7.10442254749978E-0001 * * 7.52741600593254E-0001
 17 * 7.10442254749069E-0001 * * 7.42691440203089E-0001
 18 * 7.10442254749978E-0001 * * 7.35029158913676E-0001
 19 * 7.10442254749069E-0001 * * 7.29187405916491E-0001
 20 * 7.10442254749069E-0001 * * 7.24733630756418E-0001
 21 * 7.10442254748159E-0001 * * 7.21338055288470E-0001
 22 * 7.10442254748159E-0001 * * 7.18749255371222E-0001
 23 * 7.10442254749069E-0001 * * 7.16775544266966E-0001

24 * 7.10442254749069E-0001 * * 7.15270779260209E-0001
 25 * 7.10442254749978E-0001 * * 7.14123540577930E-0001
 26 * 7.10442254749069E-0001 * * 7.13248881354048E-0001
 27 * 7.10442254749069E-0001 * * 7.12582037765969E-0001
 28 * 7.10442254749069E-0001 * * 7.12073633627369E-0001
 29 * 7.10442254749069E-0001 * * 7.11686024337723E-0001
 30 * 7.10442254748159E-0001 * * 7.11390509510238E-0001
 31 * 7.10442254748159E-0001 * * 7.11165207859267E-0001
 32 * 7.10442254747250E-0001 * * 7.10993437004618E-0001
 33 * 7.10442254746340E-0001 * * 7.10862478239505E-0001
 34 * 7.10442254746340E-0001 * * 7.10762634767889E-0001
 35 * 7.10442254745431E-0001 * * 7.10686513717519E-0001
 36 * 7.10442254745431E-0001 * * 7.10628478733270E-0001
 37 * 7.10442254746340E-0001 * * 7.10584232637302E-0001
 38 * 7.10442254744521E-0001 * * 7.10550499239616E-0001
 39 * 7.10442254744521E-0001 * * 7.10524780768537E-0001

Les dénominateurs sont toujours positifs

La borne inférieure du convexe croît toujours

La borne supérieure du convexe ne décroît pas toujours

On s'est arrêté aux valeurs : 13 et 39

et le diamètre du convexe partiel vaut alors : 8.25260212877765E-0005

Les sommes des $s(k)$ donnent comme bornes du convexe :

1 * 1.34287560969142E-0001 * * 7.40818220681831E-0001
 2 * 2.63724111761348E-0001 * * 7.13432048374671E-0001
 3 * 3.68817830902572E-0001 * * 7.10708408409118E-0001
 4 * 4.49872361504731E-0001 * * 7.10464967606640E-0001

5 * 5.11771667615903E-0001 * * 7.10444177122554E-0001
6 * 5.58974003000003E-0001 * * 7.10442417637751E-0001
7 * 5.94962179269714E-0001 * * 7.10442268565203E-0001
8 * 6.22399788770963E-0001 * * 7.10442255919588E-0001
9 * 6.43318336235097E-0001 * * 7.10442254846384E-0001
10 * 6.59266718674189E-0001 * * 7.10442254756344E-0001
11 * 6.71425827405074E-0001 * * 7.10442254749069E-0001
12 * 6.80695979101074E-0001 * * 7.10442254749069E-0001
13 * 6.87763578736849E-0001 * * 7.10442254749069E-0001
14 * 6.93151944130477E-0001 * * 7.10442254748159E-0001
15 * 6.97260054821527E-0001 * * 7.10442254748159E-0001
16 * 7.00392094358904E-0001 * * 7.10442254748159E-0001
17 * 7.02779973458746E-0001 * * 7.10442254749069E-0001
18 * 7.04600501752793E-0001 * * 7.10442254749069E-0001
19 * 7.05988479589905E-0001 * * 7.10442254749069E-0001
20 * 7.07046679280211E-0001 * * 7.10442254749069E-0001
21 * 7.07853454831820E-0001 * * 7.10442254749069E-0001
22 * 7.08468543643903E-0001 * * 7.10442254749069E-0001
23 * 7.08937489741402E-0001 * * 7.10442254748159E-0001
24 * 7.09295016066790E-0001 * * 7.10442254749069E-0001
25 * 7.09567595525186E-0001 * * 7.10442254749069E-0001
26 * 7.09775411160990E-0001 * * 7.10442254749069E-0001
27 * 7.09933850609559E-0001 * * 7.10442254749069E-0001
28 * 7.10054645459422E-0001 * * 7.10442254749069E-0001
29 * 7.10146739922493E-0001 * * 7.10442254749069E-0001
30 * 7.10216953097188E-0001 * * 7.10442254748159E-0001

31 * 7.10270483895329E-0001 * * 7.10442254748159E-0001

32 * 7.10311295983047E-0001 * * 7.10442254748159E-0001

33 * 7.10342411277452E-0001 * * 7.10442254748159E-0001

34 * 7.10366133697789E-0001 * * 7.10442254747250E-0001

35 * 7.10384219763000E-0001 * * 7.10442254746340E-0001

36 * 7.10398008649463E-0001 * * 7.10442254746340E-0001

37 * 7.10408521350473E-0001 * * 7.10442254746340E-0001

38 * 7.10416536274352E-0001 * * 7.10442254746340E-0001

39 * 7.10422646882762E-0001 * * 7.10442254745431E-0001

Les dénominateurs sont toujours positifs

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

On s'est arrêté aux valeurs : 39 et 11

et le diamètre du convexe partiel vaut alors : 1.96078735825722E-0005

Le diamètre du convexe final vaut : 9.09494701772928E-0013

Si cette quantité est négative, il faut augmenter e1 ou modifier le programme

Après utilisation de la méthode des convexes, les premières valeurs associées à la loi de Y en $w(n)$ calculées par les formules et celles associées à la loi de Y en $w(n)$ calculées par P-simulation sont respectivement :

$s_2[0] = 7.10442254749069E-0001$ $w_0[0] = 7.10442254744521E-0001$

$s_2[1] = 2.39476244098569E-0001$ $w_0[1] = 2.39476244097659E-0001$

$s_2[2] = 4.34332003696909E-0002$ $w_0[2] = 4.34332003699751E-0002$

$s_2[3] = 5.86982462772312E-0003$ $w_0[3] = 5.86982462854024E-0003$

$s_2[4] = 6.91317657508961E-0004$ $w_0[4] = 6.91317658605861E-0004$

$s_2[5] = 7.74823953402226E-0005$ $w_0[5] = 7.74823967808480E-0005$

$$s2[6] = 8.60068870818520E-0006 \quad w0[6] = 8.60069059865909E-0006$$

$$s2[7] = 9.55743662217276E-0007 \quad w0[7] = 9.55746141884789E-0007$$

$$s2[8] = 1.06343286196712E-0007 \quad w0[8] = 1.06346538652959E-0007$$

$$i_initial = 3$$

Les dénominateurs sont toujours positifs

La borne inférieure du convexe croît toujours

La borne supérieure du convexe ne décroît pas toujours

On s'est arrêté aux valeurs : 14 et 39

et le diamètre du convexe partiel vaut alors : 8.25260271994921E-0005

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

On s'est arrêté aux valeurs : 39 et 13

et le diamètre du convexe partiel vaut alors : 1.96078635781305E-0005

Le diamètre du convexe final vaut : 0.00000000000000E+0000

Si cette quantité est négative, il faut augmenter $e1$ ou modifier le programme

Après utilisation de la méthode des convexes,

les premières valeurs associées à la loi de Y en $w(n)$ calculées par les formules et celles associées à la loi de Y en $w(n)$ calculées par P-simulation sont respectivement :

$$s2[0] = 4.12951436546336E-0001 \quad w0[0] = 4.12951436547701E-0001$$

$$s2[1] = 2.67890380653625E-0001 \quad w0[1] = 2.67890380650897E-0001$$

$$s2[2] = 2.37424023639505E-0001 \quad w0[2] = 2.37424023639051E-0001$$

$$s2[3] = 6.81781909233905E-0002 \quad w0[3] = 6.81781909245274E-0002$$

$$s2[4] = 1.17846796068761E-0002 \quad w0[4] = 1.17846796083398E-0002$$

$$s2[5] = 1.56491312569074E-0003 \quad w0[5] = 1.56491312765894E-0003$$

$$s2[6] = 1.83289111450158E-0004 \quad w0[6] = 1.83289114035423E-0004$$

$s2[7] = 2.05230152788560E-0005$ $w0[7] = 2.05230186702265E-0005$
 $s2[8] = 2.27842908265621E-0006$ $w0[8] = 2.27843353111865E-0006$
 $s2[9] = 2.53220031143885E-0007$ $w0[9] = 2.53225865953517E-0007$
 $s2[10] = 2.81697923372611E-0008$ $w0[10] = 2.81774455114709E-0008$

$i_initial = 5$

Les dénominateurs sont toujours positifs

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

On s'est arrêté aux valeurs : 16 et 39

et le diamètre du convexe partiel vaut alors : $8.25260256078764E-0005$

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

On s'est arrêté aux valeurs : 39 et 14

et le diamètre du convexe partiel vaut alors : $1.96078708540881E-0005$

Le diamètre du convexe final vaut : $4.54747350886464E-0013$

Si cette quantité est négative, il faut augmenter $e1$

ou modifier le programme

Après utilisation de la méthode des convexes,

les premières valeurs associées à la loi de Y en $w(n)$

calculées par les formules et celles associées à la loi de

Y en $w(n)$ calculées par P-simulation sont respectivement :

$s2[0] = 2.40032019219370E-0001$ $w0[0] = 2.40032019219825E-0001$

$s2[1] = 1.55713876516757E-0001$ $w0[1] = 1.55713876518348E-0001$

$s2[2] = 1.38005011635187E-0001$ $w0[2] = 1.38005011635641E-0001$
 $s2[3] = 1.68321729347326E-0001$ $w0[3] = 1.68321729348690E-0001$
 $s2[4] = 2.19028014636933E-0001$ $w0[4] = 2.19028014638297E-0001$
 $s2[5] = 6.56759194471306E-0002$ $w0[5] = 6.56759194491769E-0002$
 $s2[6] = 1.14893830964462E-0002$ $w0[6] = 1.14893830999279E-0002$
 $s2[7] = 1.53180493329153E-0003$ $w0[7] = 1.53180493785143E-0003$
 $s2[8] = 1.79614234938397E-0004$ $w0[8] = 1.79614240924719E-0004$
 $s2[9] = 2.01146641068961E-0005$ $w0[9] = 2.01146719605305E-0005$
 $s2[10] = 2.23298668780222E-0006$ $w0[10] = 2.23299698917309E-0006$
 $s2[11] = 2.48155087834937E-0007$ $w0[11] = 2.48168599564433E-0007$

Ce qui suit porte sur le regime stationnaire

de la chaine incluse.

pour $t1 = 500$ on a :

$4.48822E-0001$ $3.40646E-0001$ $1.44311E-0001$ $4.71555E-0002$ $1.37423E-0002$
 $3.84532E-0003$ $1.06702E-0003$ $2.96708E-0004$ $8.28400E-0005$ $2.32124E-0005$
 $6.52295E-0006$ $1.83712E-0006$ $5.18315E-0007$ $1.46439E-0007$ $4.14198E-0008$
 $1.17263E-0008$ $3.32231E-0009$ $9.41887E-0010$ $2.67177E-0010$ $7.58288E-0011$

Fin du calcul de la loi stationnaire en $v(n)$

pour $t1 = 1\ 000$ on a :

La loi stationnaire de Y en $v(n)$ est :

$4.48822E-0001$ $3.40646E-0001$ $1.44311E-0001$ $4.71555E-0002$ $1.37423E-0002$
 $3.84532E-0003$ $1.06702E-0003$ $2.96708E-0004$ $8.28400E-0005$ $2.32124E-0005$
 $6.52295E-0006$ $1.83712E-0006$ $5.18315E-0007$ $1.46439E-0007$ $4.14198E-0008$
 $1.17263E-0008$ $3.32231E-0009$ $9.41887E-0010$ $2.67177E-0010$ $7.58288E-0011$

La loi stationnaire de Y en $w(n)$ est :

6.64006E-0001 2.55965E-0001 6.16354E-0002 1.38391E-0002 3.35261E-0003

8.74401E-0004 2.36878E-0004 6.52829E-0005 1.81513E-0005 5.07418E-0006

1.42376E-0006 4.00577E-0007 1.12934E-0007 3.18900E-0008 9.01633E-0009

2.55180E-0009 7.22813E-0010 2.04888E-0010 5.81177E-0011 1.65007E-0011

La moyenne de $Y_{\text{en_}w_n}$ calculee par Pseudo_simulation est : 4.40625000023374E-0001

La moyenne de Y en $w(n)$ en regime stationnaire calculee

par la formule globale explicite est : 4.40625000000182E-0001

L'ecart entre les deux moyennes donnees ci-dessus vaut :-2.31921148952097E-0011

La moyenne de Y en $v(n)$ en regime stationnaire calculee

par la formule globale explicite est : 8.54374167459355E-0001

La moyenne de Y en $v(n)$ en regime stationnaire calculee

par P-simulation est : 8.54374167500282E-0001

La difference entre ces deux moyennes est : -4.09272615797818E-0011

Etude de w_n

A-t-on une suite géométrique ?

Les rapports successifs valent :

2.59413E+0000 4.15289E+0000 4.45371E+0000 4.12786E+0000 3.83418E+0000

3.69135E+0000 3.62849E+0000 3.59660E+0000 3.57719E+0000 3.56392E+0000

3.55428E+0000 3.54700E+0000 3.54136E+0000 3.53691E+0000 3.53332E+0000

3.53037E+0000 3.52785E+0000 3.52540E+0000 3.52214E+0000 3.51515E+0000

Etude de v_n

A-t-on une suite géométrique ?

Les rapports successifs valent :

1.31756E+0000 2.36050E+0000 3.06032E+0000 3.43142E+0000 3.57376E+0000
 3.60380E+0000 3.59619E+0000 3.58170E+0000 3.56878E+0000 3.55857E+0000
 3.55064E+0000 3.54441E+0000 3.53946E+0000 3.53548E+0000 3.53223E+0000
 3.52955E+0000 3.52730E+0000 3.52534E+0000 3.52342E+0000 3.52087E+0000

Calcul de ngammap :

ngamma = 2.84668137043354E-0001 et ngammap = 3.51286241722119E+0000

On verifie maintenant la formule (**)

$z = 0.0000000000000000E+0000$

$s1 = 6.64006357210383E-0001$

le premier membre de la relation (**) vaut :-4.63011387939787E-0001

le second membre de la relation (**) vaut :-4.63011387926599E-0001

La difference entre les deux membres vaut : -1.31876731757075E-0011

Cette difference est assez grande si la charge est forte ;

ceci est du aux erreurs de calcul.

$z = 3.33333333333485E-0001$

$s1 = 7.56734234287251E-0001$

le premier membre de la relation (**) vaut :-3.41137813792102E-0001

le second membre de la relation (**) vaut :-3.41137813782098E-0001

La difference entre les deux membres vaut : -1.00044417195022E-0011

Cette difference est assez grande si la charge est forte ;

ceci est du aux erreurs de calcul.

ATTENTION : on modifie alpha et beta

$\alpha = 8.000000000000000E-0001$ et $\beta = 1.000000000000000E-0001$

Ce qui suit est associé à la section 2.4

On vérifie que le délai d'attente d'une cellule non prioritaire a la même loi, que l'on calcule celle-ci par P-simulation pure ou par les formules de la section 2.4

Les premières valeurs associées aux deux lois sont :

$d1[0] = 0.000000000000000E+0000$ $delai1[0] = 0.000000000000000E+0000$

$d1[1] = 6.56031804978738E-0002$ $delai1[1] = 6.56031804978738E-0002$

$d1[2] = 5.13790396915965E-0002$ $delai1[2] = 5.13790396915965E-0002$

$d1[3] = 4.31394577515221E-0002$ $delai1[3] = 4.31394577515221E-0002$

La différence entre les deux lois vaut : $2.01794136955868E-0012$

Ce qui suit est associé à la section 2.5

On vérifie que le délai entre $w(n)$ et $v(n+1)$ a la même loi, que l'on calcule celle-ci par P-simulation pure

ou par la formule du milieu de la section 2.5

Les premières valeurs associées aux deux lois sont :

$d1[0] = 0.000000000000000E+0000$ $d2[0] = 0.000000000000000E+0000$

$d1[1] = 2.93309805439094E-0001$ $d2[1] = 2.93309805439094E-0001$

$d1[2] = 1.58151109252231E-0001$ $d2[2] = 1.58151109252003E-0001$

$d1[3] = 1.01065776489691E-0001$ $d2[3] = 1.01065776489577E-0001$

La différence entre les deux lois vaut : $2.36255459640233E-0012$

Ce qui suit est encore associé à la section 2.5

La moyenne de l'augmentation de Y entre $w(n)$ et $v(n+1)$

calculée par les formules de la section 5 est : 7.26325995376101E-0001

La moyenne de l'augmentation de Y entre $w(n)$ et $v(n+1)$

calculée par la formule explicite

donnée la fin de la section 2.5 est : 7.26386488366188E-0001

La différence entre ces deux moyennes est : -6.04929900873685E-0005

Cet écart est grand car \max_X est petit

Cela permet de contrôler si \max_X suffit ou non

Voici les premières valeurs de la loi de l'augmentation de Y

calculées par les formules et par P-simulation

$k = 0 * 6.44585812165133E-0001 * * 6.44585812166952E-0001$

$k = 1 * 2.03523736689249E-0001 * * 2.03523736689249E-0001$

$k = 2 * 7.24765963859682E-0002 * * 7.24765963864229E-0002$

$k = 3 * 3.28763152527358E-0002 * * 3.28763152533611E-0002$

L'écart entre la matrice de transition donnée par les formules

et celle donnée par P-simulation est, en un certain sens : 5.64352120591138E-0010

On vérifie maintenant la formule (*) (2.1)

$i_initial = 1$ et $z = 0.00000000000000E+0000$

Le premier membre de la relation (*) (2.1) vaut : -3.67061087256388E-0001

Le deuxième membre de la relation (*) (2.1) vaut : -3.67061087255934E-0001

L'écart entre les deux membres

de la relation (*) (2.1) vaut : 4.54747350886464E-0013

Les premières valeurs associées à la loi de Y en $w(n)$ calculée

par les formules et par P-simulation sont respectivement :

$s2[0] = 9.02824592199067E-0001$ $w0[0] = 9.02824592199977E-0001$

$s2[1] = 9.22246551117496E-0002$ $w0[1] = 9.22246551116359E-0002$
 $s2[2] = 4.77703105798355E-0003$ $w0[2] = 4.77703105710248E-0003$
 $s2[3] = 1.68946467482556E-0004$ $w0[3] = 1.68946465386011E-0004$
 $s2[4] = 4.66309236690776E-0006$ $w0[4] = 4.66308742333443E-0006$
 $s2[5] = 1.09684010287520E-0007$ $w0[5] = 1.09672356043435E-0007$
 $s2[6] = 2.38404811094429E-0009$ $w0[6] = 2.35657379176837E-0009$
 $s2[7] = 1.13622669173686E-0010$ $w0[7] = 4.88532841170045E-0011$
 $s2[8] = 1.53700976682126E-0010$ $w0[8] = 1.01026893850718E-0012$
 $s2[9] = 3.59982068484651E-0010$ $w0[9] = 2.10744004913357E-0014$
 $s2[10] = 8.48591160133096E-0010$ $w0[10] = 4.42488303127113E-0016$

On constate que les valeurs calculées par les formules deviennent rapidement négatives quand i augmente. Ceci est normal et

est dû à l'instabilité des calculs effectués. De ce fait,

la méthode des convexes est extrêmement efficace.

A-t-on une suite géométrique ?

Les rapports successifs valent :

$9.78941E+0000$ $1.93059E+0001$ $2.82754E+0001$ $3.62306E+0001$ $4.25183E+0001$
 $4.65389E+0001$ $4.82378E+0001$ $4.83567E+0001$ $4.79382E+0001$ $4.76270E+0001$
 $4.75452E+0001$ $4.75805E+0001$ $4.76260E+0001$ $4.76453E+0001$ $4.76634E+0001$
 $4.76293E+0001$ $4.76352E+0001$ $4.76372E+0001$ $4.76379E+0001$ $4.76381E+0001$

Calcul de ngammap :

$\text{ngamma} = 2.09925892546039E-0002$ et $\text{ngammap} = 4.76358579625958E+0001$

$i_initial = 5$ et $z = 3.333333333333485E-0001$

Le premier membre de la relation (*) (2.1) vaut : $-6.07130268925715E-0003$

Le deuxième membre de la relation (*) (2.1) vaut : $-6.07130268925715E-0003$

L'écart entre les deux membres

de la relation (*) vaut : 0.00000000000000E+0000

Les premières valeurs associées à la loi de Y en $w(n)$ calculée

par les formules et par P-simulation sont respectivement :

$$s2[0] = 2.92302172242103E-0002 \quad w0[0] = 2.92302172241818E-0002$$

$$s2[1] = 4.26645160002863E-0002 \quad w0[1] = 4.26645160002295E-0002$$

$$s2[2] = 9.77483029677160E-0002 \quad w0[2] = 9.77483029673749E-0002$$

$$s2[3] = 2.30287726126789E-0001 \quad w0[3] = 2.30287726127472E-0001$$

$$s2[4] = 5.42886911059213E-0001 \quad w0[4] = 5.42886911061942E-0001$$

$$s2[5] = 5.42886862748446E-0002 \quad w0[5] = 5.42886862768910E-0002$$

$$s2[6] = 2.79243106644600E-0003 \quad w0[6] = 2.79243107148375E-0003$$

$$s2[7] = 9.84320286564477E-0005 \quad w0[7] = 9.84320405303940E-0005$$

$$s2[8] = 2.71207330039200E-0006 \quad w0[8] = 2.71210129243124E-0006$$

$$s2[9] = 6.36655041259988E-0008 \quad w0[9] = 6.37314940148993E-0008$$

$$s2[10] = 1.21341641581318E-0009 \quad w0[10] = 1.36898440291985E-0009$$

On constate que les valeurs calculées par les formules deviennent

rapidement négatives quand i augmente. Ceci est normal et

est dû à l'instabilité des calculs effectués. De ce fait,

la méthode des convexes est extrêmement efficace.

A-t-on une suite géométrique ?

Les rapports successifs valent :

$$6.85118E-0001 \quad 4.36473E-0001 \quad 4.24462E-0001 \quad 4.24191E-0001 \quad 1.00000E+0001$$

$$1.94414E+0001 \quad 2.83691E+0001 \quad 3.62936E+0001 \quad 4.25551E+0001 \quad 4.65538E+0001$$

$$4.82388E+0001 \quad 4.83530E+0001 \quad 4.79354E+0001 \quad 4.76263E+0001 \quad 4.75455E+0001$$

$$4.75809E+0001 \quad 4.76262E+0001 \quad 4.76455E+0001 \quad 4.76631E+0001 \quad 4.76294E+0001$$

Calcul de ngammap :

$ngamma = 2.09925892546039E-0002$ et $ngammap = 4.76358579625958E+0001$

On utilise maintenant la méthode des convexes

On va maintenant utiliser la méthode des convexes

sous une forme simplifiée

$i_initial = 1$

Les denominateurs sont toujours positifs

Les bornes successives du convexe sont :

$0 * 0.000000000000000E+0000 * * * 0.000000000000000E+0000$

$1 * 8.39639843956320E-0001 * * * 9.04837418033821E-0001$

$2 * 9.01396090103844E-0001 * * * 9.02854525174007E-0001$

$3 * 9.02803147972918E-0001 * * * 9.02824941206745E-0001$

$4 * 9.02824341126689E-0001 * * * 9.02824595672428E-0001$

$5 * 9.02824589693410E-0001 * * * 9.02824592229081E-0001$

$6 * 9.02824592175421E-0001 * * * 9.02824592199067E-0001$

$7 * 9.02824592198158E-0001 * * * 9.02824592198158E-0001$

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

Le convexe final est : $9.02824592198158E-0001 * * 9.02824592198158E-0001$

et son diamètre vaut : $0.000000000000000E+0000$

Après utilisation de la méthode des convexes simplifiée,

les premières valeurs associées à la loi de Y en $w(n)$

calculées par les formules et celles associées à la loi de

Y en $w(n)$ calculées par P-simulation sont respectivement :

$s2[0] = 9.02824592198158E-0001$ $w0[0] = 9.02824592199977E-0001$

$s_2[1] = 9.22246551090211E-0002$ $w_0[1] = 9.22246551116359E-0002$
 $s_2[2] = 4.77703105150340E-0003$ $w_0[2] = 4.77703105710248E-0003$
 $s_2[3] = 1.68946452227203E-0004$ $w_0[3] = 1.68946465386011E-0004$
 $s_2[4] = 4.66305640345333E-0006$ $w_0[4] = 4.66308742333443E-0006$
 $s_2[5] = 1.09599228192980E-0007$ $w_0[5] = 1.09672356043435E-0007$
 $s_2[6] = 2.18417840433006E-0009$ $w_0[6] = 2.35657379176837E-0009$
 $s_2[7] = -3.57560543890259E-0010$ $w_0[7] = 4.88532841170045E-0011$

On va maintenant utiliser la méthode des convexes

sous une forme simplifiée

$i_initial = 5$

Les denominateurs sont toujours positifs

La borne inférieure du convexe croît toujours

La borne supérieure du convexe décroît toujours

Sur le domaine observe, le convexe décroît régulièrement

Le convexe final est : $2.92302172237271E-0002$ * * $2.92302172241534E-0002$

et son diamètre vaut : $4.26325641456060E-0013$

Après utilisation de la méthode des convexes simplifiée,

les premières valeurs associées à la loi de Y en $w(n)$

calculées par les formules et celles associées à la loi de

Y en $w(n)$ calculées par P-simulation sont respectivement :

$s_2[0] = 2.92302172241534E-0002$ $w_0[0] = 2.92302172241818E-0002$
 $s_2[1] = 4.26645160002295E-0002$ $w_0[1] = 4.26645160002295E-0002$
 $s_2[2] = 9.77483029677160E-0002$ $w_0[2] = 9.77483029673749E-0002$
 $s_2[3] = 2.30287726126789E-0001$ $w_0[3] = 2.30287726127472E-0001$
 $s_2[4] = 5.42886911059213E-0001$ $w_0[4] = 5.42886911061942E-0001$
 $s_2[5] = 5.42886862748446E-0002$ $w_0[5] = 5.42886862768910E-0002$

$s2[6]= 2.79243106644600E-0003$ $w0[6]= 2.79243107148375E-0003$
 $s2[7]= 9.84320286564477E-0005$ $w0[7]= 9.84320405303940E-0005$
 $s2[8]= 2.71207330039200E-0006$ $w0[8]= 2.71210129243124E-0006$
 $s2[9]= 6.36655041259988E-0008$ $w0[9]= 6.37314940148993E-0008$
 $s2[10]= 1.21341641581318E-0009$ $w0[10]= 1.36898440291985E-0009$

Ce qui suit porte sur le regime stationnaire
de la chaine incluse.

Fin de la procedure transition_v(n)_w(n)

fin de la procedure transition_wn_vn_plus_un

le défaut de transition vaut : 6.91215973347425E-0011

le défaut de transition vaut : 1.55459119923762E-0005

le défaut de transition vaut : 4.82032191939652E-0011

pour $t1 = 500$ on a :

$4.00744E-0001$ $2.20610E-0001$ $1.15975E-0001$ $7.10525E-0002$ $4.80940E-0002$
 $3.43072E-0002$ $2.52222E-0002$ $1.89044E-0002$ $1.43595E-0002$ $1.10141E-0002$
 $8.51105E-0003$ $6.61523E-0003$ $5.16586E-0003$ $4.04959E-0003$ $3.18475E-0003$
 $2.51143E-0003$ $1.98509E-0003$ $1.57222E-0003$ $1.24741E-0003$ $9.91239E-0004$

Fin du calcul de la loi stationnaire en $v(n)$

pour $t1 = 1\ 000$ on a :

La loi stationnaire de Y en $v(n)$ est :

$4.00744E-0001$ $2.20610E-0001$ $1.15975E-0001$ $7.10525E-0002$ $4.80940E-0002$
 $3.43072E-0002$ $2.52222E-0002$ $1.89044E-0002$ $1.43595E-0002$ $1.10141E-0002$
 $8.51105E-0003$ $6.61523E-0003$ $5.16586E-0003$ $4.04959E-0003$ $3.18475E-0003$
 $2.51143E-0003$ $1.98509E-0003$ $1.57222E-0003$ $1.24741E-0003$ $9.91239E-0004$

La loi stationnaire de Y en $w(n)$ est :

$6.21707E-0001$ $1.45951E-0001$ $6.39343E-0002$ $4.19228E-0002$ $2.99871E-0002$

2.21328E-0002 1.66423E-0002 1.26739E-0002 9.74146E-0003 7.54048E-0003
 5.86920E-0003 4.58879E-0003 3.60091E-0003 2.83439E-0003 2.23687E-0003
 1.76926E-0003 1.40211E-0003 1.11303E-0003 8.84871E-0004 7.04407E-0004

La moyenne de $Y_{\text{en_wn}}$ calculee par Pseudo_simulation est : 1.34789426616771E+0000

La moyenne de Y en $w(n)$ en regime stationnaire calculee

par la formule globale explicite est : 1.350000000000036E+0000

L'ecart entre les deux moyennes donnees ci-dessus vaut : 2.10573383265000E-0003

La moyenne de Y en $v(n)$ en regime stationnaire calculee

par la formule globale explicite est : 2.07638648836655E+0000

La moyenne de Y en $v(n)$ en regime stationnaire calculee

par P-simulation est : 2.07422777007741E+0000

La difference entre ces deux moyennes est : 2.15871828913805E-0003

Cet ecart est grand car \max_X est petit

Cela permet de controler si \max_X suffit ou non

Etude de w_n

A-t-on une suite géométrique ?

Les rapports successifs valent :

4.25970E+0000 2.28283E+0000 1.52505E+0000 1.39803E+0000 1.35487E+0000
 1.32991E+0000 1.31312E+0000 1.30102E+0000 1.29189E+0000 1.28475E+0000
 1.27903E+0000 1.27434E+0000 1.27043E+0000 1.26713E+0000 1.26430E+0000
 1.26185E+0000 1.25972E+0000 1.25785E+0000 1.25619E+0000 1.25473E+0000

Etude de v_n

A-t-on une suite géométrique ?

Les rapports successifs valent :

1.81652E+0000 1.90222E+0000 1.63224E+0000 1.47737E+0000 1.40186E+0000
 1.36020E+0000 1.33420E+0000 1.31651E+0000 1.30373E+0000 1.29410E+0000

1.28658E+0000 1.28057E+0000 1.27565E+0000 1.27156E+0000 1.26810E+0000
 1.26515E+0000 1.26260E+0000 1.26038E+0000 1.25844E+0000 1.25672E+0000

Calcul de ngammap :

ngamma = 8.12900548966354E-0001 et ngammap = 1.23016278100840E+0000

On verifie maintenant la formule (**)(2.3)

$z = 0.000000000000000E+0000$

$s1 = 6.21707416950812E-0001$

le premier membre de la relation (**)(2.3) vaut $-4.52445078010896E-0001$

le second membre de la relation (**)(2.3) vaut $-4.52418709018275E-0001$

La difference entre les deux membres vaut : $-2.63689926214283E-0005$

Cette difference est assez grande si la charge est forte ;

ceci est du aux erreurs de calcul.

$z = 3.33333333333485E-0001$

$s1 = 6.79506195909198E-0001$

le premier membre de la relation (**)(2.3) vaut $-3.11853789877659E-0001$

le second membre de la relation (**)(2.3) vaut $-3.11835661676923E-0001$

La difference entre les deux membres vaut : $-1.81282007361006E-0005$

Cette difference est assez grande si la charge est forte ;

ceci est du aux erreurs de calcul.