



HAL
open science

Étude de performances de FIP, aide au dimensionnement d'applications

Ye-Qiong Song

► **To cite this version:**

Ye-Qiong Song. Étude de performances de FIP, aide au dimensionnement d'applications. Sciences de l'ingénieur [physics]. Institut National Polytechnique de Lorraine, 1991. Français. <NNT : 1991INPL066N>. <tel-01751013>

HAL Id: tel-01751013

<https://hal.univ-lorraine.fr/tel-01751013v1>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

[M] 1991 SONG, Y. Q

ETUDE DE PERFORMANCES DE FIP
AIDE AU DIMENSIONNEMENT D'APPLICATIONS

THESE

soutenue publiquement le 11 Juillet 1991
pour l'obtention du diplôme de
Docteur en Informatique de l'Institut National Polytechnique de Lorraine

par

Ye-qiong SONG



Membres du jury:

MM.

G. Pujolle	(président et rapporteur)	MASI-Université de Paris VI
R. Schott	(rapporteur)	CRIN-Université de Nancy I
J.P. Thomesse	(directeur)	CRIN-INPL
F. Simonot	(co-directeur)	ESSTIN-Université de Nancy I
J.P. Lobert	(examinateur)	EDF
J.F. Aubry	(examinateur)	CRAN-INPL

A mes parents

献给我的父母

Remerciements:

Le travail présenté ici a été réalisé au Centre de Recherche en Informatique de Nancy (C.R.I.N.) au sein de l'équipe "Informatique industrielle".

Je tiens à exprimer ici ma profonde gratitude à Monsieur le Professeur Jean-Pierre THOMESSE, mon directeur de recherche, pour l'accueil dans son équipe, pour avoir dirigé ces travaux, pour ses précieux conseils et ses encouragements durant tout mon travail de recherche.

Que Monsieur François SIMONOT, Maître de Conférences à l'ESSTIN, co-directeur de cette recherche, trouve ici mes remerciements pour avoir dirigé et suivi de près mes travaux. Sans ses conseils et ses aides concrètes tant au niveau théorique qu'au niveau technique, ce travail n'aurait pu être ce qu'il est.

Monsieur Guy PUJOLLE, Professeur à l'Université de Paris 6, qui m'a accueilli lors de mon DEA, me fait aujourd'hui l'honneur de présider ce jury et d'être rapporteur de mes travaux. Je l'en remercie vivement.

Je remercie Monsieur René SCHOTT, Professeur à l'Université de Nancy 1, pour avoir accepté d'être rapporteur de ces travaux et pour ses commentaires qui ont conduit à l'amélioration de cette thèse.

Je remercie également Monsieur Jean-Pierre LOBERT, chef de projet au centre d'études et de recherche à EDF, et Monsieur Jean-François AUBRY, Professeur à l'ENSEM, pour l'honneur qu'ils me font de bien vouloir juger ce travail.

Je tiens à remercier profondément Madame Françoise SIMONOT, Maître de Conférences à l'ENSEM, avec qui je partage un bureau depuis que je suis arrivé à Nancy, pour ses conseils, ses encouragements, et surtout ses aides innombrables et ses amitiés qui ont facilité énormément ce travail.

Enfin mes remerciements vont aussi à MM. Jean-Claude DERNIAME, Michel DUFNER, Pascal LORENZ, Zoubir MAMMERI, Benoît MARCHAL, Fabrice MOITESSIER, Gilles NOBLOT et Mme. Marie-Claude CARDOT, Micheline DALBOURG et tous mes collègues et amis de l'ENSEM qui par leurs conseils et leur amitié m'ont permis d'accomplir ce travail de façon efficace et agréable.

SOMMAIRE

- 0. Introduction générale**
- 1. Généralités et contexte de travail**
 - 1.1 Réseaux locaux industriels et bus de terrain
 - 1.2 Bus de terrain dans un système automatisé
 - 1.3 Présentation générale de FIP
 - 1.4 Méthodologie et critères de performances
 - 1.5 Normalisation
- 2. Description et modélisation**
 - 2.1 Présentation de la couche liaison de données de FIP
 - 2.2 Présentation des problèmes: une première approche
 - 2.3 Modélisation des problèmes complexes
 - 2.4 Modélisation générale
 - 2.5 Conclusion
- 3. Etude analytique du modèle, premier étage**
 - 3.1 Capacité infinie
 - 3.2 Capacité limitée à L
 - 3.3 Application à FIP (services périodiques)
- 4. Etude analytique du modèle, deuxième étage**
 - 4.1 Flux d'entrée du deuxième étage
 - 4.2 Deuxième étage
 - 4.3 Application à FIP (services apériodiques)
 - 4.4 Conclusion
- 5. Performances des services apériodiques obtenues par "SimulFIP"**
 - 5.1 Description de la configuration à simuler
 - 5.2 Service de messagerie apériodique
 - 5.3 Services de demande explicite apériodique
 - 5.4 Conclusion
- 6. Conclusion générale et perspectives**
- 7. Bibliographie**
- 8. Annexes**
 - A1. Définitions
 - A2. Simulateur: "SimulFIP"
 - A3. Techniques d'accès au médium à temps critique
- 9. Table des matières**

INTRODUCTION GENERALE

Introduction générale

Le travail que nous présentons dans ce mémoire est une contribution à la modélisation et l'évaluation des performances d'un bus de terrain: FIP (Flux Information Process ou Factory Instrumentation Protocol).

FIP est un bus de terrain qui assure la communication entre capteurs, actionneurs, régulateurs, et automates d'un système automatisé de production. A ce titre il doit acheminer toutes les informations liées à l'acquisition de données et à la commande du processus. Ces informations sont en général courtes (quelques octets), elles ont une durée de validité assez faible (quelques dizaines de millisecondes), ces caractéristiques conduisent à définir des protocoles spécifiques. Par ailleurs, pour des raisons de configuration, de maintenance du réseau, de compatibilité avec d'autres réseaux (MAP par exemple), il est nécessaire d'offrir des services plus traditionnels (point à point, messagerie).

Tout ceci a conduit les concepteurs de FIP à définir des protocoles particuliers qui peuvent gérer divers trafics selon plusieurs critères:

- 1) trafic d'objets identifiés cycliques ou non, en multipoint, sans acquittement.
- 2) trafic de messages traditionnels point à point avec ou sans acquittement.
- 3) trafic de messages en diffusion sans acquittement.

Certains de ces trafics sont dits à "**temps critique**", d'autres sont moins sensibles aux contraintes de temps, selon l'utilisation qui est faite des informations transmises.

Le problème primordial est de savoir avant installation, si le réseau répondra aux besoins et, satisfera les contraintes de temps. Il est donc important de pouvoir évaluer les performances de tels systèmes pour aider au dimensionnement et à la configuration du réseau et de ses stations. La coexistence des divers trafics et leur interdépendance conduit à étudier les services séparément afin de percevoir comment les trafics déterministes et aléatoires cohabitent.

Puisque la couche liaison de données, et en particulier la sous-couche MAC ("Medium Access Control") est une des principales responsables des performances d'un protocole de communication, l'étude se porte naturellement sur la modélisation de cette couche en utilisant deux moyens:

- la théorie des files d'attente
- la simulation à événements discrets.

En ce qui concerne les critères de performances, notre étude se porte principalement sur les trois plus importants:

- le **rendement** du protocole
- le **taux d'utilisation** du médium (throughput)
- le **temps de réponse** (ou délai) d'un service de la couche liaison de données.

Ce mémoire est organisé de la façon suivante:

Dans le chapitre 1: **Généralités et contexte de travail**, pour mieux situer le problème, nous commençons par introduire la position d'un bus de terrain dans un système automatisé et sa relation avec les réseaux locaux industriels. Nous présentons ensuite le protocole FIP d'une façon générale. Les critères de performances et la méthodologie pour étudier ce genre de réseaux sont décrits. A la fin de ce chapitre, nous précisons la situation actuelle de FIP dans le cadre de la normalisation sur le plan national et international.

Dans le chapitre 2: **Description et modélisation**, la couche liaison de données, l'objet à étudier, est particulièrement détaillée en présentant tous les services offerts par cette couche ainsi qu'une analyse explicite de chaque service. Nous classons les problèmes en deux catégories: problèmes simples pour tous ceux qui concernent le **trafic déterministe** et problèmes complexes pour tous ceux qui concernent le **trafic aléatoire**. Les solutions des problèmes simples tels que: construction de la table de scrutation d'un réseau FIP au niveau de la couche liaison de données, "rendement", "taux d'utilisation" ou encore "fiabilité" sont données. Quant aux problèmes complexes, ils sont modélisés et formalisés par un modèle de files d'attente "parallèle-série" en deux étages appelé **modèle général**. On notera que ce modèle représente de façon exacte les **services périodiques** et de façon approximative les **services apériodiques**.

Dans ce mémoire, nous nous intéressons particulièrement à l'analyse de ce modèle général. L'étude analytique de ce modèle est développée dans les chapitres 3 et 4.

Dans le chapitre 3: **Etude analytique du modèle, premier étage**, une étude complète du premier étage du modèle général est effectuée en utilisant la théorie des files d'attente. Les résultats obtenus concernent des paramètres tels que le nombre de trames dans une file d'attente, le délai pour qu'une trame traverse cet étage, etc. Ces résultats sont ensuite appliqués aux *services périodiques* de FIP.

Dans le chapitre 4: **Etude analytique du modèle, deuxième étage**, nous étudions d'abord le flux d'entrée de cet étage. Nous avons constaté que le flux d'entrée du deuxième étage n'est pas un processus indépendant. Cela est vraiment gênant pour l'étude de cet étage. Afin de surmonter cet obstacle, nous avons développé une méthode d'approximation. Il s'agit de remplacer un processus (ou une suite des variables aléatoires) par un autre processus (ou une suite des variables aléatoires) en évaluant la *distance en variation* entre ces deux processus (ou suite des variables aléatoires). Cette substitution est justifiée par la théorie de *robustesse quantitative* et la simulation. En utilisant cette approximation, l'étude du deuxième étage devient analogue à celle du premier étage. Les résultats ainsi obtenus sont utilisés pour évaluer les performances des *services aperiodiques* offerts par la couche liaison de données de FIP.

Dans le chapitre 5: **Performances des services aperiodiques obtenues par "SimulFIP"**, nous avons effectué la simulation de certaines configurations de FIP en utilisant le simulateur "SimulFIP" qui décrit de façon exacte tous les services offerts par la couche liaison de données de FIP. Les résultats ainsi obtenus montrent d'une part les performances de ces services, et peuvent être utilisés pour effectuer une comparaison avec certains résultats issus de notre étude analytique. La modélisation analytique menée (modèle approximatif, méthodes d'approximation...) est ainsi justifiée.

Dans l'annexe 1: **Définitions**, pour faciliter la lecture de ce mémoire, les termes spécifiques (écrits en **gras** dans ce texte) sont énumérés.

Dans l'annexe 2: **Simulateur "SimulFIP"**, nous décrivons le simulateur développé. La table de scrutation joue un rôle particulier dans FIP. Différentes configurations d'un réseau FIP manifestent des performances très variées. Afin d'évaluer les performances d'un réseau FIP de configuration quelconque, un simulateur décrivant la couche liaison de données de FIP en QNAP2 avec le configurateur en langage C a été construit. Ce logiciel permet de simuler un réseau et fournit les paramètres concernant les performances et le dimensionnement du réseau à installer.

Dans l'annexe 3: **Techniques d'accès au médium à temps critique**, ingénierie des protocoles d'accès à temps critique est discutée. Un des buts principaux d'évaluation de performances est de pouvoir réaliser la comparaison entre deux ou plusieurs systèmes ou protocoles de communication. Il est évident qu'une telle comparaison s'effectue selon

divers critères en particulier technologiques et économiques. Nous nous limitons la comparaison des trois critères "académiques", i.e: le rendement, l'utilisation et le délai. Nous établissons une brève comparaison qualitative entre deux techniques d'accès à médium: FIP et Jeton temporisé. La technique d'accès Jeton temporisé est connu sous le nom "Timed Token" utilisé par la norme IEEE802.4 et FDDI. C'est un protocole susceptible d'être utilisé dans des applications temps critique.

Dans ce travail, il y a essentiellement deux points importants:

- 1) Du point de vue théorique, la technique d'approximation que nous avons développée n'est pas habituellement utilisée lorsqu'on évalue les performances d'un réseau de files d'attente. Cela nous offre une nouvelle possibilité pour effectuer l'évaluation des performances des systèmes complexes.
- 2) Du point de vue pratique, le logiciel "SimulFIP" réalisé permet le dimensionnement correct d'un réseau FIP de configuration quelconque. Un débouché industriel de ce logiciel est tout à fait envisageable.

CHAPITRE 1

GENERALITES ET CONTEXTE DE TRAVAIL

1. Généralités et contexte de travail

Le but de ce chapitre est d'introduire le concept de bus de terrain (besoin de communication, architectures d'application, trafics, modèle "client-serveur" et "producteur-distributeur-consommateurs" ...), de présenter les services du bus de terrain FIP (couche liaison et application). La méthodologie d'évaluation de performances d'un tel protocole est aussi abordée.

1.1 Réseaux locaux industriels et bus de terrain

Les réseaux locaux industriels, dits RLI, ont très vite atteint des exigences, en matière de performances, sans commune mesure avec celles que peuvent satisfaire des réseaux télématiques traditionnels, car les caractéristiques des échanges ne sont pas identiques.

Pour tout ce qui concerne les télécommunications, il faut bien constater que nous vivons actuellement dans un monde de normes OSI/ISO adaptés à des besoins bureautiques et informatiques grand public. Ces besoins sont issus de contraintes fondamentalement différentes de celles nées de l'utilisation de réseaux locaux pour les systèmes à hautes performances en *temps réel* ou même à *temps critique* qui sont apparus depuis quelques années dans le domaine industriel. Il devient évident que les normes existantes (basées sur le modèle client-serveur) de OSI/ISO ne peuvent pas satisfaire tous les besoins de communications dans un système automatisé où les trafics à temps critique et non critique coexistent.

Face à l'augmentation de complexité du câblage et aux besoins d'échanges d'informations au niveau capteurs-actionneurs, dans les domaines de contrôle de processus, de l'automatisation, du manufacturier, des systèmes d'embarqués, etc... émerge le concept de bus de terrain (ou en anglais "field bus"). L'introduction d'un bus de terrain permet de remplacer les connexions directes (point-à-point) entre les capteurs ou les actionneurs, intelligents ou non, et des équipements de contrôle (automate programmable) par un réseau unique, ouvert et normalisé.

Les bus de terrains peuvent être considérés comme une catégorie spéciale de RLI. Ils diffèrent du réseau normalisé MAP et d'autres RLI existants (LAC, FACTOR, ...) [Lepage 90] en terme des contraintes de temps. Signalons que les bus de terrains ne reposent actuellement sur aucune norme internationale. Les travaux de la normalisation sont en cours au sein des grandes instances internationales telles que IEC (International Electro-technique Commity) et ISA (Instrumentation Society of America). Ces bus de terrains nécessitent donc une modélisation qualitative permettant la validation/vérification et une modélisation quantitative permettant l'évaluation des performances.

1.2 Bus de terrain dans un système automatisé

1.2.1 Architecture traditionnelle

Une application CIM (Computer Integrated Manufacturing) rassemble des fonctions aussi diverses que commande automatique, contrôle, suivi de production, maintenance, conduite opérateur...

Ces applications sont couramment décrites par des architectures hiérarchisées à 3 ou 5 niveaux [Decotigne 88] de noms différents selon les domaines d'application.

Voici une architecture à trois niveaux: usine - cellule - instrumentation (cf. fig.1.1)

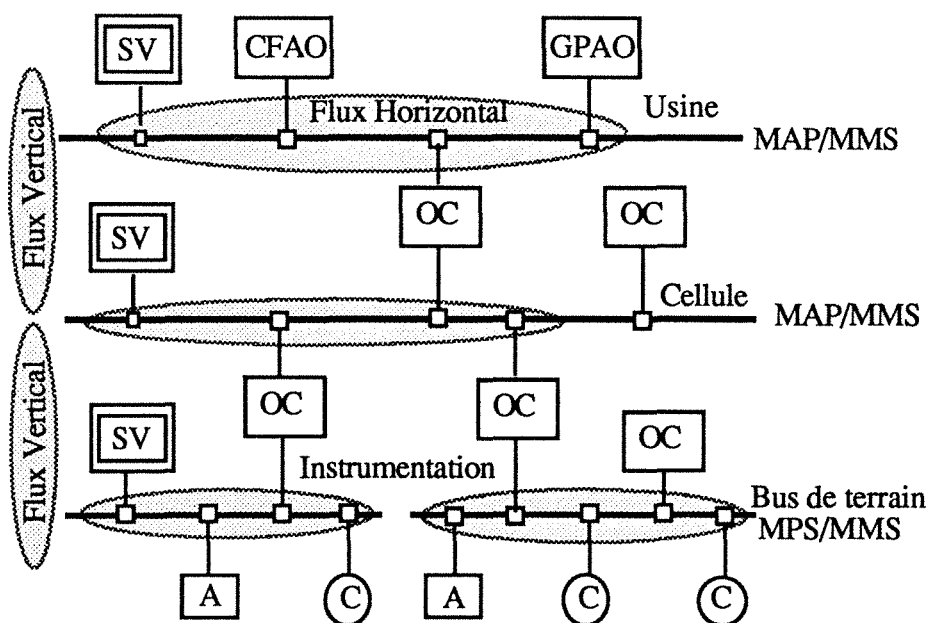


Fig.1.1 Architecture générale CIM

Le niveau usine correspond à la supervision des activités de l'ensemble de l'usine, s'y connectent les postes de conception assistée, des consoles d'interrogation, des ordinateurs de gestion ... Le trafic horizontal est très similaire à celui d'un réseau bureautique. Un temps de réponse de 2 à 5 secondes est d'ordinaire considéré comme suffisant.

Le niveau cellule correspond à un atelier ou une cellule flexible. Le réseau permet d'y assurer la supervision et la coordination des éléments d'automatisation reflex. OC: organe de contrôle qui pourrait être un AP, CN, commande de Robot, ... Il sert aussi au téléchargement des programmes dans ces divers équipements. Le trafic horizontal de ce niveau est moyen avec des pointes momentanées importantes. Les fonctions de coordination impliquent des temps de réponse plus courts de l'ordre de 0,5 à 2 secondes. Un couplage lâche est considéré comme suffisant.

Le niveau instrumentation correspond aux échanges d'informations d'une application d'automatisme centralisée ou répartie mettant en oeuvre un ou plusieurs contrôleurs ainsi que des capteurs et actionneurs. Le trafic horizontal est constitué d'une part aléatoire et d'une part déterministe. La fonction d'échange de données d'échantillonnage implique des temps de réponse très courts (temps critique) et parfois synchronisés de l'ordre de quelques millisecondes à 0,5 secondes. Un couplage serré est nécessaire.

Le niveau instrumentation est typiquement temps critique. Certains échanges doivent être exécutés à certains instants fixes (ni trop tôt ni trop tard). Ce qui est vraiment crucial comme exigence. Il est raisonnable de tolérer un certain degré de déplacement ("jitter") de ces instants. Cela nous permet de définir le terme **temps critique** [ISO 90]: *Le terme temps critique ("time-critical") est utilisé pour représenter la présence d'une fenêtre temporelle ("time-window"), durant laquelle une ou plusieurs actions spécifiées requises doivent être complétées avec un certain niveau de sûreté. Faute de compléter ces actions spécifiées durant la fenêtre temporelle risque de conduire à la faute des applications qui ont demandé ces actions, accompagné des risques pour les équipement, l'usine et voire la vie de l'être humain.* Il n'y a pas de définition de temps absolu. Bien entendu, si cette fenêtre est très large, on aura un couplage lâche et dans ce cas là, un réseau à temps non critique peut très bien être utilisé pour satisfaire totalement la demande.

On pourra donc définir des termes tels que *transaction à temps critique, transmission à temps critique et communication à temps critique.* Par exemple, la transmission est dite à temps critique quand le temps de réponse d'une donnée doit être

bornée par une fenêtre à temps limité. Au delà de ce temps, cette donnée ne sera plus valide.

1.2.2 Structuration fonctionnelle

D'un point de vue fonctionnel, on peut les représenter comme suit [Delcuvelierie 89], [Verlinde 89], [Thomesse 90] selon un modèle à trois axes, qui a été étudié et validé sur des applications concrètes [Verlinde 89-1], [Delcuvelierie 89].

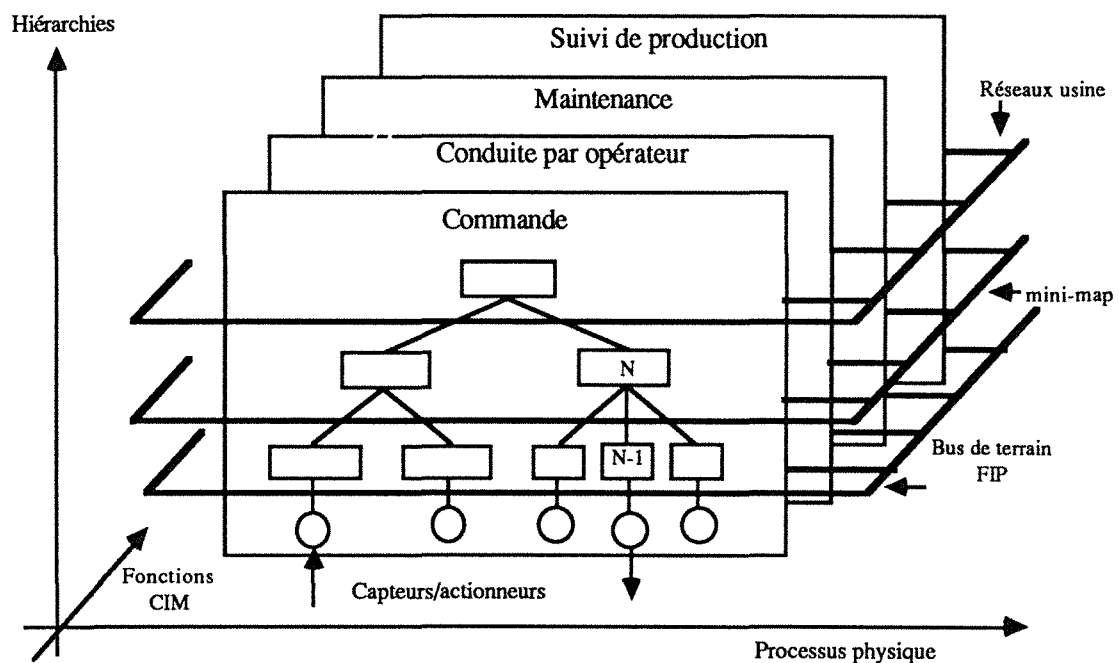


Fig. 1.2 Modèle fonctionnel

Le premier axe est l'axe hiérarchique selon lequel on peut décomposer une fonction en un certain nombre de niveaux hiérarchisés (le nombre de niveaux dépend de l'application) ; à un niveau donné, une entité offre des services accessibles du niveau supérieur ou du niveau inférieur et en général, une entité s'appuie sur les niveaux inférieurs pour interpréter les requêtes qui lui viennent du niveau supérieur.

Le second axe représente le processus physique, soit équipement par équipement, comme c'est souvent le cas dans le manufacturier, soit par ensembles de matériels ou d'équipements qui ont une signification particulièrement homogène comme dans les processus continus (par exemple : transport de fluide, gestion d'énergie...).

Le troisième axe représente les fonctions que l'on trouve dans une architecture CIM (cf. Fig. 1.1), contrôle, commande, conduite par les opérateurs, suivi de production, maintenance, etc.

Par rapport aux modèles traditionnels, on distingue donc les niveaux hiérarchiques et les fonctions, qui sont souvent confondues. Une fonction est hiérarchisée mais les fonctions ne sont pas hiérarchisées entre elles. Cette scrutation fonctionnelle est indépendante de la façon dont les entités seront implantées. Elle permet toutefois de mettre en évidence, de façon modulaire, les fonctions et les entités qui les composent. Construire une architecture opérationnelle revient à affecter les entités à des équipements réels reliés par des systèmes de communication qui devront assurer les communications entre entités définies au niveau fonctionnel.

Le rôle du bus de terrain est d'irriguer les différentes fonctions à leur premier niveau hiérarchique.

1.2.3 Types de trafics

La structure précédente permet de mettre en évidence des échanges d'information verticaux dans un plan (une fonction) ou horizontaux interplans. Les "chemins" ne dépendent pas de la sémantique des informations échangées qui sont, soit de nature "état" (information échantillonnée, variables d'état d'un processus donné ...), soit de nature événement (changement d'état).

1.2.3.1 Trafics vertical et horizontal

Trafic vertical

Le trafic vertical représente les échanges entre deux niveaux adjacents, à l'intérieur d'une même fonction. Le trafic est alors constitué de requêtes et de comptes-rendus, voire d'indications venant du niveau N-1 pour solliciter une entité de niveau N, et des réponses associées. Les requêtes, comptes-rendus, indications et réponses peuvent avoir des formes variées : consigne pour un régulateur, valeur d'une commande pour un actionneur, commande d'activation de tâche, commutation de modes de marche, demande de consignes, etc. Le trafic vertical total, à un niveau donné, est alors constitué de la somme des trafics associés à chaque fonction.

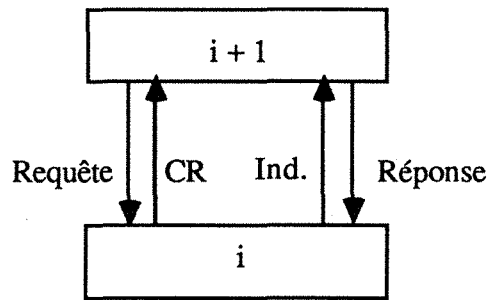


Fig. 1.3 Trafic vertical.

Le couple (requête, compte-rendu) s'applique dans le cas où le niveau supérieur est l'initiateur des services du niveau inférieur.

Le couple (indication, réponse) s'applique dans le cas où c'est le niveau inférieur qui est l'initiateur des services du niveau supérieur.

Le réalisation du service repose la plupart du temps sur l'application du modèle client-serveur. Associé à ce point de vue, on peut introduire le concept de temps de réponse d'une entité d'un niveau à une demande de service. Ce temps de réponse dépendra de la localisation de chacune des entités (délais de communication) et du délai de réaction de l'entité sollicitée, qui lui-même dépendra de la charge instantanée du système support et de son système d'exploitation. Ce temps de réponse entre entités d'une même fonction devrait être maîtrisé.

Trafic horizontal

Le trafic horizontal représente des échanges d'informations hors requêtes, comptes-rendus, etc... entre entités de même niveau dans un même plan ou dans des plans différents.

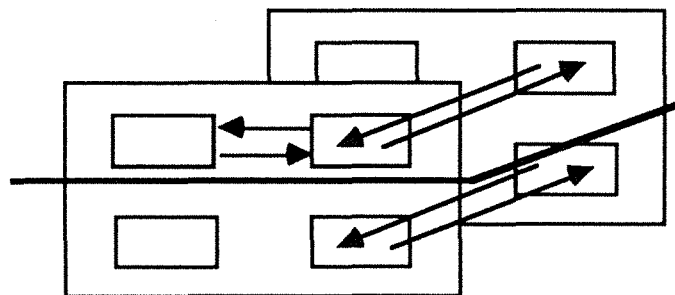


Fig. 1.4 Trafic horizontal.

Le concept de temps de réponse peut être remplacé par celui de la durée de validité des données. La durée de validité d'une information est le délai après production pendant lequel l'information a un sens pour les utilisateurs. Une même information peut ainsi avoir différentes durées de validité selon les utilisateurs.

[Exemple]

La valeur d'une pression échantillonnée toutes les 20 ms a une durée de validité égale à la période d'échantillonnage pour un régulateur, mais peut avoir une durée supérieure si elle est datée (ou au moins numérotée) pour un historique.

La durée de validité d'une information peut être tributaire de facteurs comme :

- l'instant de production,
- les délais d'acheminement,
- la fenêtre temporelle de consommation.

La connaissance de la date de production, qui implique un mécanisme de datation avec une certaine précision, peut être nécessaire pour vérifier la cohérence d'un ensemble de variables ou de données, i.e., savoir si toutes les données ont été produites dans une même fenêtre temporelle.

1.2.3.2 Communication d'état et d'événement

Cette dualité ne date pas d'aujourd'hui ; la distinction est déjà appliquée dans la conception d'applications [Galara 84].

Communication d'état

La communication d'état est basée sur la transmission d'informations d'états (calculés par leur producteur) au moment où le consommateur en a besoin. Les transmissions sont "pilotées" par l'usage des données. S'il y a événement, c'est le récepteur des états successifs qui le détectera.

Communication d'événement

La communication d'événement est basée sur la transmission d'informations, quand le producteur de l'événement (le détecteur) l'a effectivement détecté. Ces transmissions sont pilotées par les producteurs plutôt que par les consommateurs.

Liens avec trafics verticaux et horizontaux

En première approximation, on peut considérer que le trafic vertical relève d'une communication d'événements, alors que le trafic horizontal peut contenir aussi bien des communications d'état que d'événements. A la communication d'état, on peut associer la durée de validité des états considérés pour déterminer les contraintes de temps, alors qu'on associera plutôt un temps de réponse à un événement. Mais ce n'est pas toujours aussi simple, comme on le verra dans le paragraphe suivant.

1.2.3.3 Remarques sur les trafics

Une requête, un compte-rendu, une indication, une réponse comportent, en général, des informations en paramètres. Il est des cas où la transmission d'une information parfaitement identifiée peut remplacer la transmission de la requête (ou indication...) associée à l'usage qui doit en être fait. Dans ce cas, l'information est elle-même porteuse, sémantiquement parlant, de ce que doit en faire le ou les récepteurs ; en particulier dans le cas où les récepteurs sont des entités simples ou non, qui sont en attente de certaines informations pour être exécutées.

[Exemple]

C'est le cas d'un régulateur qui attend les données d'entrée tous les Δt pour calculer une commande. C'est aussi le cas du même régulateur qui, quand il reçoit une consigne, remplace l'ancienne par la nouvelle.

Dans ce cas, les mêmes services de transmission d'information serviront aux deux types de trafics - d'où la nécessité alors pour ces services de satisfaire les deux types de contraintes à la fois (temps de réponse et durée de validité).

Ceci n'est plus vrai si la fonction ou l'entité réceptrice peut faire divers usages de l'information en question. On doit alors préciser ces usages par des requêtes ou indications selon l'initiateur. On notera enfin que le modèle client-serveur est typiquement applicable au trafic vertical, alors que le modèle producteur-consommateur l'est au trafic horizontal (cf §1.2.4).

La communication temps critique n'est pas réservée à un niveau donné dans l'architecture des systèmes automatisés.

Bien sûr elle est plus cruciale dans les niveaux bas (niveau des bus de terrain, des machines en général). Mais elle est aussi présente à des niveaux tels que la commande d'un atelier (synchronisation de robots par exemple). FIP a été conçu pour satisfaire une

communication d'état d'abord, correspondant fondamentalement au flux horizontal mais permettant en même temps de fournir les services pour le trafic vertical entre capteurs, actionneurs, et les appareils de commande du premier niveau de la hiérarchie.

C'est ainsi que la valeur d'une pression par exemple, quand elle sera transmise, parviendra aussi bien au régulateur qui en a besoin (c'est du trafic horizontal) qu'au système de surveillance ou à celui de maintenance (c'est du trafic vertical).

1.2.4 Modèles Client-Serveur (CS) et Producteur-Distributeur-Consommateurs (PDC)

Les systèmes automatisés assurent des fonctions multiples. Certaines sont classiques: régulation, commande séquentielle, commande d'axe sur un robot. D'autres le sont moins, mais participent de plus en plus à l'amélioration de la productivité au sens large (suivi de production, aide à la maintenance, gestion des modes de marche).

Chacune de ces fonctions est hiérarchisée dans la mesure où l'on peut, à un certain niveau, définir des services en faisant abstraction de la façon dont ils sont réalisés. Ces fonctions ne sont pas indépendantes les unes des autres, les liens entre elles prennent la forme d'échange de données qu'on peut décrire sous forme de modèles.

Le modèle client-serveur est basé sur une association bipoint. Deux processus et deux seulement sont en relation, l'un qu'on appelle le client, l'autre le serveur. Ce modèle peut être schématisé comme dans la figure 1.5.

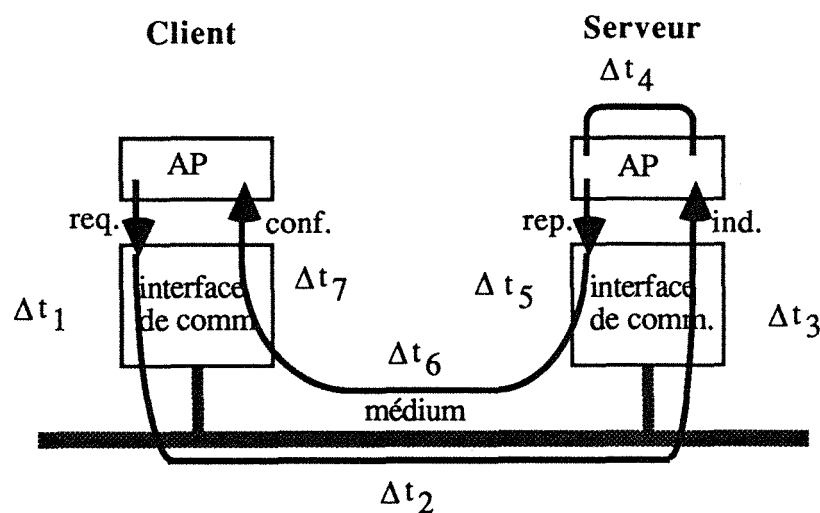


Fig. 1.5 Modèle client-serveur

Le processus client émet une requête vers le processus serveur qui à sa réception devra l'exécuter et renvoyer le résultat vers le client. Selon le modèle d'architecture fonctionnelle, on voit que le client peut être une entité de niveau N+1, le serveur une entité de niveau N et réciproquement.

Selon le modèle client-serveur, une requête sera émise par le client qui attendra la confirmation. Il est important de noter que des délais, non maîtrisables en général, interviennent. En simplifiant, on peut définir :

- Δt_1 temps de traversée des fonctions de communication chez le client pour émettre la requête
- Δt_2 temps de transmission du client vers le serveur intégrant le temps de latence
- Δt_3 analogue à Δt_1 , temps de traversée des fonctions de communication chez le serveur
- Δt_4 temps de réponse du serveur (indépendant du réseau)
- Δt_5 analogue à Δt_1 , pour transmettre la confirmation
- Δt_6 analogue à Δt_2
- Δt_7 analogue à Δt_3 , pour recevoir la confirmation.

Ces délais sont en général aléatoires. Bien que l'évaluation de performances permette de les connaître de façon plus ou moins déterminite, Δt_4 n'est toujours pas maîtrisable par le réseau (il dépend de la charge instantanée de la station qui supporte la fonction serveur). Ce n'est donc pas un modèle qui permet de garantir des contraintes temporelles.

Le modèle Producteur-distributeur-consommateur apparaît comme un complément du modèle client-serveur afin de répondre aux contraintes associées aux trafics horizontaux, qui sont la durée de validité et la cohérence des données informations échangées. En effet, ces contraintes ne sont pas prises en compte par le modèle CS. La figure 1.6 montre le modèle PDC.

Le producteur est une application responsable de la production de données qui peuvent être périodiques ou non, synchronisées ou non avec d'autres productions ou d'autres applications.

Le distributeur est responsable du transfert des données du producteur de chacune d'entre elles vers tous les consommateurs.

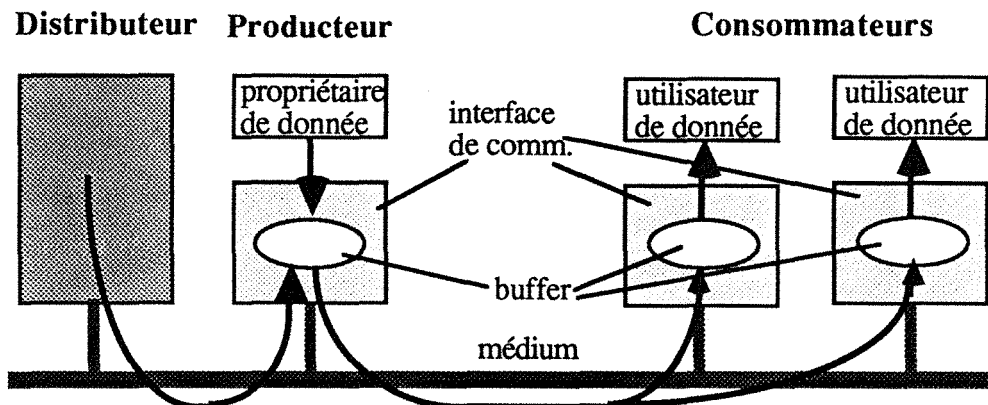


Fig. 1.6 Modèle producteur-distributeur-consommateur

Les consommateurs de données sont des applications qui, pour être exécutées, nécessitent les données reçues de correspondants. La consommation des données peut être périodique ou non et synchronisée ou non avec d'autres applications.

La distribution des données doit garantir les caractéristiques temporelles des productions par rapport aux consommations et la mise à disposition de ces données en fonction des besoins des consommateurs. La distribution peut ainsi être périodique ou non et synchronisée ou non avec d'autres applications.

De la figure 1.6, on peut constater qu'il n'y a plus de délai non maîtrisable par le réseau. Les contraintes temporelles peuvent être satisfaites. De plus, il convient de noter que ce modèle supporte la communication multipoints. Ce qui nous permet de l'utiliser pour spécifier les protocoles multipoints (multipeer/multicast).

Enfin, ce modèle PDC est particulièrement adapté aux trafics horizontaux et *peut aussi éventuellement satisfaire certains trafics verticaux.*

1.2.5 Application du modèle PDC aux bus de terrain

Nous avons vu que le modèle PDC était particulièrement adapté aux flux horizontaux soumis à des contraintes de temps strictes. S'ils ne le sont pas, le modèle client serveur peut être utilisé. Dans ce paragraphe, nous verrons que le même modèle PDC peut être utilisé pour assurer les flux verticaux dans les bus de terrain et dans FIP en particulier.

Les principaux objets échangés sur un bus de terrain sont les grandeurs numériques ou booléennes en provenance des capteurs, vers les actionneurs et de variables d'états entre les organes de commande.

[Exemple] Si nous considérons un régulateur et un capteur, par application du modèle d'architecture fonctionnelle, le régulateur devrait émettre une requête vers le capteur pour obtenir la valeur mesurée comme confirmation. Si nous ajoutons un système de supervision ou un système d'aide à la maintenance, le régulateur usuellement envoie la valeur mesurée vers ces deux autres organes. La même information est ici concernée par une transaction du type flux vertical et par une transaction du type flux horizontal. Si nous utilisons le modèle PDC, nous voyons qu'en une transaction, la valeur mesurée du capteur pourra être transmise au régulateur, au système d'aide à la maintenance et à celui de superviseur.

Règles générales :

1 - Le modèle PDC peut remplacer le modèle client/serveur quand le distributeur connaît les besoins de tout client potentiel.

Dans une application autour d'un bus de terrain, les clients doivent recevoir certaines grandeurs à des instants précis. Ces grandeurs sont répertoriées. Un distributeur peut gérer et séquencer ces transactions.

2 - Le modèle PDC est avantageux par rapport au modèle CS quand un serveur a plusieurs clients simultanés. Si plusieurs entités normalement clientes d'un même serveur doivent profiter du service au même moment pour des raisons de cohérence, le modèle PDC permet de les satisfaire en une seule transaction.

3 - Le modèle PDC impose de traduire toute demande de service en un objet particulier ou en la valeur d'un objet. Si l'on assimile les consommateurs à des clients vis à vis des objets consommés ou utilisés, le modèle PDC définit la façon dont ces clients seront alimentés par les producteurs de ces objets sans préjuger des comportements des producteurs et consommateurs. Si l'on désire influencer ou contrôler le comportement d'un producteur ou d'un consommateur, équivalent d'une demande de service, nous devons définir un ou plusieurs objets qui seront consommés par les processus concernés. Le comportement des processus sera alors le résultat de l'interprétation locale de la valeur de l'objet reçu en particulier si l'on désire démarrer/arrêter un processus à distance, ce processus devra être consommateur d'un objet dont la valeur indiquera l'action à mener. Si ce n'est pas ce processus qui est le consommateur, ce sera un ordonnanceur local.

Le producteur de cet objet n'est pas spécifié dans le modèle. Mais l'intérêt est que, si plusieurs processus doivent être démarrés/arrêtés en même temps, comme l'objet en question est transmis selon le modèle PDC lui-même, on peut être sûr qu'en cas de

bonne transmission et aux délais de propagation près, les opérations auront lieu en même temps.

1.3 Présentation générale de FIP

1.3.1 Architecture de FIP

Le bus de terrain FIP est un réseau local industriel particulier qui intéresse les niveaux bas de l'architecture des systèmes automatisés de contrôle des procédés continus, discontinus et manufacturiers. La figure 1.7 montre un exemple d'utilisation du bus de terrain FIP.

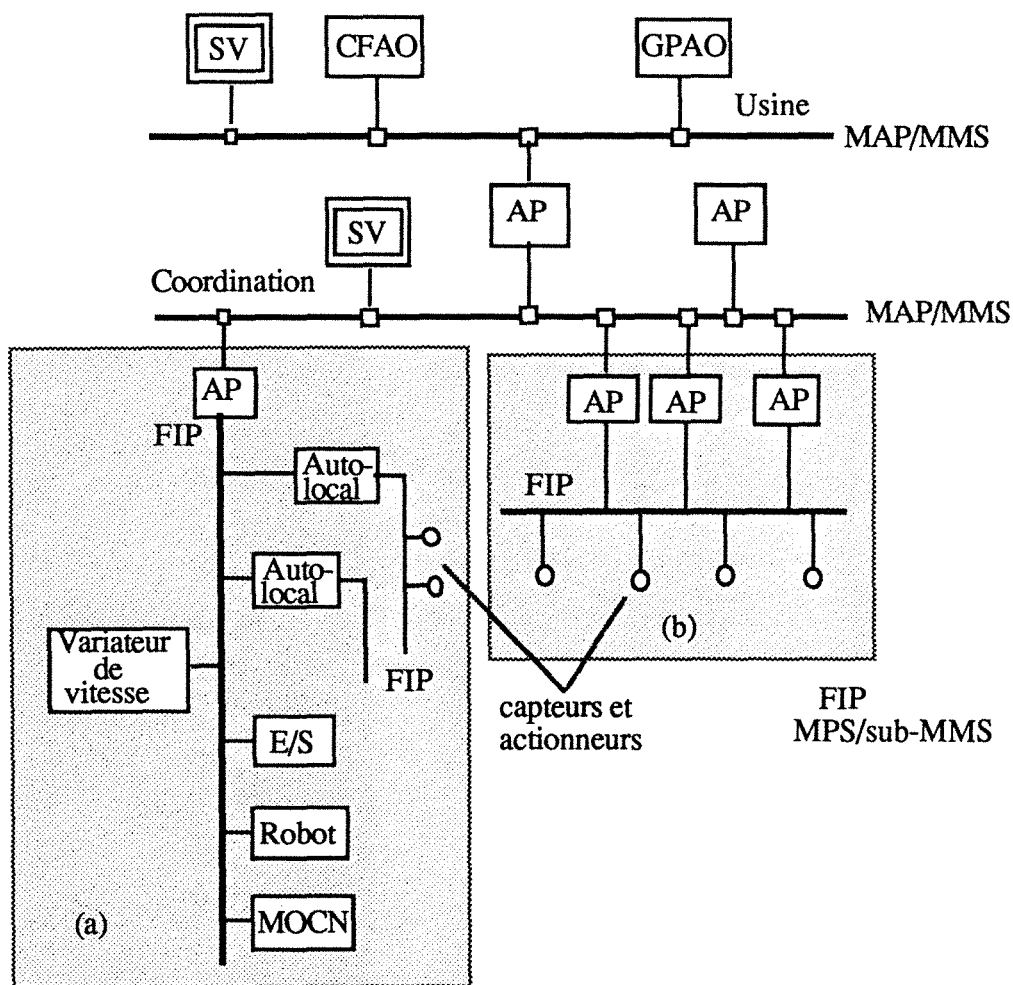


Fig. 1.7 Exemple d'un système automatisé en utilisant MAP et FIP

Dans la figure 1.7, la partie (a) montre l'utilisation de FIP au niveau instrumentation avec un automatisme centralisé sur un seul contrôleur. La partie (b) montre l'utilisation de FIP au niveau instrumentation avec un automatisme réparti sur plusieurs contrôleurs.

La topologie de ce genre de réseaux est de type bus (cf. fig. 1.8).

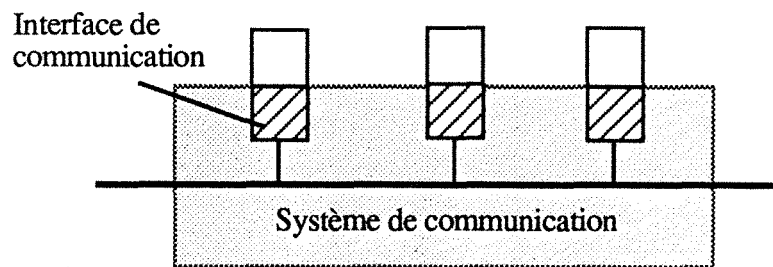


Fig. 1.8 Topologie de FIP

Depuis la normalisation du modèle O.S.I. [ISO 7498], [Zimmermann 80] tout système de communication doit y faire référence c'est-à-dire que chaque constructeur de réseau, quel qu'il soit, aime à dire que son produit est conforme au modèle même quand cela est tout à fait abusif. Mais que signifie être conforme au modèle OSI ? Normalement cela signifie que le réseau en question offre une architecture en sept couches qui sont celles définies par l'ISO. Mais cela ne veut pas dire que dans chacune des couches, les services offerts et les protocoles utilisés sont d'un type donné voire normalisés. Les bus de terrain et FIP en particulier ne dérogent pas à la règle et selon une expression consacrée maintenant, il est dit que les bus de terrain doivent offrir une architecture en 3 couches selon le modèle OSI "réduit".

Le modèle OSI réduit a été introduit au début des années 80 à propos du projet MAP [General 87]. Afin de satisfaire des exigences de temps réel, les auteurs de MAP, qui lui est conforme au modèle à 7 couches, ont estimé que pour accélérer la communication, certaines couches n'étaient pas fondamentalement nécessaires dans certains cas de figure. C'est ainsi que les couches réseau, transport, session et présentation ont disparu du modèle général donnant naissance au modèle réduit composé uniquement de la couche physique de la couche liaison de données et de la couche application [EPA-MAP].

Il faut bien être conscient qu'en supprimant des couches on diminue d'autant le "temps de traversée" de l'empilement des services mais qu'en contre-partie on n'en bénéficie plus. De plus, le problème du temps réel n'est pas seulement un problème de temps de communication mais aussi et surtout dans de nombreux cas, le problème du temps de réponse du ou des processus d'application sollicité(s). Ce temps est évidemment indépendant du nombre de couches traversées.

Pourquoi trois couches et lesquelles ? et quels services ? Quelles sont les raisons pour concevoir un bus de terrain selon l'architecture à trois couches ? et quels types de service doit-on y trouver pour offrir un aspect temps réel ou temps critique ?

L'EMUG (European MAP Users Group) a énoncé [EMUG 89] un certain nombre de spécifications qui devraient être incluses ou prises en compte dans la définition d'un réseau temps critique ou d'une architecture temps critique.

Ces éléments montrent bien qu'il ne s'agit plus d'édulcorer le modèle OSI pour satisfaire des aspects temporels mais qu'il s'agit aussi d'intégrer des mécanismes et le temps dans le modèle d'un réseau temps critique. N'oublions pas que le modèle OSI est né de l'expérience acquise dans les grands réseaux essentiellement à commutation de paquets et que le temps n'a jamais été pris en compte dans ce modèle. Il serait donc assez paradoxal de vouloir traiter du temps et en particulier respecter des contraintes temporelles en s'appuyant sur un modèle qui n'y fait jamais référence.

Quelles couches ?

Il est clair que la couche **physique** est nécessaire. On ne peut s'en passer quel que soit le support de transmission utilisé.

La couche **liaison de données** est elle aussi nécessaire pour assurer la détection d'erreurs de transmission, et fournir éventuellement une communication fiable entre deux ou plusieurs stations. Elle inclut, comme dans le modèle IEEE 802, la sous-couche "Medium Access Control" puisqu'on suppose qu'un support est partagé entre les stations connectées.

La couche **réseau** et la couche **transport** ont été introduites dans le modèle OSI pour intégrer le fait que la communication entre deux abonnés passait par des stations intermédiaires ou des noeuds intermédiaires, sortes de commutateurs encore appelés routeurs. Il fallait donc définir le routage des paquets (couche réseau) et assurer un contrôle de bout en bout entre les deux abonnés (couche transport). Ce contrôle de bout

en bout est dans l'esprit, du même type que celui qui est offert dans la couche liaison de données. La différence est que ces services s'appliquent à des objets différents : une trame dans le cas de la couche liaison de données, un message composé éventuellement de plusieurs paquets (et donc trames) dans le cas de la couche transport. Dans les deux cas, il s'agit d'offrir un service fiable de transmission d'une certaine quantité d'information.

Si la couche réseau n'est pas nécessaire, à supposer que le bus de terrain ne soit pas constitué de sous-réseaux, le rôle du contrôle de bout en bout de la couche transport disparaît. Toutefois reste le problème des messages multipaquets (fragmentation et réassemblage) qui n'est plus résolu sans couche transport. Si on fait l'hypothèse de messages courts et que l'on adapte les trames de la couche liaison de données aux plus longs messages possibles, le rôle de la couche transport disparaît totalement.

Notons que ne pas considérer de sous réseaux, n'empêche pas d'accepter que le bus de terrain puisse être formé de plusieurs segments reliés par des ponts voire de plusieurs tronçons physiques reliés par des répéteurs.

En ce qui concerne la couche **session**, elle n'apparaît pas nécessaire dans le cas des bus de terrain car son rôle dans le modèle OSI trouve son origine dans des échanges de grande quantité d'informations ce qui n'est pas le cas ici.

Un minimum de la couche **présentation** est nécessaire mais peut être implémenté statiquement plutôt que dynamiquement.

La couche **application** est évidemment nécessaire mais devra être différente des existantes aujourd'hui pour fournir les services requis par la répartition des informations, tâches et contrôle et pour satisfaire des aspects temporels. Au-dessus viendront s'ajouter des normes d'accompagnement pour normaliser soit des constituants, soit des fonctions communes à plusieurs constituants. Plus précisément il s'agirait de normaliser la fonction de communication de ces constituants ou le comportement des fonctions vu de la communication. Ceci afin d'améliorer l'interopérabilité des dispositifs, voire assurer leur interchangeabilité.

L'interface de communication FIP est donc structurée en trois couches (cf. fig. 1.9) dont les fonctions respectives correspondent à celles définies dans le modèle OSI et qui sont les suivantes:

- couche Physique,
- couche Liaison de données
- couche Application.

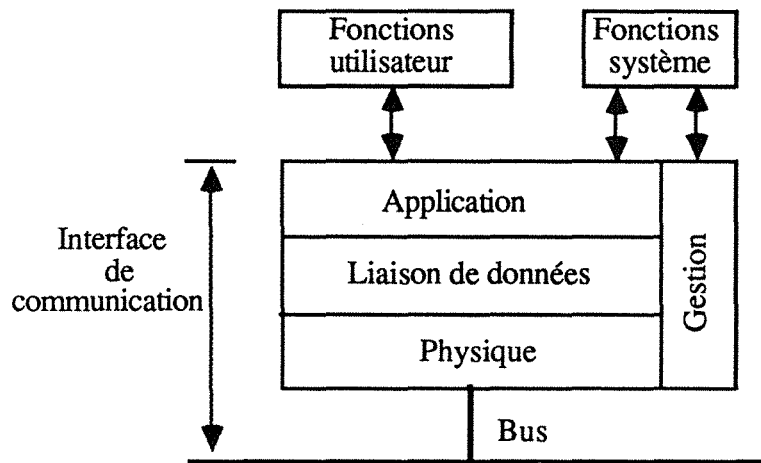


Fig. 1.9 Structure d'interface de communication FIP

1.3.2 Services offerts par FIP

Les principaux services offerts par l'interface de communication FIP sont résumés par la figure 1.10.

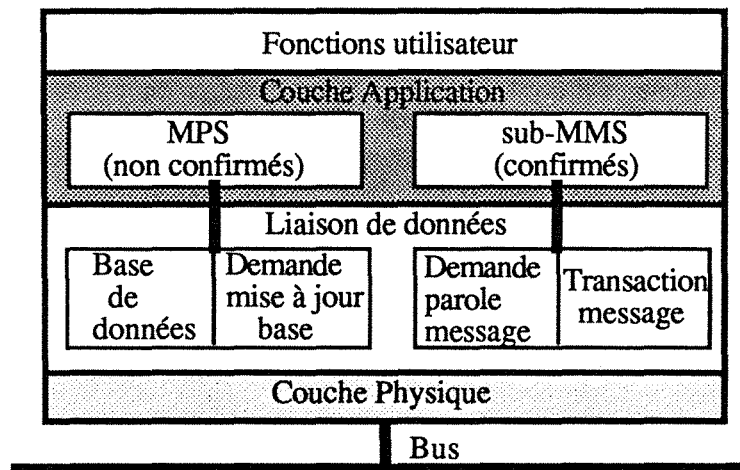


Fig. 1.10 Services offerts par l'interface

La couche liaison de données offre:

- ↳ trafic d'objets identifiés qui est
 - périodique ou
 - aléatoire

en diffusion sans acquittement,

↳ trafic de messagerie qui est

- périodique ou
- aléatoire

point à point avec ou sans acquittement ou multipoints sans acquittement.

La couche application offre à l'application de l'utilisateur l'accès à certains objets à l'aide de différents services [UTE C46-602 90].

Principaux objets:

- variable produite (qualifieur de validité temporelle comme "rafraîchissement" peut être associé)
- variable consommée (qualifieur de validité de transmission par rapport à des contraintes temporelles comme "promptitude" peut être associé)
- liste de variables
- type de la valeur d'une variable.

Principaux services:

- lecture et écriture locales de variables
- lecture et écriture distantes de variables
- demande de mise à jour de variables
- lecture de listes de variables
- indication d'émission et de transmission de variables.

Les applications à temps critique ont besoin des services périodiques/apériodiques industriels (ils sont appelés MPS au niveau de la couche Application et A/P au niveau de la couche Liaison de données). Ces services permettent:

- la mise à jour périodique en diffusion de la base de données répartie sur les différents équipements du système de commande,
- l'élaboration de status temporels associés aux données de cette base,
- l'élaboration de status de cohérence associés à des ensembles de données de cette base.

Les services de messagerie industriels sub-MMS sont principalement utilisés pour les besoins de la gestion de la configuration, des modes de marche de l'application

répartie et de la supervision. On note qu'ils sont hors temps réel (ou à temps non critique).

NB: Notre étude s'est essentiellement limitée à la couche liaison de données car, en ce qui concerne les échanges de variables simples, il n'y a pas grande différence avec la couche application. Toutefois le service de lecture de liste de variables étant très important tant du point de vue sécurité que du point de vue cohérences temporelle et spatiale, a lui aussi été modélisé et étudié (cf. §2.2.5).

1.3.3 Technique d'accès au médium de FIP

Comme les bus de terrain concernent des applications à temps critique, les techniques d'accès au médium doivent satisfaire tout d'abord les demandes d'émission urgentes (temps critique) d'une part, et essayer de ne pas priver les demandes d'émission normales (non temps critique) d'autre part. Dans FIP, deux types d'entités sont connectées sur le bus:

- Un type d'entité Arbitre de Bus (BA) qui octroie la parole à chaque producteur d'informations connecté sur le bus;
- Une entité producteur/consommateur qui émet et/ou reçoit sur le bus.

Selon les différentes types de demandes, on peut réserver le médium de transmission de manières statique ou dynamique. Le tableau 1.1 résume ces possibilités.

Tab. 1.1 Techniques d'accès selon types de demandes d'émission

Demandes d'émission	Mode de scrutation	Allocation du médium
périodique	cyclique par BA	statique
aléatoire	périodique par BA	"statique"
	apériodique	dynamique

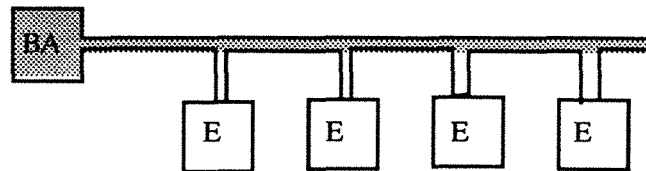
1) Technique d'accès d'objets identifiés

La communication sur le bus est effectuée en diffusion générale. Un échange de données commence par l'arbitre de bus qui émet une trame d'identifieur ou nomenclature

sur le bus. Cette trame est une identification unique de l'information désirée. Une entité producteur se reconnaît comme productrice de l'information et une (ou plusieurs) entité(s) consommateur(s) se reconnaît (reconnaissent) comme réceptrice(s) de cette même information. Ensuite, l'entité productrice émet ses données qui sont, à leur tour, reçues par la (les) entité(s) réceptrice(s).

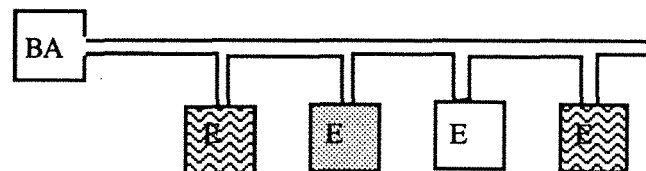
La figure 1.11 montre clairement le déroulement d'un tel échange.

Etape 1: Diffusion d'un identifieur par l'arbitre de bus BA

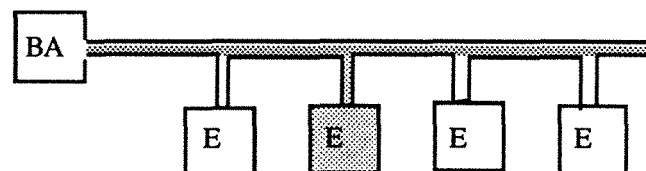


Etape 2: Reconnaissance de cet identifieur par:

- l'équipement possédant la donnée associée à cet identifieur (producteur)
- les équipements intéressés par cette donnée (les consommateurs)



Etape 3: Diffusion de la donnée par le producteur



Etape 4: Réception de donnée par tous les équipements intéressés

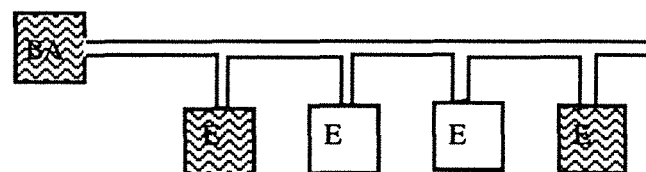


Fig. 1.11 Illustration du protocole d'accès au médium de FIP

La technique d'accès au médium de FIP est fondé sur le principe de réservation et utilise une procédure de type "tour de table" (en anglais Roll Call Polling). Puisque les instants d'arrivées des trames à temps critique sont en général déterministes, répétitifs et parfois synchronisés (un exemple est l'application fondée sur la théorie d'échantillonnage), ce protocole s'adapte très bien à ce genre d'applications. Dans FIP, une information émise par un "producteur" est identifiée à l'aide d'une désignation unique: "identifieur". La réservation du médium de transmission s'effectue à l'aide d'une table de scrutation prédéfinie lors de la phase de configuration et respectant les contraintes d'application. Dans le chapitre 2, on détaillera cette table de scrutation. En effet, cette table contient tous les identifieurs à scruter périodiquement. Cette table réside et est gérée dans une entité appelée "Arbitre de Bus (BA)". La figure 1.11 montre un BA et plusieurs entités producteur/consommateur d'informations reliées par un bus. Les informations transmises sur le bus sont en mode de diffusion générale, c'est à dire toutes les entités intéressées par une information peuvent la recevoir.

Le droit de parole de chaque entité producteur est contrôlé par l'arbitre de bus, FIP utilise une technique d'accès centralisé. En effet, le BA diffuse les identifieurs séquentiellement selon la table de scrutation. Dans un souci de sûreté de fonctionnement, le BA peut être redondant. Le producteur répond en diffusant la donnée associée à cet identifieur et les consommateurs la reçoivent. En recevant la réponse, le BA diffuse l'identifieur suivant ...

Chaque équipement (ou entité) dispose des mémoires tampons (buffer) dans lesquelles il stocke les données produites et consommées, qui sont adressées par une adresse logique codée sur 16 bits et dont les différentes valeurs correspondent aux identifieurs.

Les échanges d'informations s'effectuent en général sans acquittement avec la possibilité de disposer au niveau des équipements d'un mécanisme de détection d'erreur de transmission basé sur la gestion d'une durée de vie limitée pour les informations consommées. Pour un certain nombre d'informations du type événement, des services aperiodiques ont été définis.

Le système de communication FIP peut procéder au rafraîchissement périodique des informations dont les identifieurs figurent dans la table de scrutation du BA. En ce sens, le réseau FIP est assimilable à une base de données répartie mise à jour en temps réel.

D'autres besoins ont conduit à définir des services de messagerie. Puisque les messages arrivent de manière aléatoire, ces services génèrent un trafic aléatoire qui prend place dans les temps morts du trafic périodique.

En plus, le besoin de déclencher un transfert des objets identifiés sur demande de l'utilisateur a conduit à définir des services de "demande explicite". Puisque les demandes (ou requêtes) arrivent de manière aléatoire, ces services génèrent un trafic aléatoire qui prend place dans les temps morts du trafic périodique.

2) Technique d'accès de messages et de demandes

Pour expliquer comment un message ou une demande est transmis, on introduit d'abord la notion de "station" qui est un point de raccordement ou encore un SAP (service access point) de messagerie dans un réseau FIP.

Une "station" est un point de raccordement qui regroupe un ensemble d'équipements producteurs/consommateurs. Le nombre de producteurs dans une telle station est limité à 100 par la norme [UTE46-603 90]. Les messages et les demandes explicites générés par des dispositifs liés à cette station sont d'abord transformés en trames en traversant toutes les couches jusqu'à la couche liaison. Ces trames sont ensuite mises en files d'attente telles que F_MSGcyc (file de messages périodiques), F_MSGapér (file de messages aperiodique), F_REQi (file des requêtes), B_REQ (buffer de requête).

On voit que le protocole d'accès de FIP utilise une procédure de scrutation. Il s'adapte bien au trafic cyclique. Comme l'instant d'arrivée d'une trame "apériodique" dans une file d'attente est aléatoire et imprévisible, réserver une place en plaçant l'identifiant de cette file d'attente dans la table de scrutation gaspille sans doute la ressource (médium de transmission). Par contre, si l'identifiant de la file d'attente ne fait pas partie de la table de scrutation, les trames en attente dans cette file d'attente ne seront jamais transmises.

La solution de FIP est d'insérer l'identifiant d'une file d'attente dans la table de scrutation de façon dynamique afin de pouvoir transmettre ses trames durant les temps non occupés par le trafic déterministe.

Pour insérer un identifiant dans la table de scrutation, une station doit manifester auprès de l'arbitre de bus (l'endroit où se trouve la table de scrutation) son intention d'avoir un droit de parole supplémentaire. Cette intention est communiquée à l'arbitre de bus par l'intermédiaire d'une trame de réponse de producteur de la station.

La figure 1.12 montre les deux principaux types de trames qui transitent sur le bus: la trame d'identifieur et la trame de réponse contenant une donnée associée à un identifieur.

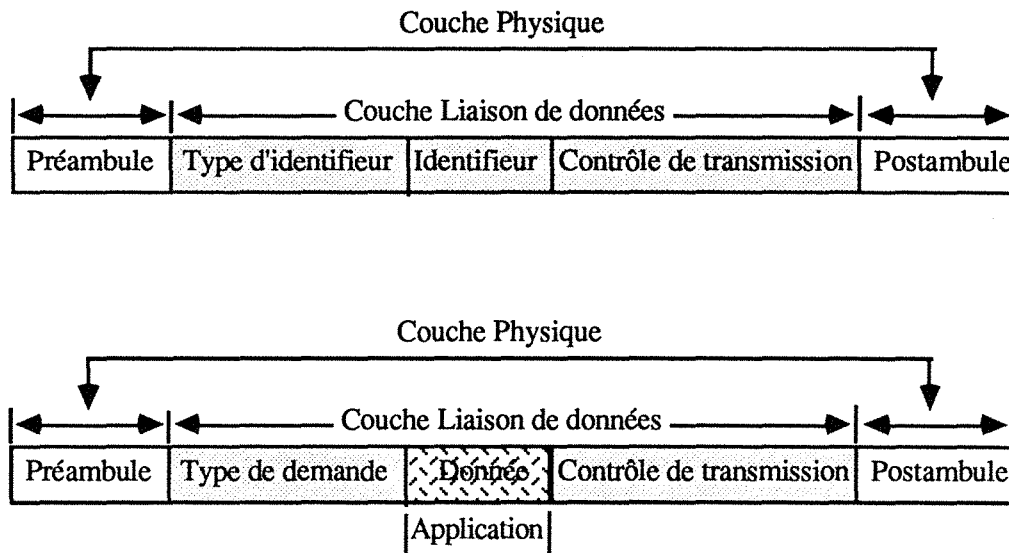


Fig.1.12 Structure des trames du protocole FIP

Les champs de contrôle "type d'identifieur" dans une trame ID et "type de demande" dans une trame de donnée jouent un rôle fondamental pour les services apériodiques.

Lors de l'envoi de la trame de réponse périodique, la station productrice a la possibilité d'ajouter une information supplémentaire dans le champ de contrôle pour informer le BA du besoin d'un droit de parole supplémentaire.

En recevant la trame réponse, le BA analyse le champ de contrôle. S'il y a une demande dans ce champ, le BA enregistre cet identifieur et puis l'insère dans la partie dynamique (i.e temps mort du trafic périodique) avec le nouveau type dans son champ "type d'identifieur". Lorsque cet identifieur est diffusé, l'entité initiatrice le reconnaît, et elle peut alors diffuser à son tour la donnée associée à cet identifieur vers le bus.

1.4 Méthodologie et critères de performances

Les protocoles de communication posent différents problèmes en vue de leur implémentation. Jusqu'à présent, la priorité a été donnée aux aspects couvrant les

problèmes de validation et vérification de ces protocoles. L'analyse des protocoles de communication OSI montre l'insuffisance accordée aux aspects quantitatifs. Parallèlement aux problèmes de validation/vérification, apparaissent les problèmes d'évaluation des performances et de dimensionnement au sein d'une architecture normalisée. En effet, une large autonomie est laissée à l'implémenteur au niveau de la paramétrisation des protocoles. Cette paramétrisation a une influence significative sur les performances des protocoles.

L'analyse doit être réalisée d'un point de vue qualitatif aussi bien que quantitatif.

La modélisation qualitative utilise souvent trois catégories de techniques:

- Techniques fondées sur les modèles des machines d'états finis ou les réseaux de Petri.
- Langage orienté modèle comme le langage formel (ESTELLE et LOTOS).
- Modèle hybride qui inclut états et construction de langage.

Ces techniques permettent en principe de tester des erreurs syntaxiques du protocole étudié (appelé validation) et de tester si le fonctionnement du système étudié est conforme à la norme (appelé vérification).

La modélisation quantitative utilise trois catégories de techniques:

- mesure d'un système existant (souvent une maquette),
- étude analytique (souvent un modèle mathématique en utilisant la théorie de files d'attente),
- simulation (souvent à événements discrets).

Pour la mesure d'un système réel, il s'agit de collecter des données statistiques sur un système en état de fonctionnement. Cela exige l'existence du système et le coût d'étude est souvent très élevé, et la durée d'étude longue. En fait, l'existence d'un système est déjà contradictoire avec cette étude, qui est souvent une phase de pré-implémentation.

La simulation d'un modèle demande:

- la connaissance parfaite du système étudié qui permet la définition d'un modèle simplifié adapté à l'étude, i.e. le modèle n'est pas forcément une description en langage de simulation du système exact. Il peut se limiter aux paramètres et aux comportements intéressants tout en conservant une certaine finesse.
- des données d'entrées. En considérant le modèle comme une boîte noire, ces données sont essentielles pour démarrer la simulation. La collecte de ces données est une étape

primordiale. On peut en déduire soit des caractères statistiques (et une distribution de probabilité) s'il s'agit d'une entrée aléatoire, soit un tableau de données si l'entrée est déterministe.

L'étude analytique consiste à élaborer un modèle mathématique, ce qui exige des hypothèses souvent éloignées de la réalité. Les résultats ainsi obtenus sont quantitatifs mais nous donnent seulement une vue approximative du système étudié. Face à la complexité croissante des systèmes à étudier, des méthodes d'approximation sont nécessaires.

Souvent les modèles mathématiques sont trop simplifiés; il reposent sur des hypothèses nécessaires à leur souplesse, qui ne garantissent pas sa validité. Dans ce cas, la simulation peut être utilisée pour valider les hypothèses nécessaires à un modèle analytique, et un modèle analytique peut proposer des alternatives raisonnables pour développer la simulation.

Dans notre étude, nous nous intéressons à une modélisation quantitative de la couche Liaison de données du protocole FIP afin d'évaluer les performances et de réaliser le dimensionnement. Il convient de signaler que cette étude quantitative est parallèle aux travaux de modélisation qualitative mentionnés précédemment. En effet, l'importance d'une analyse de performances d'un système (ou protocole) réside en principe dans les aspects suivants:

- 1) estimation des performances du système vis-à-vis de la redondance (overhead) introduite. (C'est le critère "rendement" qu'on doit évaluer).
- 2) mise au point des paramètres du système pour optimiser ses performances prévues. (il s'agit donc de dimensionnement)
- 3) comparaison de performances entre deux ou plusieurs systèmes en termes quantitatifs. (Soit on définit une fonction en regroupant les trois critères, soit on compare séparément les critères comme: délai, taux d'utilisation et rendement).

Le trafic à temps critique, en particulier le service de transfert du buffer périodique, doit être étudié avant le trafic à temps non-critique car comme nous l'avons constaté dans la présentation générale de FIP, le trafic à temps non-critique dépend du trafic à temps critique. En effet, non seulement le trafic à temps non-critique utilise le temps mort du trafic à temps critique, mais aussi toute demande de droit de parole

(appelée aussi requête) nécessaire pour déclencher les échanges à temps non-critique est véhiculée par le champ contrôle de la trame de données du service périodique.

Face à un système de communication dans lequel les trafics à temps critique (déterministes) et à temps non-critique (aléatoires) coexistent, comment peut-on apprécier ses performances? Quels sont les paramètres les plus significatifs pour le système global? Une comparaison entre différents protocoles d'accès au médium est-elle réalisable? La réponse à toutes ces questions n'est pas toujours la même du point de vue d'un utilisateur, d'un manager et d'un concepteur, car leurs préoccupations ne sont pas les mêmes. Ces trois individus ont des demandes parfois communes et parfois contradictoires. Mais quel que soit le critère de performance considéré par un individu, ce critère prend la forme "rapport coût-performance". Un utilisateur peut très bien demander un certain nombre de services avec un coût minimum. Un manager du réseau souhaite un gain maximum en investissant au minimum. Le point de vue du concepteur est alors plus équilibré. Il essaye de trouver la solution satisfaisant à la fois le manager et l'utilisateur. On peut dire que les critères de performances de l'utilisateur et du manager sont inclus dans ceux du concepteur. Dans la suite, nous regardons d'une façon générale les critères de performances définis par un concepteur. Notons que les critères concernant les trafics à temps critique et non-critique ne pèsent pas du même poids vis-à-vis des performances globales.

Les trois principaux critères utilisés pour évaluer les performances d'un protocole d'accès au médium (MAC) sont:

- le **rendement** du protocole qui s'exprime par le rapport entre la quantité d'information utile, c.à.d sans tenir compte des trames de service (gestion ou "overhead"), et la quantité totale d'information véhiculée par le médium. Ce critère est généralement utilisé pour estimer l'efficacité de la structure des trames d'un protocole d'accès.
- le **taux d'utilisation** du médium (ou "throughput" équivalent) qui s'exprime par le rapport entre la quantité d'information véhiculée par le médium (par seconde) et le débit du réseau. Ce critère est essentiellement utilisé pour évaluer globalement les performances. Un taux d'utilisation faible implique soit que le médium de transmission est peu sollicité par les stations, soit que la technique d'accès utilise "mal" la capacité du médium.
- le **délai** de transfert (ou temps de réponse) qui exprime le temps d'attente, pour une station donnée, entre l'instant d'arrivée d'une trame dans la station (souvent

mis en file d'attente) et la réception de celle-ci par la station destinataire. Ce critère est souvent exprimé en terme de moyenne ou de maximum.

Remarque 1: Un critère de puissance "Power" peut être utilisé pour remplacer "taux d'utilisation" et "délai". Il est défini tout simplement comme le rapport entre "taux d'utilisation" et "délai".

Remarque 2: On distingue le *délai de transfert* et le *délai d'accès*. Le délai d'accès d'un protocole d'accès au médium correspond au délai d'attente de la trame qui se trouve à la première place de la file d'attente. Pour ce délai, on ne compte ni le temps d'attente pour qu'une trame se déplace de la dernière à la première place de la queue (file d'attente), ni le délai de transmission. Souvent lorsque les gens apprécient un protocole, ils prennent ce délai. Par exemple, quand on parle du délai d'accès borné pour certains protocoles comme "token bus", "token ring" et FDDI, c'est toujours ce délai qui est considéré. Nous signalons que ce critère est loin d'être suffisant pour apprécier un protocole. En effet, pour un utilisateur, c'est toujours le délai de transfert (de bout en bout) qui compte.

Pour le trafic à temps critique, le taux d'utilisation du médium est défini par la table de scrutation lors de la phase de configuration. Le délai de transfert est ainsi garanti. Le seul critère qui reste à évaluer est donc le rendement du protocole. Cette étude est reportée au chapitre 2. Un critère supplémentaire est la fiabilité du service périodique. En fait, le trafic à temps critique est composé essentiellement de deux genres de communication: d'état et d'événement. Le transfert de données d'états n'exige pas une fiabilité très élevée tandis que l'erreur ou la perte de données d'événements conduira à une défaillance grave voire catastrophique [Simonot 90-4].

La figure 1.13 résume la démarche adoptée pour effectuer notre étude.

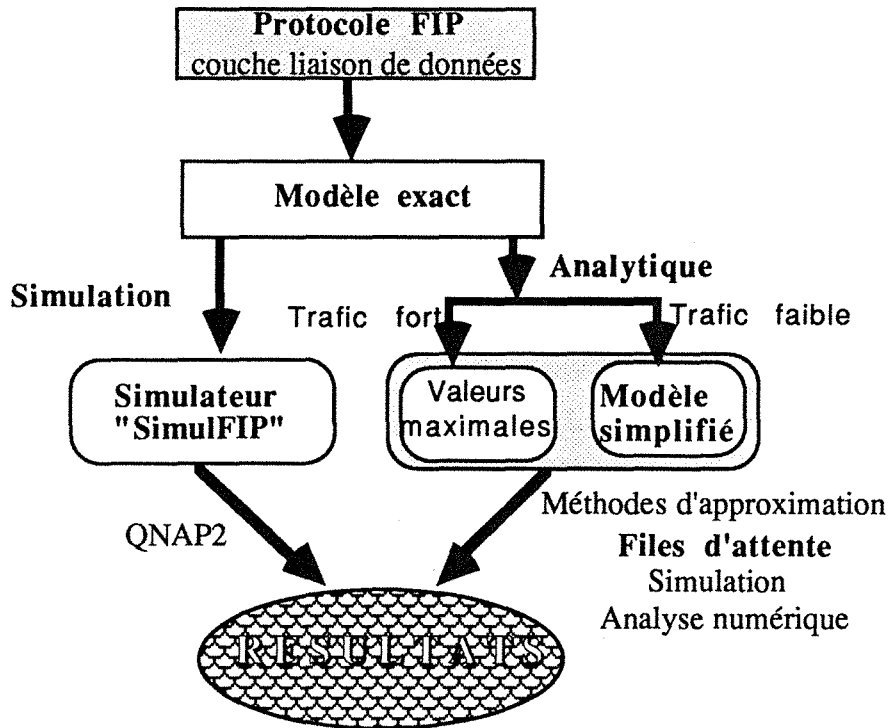


Fig. 1.13 Démarche de l'étude

Nous nous intéressons plus particulièrement à l'étude du trafic aléatoire généré en principe par des services aperiodiques tel que par exemple, la messagerie aperiodique. Etant un système ouvert, l'échange d'informations entre les différents types de réseaux est inévitable (communication avec MAP, par exemple). D'ailleurs, un certain nombre d'applications ont besoin aussi de services du genre MMS. La normalisation du bus de terrain FIP nécessite donc l'évaluation des performances des services aperiodiques et en particulier le service de messagerie.

Les moyens habituels pour étudier ce type de trafic sont sans doute:

- la théorie des files d'attente
- la simulation à événements discrets.

Pour pouvoir effectuer une étude analytique sur FIP, il est nécessaire de distinguer deux cas:

- trafic faible
- trafic fort.

Un modèle simplifié est proposé à la fin du chapitre 2 en vue d'analyser les comportements du système en cas de trafic faible. C'est un système de files d'attente à deux étages:

- le premier étage est composé de files d'attente en parallèle
- le deuxième étage est composé d'une seule file d'attente.

Les serveurs associés à chacune des files d'attente sont disponibles de manière intermittente.

Une technique d'approximation très récente est appliquée à ce modèle. Il s'agit de remplacer un processus (ou une suite de variables aléatoires) mal connu par un autre processus (ou une suite de variables aléatoires) bien connu en évaluant la *distance en variation* (l'écart) entre ces deux processus (ou suite de variables aléatoires). Les paramètres quantitatifs ainsi obtenus nous donnent une idée suffisamment précise du système étudié. L'étude est validée ensuite par la simulation (cf. §3 et §4).

Dans le cas de trafic fort, l'étude ne pose pas de difficultés particulières. Des limites existent pour la plupart de services. Le délai de transfert maximal des services aperiodiques de FIP est calculé en exemple dans le chapitre 5.

Il est souvent difficile de résoudre tous les problèmes concrets en effectuant seulement une étude analytique. Dans l'annexe2, un logiciel de conception et de préimplémentation est décrit en détail. Il s'agit d'un simulateur de FIP nommé: "SimulFIP". Ce logiciel est développé en langage de simulation QNAP2 avec l'interface utilisateur en C. Il suffit de saisir une configuration désirée, le simulateur donnera en sortie des paramètres de performances. Notons que ce logiciel peut aussi être utilisé pour valider de façon indirecte notre modèle général qui est approximatif pour les services aperiodiques de FIP(cf. §5).

1.5 Normalisation

1.5.1 Histoire de bus de terrain

La notion de bus de terrain est apparue pour la première fois dans les domaines nucléaire [Costrell 73] et militaires [Haverty 86]. A la fin des années 70, l'instance internationale IEC a publié le premier document sur les besoins fonctionnels de ce genre de système baptisé PROWAY (PROcess data hiWAY) [IEC 79]. L'étude des besoins des applications et l'étude des architectures des bus de terrain [Wood 79] [Sloman 81] a démarré à partir de cette époque. Dans le courant des années 80, sont apparues des propositions de bus de terrain en vue d'une normalisation internationale, parmi lesquelles citons les plus connues: BITBUS [Intel 84], FIP [UTE C46-60x 90], PROWAY C [ISA

85], PROFIBUS [Bosch 87] et [Simens 88], FOXBORO [Foxboro 89], ROSEMOUNT [Rosemount 88]. A la fin des années 80 et début 90, il ne reste plus que deux candidats à la normalisation internationale: FIP et PROFIBUS.

1.5.2 Normes françaises

La notion de FIP est apparue en 1984 sous forme d'un livre blanc [Galara 84]. Mais certains concepts remontent à 1978. A cette époque, l'idée d'un bus de terrain (le terme n'existait pas et on ne parlait pas encore de réseau d'instrumentation) allait naître, même si certaines idées étaient déjà plus ou moins dans l'air. Par ailleurs, des besoins de communication spécifiques au temps réel avaient été identifiés et des mécanismes (sur le papier) proposés pour y répondre [Thomesse 78-1], [Thomesse 78-2], [Thomesse 80]. Des utilisateurs, par leur retour d'expérience, avaient parfaitement reconnu le besoin. Depuis 86, FIP fait l'objet d'une série de normes expérimentales françaises UTE C46-60X:

- UTE C46-601 Architecture et présentation générale du bus de terrain FIP
- UTE C46-602 Couche application - services périodiques et apériodiques
- UTE C46-603 Couche liaison de données
- UTE C46-604 Couche physique sur paire torsadée blindée
- UTE C46-605 Gestion du bus de terrain FIP
- UTE C46-606 Couche application - services de messagerie
- UTE C46-607 Couche physique en bande de base sur fibre optique.

Parmi lesquelles C46-603 est devenue une norme finale (homologuée par AFNOR en Juin 1990).

Par ailleurs, des normes d'accompagnement spécifiques aux différentes classes de produits (capteurs, automates, régulateurs) sont en cours d'étude. Deux circuits intégrés FIPART et FULLFIP ont été conçus. De plus, un outil d'aide à la configuration "FIP-Toolbox" est également disponible [Centre 90].

1.5.3 Normalisation internationale

Les travaux sont concentrés dans le cadre de deux instances internationales: ISA SP 50 (USA) et IEC 65C WG 6. La démarche a été la suivante:

- recensement des besoins fonctionnels et des exigences d'exploitation
- analyse des normes existantes

- développement de normes expérimentales satisfaisant ces exigences
- incorporation des retours d'expérience pour élaborer la norme homologuée

Ces travaux, pour être sérieux, nécessiteront des études en profondeur (modélisation qualitative et quantitative) afin que soient préservées les caractéristiques déterminantes que FIP a proposées.

Dans l'ensemble des solutions initialement proposées, deux solutions émergent et sont au centre des travaux. FIP a maintenant une reconnaissance internationale qui ne se limite plus à la France, ni même à l'Europe. Cependant, les comités de normalisation ne souhaitent qu'une seule norme, et travaillent à un compromis où FIP serait l'un des piliers.

Parallèlement, les éléments techniques de la norme FIP ont alimenté sous forme de contributions le projet Eureka Fieldbus EU68.

Aujourd'hui, nous vivons dans une phase de comparaison, complémentation et vérification des différentes propositions. Il existe des normes de fait et des produits expérimentaux. Il est probable que la norme finale ne sera pas élaborée avant cinq ans.

CHAPITRE 2

DESCRIPTION ET MODELISATION

2. Description et modélisation

Dans ce chapitre, Nous donnons d'abord une présentation et une analyse détaillée de la couche liaison de données de FIP. Des questions concernant des performances sont posées au fur et à mesure de la présentation de chaque service fourni par cette couche. Ensuite, des solutions sont données pour des problèmes "simples" (vis-à-vis de l'évaluation de performances). En ce qui concerne les problèmes complexes, un modèle général commun à tous les services aléatoires est proposé et une formalisation de ces derniers est effectuée.

2.1 Présentation de la couche liaison de données de FIP

Dans ce paragraphe nous allons d'abord approfondir la connaissance de FIP par la présentation détaillée des différents services offerts par la couche liaison de données. Nous invoquerons au fur et à mesure tous les problèmes à résoudre au cours de notre étude. Il ne s'agit pas d'une simple reproduction du protocole mais plutôt d'une clarification (suppression des ambiguïtés). De plus, il apporte quelques précisions sur le texte de la norme NFC 46-603.

2.1.1 Vue globale

La couche Liaison de données met à la disposition de la couche Application un ensemble de services. Les différentes combinaisons de ces services offrent neuf classes de conformité (cf. tab 2.1).

Le service d'écriture de buffer et le service de lecture de buffer sont locaux à une entité. Ces deux services n'ont pas besoin de communication (ou transmission), ils ne font donc pas l'objet de notre étude.

Le service de transfert de buffer est obligatoire, car les services de demande explicite (spécifiée et libre) et de messagerie (avec et sans acquittement) s'appuient sur ce service de transfert de buffer.

Les services offerts à l'interface conceptuelle de la couche Application-Liaison de données sont en terme de primitive selon ISO 8802.2, c'est-à-dire la couche Application demande l'exécution d'une certaine tâche en envoyant une primitive <primitive>.dem à la couche Liaison de données. Cette dernière répond à une telle demande avec la primitive

<primitive>.conf. De plus, la couche Liaison de données peut informer la couche Application en lui passant une primitive <primitive>.ind.

Tab 2.1 Classes de conformité offertes par la couche Liaison de données de FIP:

classe service	1	2	3	4	5	6	7	8	9
transfert buffer	X	X	X	X	X	X	X	X	X
écriture buffer	X	X	X	X	X	X	X	X	X
lecture buffer		X	X	X	X	X	X	X	X
demande explicite libre				X	X			X	X
demande explicite spécifiée			X		X				X
message sans acquittement						X	X	X	X
message avec acquittement							X	X	X

Les différents services se déroulent dans les fenêtres ci-dessous (fig.2.1):

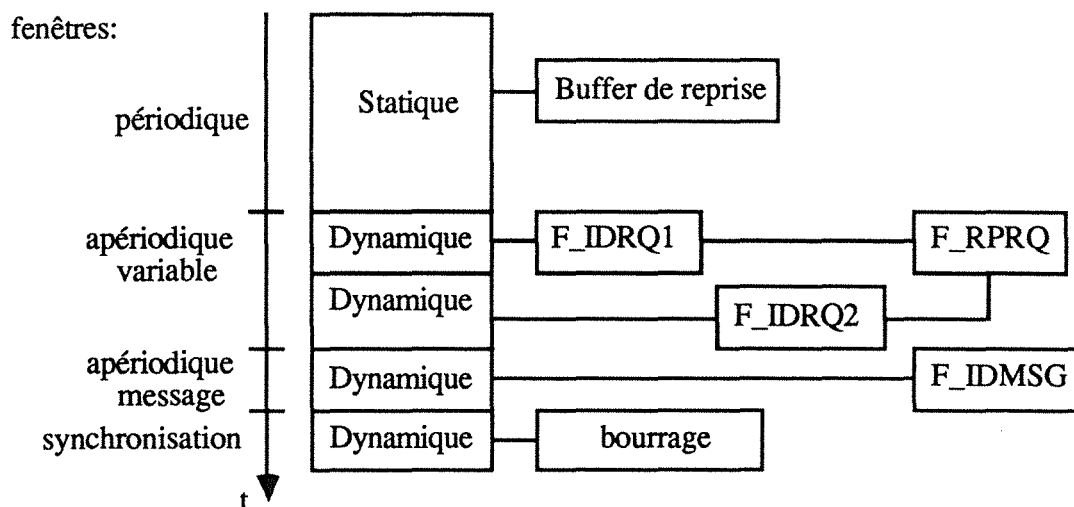


Fig.2.1 Les fenêtres dans un cycle élémentaire de FIP

Les différents services se déroulent dans des fenêtres différentes. On peut distinguer trois fenêtres: périodique durant laquelle se déroule le service de transfert de buffer *cyclique*; aperiodique variable durant laquelle se déroule le service de demande explicite *aperiodique* et le transfert de buffer; aperiodique message durant laquelle se déroule le service de messagerie *aperiodique*.

Si on examine les services de plus près, on s'aperçoit qu'il existe des options différentes. Pour éclaircir le fonctionnement de ces options et aussi pour faciliter la description, on les nomme comme des services et non pas comme des options d'un certain service. On énumère ci-dessous ces services, leurs issues et les fenêtres correspondantes. Le nombre qui précède chaque terme désigne la priorité.

- 0- Le "service de transfert cyclique **cyc**" issu du service de transfert de buffer qui se déroule durant la "fenêtre de scrutation cyclique **Fcyc** de durée **Tcyc**".
 - 1- Le "service de messagerie périodique **mp**" issu du service de messagerie qui se déroule durant la "fenêtre de messagerie périodique **Fmp** de durée **Tmp**".
 - 1- Le "service de demande explicite spécifiée périodique **rqp**" issu du service de demande explicite et du service de transfert de buffer qui se déroule durant la "fenêtre explicite périodique **Frqp** de durée **Trqp**".
 - 2- Le "service de demande explicite spécifiée aperiodique **rqs**" issu du service de demande explicite et du service de transfert de buffer qui se déroule durant la "fenêtre explicite aperiodique **Frqa** de durée **Trqa**" et la fenêtre **Fcyc**.
 - 2- Le "service de demande explicite libre urgente **rqu**" issu du service de demande explicite et du service de transfert de buffer qui se déroule durant la "fenêtre explicite aperiodique **Frqa** de durée **Trqa**" et la fenêtre **Fcyc**.
 - 3- Le "service de demande explicite libre normale **rqn**" issu du service de demande explicite et du service de transfert de buffer qui se déroule durant la "fenêtre explicite aperiodique **Frqa** de durée **Trqa**" et la fenêtre **Fcyc**.
 - 2- Le "service de messagerie aperiodique **msg**" issu du service de messagerie qui se déroule durant la "fenêtre de messagerie aperiodique **Fmsg** de durée **Tmsg**" et la fenêtre **Fcyc**.
 - *4- Le "service de synchronisation **syn**" qui se déroule durant "fenêtre de bourrage **Fbrg** de durée **Tbrg**".
- note:** La fenêtre de bourrage **Fbrg** est optionnelle.

Le fonctionnement de l'arbitre de bus comporte deux niveaux d'enchaînement:

- enchaînement de ces fenêtres pour composer un cycle élémentaire
 - enchaînement de cycles élémentaires pour former un macro cycle.
- De plus, deux principes d'enchaînement de cycles élémentaires sont possibles:
- avec synchronisation
 - ou sans synchronisation.

Si la fonction de scrutation synchronisée n'est pas demandée, la fenêtre de bourrage n'existe pas.

Dans cette thèse, on étudie uniquement l'option avec synchronisation. Puisqu'on se place dans un environnement à temps critique, l'option sans synchronisation ne peut satisfaire les exigences du temps critique.

Nous rappelons que la sous-couche MAC (Medium Access Control) de FIP utilise un protocole centralisé. Deux types d'entités (ou stations) dit *arbitre de bus* (*station maître*) et *producteur/consommateur* (*station esclave*) sont raccordées à un support de transmission de topologie bus. L'arbitre de bus possède une table de scrutation composée d'une liste d'identifieurs (partie statique), émis cycliquement avec une période d'un macro cycle. Un macro cycle comprend un ou plusieurs cycles élémentaires (ou micro cycle) qui est lui même construit par des fenêtres différentes comme montrées dans la figure 2.1.

La figure 2.2 donne un exemple d'une table de scrutation.

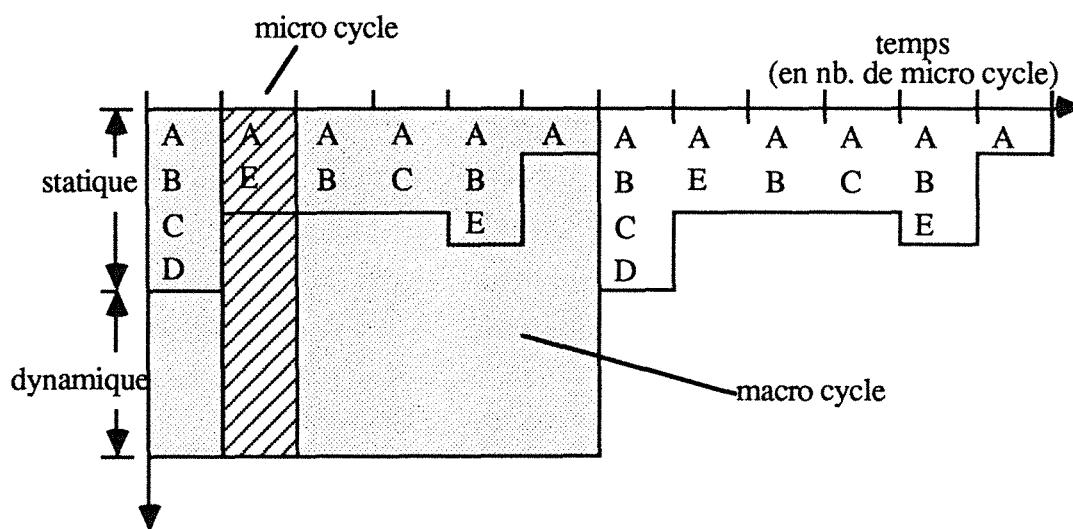


Fig.2.2 Table de scrutation

Sur la figure 2.2, on a supposé que le BA fonctionne en mode synchrone. Dans ce mode, la durée d'un cycle élémentaire (ou micro cycle) T_{bc} et celle d'un macro cycle T_m se calculent de la manière suivante:

- T_{bc} est le PGCD des périodes de scrutation de toutes les variables produites dans le système.
- T_m est le PPCM des périodes de scrutation de toutes les variables produites dans le système.

Pour $n = 0, 1, \dots, T_m/T_{bc}-1$, on a la relation ci-dessous:

$$T_{bc} = T_{cyc}(n) + T_{mp}(n) + T_{rqp}(n) + T_{rqa}(n) + T_{msg}(n) + T_{brg}(n).$$

Il est évident que la construction de cette table dépend d'une part de l'application et d'autre part de l'implémenteur. L'algorithme de gestion et le dimensionnement de ces fenêtres doivent permettre un droit d'accès égalitaire entre les différentes entités afin d'éviter la situation de famine pour certaines entités.

Questions générales: La table de scrutation joue un rôle extrêmement important sur les performances des services aléatoires. Comment construire une telle table? Comment dimensionner les différentes fenêtres d'un cycle élémentaire? Quelle est l'influence de la configuration de cette table sur l'efficacité de chaque service? Comment dimensionner les buffers et les files d'attente?

Pour répondre à ces questions générales, nous analysons en détail les services de la couche liaison de données en mettant évidence son rôle, son fonctionnement, les ressources requises, les moments de déroulement et abordons enfin la question de performances. Il faut signaler que nous avons choisi une langue naturelle (français) au lieu d'un langage formel pour présenter le protocole afin de faciliter la lecture. Une description plus ou moins formelle (en machine d'états finis et en algorithmique) peut être trouvée dans la norme FIP [NFC46-603].

2.1.2 Services de transfert cyclique

La vocation première de FIP est d'assurer le service de transfert cyclique de variables identifiées ainsi que l'écriture et la lecture de ces variables par la couche Application.

Chaque entité raccordée au bus est configurée avec deux types de buffers:

- B_DATprod (ID, VAL) est un producteur d'information
- B_DATcons (ID, VAL) est un consommateur d'information
où ID est l'identifiant de la variable et VAL est sa valeur.

Les primitives du service de transfert cyclique sont:

- L_SENT.indication (ID): signale à la couche Application qu'une variable identifiée produite vient d'être émise sur le bus.
- L_RECEIVED.indication (ID): signale à la couche Application qu'une variable identifiée consommée vient d'être reçue correctement.

L'entité prod/cons configurée pour ce service possède deux buffers: B_DATprod et B_DATcons. Le transfert de buffer se déroule en deux phases pour toute variable durant T_{cyc}:

- L'arbitre de bus émet la trame identifiant (ID_DAT) associée à la variable et arme une temporisation T1.
- L'entité productrice émet la trame de réponse correspondante (RP_DAT) et génère une primitive L_SENT.ind vers la couche Application. La valeur de la variable est prélevée par toutes les entités consommatrices et une primitive L_RECEIVED.ind est générée vers chacune des couches Application concernées.
- L'arbitre de bus scrute l'identifiant suivant s'il reçoit la trame de réponse (RP_DAT) ou si la temporisation T1 expire.

Question 1: Pour ce service, il n'y a pas de problèmes concernant de délai de transfert car la table de scrutation dans le BA est prédéfinie à la configuration. Chaque producteur peut donc envoyer sa valeur au moment exact où il faut. Le problème de dimensionnement n'existe pas non plus puisqu'il ne s'agit que de deux buffers de dimension 1. Par contre, le rendement et le taux d'utilisation peuvent être calculés à partir de la table de scrutation. Comment calculer ces deux paramètres ? (réponse au §2.2.3 et §2.2.4)

2.1.3 Services de demande explicite

Ces services permettent la demande explicite de diffusion d'un ou plusieurs identifiants associés chacun à une variable afin de déclencher l'échange de cette ou ces variables. L'utilité d'un tel échange surgit par exemple si, pendant la fenêtre périodique,

des données importantes sont perdues (données d'événement telle qu'une alarme, une synchronisation, etc...).

Ce service est invoqué par la couche Application de façon aléatoire. Deux options qui résultent des utilisations possibles de ce service sont offertes:

- demande explicite **spécifiée** (§2.1.2.1 et §2.1.2.2)
- demande explicite **libre** (§2.1.2.3 et §2.1.2.4).

⇒ demande explicite **spécifiée**. Les primitives du service à l'interface Application-Liaison sont les suivantes:

- L_SPEC_UPDATE.demande (ID_SPEC, SUITE) permet à la couche Application de demander la mise en circulation d'une ou plusieurs variables identifiées en précisant l'ID qui permettra la transmission de la demande vers BA. Un buffer B_REQ est associé à l'identifieur spécifié. Toute demande antérieure sur ce même ID est alors écrasée.
- L_SPEC_UPDATE.confirmation (ID_SPEC, STATUS) fait suite à une demande L_SPEC_UPDATE.dem et fournit à la couche Application le compte rendu de l'échange sous forme d'un STATUS.

Pour cette option, deux modes de travail sont offerts pour satisfaire des exigences différentes (le choix est fait à la configuration):

→ mode **périodique**: l'arbitre de bus scrute cycliquement l'entité pour voir si l'entité a des IDs à envoyer vers le BA; s'il n'y en a pas, le BA passe à l'élément suivant pour continuer la scrutation, s'il y en a, après la réception de ces IDs appelé liste d'ID (trame RP_RQ1), le BA envoie ces IDs un par un donc la demande est satisfaite durant la scrutation périodique réservée pour ce service (Trqp).

→ mode **apériodique**: contrairement au mode périodique, une demande arrivée de façon aléatoire attend d'abord dans la station. En recevant l'identifieur spécifié, la demande est véhiculée vers BA par le champ de contrôle de la trame réponse (RP_DAT_RQ1) de cette variable que la station possède durant la scrutation périodique (Tcyc). Le BA mémorise cette demande et puis envoie l'identifieur (ID_RQ1) durant la scrutation apériodique (Trpa). La station lui répond en envoyant la trame réponse (RP_RQ1) qui contient la suite d'IDs demandés. Le BA envoie ces IDs un par un, la demande est donc satisfaite durant Trqa.

↳ demande explicite libre. Les primitives du service à l'interface Application-Liaison sont les suivantes:

- L_FREE_UPDATE.demande (SUITE, PRIORITE) permet à la couche Application de demander la mise en circulation d'une ou plusieurs variables identifiées. La demande est véhiculée par la première trame réponse (configurée pour ce service) qu'elle rencontre.
- L_FREE_UPDATE.confirmation (STATUS) fait suite à une demande L_FREE_UPDATE.dem et fournit à la couche Application le compte rendu de l'échange sous forme d'un STATUS.

Deux priorités sont offertes pour satisfaire des exigences différentes (le choix est fait à la configuration):

➔ priorité urgente: une demande urgente arrivée de façon aléatoire attend d'abord dans F_REQ1 de la station. La demande est véhiculée vers BA par le champ de contrôle de la trame réponse (RP_DAT_RQ1) de cette variable que la station possède durant la scrutation périodique (Tcyc). Le BA mémorise cette demande et puis envoie l'identifieur (ID_RQ1) durant la scrutation apériodique (Trpa). La station lui répond en envoyant la trame réponse (RP_RQ1) qui contient la suite d'identifieurs demandés. Le BA envoie ces identifieurs un par un la demande est donc satisfaite durant Trqa.

➔ priorité normale: une demande normale arrivée de façon aléatoire attend d'abord dans F_REQ2 de la station. La demande est véhiculée vers BA par le champ de contrôle de la trame réponse (RP_DAT_RQ2) de cette variable que la station possède durant la scrutation périodique (Tcyc) à condition que la file F_REQ1 soit vide. Le BA mémorise cette demande et puis envoie l'identifieur (ID_RQ2) durant la scrutation apériodique (Trpa). La station lui répond en envoyant la trame réponse (RP_RQ2) qui contient la suite d'IDs demandés. Le BA envoie ces IDs un par un la demande est donc satisfaite durant Trqa.

La couche Application peut donc effectuer une demande explicite à travers deux primitives différentes. La différence entre les demandes spécifiées et libres réside dans la manière d'adresser une demande d'échange explicite. L_SPEC_UPDATE.dem fixe l'identifieur sur lequel la demande doit être effectuée. Par contre, dans le cas

L_FREE_UPDATE.dem, la couche liaison véhicule la demande sur le premier ID qui arrive (et qui est configuré pour ce service).

Il convient de remarquer les points suivants:

- Une demande explicite spécifiée est satisfaite durant la scrutation périodique (Trqp) ou apériodique (Trqa) selon la configuration. Seule la priorité urgente (priorité 1) est offerte. La demande est écrasée par une nouvelle arrivée.
- Une demande explicite libre est satisfaite durant uniquement la scrutation apériodique (Trqa). Deux niveaux de priorité, urgente ou normale (priorité 2), sont possibles. Les demandes sont mises en file d'attente.

2.1.3.1 Demande explicite spécifiée périodique

Une entité configurée pour satisfaire L_SPEC_UPDATE périodique possède un ou plusieurs buffers B_REQ(ID, SUITE, ETAT) où:

- ID est l'identifieur lié au buffer avec un indicateur RQ_INHIBE = VRAI pour indiquer que cet ID est configuré pour le mode périodique.
- SUITE est la suite d'identifieurs des variables à transmettre sur le bus.
- ETAT = VRAI si le buffer est rempli d'une SUITE; sinon FAUX.

L'arbitre de bus possède un buffer de reprise B_reprise (SUITE) dont la taille maximale est 64 IDs.

Le service de demande explicite spécifiée périodique est satisfait durant la scrutation explicite périodique (Trqp).

1) Sur réception d'une primitive L_SPEC_UPDATE.dem (IDk, SUITE), l'entité producteur test sur B_REQ (ID, SUITE, ETAT) correspondant. Si ETAT = FAUX, ce buffer est vide a priori, le buffer est mis à jour avec la nouvelle SUITE et ETAT prend la valeur VRAI. Si ETAT = VRAI, le buffer est déjà occupé par une SUITE qui attend d'être envoyée vers le BA. Alors cette ancienne SUITE est écrasée par la nouvelle et la primitive L_SPEC_UPDATE.conf (IDk, écrase) est envoyée vers la couche Application. La valeur de ETAT reste VRAI.

2) BA diffuse ID_RQ1 (IDk) et arme une temporisation T1.

3) Sur réception de ID_RQ1 (IDk) pendant la scrutation périodique, l'entité producteur teste ETAT. Si ETAT = FAUX, elle ne répond rien. Si ETAT = VRAI, elle envoie la SUITE vers BA et (en même temps) L_SPEC_UPDATE.conf (IDk, ok) vers la couche Application et BA stocke la SUITE dans B_reprise.

Le service de demande explicite est terminé, on passe au service de transfert de buffer:

4) Si BA reçoit RP_RQ1, il stocke la SUITE dans B_reprise et commence tout de suite à vider ce B_reprise en envoyant les identifiants de la SUITE un par un jusqu'au dernier, la demande est satisfaite durant la scrutation périodique. Sinon, BA diffuse l'identifiant suivant, lors de l'expiration de T1.

Question 2: Quelle est la probabilité d'écrasement? Si elle est trop importante, ce service ne peut plus fonctionner correctement. Le délai d'attente d'une demande est lui aussi un paramètre intéressant. (réponse dans §3.3)

2.1.3.2 Demande explicite spécifiée apériodique

Une entité configurée pour satisfaire L_SPEC_UPDATE apériodique possède un ou plusieurs buffers B_REQ(ID, SUITE, ETAT) où:

- ID est l'identifiant lié au buffer avec un indicateur RQ_INHIBE = FAUX pour indiquer que cet ID est configuré pour le mode apériodique.
- SUITE est la suite d'identifiants des variables à transmettre sur le bus.
- ETAT = VRAI si le buffer est rempli d'une SUITE; sinon FAUX.

L'arbitre de bus possède une file d'attente F_IDRQ1(liste_ID) avec un mécanisme d'antiduplication et une file apériodique en cours F_RPRQ (SUITE). Où liste_ID est une liste composée d'IDs qui ont véhiculé une demande explicite urgente (priorité = 1).

Le service de demande explicite spécifiée apériodique est satisfait durant la scrutation explicite apériodique (Trqa).

1) A la réception d'une primitive L_SPEC_UPDATE.dem (IDk, SUITE), l'entité producteur teste le B_REQ (ID, SUITE, ETAT) correspondant. Si ETAT = FAUX, ce buffer est vide a priori, le buffer est mis à jour avec la nouvelle SUITE et ETAT prend la valeur VRAI. Si ETAT = VRAI, le buffer est déjà occupé par une SUITE qui attend d'être envoyée vers le BA. Alors cette ancienne SUITE est écrasée par la nouvelle et la primitive L_SPEC_UPDATE.conf (IDk, écrase) est envoyée vers la couche Application. La valeur de ETAT reste VRAI.

2) Lorsque l'entité où se situe B_REQ reçoit la trame ID_DAT (IDk), elle vérifie le drapeau ETAT du buffer B_REQ, si ETAT = VRAI, l'entité émet RP_DAT_RQ1 en prévenant ainsi le BA qu'elle veut une circulation apériodique. Le BA stocke IDk dans la file F_IDRQ1. Si ETAT = FAUX, l'entité émet RP_DAT normale.

3) Lorsque la scrutation périodique est finie, BA aborde la fenêtre apériodique variable. BA émet ID_RQ1 (ID) où ID est le premier dans la file fifo F_IDRQ1.

4) L'entité ayant émis une demande sur cet ID par exemple ID_k répond avec la trame RP_RQ1 (SUITE), en même temps, la couche Liaison envoie vers la couche Application L_SPEC_UPDATE.conf (ID_k, ok) en lui indiquant que la demande a été prise en compte par le BA.

Le service de demande explicite est terminé, on passe au service de transfert de buffer:

5) BA reçoit RP_RQ1 et stocke la SUITE dans F_RPRQ. Le BA commence à scruter les IDs appartenant à SUITE comme dans la fenêtre périodique en émettant ID_DAT sur chaque ID et attend la trame RP_DAT correspondante. Puis, BA passe à ID suivant dans la fifo ("fist-in, first-out") F_IDRQ1. Si F_IDRQ1 est vide, BA passe au premier ID de la file F_IDRQ2.

6) Lorsque la largeur maximale de la fenêtre apériodique variable est atteinte, le transfert en cours est suspendu par une primitive de la couche gestion, et le BA aborde une autre fenêtre.

Question 3: Quel est le délai de transfert d'une telle demande? Ce délai de transfert est composé d'un délai d'attente et d'un délai de transmission de tous les identifiieurs de la SUITE. (réponse dans §4.3)

2.1.3.3 Demande explicite libre urgente

Ce service est satisfait durant la scrutation explicite apériodique (Trqa).

Une entité configurée pour satisfaire L_FREE_UPDATE de priorité urgente possède une file d'attente F_REQ1 (ID, SUITE, PLACE) pour stocker les suites. Où:

- ID référence le premier identifieur rencontré et configuré pour ce service sur lequel la demande a été transmise au BA, on note ID_p cet identifieur. ID = NIL à l'état initial.
- SUITE est une suite d'identifieurs ou une concaténation des SUITES d'identifieurs correspondants à des demandes consécutives.
- PLACE indique le nombre de places inoccupées de la file F_REQ1. PLACE = 0 si la file F_REQ1 est saturée. On note que PLACE ≤ 64.

On note que cette file d'attente doit être munie d'un mécanisme d'antiduplication afin de supprimer les identifieurs redondants.

Le BA possède une file d'attente F_IDRQ1 avec un mécanisme d'antiduplication et une file F_RPRQ.

Suite à une primitive L_FREE_UPDATE.dem de priorité urgente:

- 1) L'entité initiatrice teste PLACE. Si PLACE < nombre d'identifieurs contenus dans la demande présente (après enlèvement des identifieurs redondants), la file F_REQ1 est "pleine", la couche liaison envoie L_FREE_UPDATE.conf(saturé) à la couche application. Sinon, on stocke la SUITE correspondant à cette demande dans F_REQ1.
- 2) En recevant un identifieur durant Tcyc, l'entité initiatrice teste l'indicateur ID. Si ID=NIL, l'entité initiatrice force ID = IDp et envoie RP_DAT_RQ1. Si ID <> NIL, F_REQ1 est déjà lié à un identifieur (IDp par exemple), l'entité initiatrice répond RP_DAT si l'identifieur reçu n'est pas IDp ou RP_DAT_RQ1 si l'identifieur reçu est IDp pour éviter le blocage due à la perte de trame RP_DAT_RQ1 avant d'arriver au BA. Les autres ID_DATs qui ne sont pas IDp sont ainsi inhibés par ID=IDp, donc ils ne peuvent plus véhiculer les demandes.

NB: La SUITE stockée dans les files F_REQ1 peut être la concaténation de plusieurs sous-suites. Chaque sous-suite est associée à une L_FREE_UPDATE.dem. La SUITE peut être ainsi augmentée même après l'émission de la trame réponse RP_DAT_RQ1. Cependant, si une sous-suite conduisait à dépasser la taille maximale de la file F_REQ1, la demande correspondante serait refusée. Dans ce cas, la couche Liaison envoie la primitive L_FREE_UPDATE.conf (saturé) à la couche Application.

- 3) En recevant les RP_DAT_RQ1, le BA stocke les ID_RQ1(IDp) dans la file F_IDRQ1. Il commence à les émettre dès que la fenêtre explicite aperiodique est atteinte.
4. Lorsque la trame ID_RQ1 arrive, l'entité émet la trame RP_RQ1, met ID = NIL et envoie autant de primitives L_FREE_UPDATE.ind (ok) que de sous-suites présentées dans une SUITE.

Question 4: Comment évaluer le délai de transfert d'une demande explicite libre de priorité urgente? (cf. ch.4)

2.1.3.4 Demande explicite libre normale

Ce service est satisfait durant la scrutation explicite aperiodique (Trqa).

Une entité configurée pour satisfaire L_FREE_UPDATE de priorité normale possède une file d'attente F_REQ2 (ID, SUITE, PLACE) avec un mécanisme d'antiduplication pour stocker les suites.

Le BA possède une file d'attente F_IDRQ2 avec un mécanisme d'antiduplication et une file F_RPRQ.

Suite à une primitive L_FREE_UPDATE.dem de priorité normale:

- 1) L'entité initiatrice teste PLACE. Si PLACE < nombre d'identifieurs contenus dans la demande présente (après enlèvement des identifieurs redondants), la file F_REQ2 est "pleine", la couche liaison envoie L_FREE_UPDATE.conf(saturé) à la couche application. Sinon, on stocke la SUITE correspondant à cette demande dans F_REQ2.
- 2) En recevant un identifieur durant Tcyc, l'entité initiatrice teste l'indicateur ID de la file F_REQ1. Si ID de F_REQ1 = NIL, la file F_REQ1 est vide, ou si ID de F_REQ1 = IDk et l'identifieur présent n'est pas IDk, l'entité initiatrice peut alors tester ID de F_REQ2. Si ID de F_REQ2 = NIL, l'entité initiatrice force ID de F_REQ2 = IDp et envoie RP_DAT_RQ2. Si ID <> NIL, F_REQ2 est déjà lié à un identifieur (IDp par exemple), l'entité initiatrice répond RP_DAT si l'identifieur reçu n'est pas IDp ou RP_DAT_RQ2 si l'identifieur reçu est IDp pour éviter le blocage due à la perte de trame RP_DAT_RQ2 avant d'arriver au BA. Les autres ID_DATs qui ne sont pas IDp sont ainsi inhibés par ID=IDp, donc ils ne peuvent plus véhiculer les demandes.
- 3) En recevant les RP_DAT_RQ2, le BA stocke les ID_RQ2 (IDp) dans la file F_IDRQ2. Il commence à les émettre dès que la fenêtre explicite apériodique est atteinte et la file F_REQ1 déjà vidée.
4. Lorsque la trame ID_RQ2 arrive, l'entité émet la trame RP_RQ2, met ID = NIL et envoie autant de primitives L_FREE_UPDATE.ind (ok) que les sous-suites présentées dans une SUITE.

Question 5: Comment évaluer le délai de transfert d'une demande explicite libre de priorité normale? (cf. ch.4)

2.1.4 Service de messagerie

Les services de messagerie permettent l'échange de messages avec ou sans acquittement.

L'initiateur de la demande d'échange de message est l'entité application (source du message). La demande est satisfaite durant la scrutation de messagerie périodique ou apériodique selon la configuration.

Un message FIP porte son adresse source (SADR) et son adresse destinataire (DADR). L'adresse SADR représente la SAP (Service Access Point) d'où le message est issu. L'adresse DADR donne le SAP auquel le message est destiné.

Les primitives du service entre la couche Liaison et Application sont:

- L_MESSAGE.dem (REFERENCE_RESSOURCE, DADR, SADR, MESS)
- L_MESSAGE.ind (DADR, SADR, MESS)
- L_MESSAGE.conf (REFERENCE_RESSOURCE, DADR, SADR, STATUS)
- L_MESSAGE_ACK.demande (REFERENCE_RESSOURCE, DADR, SADR, MESS)
- L_MESSAGE_ACK.ind (DADR, SADR, MESS)
- L_MESSAGE_ACK.conf (REFERENCE_RESSOURCE, DADR, SADR, MESS)

où REFERENCE_RESSOURCE est un identifieur lié à une ressource de messagerie. Dans le cas du transfert de message périodique, l'identifieur, configuré en messagerie périodique, se réfère à une file d'attente F_MSGcyc. Dans le cas d'un transfert de message aperiodique, cet identifieur se réfère à une file F_MSGaper allouée dynamiquement.

2.1.4.1 Service de messagerie périodique

L'entité Liaison où se situe le SAP (Service Access Point) possède une file d'attente fifo F_MSGcyc (ID, M, S, ETAT) où

- ID désigne l'identifieur lié à la file d'attente
- M a la structure "mess (SADR, DADR, MESS)"
- S est "ack" ou "noack" selon le type de message
- ETAT = 1, si la file F_MSGcyc est pleine.

Le BA possède une fenêtre de scrutation de messagerie périodique Tmp (ID_MSG).

Dans ce cas, la demande de transfert de message est alors satisfaite durant la fenêtre de messagerie périodique (Tmp) par configuration. Les ressources sont allouées statiquement à la configuration. L'entité effectue ses demandes de transfert de message directement.

Question 6: une étude des performances, i.e. délai, nombre moyen de messages dans la file et taille optimale de la file, est intéressante. Comment évaluer tous ces paramètres? (cf. ch.3)

2.1.4.2 Service de messagerie aperiodique

Dans ce cas, l'échange d'un message est plus compliqué. L'entité source doit tout d'abord manifester son intention d'émettre un message auprès de l'entité BA.

L'entité source possède:

1) une file d'attente F_MSGaper (REF, MESS, S, PLACE) où:

- REF est un tableau qui contient les identifiants déjà associés à cette file
- PLACE est le nombre de places occupées par des messages

2) un ensemble d'identifiants produits qui sont configurés pour supporter ce service:

ID (MSG) où:

- MSG = 1 si l'identifiant est déjà associé à la file F_MSGaper.

L'entité BA possède une file d'attente fifo F_IDMSG munie d'un mécanisme d'antiduplication. On stocke les identifiants qui ont véhiculé une requête de message, i.e: les identifiants ayant l'indicateur MSG=1.

Le déroulement est le suivant:

- 1) Suite à une primitive L_MSG.dem (ou L_MSG_ACK.dem), si la file F_MSGaper n'est pas pleine (PLACE < taille maximale de la file F_MSGaper), l'entité source stocke le message dans la file F_MSGaper. Sinon, le message est refusé et la couche liaison envoie à la couche application L_MSG.conf(refusé).
- 2) En recevant un identifiant configuré pour le service de messagerie aperiodique durant Tcyc, l'entité source teste l'indicateur MSG de cet identifiant. Si MSG=1, l'entité source envoie encore une fois RP_DAT_MSG pour éviter le blocage dû à la perte de la trame RP_DAT_MSG envoyée précédemment. Si MSG=0 et ssi PLACE > le nombre d'identifiants dans REF, l'entité source envoie RP_DAT_MSG.
- 3) Le BA reçoit les RP_DAT_MSG et enlève les demandes dupliquées. Il stocke les ID_MSG dans la file F_IDMSG et puis "passe la main" à l'entité source en émettant la trame ID_MSG durant la fenêtre de message aperiodique.
- 4) En recevant ID_MSG, l'entité source émet la trame réponse RP_MSG dont le champ contrôle étendu contient les adresses DADR et SADR.
- 5*) Si c'est un message avec acquittement, l'entité réceptrice envoie RP_ACK.
- 5) L'entité source "redonne la main" au BA en émettant RP_FIN, une primitive L_MSG.conf (ou L_MSG_ACK.conf) est fournie à la couche Application.

Question 7: La probabilité de rejet d'une demande est un paramètre sensible pour un utilisateur. Aucun utilisateur ne pourrait utiliser un système où ses demandes seraient très

souvent refusées. La capacité de la file F_IDMSG, la capacité d'évacuation du canal (i.e. la largeur de la fenêtre de messagerie aperiodique) et le rapport entre les deux jouent un rôle important sur les performances du service. Comment les évaluer?

2.1.5 Structure de trames et Temporisation

Pour évaluer les performances du protocole FIP, la connaissance de la structure des différentes trames et la taille des temporisations est indispensable.

2.1.5.1 Structure de trames

La structure de base d'une trame est définie par:

	<----- FSS ----->			<--- FES --->		
	Préambule	FSD EB+	Contrôle	Données	FCS	EB- FED EB-
CYC:	8bits	5+1 bits	8bits	8n (0≤n≤128) bits	16bits	7bits
REQ:			8bits	16n (0≤n≤64) bits		
MSG:			8+48 bits	8n (0≤n≤256) bits		

La taille des différentes trames est ainsi donnée par la suite:

ID_*:	40 + 21 bits
RP_DAT_*:	24+8n (0≤n≤128) + 21 bits
RP_RQi:	24+16n (0≤n≤64) + 21 bits
RP_MSG:	24+48+8n (0≤n≤256) + 21 bits
RP_ACK:	24 + 21 bits
RP_FIN:	24 + 21 bits

2.1.5.2 Temporisation

Une temporisation de référence T0 est définie comme correspondant au temps de silence maximal admissible sur un segment réseau. La temporisation T0 est un paramètre global système. Elle est liée au temps de retournement d'une station.

Cinq temporisations T1, T3, T4, T5, T6 sont définies sur la base de T0. Une temporisation supplémentaire T2 est aussi utilisée et ne fait pas référence à T0.

Tab 2.2 Temporisations utilisées dans FIP:

Nom de tempo.	Localisation	Fonction
T1 = T0	BA	Surveille l'absence de RPxx après IDxx
T2	BA	Surveille le remplissage de la fenêtre de synchro. par des bourrages
T3 = T0	BA	Surveille l'absence de IDxx après RPxx basculement des BA
T4 = T0	Station cons.	Surveille l'absence de RPxx après IDxx
T5 = 2*T0	BA	Surveille le manque de RP_FIN après ID_MSG
T6 = T0	Station source	Surveille l'absence de RP_ACK après un RP_MSG_ACK

2.1.6 Synthèse des questions

Pour atteindre les performances prévues du protocole FIP, la paramétrisation joue le rôle décisif. Le but de l'évaluation des performances est de trouver le meilleur rapport entre "temps de réponse" et "débit réel" en déterminant les paramètres sensibles tels que la largeur des fenêtres, la capacité de files d'attente, ...etc, tout en minimisant le temps d'attente. C'est pourquoi la question cruciale commune à tous les services est: comment évaluer le délai de transfert? La réponse à cette question n'est pas simple pour les services aléatoires. D'ailleurs, des configurations différentes donneront sans doute des performances différentes.

Deux catégories de problème apparaissent:

- Les problèmes "simples" vis-à-vis de l'évaluation de performances que l'on peut résoudre immédiatement (§2.2) tels que la construction de la table de scrutation, l'optimisation de la largeur des fenêtres Tmp et Trqp, le rendement du protocole, etc.
- Les problèmes "complexes" qui ont besoin d'une modélisation approfondie afin d'obtenir le délai de transfert de chacun des services, la probabilité de rejet de messages

(ou la probabilité d'écrasement d'une demande explicite spécifiée), etc. Ces problèmes font l'objet de toute notre étude.

2.2 Analyse des problèmes "simples"

2.2.1 Construction de la table de scrutation

On montre d'abord les règles à suivre pour construire une table de scrutation. On voit ensuite apparaître le besoin d'optimisation des largeurs des deux fenêtres Tmp et Trqp.

Dans une table de scrutation, les identifiants rangés sont de trois types: ID_DAT, ID_MSG et ID_RQ1 qui forment la partie statique (cf. Fig2.3). La partie non utilisée (appelée aussi temps mort) par la partie statique est utilisée pour les transactions des services aperiodiques tels que messagerie aperiodique et demande explicite aperiodique.

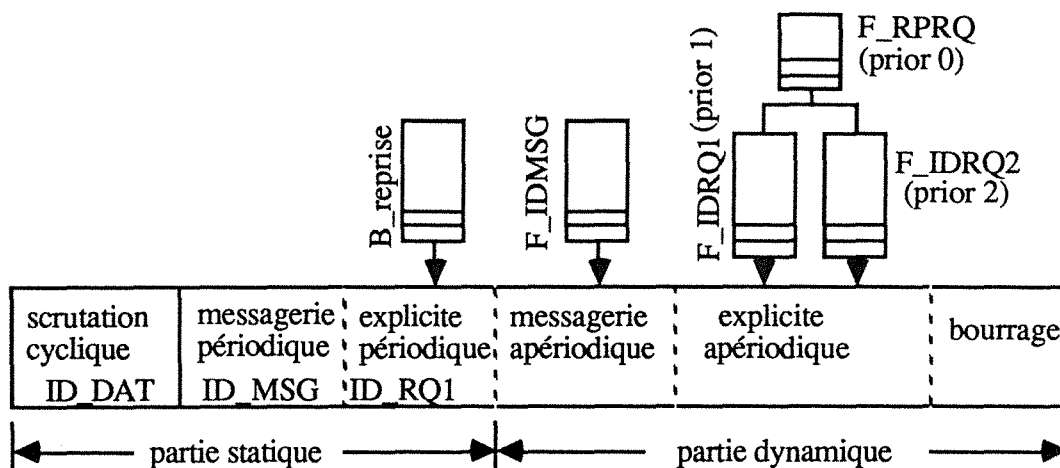


Fig.2.3 Structure et comportement d'un cycle élémentaire

Sur la figure 2.3, nous montrons le cycle élémentaire le plus complet ou "maximum". On note qu'un cycle élémentaire "minimum" ne contient qu'une fenêtre de scrutation cyclique. La fenêtre de bourrage de synchronisation intervient pour remplir le médium de transmission afin qu'il n'y ait pas une durée de silence trop longue ($>T_0$). Ces trames de bourrage ne portent aucune information utile.

La figure 2.3 met en évidence la relation entre différentes ressources qui résident dans l'arbitre de bus et ses fenêtres correspondantes. Les lignes pointillées signifient que leur positions peuvent "bouger" selon la charge correspondante. Mais une borne maximale existe.

La configuration de la table de scrutation dépend évidemment de plusieurs facteurs tels la période de chaque variable à scruter, le débit du réseau, la coordination entre un certain nombre de variables, la durée de la validité de chaque variable, ...etc. Dans ce paragraphe, nous n'avons pas l'intention de configurer une table pour une application particulière. A partir des deux paramètres les plus importants: période et débit, nous donnons quelques règles simples permettant de construire une table de scrutation simplifiée mais suffisante pour évaluer dans la suite les performances du réseau.

Etant donné le débit de transmission, la capacité du médium exprimé en nombre de bits peut être calculée par l'égalité 2.1:

$$\text{Capacité} = T_{bc} \cdot \text{Débit} \quad (2.1)$$

Bien entendu, on sait que la durée d'un cycle élémentaire se calcule de la manière suivante:

$$T_{bc} = \text{PGCD} (\text{période de scrutation de toutes les variables produites}) \quad (2.2)$$

et un macro cycle:

$$T_m = \text{PPCM} (\text{période de scrutation de toutes les variables produites}) \quad (2.3)$$

Le nombre de cycles élémentaires dans un macro cycle (T_m/T_{bc}) est un nombre entier.

En respectant ces relations, on arrive à construire un cadre ainsi qu'il est montré sur la figure 2.4.

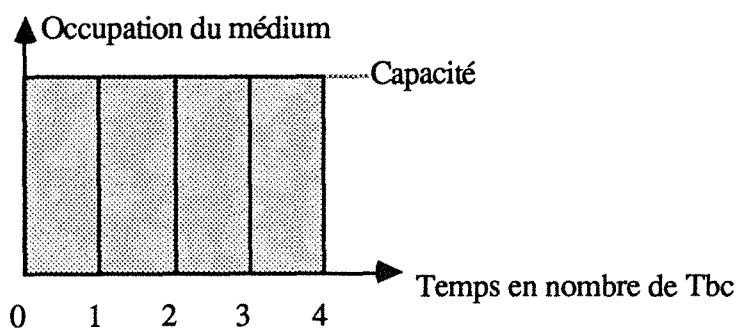


Fig.2.4 Illustration d'un cadre avec $T_m=4T_{bc}$

Maintenant, il ne reste plus qu'à le remplir en posant les trois types de paquets d'épaisseurs variables avec trois priorités différentes dans l'ordre décroissant:

ID_DAT	:	61 bits	
Temps inter trames	:	r bits	
RP_DAT_*	:	45+8n bits (n≤128)	
Temps inter trames	:	r bits	
ID_MSG	:	61 bits	
Temps inter trames	:	r bits	
RP_MSG_NOACK [RP_MSG_ACK]	:	93+8n bits (n≤256)	
Temps inter trames	:	r bits	
[RP_ACK+]	:	45 bits	
[Temps inter trames]	:	r bits	
RP_FIN	:	45 bits	
Temps inter trames	:	r bits	
ID_RQ1	:	61 bits	
Temps inter trames	:	r bits	
RP_RQ1	:	45+16n bits (n≤64)	
Temps inter trames	:	r bits	
ID_DAT(1)	}		où n est le nombre d'IDs à scruter explicitement
Temps inter trames			
RP_DAT(1)			
Temps inter trames			
.			
ID_DAT(n)	}		
Temps inter trames			
RP_DAT(n)			
Temps inter trames			

- NB:**
1. Le temps de silence inter trames maximal admissible sur un segment de réseau est noté T_0 . T_0 est lié notamment au temps de retournement. Le temps inter trames est supposé égal à r bis/Débit.
 2. Une transaction de messages avec acquittement peut être reprise plusieurs fois en cas d'erreur, i.e: RP_ACK-, RP_MSG_ACK, RP_ACK-, ..., RP_ACK+, RP_FIN.

Le rangement des identifiants de ces variables est déterminé par l'application et par l'implémenteur. Si une application a besoin de plusieurs valeurs de variables en même temps (dans un même cycle élémentaire), des problèmes sérieux se poseront car la capacité du médium risque d'être insuffisante. Une hypothèse réaliste est que la durée de validité des valeurs des variables de grandes périodes est suffisamment longue, ainsi peut-on les ranger dans des cycles élémentaires voisins (avant ou après) du cycle où elles

devraient résider. Avec ce compromis, on peut aussi réduire considérablement la longueur du macro cycle T_m . Par exemple, pour $T_m = \text{PPCM}(2, 23, 25)$, si la durée de validité des valeurs des variables dont leurs périodes sont 23 et 25 est grande, on peut prendre $T_m = 24$ au lieu de sa valeur exacte 1150. On obtient par conséquent une table de scrutation d'une dimension modeste.

[Exemple] Dans un bus de terrain FIP, on a besoin de transférer trois variables v_1 , v_2 , et v_4 avec les périodes de 1, 2 et 4 ms et les tailles de $n=2$, $n=5$ et $n=10$ respectivement. En plus, une ressource de messagerie périodique F_MSGcyc (file d'attente) est associée à la variable v_2 qui ne supporte que la messagerie sans acquittement, une ressource de demande explicite spécifiée périodique B_REQ (buffer) est associée à v_4 . On suppose que $r = 10$ bits

Selon les règles ci-dessus, on a:

$$T_{bc} = \text{PGCD}(1, 2, 4) = 1 \text{ ms}$$

$$T_m = \text{PPCM}(1, 2, 4) = 4 \text{ ms}$$

et le cadre de la figure 2.4 peut être rempli comme montré dans la figure 2.5.

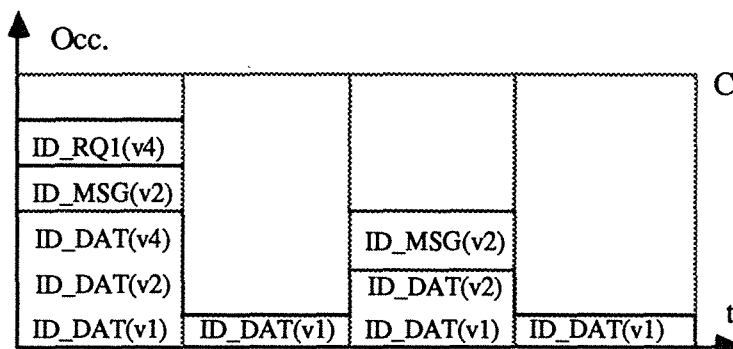


Fig. 2.5 Exemple d'une table de scrutation

Dans cet exemple, pour assurer le bon fonctionnement, le débit de transmission du réseau requis se calcule en utilisant l'égalité 2.1

$$\text{Débit} = \max_i (\text{nombre de bits dans } T_{bc}(i)) / T_{bc}; \quad i = 0, 1, T_m/T_{bc}-1$$

Il est évident que le nombre maximal de bits est la somme de bits nécessaire pour compléter les transactions de: v_1 , v_2 , v_4 , un message et un transfert de buffers explicite.

- Pour la fenêtre de scrutation cyclique qui est composée de trois transactions de transfert de variables v1, v2, v4, la dimension de cette fenêtre est:

$$\begin{aligned} T_{cyc} &= [61 + 10 + (45 + 2*8) + 10] \\ &\quad + [61 + 10 + (45 + 5*8) + 10] \\ &\quad + [61 + 10 + (45 + 10*8) + 10] \\ &= 514 \text{ bits.} \end{aligned}$$

- Pour la fenêtre de messagerie périodique qui est composée d'une transaction de transfert d'un message sans acquittement, la dimension de cette fenêtre est:

$$T_{mp} = [61 + 10 + (93 + 256*8) + 10 + 45 + 10] = 2232 \text{ bits}$$

Puisqu'on ne connaît pas la taille de message à transférer a priori, on est obligé de prendre sa taille maximale 256 octets [UTE-603 90].

- Pour la fenêtre explicite périodique qui est composée d'une transaction constituant d'un transfert d'une requête suivie par n (≤ 64) transferts des variables, la dimension de cette fenêtre est:

$$T_{rqp} = [61 + 10 + (45 + 16n) + 10] + n*[61 + 10 + (45 + 8m) + 10].$$

On a toujours $n \leq 64$ et $m \leq 128$. Mais dans notre exemple, il est évident que $n \leq 3$ et m prend sa valeur dans l'ensemble {2, 5, 10}. La dimension maximale est donc:

$$\begin{aligned} T_{rqp} &= [61 + 10 + (45 + 3*16) + 10] + [61 + 10 + (45 + 2*8) + 10] \\ &\quad + [61 + 10 + (45 + 5*8) + 10] \\ &\quad + [61 + 10 + (45 + 10*8) + 10] \\ &= 174 + 514 = 688 \text{ bits.} \end{aligned}$$

Le débit requis est:

$$\text{Débit} = (514 + 2232 + 688)/10^3 = 3,464 \text{ Mbits/s.}$$

Pour transférer 3 variables identifiées cycliquement, assurer le transfert d'un message et d'une demande explicite périodique, on a besoin d'un réseau de 3,464 Mbps comme débit. On peut s'étonner d'un tel débit. En effet, pour assurer la scrutation cyclique de trois variables, un débit de 514 Kbps suffirait. Par contre, pour assurer de plus une messagerie périodique et une demande explicite périodique, en considérant le cas le plus défavorable, un débit de 3,464 Mbps est nécessaire. On voit ici que les services de

messagerie périodique et de demande explicite périodique sont très gourmands en capacité de transmission.

Heureusement, dans la pratique il n'est pas nécessaire de prendre le débit maximal. Deux arguments mis en évidence ci-dessous nous permettent de choisir un débit plus faible que le débit maximal.

- En général, dans une fenêtre de messagerie périodique, on trouve plusieurs ID_MSG. Dans une fenêtre explicite périodique, la présence de plusieurs ID_RQ1 est aussi possible. Mais puisque les messages sont arrivés dans une station de façon aléatoire, un message n'est pas toujours présent lorsque cette station est scrutée par le BA. Et encore, même s'il se présente, sa taille n'est pas forcément la taille maximale: 256 octets. On trouve une situation similaire dans une fenêtre explicite périodique. Tout ceci peut conduire à une réduction considérable du débit requis.

- Dans le protocole FIP, le service le plus prioritaire est celui de scrutation cyclique des variables identifiées. Si les fenêtres de messagerie périodique et explicite périodique sont sous dimensionnées, lorsque la fin de ces fenêtres est atteinte, la scrutation en cours sera interrompue par un signal de synchronisation de déclenchement du cycle élémentaire suivant.

Lors de la phase de conception (implémentation), un concepteur (implémenteur) doit trouver un compromis optimal entre la "performance" et le "coût", c.à.d entre la taille de la fenêtre et le débit, en prenant en compte l'aspect aléatoire des trafics générés par les services de messagerie périodique et explicite périodique, d'où la nécessité d'optimiser les fenêtres "mp" et "rqp".

Le sous dimensionnement de ces fenêtres pose tout de même un problème lorsque toutes les files (ou buffers) sont lourdement chargées. Les files (ou buffers) rangées à la fin de la fenêtre mp (ou rqp) ne seront jamais scrutées.

Pour y remédier, on propose d'inverser l'ordre de rangement des ID_MSGs (ou ID_RQ1) alternativement pour que l'ordre de scrutation de deux cycles élémentaires (ou même deux macro cycles) consécutifs soit inversé. Cette politique éliminera le phénomène de blocage de certaines files (ou buffers). On peut même imaginer d'autres politiques similaires. Par exemple, ranger ces ID_MSGs (ou ID_RQ1s) dans un ordre pseudo-aléatoire ou dans l'ordre d'importance.

2.2.2 Dimensionnement des fenêtres de messagerie périodique (T_{mp}) et de demande explicite périodique (T_{rqp})

Dans ce paragraphe, nous montrons comment dimensionner correctement ces deux fenêtres.

2.2.2.1 Fenêtre de messagerie périodique

On suppose que M files d'attente F_MSGcyc doivent être scrutées une fois par cycle. On détermine la largeur de la fenêtre L_{mp} (en nombre d'octets) afin que la probabilité pour qu'il y ait au moins une F_MSGcyc qui n'a pas pu être scrutée durant cette fenêtre soit inférieure à un certain seuil α . On note que T_{mp} = L_{mp}/débit.

Soit

M: le nombre de stations pouvant envoyer des messages;

L_i: la taille d'un message de la station S_i avec $1 \leq L_i \leq 256$ octets (*En réalité* $(6l+r+45+r) \text{ bits} \leq L_i \leq [6l+r+m(93+8 \times 256+r+45+r)+45+r] \text{ bits}$ où m est le nombre maximal de reprises autorisées au cas où on reçoit RP_ACK-);

$\pi_i(0)$: la probabilité qu'il n'y ait aucun message dans S_i. (cf §3 pour le calcul de $\pi_i(0)$);

ρ_i : la probabilité qu'il y ait au moins un message dans S_i, $\rho_i = 1 - \pi_i(0)$.

On a alors la somme des longueurs des messages:

$$L = \sum_{i=1}^M X_i * L_i = \sum_{i=1}^M T_i \quad (2.4)$$

où:

$$X_i = \begin{cases} 1, & \rho_i, \\ 0, & 1 - \rho_i \end{cases} \quad \text{et} \quad 0 \leq T_i \leq 256$$

Dans une application FIP, les messages sont en général de petite taille. On suppose qu'il n'y a jamais de segmentation des grands messages utilisateurs, on voit alors que les conditions suivantes sont remplies:

H1) Les variables aléatoires L_i sont indépendantes;

H2) Les variables aléatoires X_i sont indépendantes;

H3) X_i et L_i sont indépendantes.

But: trouver, pour un seuil α donné ($0 < \alpha < 1$), la dimension de la fenêtre: L_{mp}(α) (en nombre d'octets) tel que:

$$P[L > L_{mp}(\alpha)] \leq \alpha \quad (2.5)$$

Démonstration: De l'équation (2.4), on déduit la fonction génératrice de L:

$$G_L(z) = \prod_{i=1}^M [1 - \rho_i + \rho_i G_{L_i}(z)] \quad (2.6)$$

On peut en déduire certains moments de L comme $E(L)$, $V(L)$ et $E(L - E(L))^3$.

Théoriquement, nous avons une solution complète du problème. Mais du point de vue pratique, le calcul est difficile à effectuer. Aussi nous proposons une méthode approchée mais simple.

Une méthode simple de calcul du $L_{mp}(\alpha)$:

Si nous prenons:

$$L_{mp}(\alpha) = t_1 + t_2 + \dots + t_M$$

Nous calculons d'abord:

$$P[T_1 \leq t_1, T_2 \leq t_2, \dots, T_M \leq t_M] = \prod_{i=1}^M P[T_i \leq t_i] \quad (\text{indépendance})$$

$$\leq P[T_1 + T_2 + \dots + T_M \leq t_1 + t_2 + \dots + t_M] = P[L \leq L_{mp}(\alpha)].$$

Pour que $P[L > L_{mp}(\alpha)] \leq \alpha$, il faut et il suffit que $P[L \leq L_{mp}(\alpha)] > (1 - \alpha)$. Puisque l'on a l'inégalité ci-dessus, il suffit que:

$$\prod_{i=1}^M P[T_i \leq t_i] \geq 1 - \alpha, \quad \text{i.e.} \quad P[T_i \leq t_i] \geq (1 - \alpha)^{1/M}$$

Puisque $P[T_i > t_i] = \rho_i * P[L_i > t_i]$, nous avons:

$$P[L_i > t_i] \leq \varepsilon / \rho_i \quad \text{inégalité qui définit le nombre } t_i \quad (2.7)$$

où

$$\varepsilon = 1 - (1 - \alpha)^{1/M} \approx \alpha / M.$$

Il suffit donc de connaître la loi de L_i et pour un seuil α donné, nous calculons t_i selon l'équation (2.7) puis $L_{mp} = t_1 + t_2 + \dots + t_M$ qui fournit une valeur par excès au seuil α . La connaissance d'une application réelle conduit à identifier la loi de L_i (de façon explicite ou par énumération, histogramme...) et permet donc de déterminer une largeur de fenêtre $L_{mp}(\alpha)$ satisfaisant la contrainte de l'équation (2.5).

2.2.2.2 Fenêtre de demande explicite périodique

Si l'on suppose que la taille des variables qu'une demande souhaite voir véhiculer est constante (m octets), on peut considérer approximativement qu'une demande

déclenchera une transaction qui durera $(n+1)m$ avec $n \leq 64$, où n est le nombre de variables.

Avec M : le nombre de stations pouvant envoyer des demandes explicites;

$L_i = m(n+1)$: la taille d'une demande de la station S_i avec $1 \leq L_i \leq 65m$;

ρ_i : la probabilité d'avoir une demande dans S_i .

On peut appliquer les formules précédentes pour obtenir la largeur de la fenêtre L_{rqp} acceptable.

Lorsque m n'est pas constante, la taille d'une variable devient une variable aléatoire supplémentaire. Le calcul de L_i serait possible moyennant la connaissance de ces variables aléatoires. Or, elle serait directement issue des caractéristiques d'une application particulière. En l'absence de telles informations, ce problème général n'a pas été traité dans ce travail.

2.2.3 Rendement du protocole FIP

Dans ce paragraphe, nous décrivons la manière de calculer un des principaux critères de performances: le rendement du protocole. Puisqu'il existe quatre types de transactions utiles: transfert de variables identifiées, de message sans acquittement, de message avec acquittement et de demande explicite, il devient évident que le calcul du rendement global du protocole FIP dépend de chaque transaction particulière. Pourtant, la méthode de calcul reste identique pour ces quatre transactions.

Le rendement est défini comme le rapport entre le nombre de bits d'informations utiles transmis et le nombre total de bits transmis, plus les temps morts entre les trames composées de la transaction.

Pour une transaction de transfert de variables identifiées:

ID_DAT	:	61 bits
Temps inter trames	:	r bits
RP_DAT_*	:	$45+8n$ bits ($n \leq 128$)
Temps inter trames	:	r bits

le rendement de cette transaction est:

$$\eta_{cyc} = \frac{8n}{61+2r+45+8n} \quad (2.8a)$$

Pour une transaction de transfert de messages sans acquittement:

ID_MSG	:	61 bits
Temps inter trames	:	r bits
RP_MSG_NOACK	:	93+8n bits (n≤256)
Temps inter trames	:	r bits
RP_FIN	:	45 bits
Temps inter trames	:	r bits

le rendement de cette transaction est:

$$\eta_{m_nak} = \frac{8n}{61+r+93+r+45+8n+r} \quad (2.8b)$$

Pour une transaction de transfert de messages avec acquittement et m reprises:

ID_MSG	:	61 bits	} m + 1
Temps inter trames	:	r bits	
RP_MSG_ACK	:	93+8n bits (n≤256)	
Temps inter trames	:	r bits	
RP_ACK-	:	45 bits	
Temps inter trames	:	r bits	
.	:		
.	:		
RP_MSG_ACK	:	93+8n bits (n≤256)	
Temps inter trames	:	r bits	
RP_ACK+	:	45 bits	
Temps inter trames	:	r bits	
RP_FIN	:	45 bits	
Temps inter trames	:	r bits	

le rendement de cette transaction est:

$$\eta_{m_ack} = \frac{8n}{61+r+(m+1)(93+8n+r+45+r)+45+r} \quad (2.8c)$$

Pour une transaction de transfert de demande explicite:

ID_RQi (i = 1, 2)	:	61 bits
Temps inter trames	:	r bits
RP_RQi	:	45+16n bits (n≤64)
Temps inter trames	:	r bits

le rendement de cette transaction est:

$$\eta_{req} = \frac{16n}{61+2r+45+16n} \quad (2.8d)$$

[Exemple] Supposons qu'un superviseur ait collecté les informations suivantes concernant la charge du réseau. Le trafic est composé de:

- 50% de transactions de transfert de variables identifiées de taille moyenne 10 octets
- 20% de transactions de transfert de messages sans acquittement de taille de 100 octets
- 10% de transactions de transfert de messages avec acquittement de taille de 200 octets
- 5% de transactions de transfert de demandes explicite de nombre moyen de 6 octets
- 15% de transactions de transfert de bourrages.

Avec le temps inter trames $r = 42 \mu s = 42 \text{ bits/1 Mbps}$ [Centre 91], si la transmission est fiable (i.e. pas de reprise de messages avec acquittement), on a le rendement global:

$$\eta = \frac{50\% \eta_{cyc} + 20\% \eta_{m_nak} + 10\% \eta_{m_ack} + 5\% \eta_{req}}{1-15\%} \approx 0,3286$$

2.2.4 Taux d'utilisation du réseau FIP

Un autre critère important est le "taux d'utilisation" ou charge réelle ou encore débit réel ("throughput") qui peut être défini dans FIP comme le rapport entre le nombre de bits véhiculés par le médium par seconde (y compris les temps inter trames équivalent en nombre de bits, les trames de bourrage ne comptent pas) et le débit du réseau. Un taux faible implique soit que le réseau est peu sollicité par les stations, soit que la technique d'accès utilise mal la capacité du réseau.

Cinq types de trafic coexistent dans un réseau FIP: trafic généré par le service de transfert cyclique, par le service de messagerie périodique, par le service de demande explicite périodique, par le service de messagerie aperiodique et par les services de demande explicite aperiodique. Comme pour le rendement, le calcul du taux d'utilisation dépend aussi de chaque type de trafic particulier. Le taux d'utilisation global est simplement la somme de tous ces taux particuliers. Pour le premier service, le calcul est simple puisque les transactions correspondantes se déroulent durant la fenêtre cyclique T_{cyc} , et que le trafic généré a une caractéristique déterministe quand la transmission est fiable. Par contre, pour les autres services, le calcul sera plus compliqué car ces derniers dépendent de nombreux facteurs tels que l'intensité du flux d'arrivée dans une station et la fréquence de scrutation de cette station par l'arbitre de bus. Nous les étudierons dans les chapitres 3 et 4.

Pour le trafic déterministe généré par le service de transfert cyclique, le taux d'utilisation peut être obtenu facilement à partir de la table de scrutation.

Le taux d'utilisation correspondant à ce trafic est alors:

$$\rho_{cyc} = \frac{1}{T_m} \sum_{i=0}^{(T_m/T_{bc})-1} T_{cyc}(i) \quad (2.9)$$

2.2.5 Fiabilité de transfert cyclique de variables identifiées

Jusqu'à présent, on a toujours supposé que la transmission dans un réseau FIP était fiable. En réalité, sur la plupart des voies de communication, il existe une probabilité d'erreur de l'ordre de 10^{-5} à 10^{-6} par bit. A l'aide de la technique de codage et de transmission par paquet (trame), cette probabilité d'erreur peut être diminuée jusqu'à 10^{-8} voire 10^{-10} par trame [Nussbaumer 87].

Par rapport à l'erreur de transmission, une autre source d'erreur appelée "erreur à la réception" est beaucoup plus importante. Il s'agit d'erreurs dues à la non-validité temporelle des données transmises.

Pour un transfert cyclique de variables identifiées, nous avons vu qu'une erreur ou perte de trames ne conduit pas en général à un défaut grave du point de vue de la couche application. Du point de vue de la couche liaison de données, l'arbitre de bus envoie tout simplement l'identifieur suivant de la table de scrutation lors de l'expiration de la temporisation T1. Ceci est peut-être vrai lorsqu'il s'agit de transmission de données d'états comme, par exemple l'échantillonnage d'états des capteurs. En effet, une erreur de ce genre de données sera détectée mais non pas corrigée. Par contre, lorsqu'il s'agit de transmettre un certain nombre de données importantes du point de vue de l'application (comme par exemple, une alarme, un signal de synchronisation ou encore une collection de données pour effectuer un calcul dans un automate afin de prendre une décision importante), une erreur ou perte d'une trame dans ce cas là conduira à un défaut grave voire catastrophique.

Pour améliorer la fiabilité de ce genre de transferts, FIP adopte un mécanisme de reprise dans le cas où une erreur est détectée au niveau application.

Un service de la couche application dit "service d'accès à liste de variables" exige la cohérence temporelle (i.e: toutes les variables reçues par un consommateur de cette liste doivent être validées avant d'être consommées) et spatiale (i.e: tous les consommateurs répartis doivent posséder la même liste, ceci est réalisé par l'échange de

quelques variables de cohérence spatiale). Pour augmenter la fiabilité de "multicast" d'une liste de variables, chaque consommateur a la possibilité de déclencher une demande explicite de transfert d'une liste de variables mal reçues. Ainsi réalise-t-on la retransmission des variables erronées.

Habituellement, on considère que la technique de retransmission avec acquittement dans un protocole de communication "multicast" gaspille du temps. Contrairement à ce qui est fait dans la communication point à point pour laquelle la technique de retransmission est bien étudiée sous plusieurs noms comme "Go-Back-N ARQ", "Stop-and-Wait ARQ" et "Selective-Repeat ARQ", très peu d'études ont été faites pour les protocoles "multicast". En effet, l'acquittement individuel habituel (c'est à dire que chaque récepteur envoie un acquittement à la réception d'une variable) est un gaspillage de temps. Dans FIP, puisque l'on se situe dans des applications à temps critique, l'acquittement groupé (c'est à dire qu'un récepteur envoie un acquittement seulement lorsqu'il reçoit la liste de variables) est adopté pour économiser du temps.

Un exemple ci-dessous (fig.2.6) montre clairement la relation étroite entre le service de la couche application: accès à une liste de variables, et le service de la couche liaison de données: demande explicite spécifiée périodique.

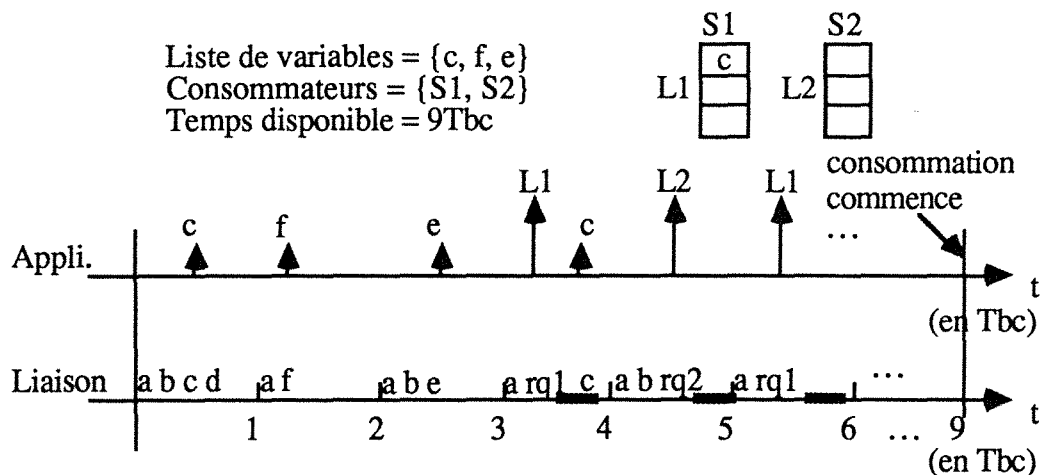


Fig. 2.6 Relation entre "accès à liste" et "explicite spécifié périodique"

Un modèle général est étudié en détail. Il s'agit d'une étude de fiabilité de plusieurs stratégies de retransmission proposées pour la communication "multicast" dans un réseau à temps critique [Simonot 90-4].

Le modèle de la stratégie qui correspond exactement à notre exemple dans la figure 2.6 est montré dans la figure 2.7.

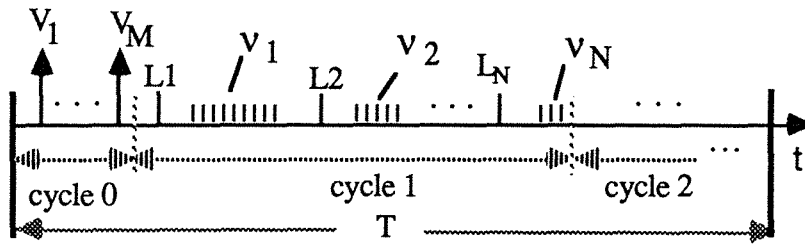


Fig. 2.7 Modèle général de la stratégie "acquittement groupé"

Dans la figure 2.7, M est le nombre de variables dans la liste et N le nombre de consommateurs de la liste. Et on suppose que toutes les trames circulant dans le réseau ont une taille constante: 1 slot.

Il est évident que sans erreur, cette stratégie (avec acquittement groupé) consomme M+N slots de temps tandis qu'une stratégie banale (avec acquittement individuel) en consomme MN slots.

Nous nous intéressons aux deux variables suivantes:

- v: nombre de variables à retransmettre durant le cycle 1, avec $v = v_1 + v_2 + \dots + v_N$.
- Y: le nombre de variables erronées chez au moins un consommateur après la première retransmission (i.e: à la fin du cycle 1).

Si on ignore la petite probabilité d'erreur de transmission et si on considère seulement la probabilité d'erreur à réception p (et $q=1-p$) chez un consommateur, on peut obtenir les inégalités suivantes [Simonot 90-4]:

$$E(X_1) \leq E(v) \leq E(X_1) + M(N-1)(1-\pi_0) \tag{2.10}$$

$$E(X_2) - \sqrt{E[(X_2)^2]} \sqrt{1-\pi_0} \leq E(Y) \leq E(X_2) \tag{2.11}$$

où:

$$\pi_0 = (q^N + Npq^{N-1})^M$$

$$E(X_1) = M(1-q^N)$$

$$E(X_2) = M(1-q^N)^2$$

$$E[(X_2)^2] = M(1-q^N)^2[(M-1)((1-q^N)^2 + 1)]$$

[Exemple]

On suppose que M=10, N=2 and p=0.1. D'après les inégalités (2.10) et (2.11), on peut déterminer le nombre moyen d'erreurs après la première retransmission: E[Y]

$$E[X_2] = 10(1-(1-0.1)^2)^2 = 0.3610$$

$$1 - \pi_0 = 0.0956$$

On déduit les bornes:

$$0.1472 \leq E[Y] \leq 0.3610.$$

et le temps écoulé t :

$$t = M + E(v) + N$$

$$E[X_1] = 1.9 \leq E(v) \leq 2.856$$

$$13.9 \leq t \leq 14.856$$

Le nombre moyen d'erreurs avant la retransmission était $E[X_1] = 1.9$ et après la retransmission $E[Y] \leq 0.3610$. Le temps écoulé pour transmettre M variables est $M=10$, plus les acquittements de $N=2$ consommateurs, plus le temps pour effectuer la retransmission $E(v)$, ce qui donne les bornes inférieure et supérieure de t .

2.3 Modélisation des problèmes complexes

Dans ce paragraphe, on analyse les problèmes service par service. Différent du service de transfert cyclique de variables, ces services sont déclenchés sur demande invoquée par la couche application. Les modèles de performances en sont déduits. Ce sont des modèles en files d'attente non classique car le serveur est intermittent. La disponibilité est décrite par un interrupteur $I_{\text{service}}(t)$ qui ferme lorsque $I_{\text{service}}(t) = 1$. On définit:

pour $0 \leq t < T_m$

$$I_{\text{service}}(t) = I_{\text{service}}(t+kT_m) = \begin{cases} 1, & \text{durant un certain moment dans } T_{\text{service}} \\ 0, & \text{ailleurs} \end{cases}$$

où "service" prend la "valeur" dans l'ensemble {cyc, mp, rqp, rqa, rqu, rqn, msg}.

Le temps de service est toujours proportionnel à la longueur des trames.

Dans le §2.3.1, on établit un modèle exact afin de répondre ensuite à la question 2 (cf.§2.1.3.1).

Dans le §2.3.2, on établit un modèle exact afin de répondre à la question 6 (cf.§2.1.4.1).

Dans le §2.3.3, on établit un modèle exact et puis un modèle approximatif afin de répondre à la question 7 (cf.§2.1.4.2).

Dans le §2.3.4, on établit un modèle approximatif afin de répondre à la question 3.

Dans le §2.3.5, on établit un modèle approximatif afin de répondre aux question 4 et 5 (cf.§2.1.3.3 et §2.1.3.4).

2.3.1 Service de demande explicite spécifiée périodique

Le service de demande de transfert de buffer explicite spécifiée périodique a besoin d'un buffer de requête B_REQ. ID_RQ1 arrive périodiquement selon la table de scrutation (durant la fenêtre explicite périodique). Mais les demandes arrivent aléatoirement. Si la trame ID_RQ1 arrive et trouve B_REQ vide, le BA envoie le ID_RQ1 suivant après l'expiration de temporisation T1; s'il trouve B_REQ non vide, la station source répond en émettant la trame RP_RQ1 (SUITE), où SUITE est une liste d'identifieurs dont le nombre varie entre 0 et 64. Cette SUITE sera stockée dans le buffer B_reprise. Ces identifieurs sont ensuite envoyés immédiatement, un par un, durant la fenêtre explicite périodique jusqu'à ce que B_reprise soit vide.

Le modèle exact de ce service est donné dans la figure 2.8

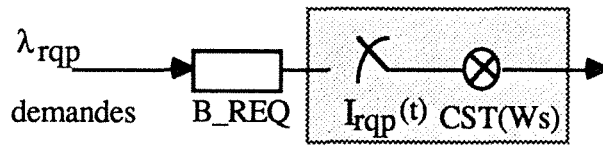


Fig 2.8 Modèle du service de demande explicite spécifiée périodique

C'est un modèle de file d'attente avec serveur intermittent où encore un système autonome. La disponibilité du serveur est décrite par un interrupteur $I_{rqp}(t)$ qui ferme lorsque $I_{rqp}(t) = 1$. On définit:

pour $0 \leq t < T_m$

$$I_{rqp}(t) = I_{rqp}(t+kT_m) = \begin{cases} 1, & \text{durant un certain moment dans } T_{rqp} \\ 0, & \text{ailleurs} \end{cases}$$

Le service commence de manière autonome, indépendamment du flux d'entrée des demandes.

La durée de service W_s est égale à la durée d'une transaction pour compléter une demande explicite spécifiée périodique, i.e: le temps de transmission d'une "macro trame":

$$\{(ID_RQ1+r+RP_RQ1+r) + [(ID_DAT(1)+r+RP_DAT(1)+r + \dots + ID(n)+r+RP_DAT(n)+r)]\}.$$

CST(x) signifie que le temps de service est constant proportionnel à x.

λ_{rqp} est l'intensité du flux d'arrivée des demandes.

La capacité du buffer B_REQ est limitée à 1. Une nouvelle demande écrasera l'ancienne en attente dans le buffer.

On s'intéresse à trois paramètres:

- La probabilité que B_REQ soit vide
- La probabilité d'écrasement
- Le temps moyen d'attente d'une demande.

Ces trois paramètres sont évalués dans le chapitre 3.

2.3.2 Service de messagerie périodique (avec ou sans acquittement)

Le service de transfert de messages périodique a besoin d'une file d'attente fifo (first in first out): F_MSGcyc. ID_MSG arrive périodiquement avec au moins la période T_m selon la table de scrutation (durant la fenêtre messagerie périodique). Les messages arrivent aléatoirement. Si la trame ID_MSG arrive et trouve F_MSGcyc vide, l'entité source envoie la trame RP_FIN; si la trame ID_MSG trouve F_MSGcyc non vide, l'entité source lui répond par l'envoi du premier message en attente dans la file F_MSGcyc (trame RP_MSG_ACK ou RP_MSG_NOACK). S'il s'agit d'un message sans acquittement, l'entité source envoie vers l'arbitre de bus une trame RP_FIN pour terminer cette transaction. S'il s'agit d'un message avec acquittement, l'entité source attend une réponse RP_ACK+ ou RP_ACK-. Si l'entité source reçoit RP_ACK+, elle envoie une trame RP_FIN pour terminer la transaction. Si la réponse est RP_ACK-, des reprises sont possibles avec un nombre limité. Elle envoie RP_FIN lorsqu'une réponse RP_ACK+ est enfin arrivée ou le nombre maximal de reprises autorisé est atteint.

Le modèle de ce service est montré dans la figure 2.9.

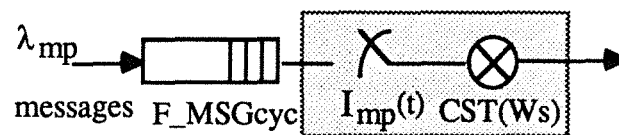


Fig. 2.9 Modèle du service de messagerie périodique

C'est un système autonome. La durée de service W_s est égale à la durée d'une transaction de transfert périodique d'un message, i.e: le temps de transmission d'une

"macro trame": {(ID_MSG+RP_MSG_{xx}+RP_FIN)}, où RP_MSG_{xx} peut être soit RP_MSG_NOACK, soit (RP_MSGACK+RP_ACK+).

λ_{mp} est l'intensité du flux d'arrivée des messages.

$I_{mp}(t)$ représente la disponibilité du serveur qui ferme lorsque $I_{mp}(t) = 1$. On définit:

pour $0 \leq t < T_m$

$$I_{mp}(t) = I_{mp}(t+kT_m) = \begin{cases} 1, & \text{durant un certain moment dans } T_{mp} \\ 0, & \text{ailleurs} \end{cases}$$

La capacité de la file F_MSG_{cyc} est limitée à L. Un nouvel arrivé sera rejeté s'il trouve L messages devant lui (la file est déjà pleine).

On s'intéresse aux trois paramètres:

- La probabilité que F_MSG_{cyc} soit vide
- La probabilité de rejet
- Le temps moyen d'attente d'un message.

Ces paramètres sont évalués dans le chapitre 3.

2.3.3 Service de messagerie apériodique (avec ou sans acquittement)

On suppose que la transmission est fiable, et la réception toujours correcte. Le mécanisme de reprise en cas d'erreur ne sera donc pas examiné. De plus, on ne distingue pas les messages avec acquittement et sans acquittement. Lorsqu'on parle d'une transaction de transfert d'un message, il n'y aura alors que ID_MSG et RP_MSG (RP_FIN et les temps inter trames sont incorporées dans RP_MSG).

Le service de transfert de messages apériodique nécessite une file d'attente F_MSG_{Gaper} dans la station émettrice, une file d'attente F_IDMSG munie d'un mécanisme d'antiduplication dans l'arbitre de bus, une file d'attente F_MSG_{reçu} dans la(les) station(s) destinataire(s) et une fenêtre de messagerie apériodique qui fournit un canal de transmission pour ce trafic. La dimension de cette fenêtre dans un cycle élémentaire quelconque ne doit jamais dépasser le temps mort laissé par les trafics cycliques (i.e: la partie dynamique, cf. fig.2.1 ou fig.2.3).

Une transaction de ce service est composée de deux phases:

1) prévenir l'arbitre de bus de la présence d'une demande d'émission d'un message de la station source par ajout d'un ID_MSG dans la file F_IDMSG. Cette demande est

véhiculée par le champ de contrôle de la première trame RP_DAT qu'elle rencontre durant T_{cyc} .

2) le transfert effectif du message déclenché par la réception de ID_MSG durant T_{msg} . Cet ID_MSG est celui qui est transmis durant la première phase ci-dessus.

La figure 2.10 montre clairement la transaction de ce service par le diagramme temporel.

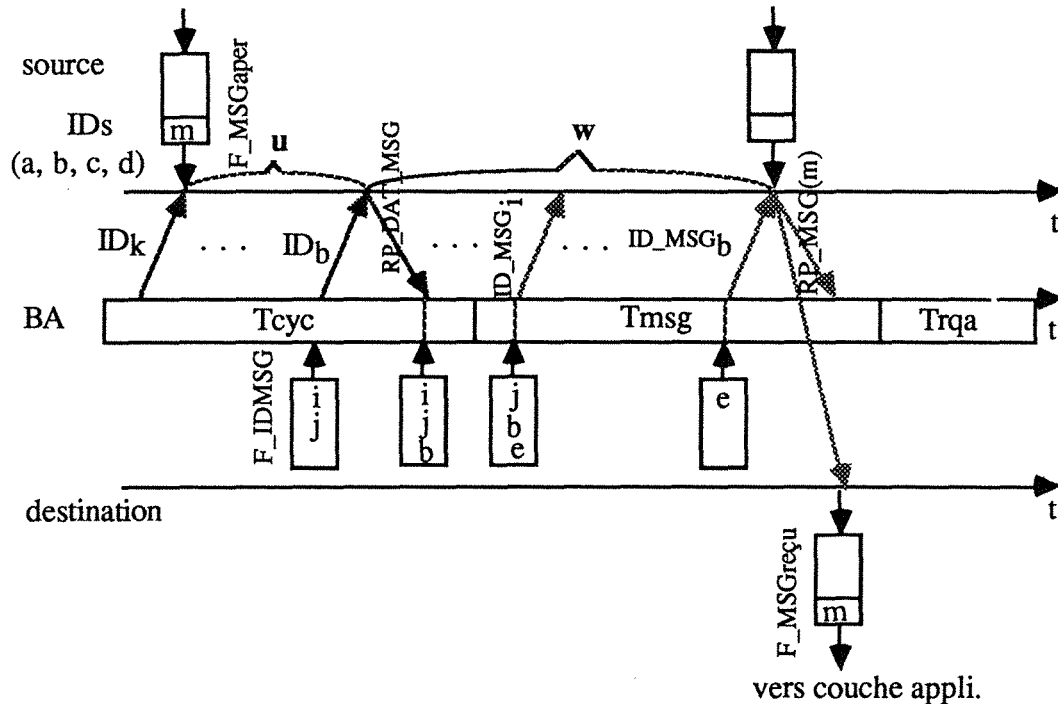


Fig. 2.10 Diagramme temporel d'une transaction de messagerie apériodique

Un message "m" arrivé dans la station source est stocké d'abord dans la file $F_MSGaper$. Ce message attend un délai "u" avant de recevoir ID_b , l'un des quatre identifiants (a, b, c, d) disponibles de cette station source. La station source prévient le BA de la demande de droit de parole supplémentaire en répondant RP_DAT_MSG . Le BA stocke ID_MSG_b , puis commence à diffuser les $IDMSGs$ dès que la fenêtre T_{msg} est atteinte. La station source transmet alors le message "m" en recevant ID_MSG_b . La station destinataire reçoit ce message "m" après un délai de transmission " W_s ". On voit aussi que le délai d'attente d'un message dans la file $F_MSGaper$ est composé de deux délais u et w, et le délai total est $u+w+W_s$.

L'établissement d'un modèle de performances de ce service n'est pas simple. De plus, la résolution de ce dernier n'est pas évidente. Le fait de stocker le message dans $F_MSGaper$, d'envoyer une demande de ce message vers l'arbitre de bus et d'attendre un

ID_MSG nous oblige à établir un système de files d'attente avec des retours (feedback) des trames (cf. figure 2.11).

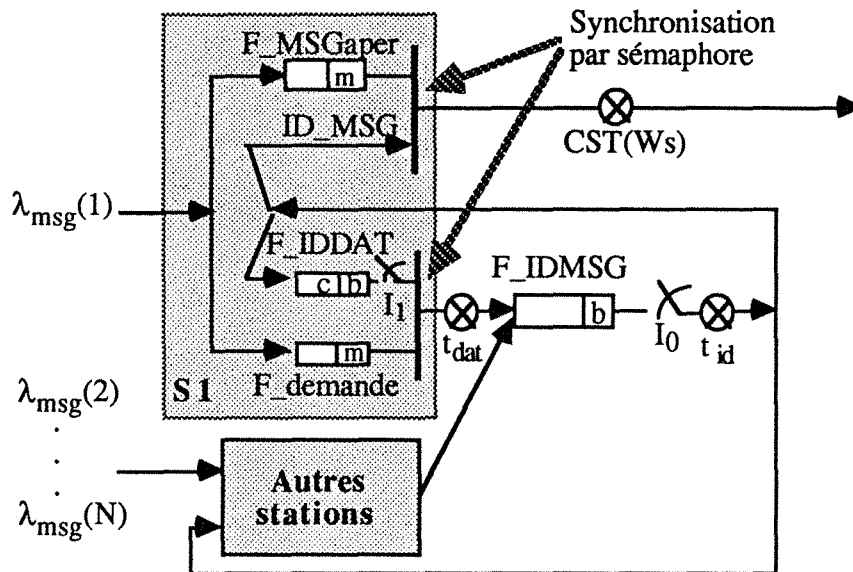


Fig. 2.11 Modèle exact du service de messagerie aperiodique

Sur la figure 2.11, un message entré dans la station source S_1 crée deux trames:

- une trame message est stockée dans la file d'attente $F_MSGAper$
- une autre trame "fictive" appelée trame de demande est stockée dans la file d'attente $F_demande$ afin de demander le droit d'émission de ce message.

La trame de demande sera transportée par une trame de réponse du service de transfert cyclique. Chaque station possède un certain nombre d'identifieurs qui agissent comme des transporteurs des demandes. Ces identifieurs sont stockés dans la file F_IDDAT . Un mécanisme de synchronisation de type sémaphore [Fdida 86] et [Mailles 87] est adopté. Une demande en attente dans $F_demande$ ne peut quitter cette file que lorsqu'il y a encore une trame dans F_IDDAT et que l'interrupteur $I_1(t)$ est fermé. Cette demande est ensuite stockée comme une trame ID_MSG dans la file F_IDMSG qui se situe dans la station BA. Cette trame ID_MSG retournera dans la station source durant l'intervalle de temps où l'interrupteur $I_0(t)$ ferme. En arrivant dans la station source, cette trame ID_MSG en crée de nouveau deux:

- une trame redevient la trame ID_DAT , et entre dans la file F_IDDAT . Elle est donc de nouveau disponible pour transporter une autre demande;

- l'autre autorise le message en attente en tête de la file F_MSG aper de quitter la file.

Le message est enfin arrivé à la destination en traversant le médium de transmission qui est représenté par un temps de service proportionnel à la longueur de cette trame de message.

On note qu'il est aussi nécessaire de distinguer des trames de demande provenant des différentes stations sources afin de les acheminer vers les endroits d'où elles sortent.

L'étude analytique de ce genre de système est délicate. Par contre, une étude par simulation est faisable (cf. §5).

Heureusement, nous avons constaté que la circulation d'une trame ID_MSG est toujours suivie par une circulation de la trame RP_MSG . Si la transmission est fiable, on peut considérer le couple (ID_MSG+RP_MSG) comme une seule trame et on l'appelle trame MSG . Cela nous permet de supprimer un retour de ID_MSG vers la station source. De plus, on peut très bien imaginer qu'un message entré dans sa file d'émission, au lieu de manifester sa présence par l'envoi d'une demande vers le BA et attendre une invitation (ID_MSG), est transporté directement vers le BA par le champ contrôle d'une trame RP_DAT_MSG . Et on laisse une des places de F_MSG aper correspondante en état bloqué tant que ce message n'est pas sorti de la file F_IDMSG . Dans ce cas, la file F_IDMSG contiendra des trames MSG au lieu des trames ID_MSG . On nomme cette file " F_MSG ". Le dual (F_MSG aper, ID_MSG) est ainsi disparu de la figure 2.11, et on obtient un modèle plus simple avec blocage (cf. fig.2.12).

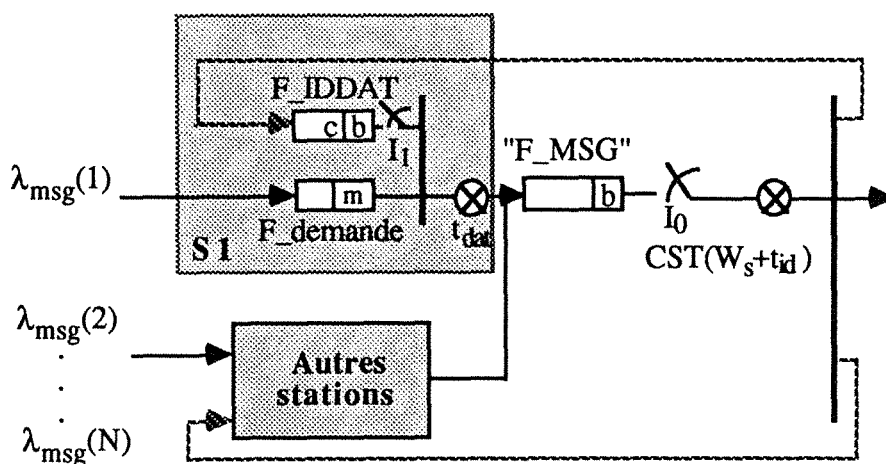


Fig. 2.12 Modèle de messagerie a périodique avec blocage

Ce modèle ne change absolument pas le délai de transfert (délai total) d'un message. En effet, au lieu d'attendre toujours dans la file $F_MSGaper$ de la station source, un message attend maintenant un délai u (cf. fig.2.10) dans la file d'attente $F_demande$, puis un délai w (cf. fig.2.10) dans la file F_MSG de l'arbitre de bus. Une place de la file $F_MSGaper$ reste bloquée tant que ce message n'est pas complètement sorti.

Remarque: Puisque le contrôle du flux de messages apériodiques s'effectue par le nombre limité d'identifieurs configurés pour ce service et la capacité des files d'attente $F_MSGaper$, la dimension de la file F_IDMSG se calcule alors de la manière suivante:

$$\text{Taille de } F_IDMSG = \min\{\text{nombre total de places des } F_MSGaper, \\ \text{nombre total d'identifieurs configurés pour ce service}\}.$$

Avec ce dimensionnement, il est certain qu'il n'y aura jamais de dépassement de cette file.

Un modèle avec blocage n'est pas facile à étudier. A titre d'exemple, on analyse un cas très simple de ce genre de modèle sur la figure 2.13.

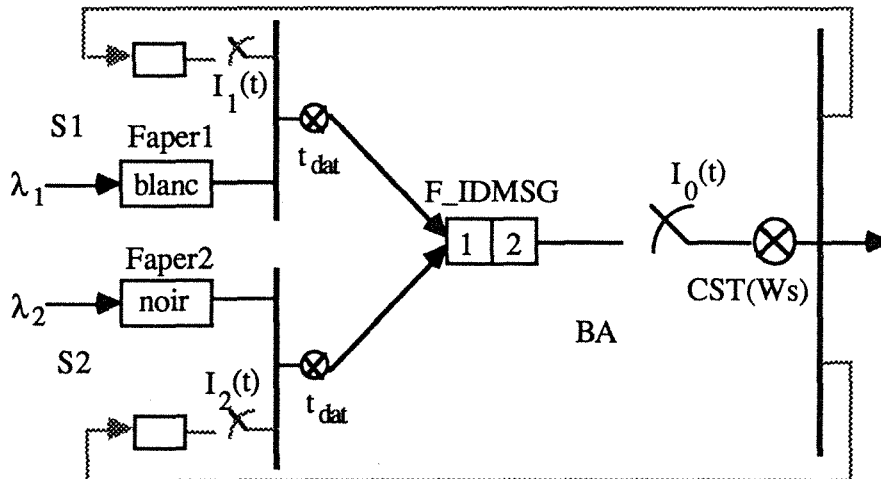


Fig. 2.13 Modèle avec blocage (deux stations)

Sur la figure 2.13, on peut voir deux stations. Chaque station source S_i possède une file $F_MSGaper$ d'une seule place et un seul identifieur. On note les $F_MSGaper$ comme $Faper1$ et $Faper2$ colorés en blanc et noir. Une file F_IDMSG de capacité limitée à 2 suffira. On considère qu'un message, au lieu de rester dans $Faper$ pour attendre ID_MSG , est transporté directement vers F_IDMSG lorsque l'interrupteur de cette

station est fermé. Faper reste bloqué tant que ce message (avec une couleur pour indiquer la provenance de celui-ci) n'est pas sorti de la file F_IDMSG. On suppose que les messages ont une taille constante: 1 slot, et les transporteurs RP_DAT aussi. Les flux d'arrivée sont Poissonniens avec les intensités λ_1 et λ_2 , et les interrupteurs $I_0(t)$, $I_1(t)$ et $I_2(t)$ réagissent de la manière suivante (figure 2.14).

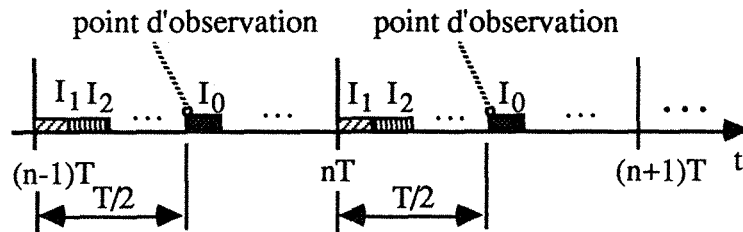


Fig.2.14 Moments de fermeture des interrupteurs

On définit:

- Etat du système: $E_i(n) = (\text{Faper1}, \text{Faper2}, \text{F_IDMSG}(1), \text{F_IDMSG}(2))$,
où i est le numéro de l'état,
et n est l'instant d'observation juste avant le départ d'un client (message).
- Probabilité de transit d'états: $p_{ij}(n) = P[E_j(n+1) \mid E_i(n)]$.

En régime stationnaire, les $p_{ij}(n)$ sont indépendantes de l'instant n .

En observant globalement le système aux instants d'observation (cf. Fig.2.14), on a dix états possibles suivants:

$$\begin{aligned}
 E_1 &= (0, 0, 0, 0); E_2 = (b, 0, 0, 0); E_3 = (0, n, 0, 0); E_4 = (0, 0, 0, b); \\
 E_5 &= (0, 0, 0, n); E_6 = (b, n, 0, 0); E_7 = (b, 0, 0, n); E_8 = (0, n, 0, b); \\
 E_9 &= (0, 0, b, n); E_{10} = (0, 0, n, b).
 \end{aligned}$$

La matrice des probabilités de transition d'états \mathbf{P} est donnée ci-dessous:

	1	2	3	4	5	6	7	8	9	10
1	a^2b^2	$a^2(1-b^2)$	$b^2(1-a^2)$	0	0	$(1-a^2)(1-b^2)$	0	0	0	0
2	0	0	0	b^2	0	0	0	$1-b^2$	0	0
3	0	0	0	0	a^2	0	$1-a^2$	0	0	0
4	ab^2	$b^2(1-a)$	$b(1-b)a$	0	0	$b(1-a)(1-b)$	0	0	0	0
			$+a(1-b)$			$+(1-b)(1-a)$				
5	a^2b	$a^2(1-b)$	$a(1-a)b$	0	0	$a(1-b)(1-a)$	0	0	0	0
			$+b(1-a)$			$+(1-a)(1-a)$				
6	0	0	0	0	0	0	0	0	1/2	1/2
7	0	0	0	b	0	0	0	$1-b$	0	0
8	0	0	0	0	a	0	$1-a$	0	0	0
9	0	0	0	b	0	0	0	$1-b$	0	0
10	0	0	0	0	a	0	$1-a$	0	0	0

où:

$$a = \exp(-\lambda_1 T/2), b = \exp(-\lambda_2 T/2).$$

En régime stationnaire, on a:

$$\pi = \pi P \tag{2.12}$$

où π est un vecteur des distributions de probabilités stationnaires, on peut ainsi en déduire les paramètres intéressants: délai total, probabilité de rejet etc...

Sur un simple exemple, on peut facilement constater la complexité de la résolution du système. Dans le cas général où:

- 1 - le nombre de stations n'est plus limité à 2,
- 2 - le nombre de places d'une file $F_MSGaper$ n'est plus limité à 1,

on constate que le nombre d'état possibles s'accroît très rapidement et que la méthode décrite ci-dessus devient inexploitable, même si ce genre de modèle dit avec blocage a déjà été étudié et a donné certains résultats [Gross 85] et [Perros 89]. Nous nous apercevons que cette modélisation est en fait peu adaptée à la résolution de notre problème. La difficulté tient au fait que nous sommes obligés de distinguer les trames dans chaque file, ce qui contribue à l'augmentation du nombre d'états.

Dans la suite, on propose un modèle approximatif sans blocage en vue de résoudre le problème. L'idée est d'analyser séparément le problème pour deux cas: **trafic faible** et **trafic fort**.

En effet, pour un message provenant d'une station source quelconque, le délai moyen d'attente dans la file d'attente commune F_IDMSG est identique. On n'a pas

besoin de distinguer les messages de différentes provenances (indiqué par des couleurs différentes dans notre modèle avec blocage) à condition qu'il n'y ait aucune occurrence de phénomène de "blocage du à la pénurie d'identifieurs (transporteurs des demandes). La réalisation de cette condition suppose un nombre de transporteurs infini, ou de façon plus réaliste, l'intensité du flux d'arrivée de message dans une station source λ_i est faible pour qu'on puisse considérer que le nombre de message dans la file F_MSGaper est très faible devant le nombre d'identifieurs configurés pour ce service.

En résumé, le modèle approximatif est quasi exact dans le cas où le trafic est suffisamment faible par rapport à la capacité d'évacuation de la file F_MSGaper. Un message est déjà sorti avant qu'un nouveau arrive ce qui ne conduit pas à la pénurie d'identifieurs.

Dans ce cas, nous pouvons étudier un modèle sans blocage (Fig.2.15) au lieu d'un modèle avec blocage.

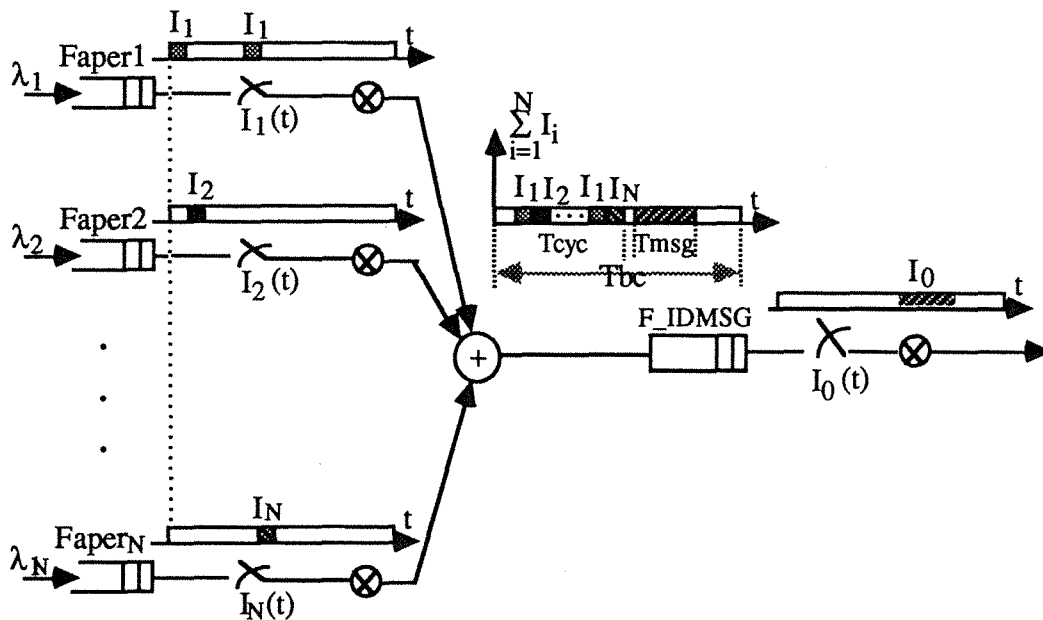


Fig.2.15 Modèle approximatif (sans blocage)

C'est un modèle de files d'attente en parallèle (les files du premier étage) et série (en tandem avec la file du deuxième étage).

En fait, trois paramètres "physiques" interviennent sur les performances de ce modèle:

- la capacité de la file F_MSGaper,

- le nombre d'identifieurs configurés pour ce service durant un cycle élémentaire T_{bc} ou un macro cycle T_m (i.e: la capacité d'évacuation du premier étage),
- la largeur de la fenêtre de messagerie aperiodique T_{msg} (i.e: la capacité d'évacuation du deuxième étage).

Pour mieux exploiter la capacité du médium de transmission ou plus précisément la capacité d'évacuation de messages de la fenêtre de messagerie aperiodique, on configure le système de telle manière qu'il n'y ait pas de goulet d'étranglement à l'issue du premier étage (la situation de famine du support de transmission n'existe pas). Le résultat est donc que les files du premier étage sont toujours faiblement chargées tant que la condition de stationnarité est respectée.

La condition de stationnarité dépend de la capacité d'évacuation du deuxième étage ou plus précisément de la dimension de la fenêtre de messagerie aperiodique. Si la taille moyenne d'un message est m , cette condition peut s'exprimer par l'équation suivante:

$$\rho_{msg} = \frac{T_m \left(\sum_{j=1}^N \lambda_j \right)}{(T_m/T_{bc}) - 1 - \sum_{i=0} (L_{msg}(i)/m)} < 1 \quad (2.13)$$

i.e: le nombre de messages arrivés durant un macro cycle T_m ne dépasse pas la capacité d'évacuation des messages de la file du deuxième étage.

L'avantage d'un modèle sans blocage réside dans le fait qu'on peut étudier les deux étages séparément.

Dans le cas du trafic fort, le délai d'attente w dans F_{IDMSG} augmente. Si dans une station source, le nombre d'identifieurs qui peuvent transporter des demandes est largement supérieur au nombre de places dans $F_{MSGaper}$, le délai u dans $F_{MSGaper}$ deviendra négligeable par rapport à w , et l'étude de la seule file du deuxième étage suffira. Si le nombre d'identifieurs qui peuvent transporter une demande est nettement inférieur au nombre de places dans $F_{MSGaper}$, il y aura de plus en plus de messages qui resteront bloqués dans $F_{MSGaper}$. Quand le système est stable, le flux d'entrée du deuxième étage tend vers une limite, et le délai d'attente dans la file F_{IDMSG} aura sa

valeur maximale. On peut connaître l'intensité du flux d'entrée du deuxième étage, l'évaluation du délai maximal sera possible (cf. §5).

2.3.4 Service de demande explicite spécifiée aperiodique

Le service de demande de transfert de variables explicite spécifiée aperiodique nécessite les ressources suivantes:

- un buffer de requête B_REQ dans la station source,
- deux files d'attente dans l'arbitre de bus F_IDRQ1 qui contient des ID_RQ1 et F_RPRQ qui contient la suite d'identifieurs,
- une fenêtre explicite aperiodique dont la dimension ne peut jamais dépasser le temps mort laissé par le trafic periodique.

On signale que cette fenêtre est partagée entre ce service et le service explicite libre.

Une transaction de ce service est composée de deux phases:

- 1) indication à l'arbitre de bus de la présence d'une demande en ajoutant un ID_RQ1 dans la file F_IDRQ1.
- 2) le transfert effectif de la SUITE d'identifieurs (RP_RQ1), puis la scrutation des variables correspondantes.

NB: On aurait pu dissocier le transfert de SUITE et l'émission des variables correspondantes. Pour les mêmes raisons qu'au paragraphe précédent (§2.3.3), nous les associons en une phase unique.

Pour accomplir la première phase, une variable identifiée est spécifiée pour transporter dans son champ contrôle la demande d'accès (ID_RQ1) vers l'arbitre de bus ou plus précisément dans la file F_IDRQ1. Puisque cette variable est reçue en général periodiquement par la station source, le modèle de cette phase est similaire à celui du service de demande explicite periodique.

En ce qui concerne la deuxième phase, le transfert effectif se déroule durant la fenêtre aperiodique. Il est possible que le transfert des variables de la SUITE soit interrompu à cause de l'expiration de cette fenêtre et soit repris dans la fenêtre aperiodique suivante à l'aide de deux primitives de la couche gestion [UTE C46-605 90]: L_SUSPEND_* et L_CONTINUE*.

Si on examine ce qui se passe dans la file F_IDRQ1, on s'aperçoit qu'il existe une influence entre le trafic généré par ce service et celui de demande explicite libre de priorité urgente (même priorité que "spécifié"). Il sera difficile d'évaluer ses performances à

partir d'un modèle exact (dépendance entre les files d'attente, blocage, interférence entre différents services, ...) sauf par simulation (cf. ch.5).

Grâce à une analyse similaire à celle du service de messagerie aperiodique, on propose un modèle approximatif pour l'évaluation du délai de transfert (Fig.2.16).

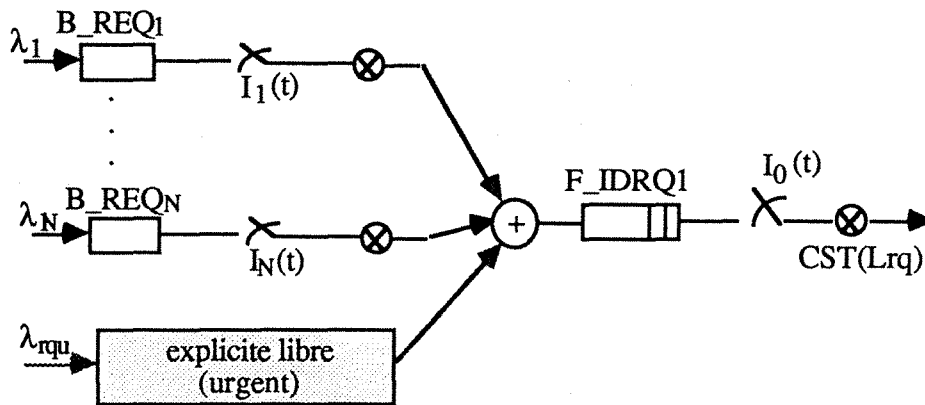


Fig. 2.16 Modèle approximatif du service explicite spécifié aperiodique

Deux phases de la transaction correspondent à la traversée des deux étages du modèle. Lorsque l'identifiant spécifié associé à un certain B_REQ arrive, on considère qu'au lieu d'emporter une demande, la trame RP_DAT_RQ1 emporte directement la SUITE d'identifiants vers F_IDRQ1. Dans ce cas, celles qui sont en attente dans F_IDRQ1 seront des "macro trames" composées de:

$$\{(ID_RQ1+RP_RQ1) + n(ID_DAT+RP_DAT)\}$$

et le temps de service Lrq sera la durée de transmission d'une "macro trame". On note que la file F_RPRQ n'apparaît pas dans notre modèle, et l'influence du trafic généré par le service de demande explicite libre de priorité urgente est représentée comme une contribution du flux supplémentaire fixe du paramètre λ_{rq} .

Ce modèle représente bien la réalité lorsque le trafic est faible. Dans le cas contraire, on peut négliger le délai d'attente dans B_REQ. On suppose qu'il y a toujours une demande dans B_REQ. En évaluant le flux d'entrée maximal de la file F_IDRQ1, on peut calculer le délai maximal de ce service.

2.3.5 Service de demande explicite libre (urgente ou normale)

Ce service nécessite les ressources suivantes:

- deux files dans la station source F_REQ_i ($i=1$ ou 2 , qui indique deux priorités différentes),
- trois files dans l'arbitre de bus F_RPRQ , F_IDRQ_i ($i=1$ ou 2).
- une fenêtre "explicite aperiodique" comme canal de transmission. On note que cette fenêtre est partagée par le service de demande explicite spécifiée aperiodique.

Une transaction de ce service est composée également de deux phases:

- 1) indication à l'arbitre de bus de la présence d'au moins une demande en ajoutant un ID_RQ1 dans la file F_IDRQ1 .
- 2) le transfert effectif de la SUITE d'identifieurs (RP_RQ1), puis la scrutation des variables correspondantes. On note que cette SUITE peut être la concaténation de plusieurs SUITES.

La première phase se déroule durant la fenêtre cyclique tandis que la deuxième se déroule durant la fenêtre explicite aperiodique.

En général, dans une station source, le nombre d'identifieurs configurés pour ce service est largement supérieur au nombre de files d'attente F_REQ_i qui vaut 1 pour chaque priorité.

On peut établir un modèle similaire à celui du service de demande explicite spécifiée aperiodique.

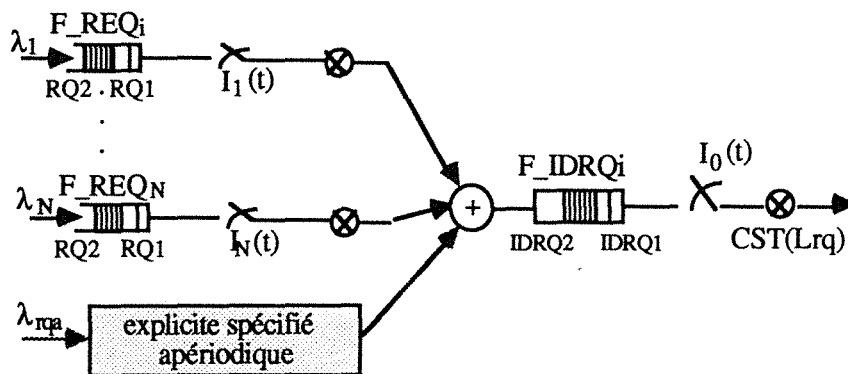


Fig. 2.17 Modèle approximatif du service explicite libre (urgent et normal)

Dans ce modèle, on a regroupé F_REQ1 et F_REQ2 en une seule file F_REQ avec deux priorités. On fait de même pour F_IDRQ_i . Comme pour les autres services précédents, on considère toujours que la SUITE est emportée directement dans F_IDRQ_i . Il convient de signaler que cette approche s'accorde presque toujours à la réalité même si le trafic est assez élevé. Cela est dû au fait qu'un ID_RQ_i arrive dans une station source

emporte toutes les SUITES accumulées dans la file F_REQ_i (d'où vient la concaténation des SUITES).

2.4 Modélisation générale

2.4.1 Modèle commun de tous les services aléatoires

A travers l'analyse de tous les services aléatoires, on peut établir un modèle commun permettant l'évaluation des performances de tous ces services (Fig. 2.18)

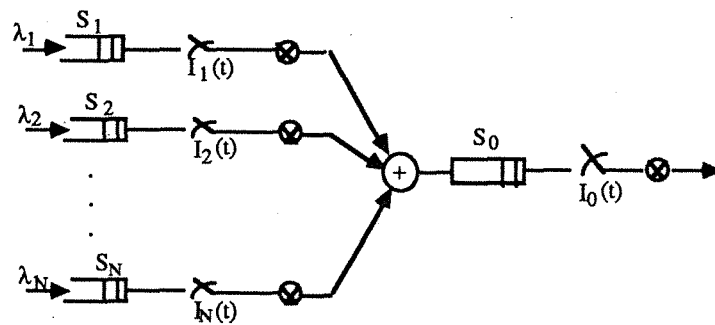


Fig. 2.18 Modèle commun

C'est un modèle de files d'attente parallèle-série en deux étages et avec serveurs intermittents. Le premier étage correspond au transfert des demandes de droit de parole et le deuxième étage correspond au transfert effectif des différentes trames d'informations. Pour éviter le goulet d'étranglement à la sortie du premier étage, le système est en général conçu de telle manière que la capacité d'évacuation de flux d'informations du premier étage est nettement supérieure à celle du deuxième étage. La condition stationnaire du système, lorsque les files ont une capacité infinie, est que le flux total en entrée ne dépasse pas la capacité d'évacuation du deuxième étage.

- Pour le service de messagerie périodique, il suffit d'étudier une file d'attente du premier étage avec capacité limitée à L et avec rejet en cas de saturation de cette file. Il n'y a pas d'attente au deuxième étage.
- Pour le service de demande explicite spécifiée périodique, une étude sur la file du premier étage avec capacité limitée à 1 et avec écrasement en cas de saturation de cette file. Là encore, il n'y a pas d'attente au deuxième étage.

- Pour le service de messagerie apériodique, on doit étudier les deux étages pour évaluer ses performances dans le cas du trafic faible.
- Pour le service de demande explicite spécifiée apériodique, le modèle complet doit également être étudié afin d'avoir une idée de ses performances dans le cas du trafic faible. L'influence du service de demande explicite libre de priorité urgente est représentée par un flux supplémentaire fixe. La capacité des files du premier étage est limitée à 1.
- Pour le service de demande explicite libre (urgent et normal), on doit étudier ce modèle avec deux priorités de clients. On verra que le modèle reflète bien la réalité même si le trafic est assez élevé.

2.4.2 Principe d'étude du modèle

L'étude de ce modèle sera décomposée en trois étapes:

2.4.2.1 Etude du premier étage

Le modèle de base de cet étage est:

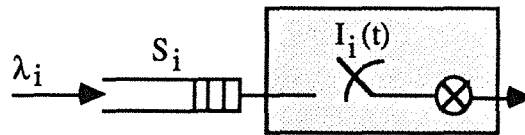


Fig. 2.19a Modèle de base du premier étage

Avec la distribution de nombre de clients dans la file:

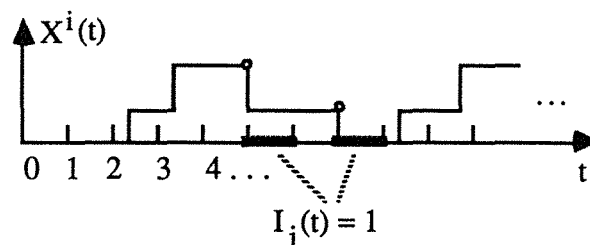


Fig. 2.19b Distribution de clients dans une file du premier étage

où

- $X^i(t)$: le nombre de clients dans la file à l'instant t ;
- λ_i : l'intensité du flux d'arrivée de clients = nombre moyen par unité de temps.

On note que les clients (les demandes) ne peuvent sortir du premier étage que durant $T_{cyc}(n)$, $n = 0, 1, \dots, T_m/T_{bc} - 1$. Ils arrivent à n'importe quel instant.

Cet étage sera étudié dans le chapitre 3.

2.4.2.2 Etude du flux d'entrée du deuxième étage

Ce flux est la superposition des flux de sortie des files durant un macro cycle T_m du premier étage.

Il suffit d'étudier le flux de sortie d'une file S_i du premier étage. Cette étude sera détaillée dans le chapitre 4.

2.4.2.3 Etude du deuxième étage

Le modèle de base de cet étage est:

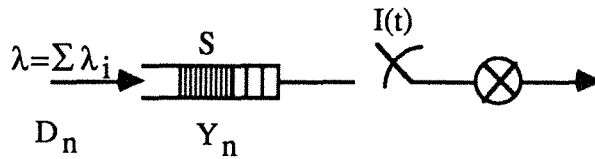


Fig. 2.20a Modèle de base du deuxième étage

Avec la distribution du nombre de clients dans la file:

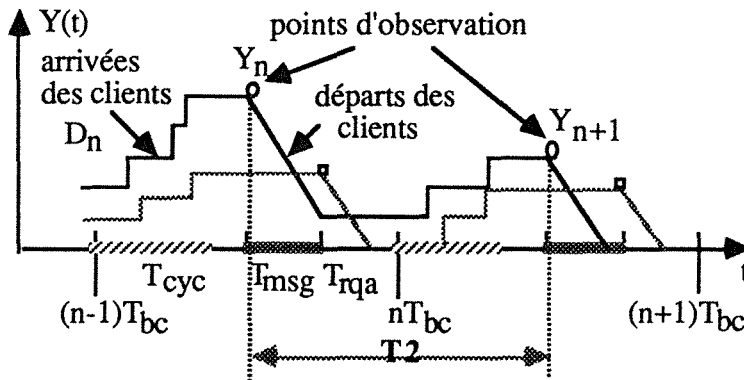


Fig. 2.20b Distribution de clients dans la file du deuxième étage

où

- $Y(t)$ est le nombre de clients dans la file à l'instant t
- Y_n est le nombre de clients dans la file à l'instant (nT_2) juste avant la fermeture de l'interrupteur $I(t)$
- T_2 est l'intervalle de temps qui sépare deux fermetures consécutives de l'interrupteur
- D_n est le nombre de clients qui sont arrivés durant $[(n-1)T_2, nT_2]$
- λ est l'intensité du flux d'arrivée des clients.

Afin d'effectuer l'étude analytique, on pose les hypothèses supplémentaires:

Hyp1:

Dans notre étude analytique, nous supposerons que la taille d'une macro trame (ID_DAT+RP_DAT_*) est toujours constante et dure 1 slot.

Hyp2:

Il existe une influence de la fenêtre messagerie aperiodique sur la fenêtre explicite aperiodique (mais pas réciproquement) car en pratique, on définit une largeur de Tmsg maximale. La largeur réelle de cette fenêtre dépend du trafic du service correspondant. Si le trafic est faible, Tmsg réelle n'atteindra peut-être pas sa dimension maximale. Dans ce cas, Trqa peut être grande car $Trqa(n) = Tbc - Tcyc(n) - Tmp(n) - Trqp(n) - Tmsg(n)$. Dans l'étude analytique qui sera menée dans le chapitre 4, on supposera que les dimensions de Trqa et Tmsg sont fixes.

2.5 Conclusion



Tous les problèmes concernant les performances et le dimensionnement ont été abordés à travers l'analyse et la modélisation des tous les services de la couche liaison de données du protocole FIP.

- Dans le premier paragraphe, nous avons donné une présentation détaillée de la couche liaison de données.
- Dans le deuxième paragraphe, les problèmes les plus simples ont été résolus:
 - la méthode de construction de la table de scrutation est donnée d'abord avec des formules permettant de dimensionner les fenêtres,
 - le rendement du protocole est calculé,
 - la manière de calculer le taux d'utilisation ("throughput" équivalent) est abordée,
 - la fiabilité du service de transfert cyclique de listes de variables est également discutée.
- Dans le troisième et le quatrième paragraphe, nous avons établi un modèle général commun pour tous les "services aléatoires". C'est un modèle exact pour certains services et approximatif pour d'autres. L'intérêt d'avoir un modèle commun est évident.

Ce modèle s'appuie sur des files d'attente parallèle-série à deux étages avec serveurs intermittents et temps de service déterministe. Ce modèle a été construit pour évaluer les performances du réseau FIP. Mais il peut aussi être utilisé à d'autres fins. Par exemple, on pourra modéliser un "bridge" (fig.2.21) d'interconnexion de réseaux locaux, ou un concentrateur avec multiplexage statistique [Malgari 88] (fig.2.22).

Application 1: interconnexion de deux réseaux locaux (polling, token-bus ou token-ring).

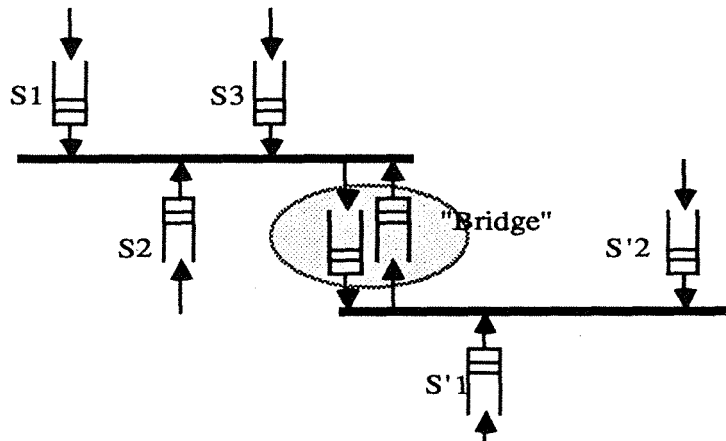


Fig. 2.21 Interconnexion de deux "LANs" par un "bridge"

On suppose que ce bridge a des files d'attente pour stocker provisoirement des messages. Les deux réseaux locaux adoptent un protocole de scrutation.

Application 2: Concentrateur de communication.

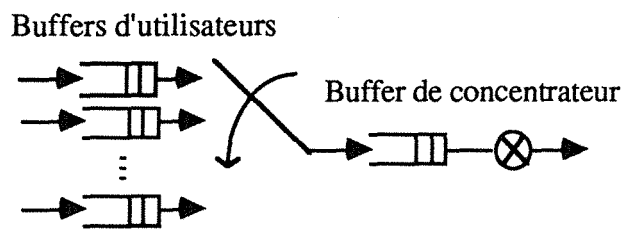


Fig. 2.22 concentrateur de communication

Les buffers d'utilisateurs sont scrutés statiquement.

CHAPITRE 3

ETUDE ANALYTIQUE DU MODELE, PREMIER ETAGE

3. Etude analytique du modèle, premier étage

Dans ce chapitre, nous effectuons une étude analytique du premier étage du modèle général établi dans le chapitre précédent. Nous supposons que les stations du premier étage sont homogènes, ce qui nous permet d'étudier une seule station.

La figure 3.1 montre le modèle à étudier.

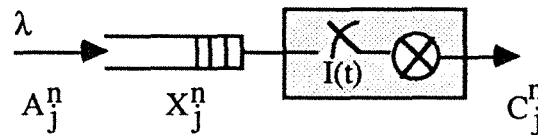


Fig. 3.1 Modèle de base du premier étage

On suppose que le temps de service des clients est constant: 1 slot. Et le taux d'arrivée des clients est λ .

La figure 3.2 donne un exemple de la distribution du nombre de clients dans la file d'attente (le client en service est exclus). Les clients ne peuvent sortir de la file qu'à des instants discrets (fin de chaque slot). En revanche, ils arrivent à n'importe quel instant.

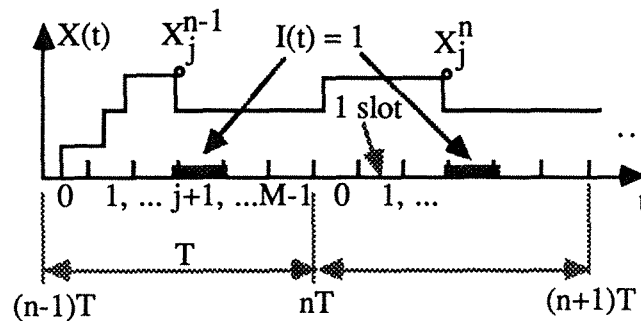


Fig. 3.2 Distribution des clients dans la file d'attente

On définit d'abord les notations utilisées:

- $I(t)$: interrupteur représentant les moments disponibles du serveur. Le serveur est disponible lorsque l'interrupteur ferme, i.e. lorsque $I(t) = 1$. D'ailleurs, on trouve toujours un T ($T_{bc} \leq T \leq T_m$, cf. §2) tel que $I(t) = I(t+nT)$.
- n : indice désignant le numéro du cycle T , $n = 0, 1, \dots, \infty$.
- j : indice désignant le numéro du slot, $j = 0, 1, \dots, M-1$.
- M : nombre de slots dans un cycle T .
- $X(t)$: nombre de clients dans la file à l'instant t .
- X_j^n : nombre de clients dans la file à la fin du $j^{\text{ème}}$ slot du $n^{\text{ème}}$ cycle.
- A_j^n : nombre de clients qui sont arrivés durant le $j^{\text{ème}}$ slot du $n^{\text{ème}}$ cycle.
- C_j^n : nombre de clients qui sont sortis durant le $j^{\text{ème}}$ slot du $n^{\text{ème}}$ cycle.

L'étude du processus $(X(t) | t \geq 0)$ peut s'effectuer en observant ce processus à des instants particuliers. Le choix de ces *instants d'observation* dépend de la fonction $I(t)$.

Il existe deux modes différents de fermeture de l'interrupteur $I(t)$ qui résultent toutes les possibilités:

- **régulier**. Dans ce cas particulier, l'intervalle de temps entre deux fermetures consécutives de $I(t)$ est toujours constant. Lors de l'étude, le système n'est observé qu'aux instants nT .

- **irrégulier**. Dans ce cas général, l'intervalle de temps entre deux fermetures consécutives n'est pas constant. Lors de l'étude, on est obligé de considérer tous les M slots intervenant pendant le cycle T .

Pour le cas général, en supposant que la file d'attente est de capacité infinie, il est clair que l'évolution du nombre de clients dans la file d'attente peut être décrite par une équation de récurrence [Hayes 84]:

$$X_{j+1}^n = X_j^n - e_j * U(X_j^n) + A_{j+1}^n, \quad j=0, 1, \dots, M-1 \quad (3.1)$$

avec

$$X_0^{n+1} = X_M^n$$

où:

$$e_j = \begin{cases} 1 & \text{si l'interrupteur ferme durant le slot } j \\ 0 & \text{sinon} \end{cases}$$

et

$$U(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Dans le cas particulier où l'interrupteur $I(t)$ ferme régulièrement, on observe le système à chaque moment de fermeture. L'équation 3.1 est réduite à l'équation 3.2:

$$X_{n+1} = X_n - U(X_n) + A_{n+1} \quad (3.2)$$

où:

- X_n : le nombre de clients dans la file à l'instant nT juste avant la fermeture de $I(t)$;
- A_n : le nombre de clients qui sont arrivés durant $[(n-1)T, nT]$.

Lorsque la capacité de la file d'attente est limitée à L , les équations 3.1 et 3.2 doivent être légèrement modifiées.

- Pour le cas général, on a:

$$X_{j+1}^n = \min[X_j^n - e_j * U(X_j^n) + A_{j+1}^n, L] \quad j = 0, 1, \dots, M-1 \quad (3.3)$$

- Pour le cas particulier, on a:

$$X_{n+1} = \min[X_n - U(X_n) + A_{n+1}, L] \quad (3.4)$$

C'est un modèle de file d'attente avec serveur disponible intermittent. Le temps de service est déterministe. Ce modèle n'est pas une file d'attente classique. Le serveur commence son service de manière autonome, indépendamment du flux d'entrée des clients.

L'étude des systèmes d'attente évoqués s'effectuera en trois parties:

3.1 Capacité infinie

3.1.1 Cas simple: l'interrupteur $I(t)$ ferme régulièrement

3.1.2 Cas général: l'interrupteur $I(t)$ ferme irrégulièrement

3.2 Capacité limitée à L

3.2.1 Cas simple: l'interrupteur $I(t)$ ferme régulièrement

3.2.2 Cas général: l'interrupteur $I(t)$ ferme irrégulièrement

3.3 Application aux services périodiques de FIP

3.3.1 Performances du service de messagerie périodique

3.3.2 Performances du service de demande explicite spécifiée périodique

3.1 Capacité infinie

La capacité infinie n'est qu'une approximation de la réalité, mais les résultats obtenus dans ce cas permettent tout de même d'évaluer de façon simple et satisfaisante le fonctionnement du système. Nous étudierons par la suite le cas de la capacité limitée (cf. §3.2).

Les deux paramètres les plus intéressants sont:

- la distribution de probabilité stationnaire du nombre de clients dans la file,
- la distribution de probabilité stationnaire du délai d'attente.

A partir de ces deux distributions, nous pouvons déduire d'autres paramètres présentant un intérêt pratique tels que le nombre moyen de clients dans la file d'attente, le délai moyen d'attente d'un client.

Nous allons d'abord étudier le cas simple où l'interrupteur ferme régulièrement, et puis le généraliser au cas général où l'interrupteur ferme irrégulièrement. L'étude utilise la théorie des chaînes de Markov induites ("imbedded Markov chain").

3.1.1. Cas simple: l'interrupteur ferme régulièrement

3.1.1.1 Description du problème

A chaque instant $t_n = nT$, l'interrupteur ferme et permet le départ éventuel d'un client (cf. figure 3.3).

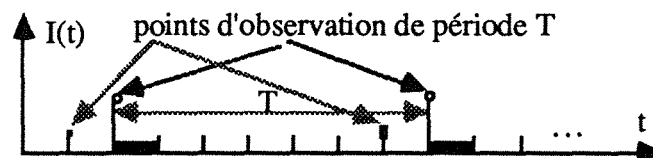


Fig. 3.3 Description d'un interrupteur régulier

On suppose que le flux d'arrivée des clients est un processus à Accroissements Indépendants et Stationnaires (A.I.S).

On s'intéresse aux grandeurs suivantes :

$X(t)$ = nombre de clients dans la file d'attente à l'instant t .

$W_q(k)$ = temps d'attente du $k^{\text{ème}}$ client (ou Délai).

ainsi qu'à l'existence des lois limites de ces deux grandeurs lorsque $k \rightarrow +\infty$ et $t \rightarrow +\infty$.

3.1.1.2 Résultats obtenus

A l'issue de l'étude qui se déroulera dans le §3.1.1.3, deux catégories de résultats sont obtenus.

1) En ce qui concerne le nombre de clients dans la file:

- Le processus décrit par l'équation 3.2 est une chaîne de Markov admettant une distribution limite. La distribution de probabilité limite (et stationnaire) du nombre de clients dans la file d'attente aux instants nT^- (juste avant la fermeture de l'interrupteur) est donnée par sa fonction génératrice:

$$P(z) \equiv E[z^X] = \frac{(1-\rho)(1-z)A(z)}{A(z)-z} \quad (3.5)$$

où:

$A(z) \equiv E(z^A)$ est la fonction génératrice de la variable aléatoire A_n et $\rho = \lambda T (< 1)$.

La fonction génératrice $P(z)$ permet de reconstituer la distribution limite de probabilité: $(\pi(k))_{k \geq 0}$.

- On en déduit le nombre moyen de clients à l'instant nT^- :

$$E[X_{\infty}] = (1/2)[E(A) + V(A)/(1-E(A))] \quad (3.6)$$

Et le nombre moyen à l'instant nT^+ :

$$E[M_{\infty}] = E[X_{\infty}] - (1-\pi(0)) = E[X_{\infty}] - \rho \quad (3.7)$$

Le nombre moyen de clients dans la file \bar{N}_q est donné par:

$$\bar{N}_q \equiv \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{\tau}^{\tau+T} E[X(t)] dt = E[X_{\infty}] - \frac{\rho}{2} \quad (3.8)$$

En particulier, lorsque le processus du flux d'arrivée est Poissonnien de paramètre λ :

$$\bar{N}_q = \frac{\rho}{2} + \frac{\rho^2}{2(1-\rho)} \quad (3.9)$$

2) En ce qui concerne le délai d'attente d'un client:

- La distribution de probabilité limite du délai d'attente W_q est donnée par sa transformée de Laplace:

$$W_q(s) = \{[1-\rho] + e^{sT}[P(e^{-sT}) - (1-\rho)]\} \frac{e^{-\lambda T[1-e^{-sT}] - e^{-sT}}}{T\{s - \lambda[1-e^{-sT}]\}} \quad (3.10)$$

où:

$P(e^{-sT})$ est donné par l'équation (3.5) en remplaçant z par e^{-sT} .

- On en déduit le délai d'attente moyen d'un client:

$$E(W_q) = \frac{T}{2} + \frac{\lambda T^2}{2(1-\rho)} \quad (3.11)$$

Remarque: Le nombre moyen de clients en attente dans la file \bar{N}_q et le délai moyen d'attente d'un client $E(W_q)$ sont liés par la formule de Little $E(N_q) = \lambda E(W_q)$.

3.1.1.3 Démonstrations

Ce paragraphe est subdivisé en trois parties:

- 1) Calcul du nombre de clients dans la file d'attente
- 2) Calcul du délai d'attente
- *3) Calcul du nombre et du délai *d'un autre point de vue*

1) Etude du nombre de clients dans la file

Le principe d'étude est comparable à celui utilisé dans M/G/1 [Kleinrock 75] ou encore dans le TDM (Time Division Multiplexing) [Hayes 84].

On note:

$X_n = X(nT^-)$: nombre de clients dans la file juste avant la fermeture de l'interrupteur.

$M_n = X(nT^+)$: nombre de clients dans la file juste après la fermeture de l'interrupteur.

A_{n+1} : nombre de clients qui sont arrivés durant $]nT, (n+1)T[$.

où T est la durée constante séparant deux fermetures consécutives de l'interrupteur.

$(v(t) | t \geq 0)$ ($v(0)=0$): nombre de clients qui sont arrivés durant $]0, t[$, on a alors

$$v((n+1)T) - v(nT) = A_{n+1}.$$

Le modèle peut être décrit par l'équation de récurrence 3.2. On la réécrit ci-dessous:

$$X_{n+1} = (X_n - 1)^+ + A_{n+1} \quad , \quad x^+ = \max(0, x)$$

Cette équation apparaît dans l'étude de M/G/1. Signalons que les variables aléatoires X_n et A_n n'ont pas la même signification dans M/G/1 et dans notre modèle. En effet, dans M/G/1:

- X_n est le nombre de clients dans le système à l'instant où le $n^{\text{ème}}$ client quitte le système.

- A_n est le nombre de clients qui sont arrivés durant le service du $n^{\text{ème}}$ client.

On peut démontrer que la distribution limite existe. La condition de stationnarité est $\rho = \lambda T < 1$, où λ est le taux d'arrivée de clients. En fait, si les variables aléatoires $A_1, A_2, \dots, A_n, \dots$ sont indépendantes et identiquement distribuées de même loi que A alors $(X_n)_{n \geq 0}$ est une chaîne de Markov homogène. En particulier: si $(v(t) | t \geq 0)$ est un processus à Accroissements Indépendants et Stationnaires, alors les variables aléatoires A_n sont indépendantes et identiquement distribuées.

Il est clair que:

$$\begin{aligned} M_n &= \max(X_n - 1, 0) = (X_n - 1)^+ \text{ et } M_n \text{ vérifie :} \\ M_n &= \max(M_n + A_{n+1} - 1, 0) \end{aligned} \quad (3.12)$$

La connaissance de X_n fournit M_n ; par contre, si $M_n = 0$, on ne peut reconstituer X_n qui peut être 0 ou 1. Il est donc préférable d'étudier X_n et d'en déduire M_n .

$$\text{On note que } X(t) = M_n + v(t) - v(nT); \quad \forall t \in]nT, (n+1)T] \quad (3.13)$$

L'étude de la chaîne (X_n) est faite dans la littérature [Kleinrock 75], [Gross 85]. Nous prenons directement les résultats. La distribution limite existe pour cette chaîne de Markov quand $\rho = \lambda T < 1$. La fonction génératrice $P(z)$ s'écrit :

$$P(z) \equiv E[z^X] = \{(1-\rho)(1-z)A(z)\} / \{A(z)-z\} \quad (3.14)$$

où

$A(z) \equiv E(z^A)$ est la fonction génératrice de la variable aléatoire A ou encore de $v(T)$.

On sait que la distribution limite de X_n est définie par:

$$\pi(k) \equiv \lim_{n \rightarrow +\infty} P[X_n = k]$$

et on définit la distribution stationnaire de M_n ($\tilde{\pi}(k)$):

$$\tilde{\pi}(k) \equiv \lim_{n \rightarrow +\infty} P[M_n = k] = \begin{cases} \pi(k+1) & \text{si } k > 0 \\ \pi(0) + \pi(1) & \text{si } k = 0 \end{cases}$$

M_∞ a donc pour distribution limite $(\tilde{\pi}(k))$.

En dérivant l'équation (3.14), nous obtenons le nombre moyen de clients dans la file à la fin de chaque slot :

$$E[X_\infty] = (1/2)[E(A) + \text{Var}(A)/(1-E(A))] \quad (3.15)$$

C'est la formule de Pollaczek-Khinchin.

On peut obtenir également le nombre moyen de clients juste après la fermeture de l'interrupteur:

$$E[M_\infty] = E[X_\infty] - (1-\pi(0)) = N_q - \rho \quad (3.16)$$

Une observation intéressante est que l'équation (3.2) est exactement la même que celle de M/G/1. Est-ce que cela veut dire que M/G/1 (sans considérer le client qui est dans le service) et notre modèle ont les mêmes performances? La réponse est non. C'est évident qu'à cause de l'interrupteur, les clients dans notre modèle sont obligés d'attendre plus longtemps pour quitter le système.

Une étude en détail de M/G/1 montre que l'observation du nombre de clients aux instants où un client quitte le système sera toujours la même que les observations à n'importe quel instant aléatoire [Gross 85]. C'est une propriété très particulière de M/G/1.

Par contre, on peut démontrer qu'un système autonome comme notre modèle n'a pas cette propriété intéressante. Si on observe le modèle de la figure 3.1 à des instants quelconques, l'état stationnaire n'existe plus.

En effet, $X(t)$ pour un t quelconque n'admet pas de distribution limite quand $t \rightarrow +\infty$ (i.e. $X(t)$ ne converge pas en loi).

Le nombre de clients dans la file à n'importe quel instant t est donné par l'équation 3.13. En posant $t_n = nT + h$ et en prenant l'espérance de l'équation (3.13), on obtient le nombre moyen de clients dans la file observé à l'instant h .

$$E[X(t_n)] = E[M_n] + E[v(h)] \quad (3.17)$$

$E[M_n]$ est le nombre moyen de clients quand on observe le système juste après la fermeture de l'interrupteur, il est donné par l'équation 3.16. En particulier, quand $(v(h))$ est un processus Poissonien, on a :

- $E[v(h)] = \lambda \cdot h.$
- $E[X_\infty] = \rho + \frac{\rho^2}{2(1-\rho)}$
- $E[M_\infty] = E[X_\infty] - \rho = \frac{\rho^2}{2(1-\rho)}$

$E[X(t_n)]$ a la forme présentée sur la figure 3.4.

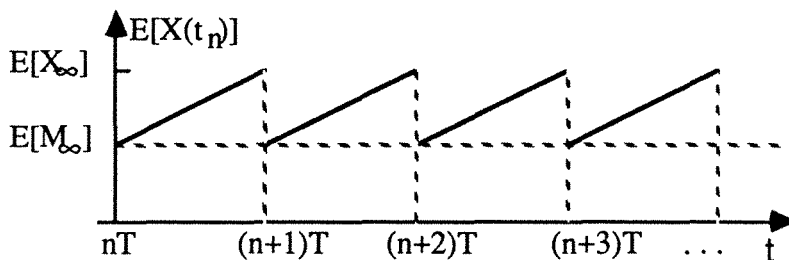


Fig. 3.4 Espérance du nombre de clients en fonction du temps

L'espérance est fonction du temps h . Si les points d'observation qu'on a choisi sont les débuts de chaque slot, $h = 0$ donc l'espérance obtenue est bien $E[M_\infty]$. Pour obtenir la moyenne sur h , il suffit de "moyenner" l'équation 3.17 sur l'intervalle $[0, T]$ et ce qui nous donne:

$$\bar{N}_q \equiv \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^{\tau+T} E[X(t)] dt = \frac{\rho}{2} + \frac{\rho^2}{2(1-\rho)} \tag{3.18}$$

On a constaté que le nombre moyen de clients obtenu par la simulation correspond parfaitement à cette équation [Song 88].

2) Etude du délai d'attente

Le délai d'attente d'un client dans la file est composé des éléments suivants:

- Un nouveau client arrivé trouve le système vide. S'il arrive h secondes après le début de l'intervalle considéré, un délai de $(T - h)$ secondes aura lieu avant qu'il quitte le système. T secondes est la longueur d'un intervalle séparant deux fermetures consécutives de l'interrupteur.
- Soit L le nombre de clients qui sont arrivés pendant l'intervalle $(0, h)$. Donc, le temps pour qu'ils quittent tous le système est $L \cdot T$.

- Soit X_n le nombre de clients dans la file à la fin de l'intervalle précédent. Compte tenu qu'un client quitte la file au début de l'intervalle présent, le temps pour qu'ils quittent tous la file est alors $T*(X_n - 1)^+$.

Ainsi, en supposant la distribution limite atteinte (ce qui permet de supprimer l'indice n):

$$W_q = (X_{\infty} - 1)^+ * T + L * T + (T - h) \quad (3.19)$$

La transformée de Laplace du délai peut être obtenue à partir de cette équation. Le premier terme est indépendant des autres, on a la fonction génératrice de $(X_{\infty} - 1)^+$:

$$E[z^{(X-1)^+}] = 1 - \rho + z^{-1}[P(z) - (1-\rho)] \quad (3.20a)$$

où $P(z)$ est la f.g de X_{∞} .

La transformée de Laplace de $(X_{\infty} - 1)^+ * T$ devient:

$$\tilde{P}(s) = E[e^{-sT(X-1)^+}] = P[e^{-sT}] \quad (3.20b)$$

Considérons maintenant la variable aléatoire: $f = LT + (T-h)$

On note que les variables aléatoires L et h ne sont pas indépendantes. Un calcul de la probabilité conditionnelle nous donne $\Pr[t \leq f \leq t+dt]$. La transformée de Laplace de f est alors:

$$F(s) = \frac{e^{-\lambda T[1-e^{-sT}]} - e^{-sT}}{T\{s - \lambda[1 - e^{-sT}]\}} \quad (3.21)$$

La transformée de Laplace de W_q est donc:

$$W_q(s) = \tilde{P}(s)F(s) \quad (3.22)$$

En dérivant l'équation 3.22 au point $s = 0$, on obtient le délai moyen.

On peut aussi calculer le délai moyen directement. Puisque l'espérance de la somme des variables aléatoires est la somme des espérances de chaque variable même si les variables ne sont pas indépendantes. On peut prendre l'espérance de l'équation 3.19.

$$E[W_q] = E[(X_{\infty} - 1)^+]*T + E[L]*T + 2*T - E[h] \quad (3.23)$$

on sait que h est uniformément distribué dans $[0, T]$ et on a :

- $E[h] = T/2$;
- $E[L] = \lambda * T/2$;
- $E[(X_n - 1)^+] = E[X_\infty] - \rho = E[M_\infty]$.

On obtient le délai moyen d'un client dans la file:

$$E(W_q) = \frac{T}{2} + \frac{\lambda T^2}{2(1-\rho)} \quad (3.24)$$

Cette équation est confirmée par la simulation [Song 88].

On a remarqué que les équations 3.18 et 3.24 sont reliées par la formule de Little.

*3). Etude de la chaîne induite (x_i) --- une autre approche

On a déjà étudié le nombre de clients dans le système aux instants $(nT)^+$ et $(nT)^-$.

On peut aussi introduire une chaîne de Markov à des instants où un client quitte la file.

Soit:

- x_i : nombre de clients à l'instant où le $i^{\text{ème}}$ client quitte la file.
- a_i : nombre de clients arrivés pendant un intervalle T .
- \hat{a}_i : nombre de clients arrivés pendant le temps aléatoire Y .
- Y : temps d'attente quand un client arrive et trouve un système vide (un client qui est arrivé dans un système vide est obligé d'attendre le début de l'intervalle suivant).

On a la relation de récurrence suivante :

$$x_{i+1} = x_i - U(x_i) + a_{i+1}U(x_i) + \hat{a}_{i+1}[1-U(x_i)] \quad (3.25)$$

Notons que x_i a la même signification que dans la file M/G/1.

Prenons l'espérance de l'équation (3.25) et *supposons l'existence d'un état stationnaire*, nous avons :

$$1 - P_0 = E[\hat{a}]/(1 - E[a] + E[\hat{a}]) \quad (3.26)$$

P_0 est la probabilité que le système soit vide. Enlevant la dépendance de i , la fonction génératrice de la probabilité d'avoir x clients dans le système est alors :

$$\begin{aligned}
P(z) &= E[z^X] = E[z^{X-U(x)+aU(x) + \hat{a}[1-U(x)]}] \\
&= P_0 E[z^{\hat{a}}] + \sum P(x=i) z^{i-1+a} \\
&= P_0 \hat{A}(z) + z^{-1} [\sum P(x=i) z^i - P_0] A(z) \\
&= P_0 \hat{A}(z) + z^{-1} [P(z) - P_0] A(z)
\end{aligned} \tag{3.27}$$

où $A(z)$ et $\hat{A}(z)$ sont respectivement la fonction génératrice de a et \hat{a} .

A partir de l'équation (3.27), on trouve :

$$P(z) = P_0 [A(z) - z\hat{A}(z)] / [A(z) - z] \tag{3.28}$$

Les probabilités marginales à l'instant où un client quitte le système peuvent être obtenues à partir de cette fonction génératrice.

Le calcul de $A(z)$ est simple quand le flux d'arrivée est Poissonnien :

$$A(z,t) = E[z^{a(t)}] = \sum_{n=0}^{\infty} z^n \frac{(\lambda t)^n e^{-\lambda t}}{n!} = e^{\lambda t(z-1)}$$

$$A(z) = A(z,T) = e^{\rho(z-1)} \tag{3.29}$$

Par contre, le calcul de $\hat{A}(z)$ n'est pas immédiat. On définit d'abord deux variables aléatoires: I et Y .

- I : durée de la période d'oisiveté durant laquelle le système est vide ("idle period"). I a une distribution exponentielle de taux λ car cette période se termine par l'arrivée d'un client.

- Y : intervalle entre la fin de la période I et le début du cycle suivant.

La relation entre I et Y est présentée sur la figure 3.5

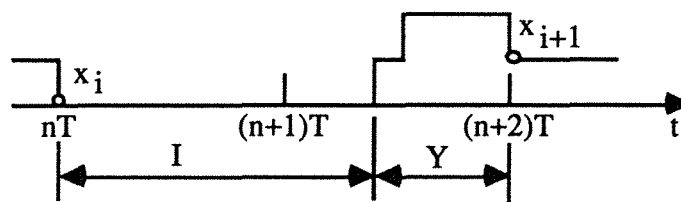


Fig. 3.5 Relation entre I , Y , et T

Selon la figure, on trouve que:

$$0 \leq Y = (i+1)T - I \leq T, \quad iT \leq I \leq (i+1)T \quad (3.30)$$

Y peut être calculé de la façon suivante :

$$\begin{aligned} P[Y \leq y] &= \sum_i P[Y \leq y, iT \leq I \leq (i+1)T] \\ &= \sum_i P[(i+1)T - y \leq I, iT \leq I \leq (i+1)T] \end{aligned} \quad (3.31)$$

pour $0 \leq y \leq T$:

$$\begin{aligned} P[Y \leq y] &= \sum_i P[(i+1)T - y \leq I \leq (i+1)T] \\ &= \sum_i [e^{-\lambda[(i+1)T - y]} - e^{-\lambda(i+1)T}] \\ &= (e^{\lambda y} - 1)e^{-\lambda T} / (1 - e^{-\lambda T}) \end{aligned}$$

pour $y \geq T$:

$$P[Y \leq y] = 1$$

$$F_Y(y) = P[Y \leq y] = \begin{cases} (e^{\lambda y} - 1)e^{-\lambda T} / (1 - e^{-\lambda T}), & 0 \leq y \leq T \\ 1, & y \geq T \end{cases} \quad (3.32)$$

La densité de probabilité de Y pour $0 \leq y \leq T$ est alors:

$$f_Y(y) = dF_Y(y)/dy = \lambda * e^{\lambda y} * e^{-\lambda T} / (1 - e^{-\lambda T}) \quad (3.33)$$

Les deux premiers moments sont :

$$\begin{aligned} E[Y] &= \int_0^T y f_Y(y) dy = \frac{e^{-p}}{\lambda(1-e^{-p})} \int_0^T \lambda y e^{\lambda y} d(\lambda y) = \frac{T}{1-e^{-p}} - \frac{1}{\lambda} \\ E[Y^2] &= \int_0^T y^2 f_Y(y) dy = \frac{2}{\lambda^2} + \frac{T^2 - \frac{2T}{\lambda}}{1-e^{-p}} \end{aligned}$$

La fonction génératrice de \hat{a} est :

$$\hat{A}(z) = E[z^{\hat{a}}] = [e^{-\rho}(e^{\rho z}-1)]/[z(1-e^{-\rho})] \quad (3.34)$$

Le nombre moyen de clients est :

$$E[x_{\infty}] = P'(1) = \{P_0[2\hat{A}'(1)+\hat{A}''(1)-A''(1)] + A''(1)\}/\{2[1-A'(1)]\} \quad (3.35)$$

où :

$$P_0 = [1-A'(1)]/[1-A'(1)+\hat{A}'(1)];$$

et

$$A'(1) = \lambda T = \rho;$$

$$A''(1) = \lambda^2 T^2 = \rho^2;$$

$$\hat{A}'(1) = \lambda E[Y] = \rho/(1-e^{-\rho}) - 1;$$

$$\hat{A}''(1) = \lambda^2 E[Y^2] = 2 + (\rho^2 - 2\rho)/(1-e^{-\rho}).$$

Insérant dans l'équation (3.35), nous avons le nombre moyen de clients dans la file à l'instant où un client quitte le système :

$$E(x_{\infty}) = \rho^2/[2(1-\rho)] + \rho/2 \quad (3.36)$$

C'est aussi le résultat que l'on a trouvé dans le paragraphe précédent (l'équation 3.18).

En utilisant la formule de Little, pour le délai moyen, la même équation que (3.24) peut être trouvée.

Remarque: Selon l'analyse ci-dessus, on a l'impression que la distribution stationnaire du nombre de clients dans la file existe partout sur l'axe des temps, ce qui n'est pas du tout le cas en réalité. Pourtant, l'équation (3.36) est bien identique à l'équation (3.18). L'explication est que dans l'analyse menée dans cette dernière partie, on observe le système aux instants de départ d'un client, et ces instants font bien partie des instants de fermeture de l'interrupteur. C'est pour cela que les résultats obtenus à partir de deux points de vue différents coïncident.

3.1.2 Cas général: l'interrupteur ferme irrégulièrement

C'est un cas plus général que le cas précédent. On se limite à l'analyse de la distribution de probabilité limite du nombre de clients. A partir de ce résultat, on peut déduire le nombre moyen de clients dans la file. Le délai moyen est fourni par la formule de Little.

Supposons T composé de M slots, c d'entre eux sont dédiés à une station particulière (i.e: l'interrupteur de cette station ferme durant ces c slots). Nous introduisons une chaîne de Markov à la fin de chaque slot. *Nous avons en tout M chaînes de Markov homogènes (homogène lorsqu'on observe le système avec l'intervalle T ou encore M slots).*

Soit:

- X_j^n : nombre de clients dans la file à la fin du j^{ème} slot du n^{ème} cycle.
- A_j^n : nombre de clients qui sont arrivés durant le j^{ème} slot du n^{ème} cycle.

L'évolution du système est décrite par l'équation 3.1, on la réécrit ci-dessous:

$$X_{j+1}^n = X_j^n - e_j * U(X_j^n) + A_{j+1}^n, \quad j=0, 1, \dots, M-1 \quad (3.37)$$

avec $X_0^{n+1} = X_M^n$

On calcule maintenant la fonction génératrice de X_j^n . En supposant l'existence d'une distribution limite et avec la condition de stationnarité $\lambda T < c$, on peut supprimer l'indice n.

Soit $P_j(z)$ la fonction génératrice de X_j , nous avons :

$$P_{j+1}(z) = \begin{cases} P_j(z)A(z,T), & \text{pour } e_j=0 \\ [z^{-1}P_j(z) + (1-z^{-1})P_{j0}]A(z,T), & \text{pour } e_j=1 \end{cases} \quad (3.38)$$

où:

P_{j0} est la probabilité que le système soit vide quand le j^{ème} slot commence. $\{P_{j0}\}$ est une série de c inconnues.

En définissant:

$$e_j = \begin{cases} z, & \text{pour } e_j = 0 \\ 1, & \text{pour } e_j = 1 \end{cases} \quad \text{et} \quad \delta_j = \begin{cases} 0, & \text{pour } e_j = 0 \\ 1, & \text{pour } e_j = 1 \end{cases}$$

l'équation (3.38) peut s'écrire matriciellement :

$$\begin{bmatrix} P_1(z) \\ P_2(z) \\ \vdots \\ P_M(z) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & \varepsilon_M \\ \varepsilon_1 & 0 & \dots & 0 & 0 \\ 0 & \varepsilon_2 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \varepsilon_{M-1} & 0 \end{bmatrix} \begin{bmatrix} P_1(z) \\ P_2(z) \\ P_3(z) \\ \vdots \\ P_M(z) \end{bmatrix} \frac{A(z)}{z} + (1-z^{-1})A(z) \begin{bmatrix} P_{M0} \delta_M \\ P_{10} \delta_1 \\ \vdots \\ P_{M-10} \delta_{M-1} \end{bmatrix}$$

Ou encore:

$$\boxed{[\mathbf{I} - z^{-1}A(z)\mathbf{E}(z)]\mathbf{P}(z) = (1 - z^{-1})A(z)\mathbf{P}_0} \quad (3.39)$$

La partie gauche de l'équation (3.39) est inversible. Le vecteur \mathbf{P}_0 peut être déterminé en utilisant le théorème de Rouché [Hayes 84].

Les M fonctions génératrices $P_j(z)$ étant connues, nous pouvons calculer les paramètres moyens comme par exemple $E[N_q]$ et $E[W_q]$ de la même façon que dans le cas simple (cf. §3.1.1), mais ce n'est pas toujours faisable en raison de la complexité du calcul. Par exemple, pour $c=9$ et $M=600$, nous devons manipuler une matrice de dimension $[600 \times 600]$!

Pour surmonter cet obstacle de calcul, au lieu de considérer une file de capacité infinie, on peut considérer une file de capacité limitée mais suffisamment grande pour que la probabilité de dépassement de cette capacité soit négligeable. On verra dans la suite (cf §3.2) que l'analyse sera simplifiée et les résultats sont très bien approchés.

3.2 Capacité limitée à L

En réalité, une file d'attente de capacité infinie n'existe pas. Nous considérons dans ce paragraphe la file d'attente de capacité limitée à L . L'analyse du paragraphe précédent (§ 3.1) est basée sur l'hypothèse de capacité infinie. La distribution de probabilité peut être calculée à partir de sa fonction génératrice. En pratique, on souhaite qu'une file d'attente ait une capacité suffisamment grande afin que la probabilité de

dépassement de capacité ("overflow") soit négligeable dans des conditions de fonctionnement normal. Quand la capacité de la file n'est pas assez grande, la probabilité de dépassement ne peut être négligée et l'analyse du paragraphe précédent n'est plus applicable. Dans ce cas, nous fournissons une autre approche.

Nous indiquons qu'avec une file d'attente de capacité limitée, l'analyse du système sera grandement simplifiée quel que soit le cas (général ou simple). En effet, grâce à la capacité limitée, nous traiterons une matrice de probabilités de transition de dimension finie au lieu d'une matrice de dimension infinie.

Comme dans §3.1, nous étudions d'abord le cas simple et puis le cas général.

3.2.1 Cas simple: l'interrupteur ferme régulièrement

3.2.1.1 Description

On considère d'abord le cas simple. On note que la signification du "cas simple" est légèrement modifiée. On autorise $b \geq 1$ (au lieu de 1 dans le "vrai cas simple") clients à sortir dès que l'interrupteur $I(t)$ ferme. *Ceci permet de réutiliser certains résultats obtenus ici pour l'étude du deuxième étage (cf. §4).* Le cas qui nous concerne devient un cas particulier avec $b = 1$. Le système à étudier peut être décrit par:

$$X_{n+1} = \min[(X_n - b)^+ + A_{n+1}, L], \quad \text{avec } (x^+ = \max(0, x)) \quad (3.40)$$

Soit:

- X_n = nombre de clients dans la file à l'instant nT^- (juste avant le départ).
- A_n = nombre de clients qui sont arrivés durant $[(n-1)T, nT]$, $n \geq 1$.
- b = capacité d'évacuation du système durant une période T .
- $O_n = \min(b, X_n)$ = nombre de clients qui sont sortis durant $[nT^-, nT^+]$.
- L = capacité de la file d'attente.

Remarque: On a supposé que O_n clients sortent instantanément, ce qui n'est pas le cas pour la file d'attente du deuxième étage. En effet, pour les clients du deuxième étage, puisque la transmission d'un client prend un certain temps (délai de transmission), les clients sortent un par un (cf. §4).

3.2.1.2 Résultats obtenus

Les résultats sont de trois types:

- La distribution limite de probabilité du nombre de clients dans la file $\pi(k)_{0 \leq k \leq L}$ est obtenue directement par l'analyse numérique. Pour cela, un programme écrit en C en utilisant la méthode de surrelaxation est développé.

- La probabilité de rejet ou de dépassement de capacité ("overflow") Π_{over} est donnée:

$$\Pi_{\text{over}} = 1 - \frac{b}{E(A)} + \frac{1}{E(A)} \sum_{k=0}^{b-1} (b-k)\pi(k) \quad (3.41)$$

- Le nombre moyen \bar{N}_q est également donné:

$$\bar{N}_q = \sum_{j=0}^L j\pi(j) - \frac{E(A)}{2} \approx \sum_{j=b}^L (j-b)\pi(j) + \frac{E(A)}{2} \quad (3.42)$$

et d'après la formule de LITTLE [Little 61], le délai moyen d'attente $E[W_q]$ peut être déterminé par:

$$\bar{N}_q = \frac{E(A)}{T} (1 - \Pi_{\text{over}}) E(W_q) \quad (3.43)$$

3.2.1.3 Démonstrations

La chaîne $(X_n)_{n \geq 0}$ est irréductible, apériodique et récurrente positive (cela se vérifie facilement), il existe alors une distribution stationnaire: $\pi(k)_{0 \leq k \leq L}$ qui est aussi la distribution limite. De plus, la loi des grands nombres est valable pour toute fonction bornée φ [Asmussen 87, p16]:

$$\lim_{m \rightarrow +\infty} \frac{1}{m} \sum_{j=1}^m \varphi(X_j) = \sum_{k=0}^L \varphi(k)\pi(k) \quad (3.44)$$

Nous cherchons d'abord la probabilité de dépassement pour une file d'attente de capacité limitée à L , avec $L \geq b$.

Durant $]0, mT]$, la proportion de clients refusés est:

$$\frac{(A_1 + A_2 + \dots + A_m) - (O_1 + O_2 + \dots + O_m) - X(mT)}{A_1 + A_2 + \dots + A_m}$$

où

$$X(mT) \leq L,$$

et

$$\lim_{m \rightarrow +\infty} \frac{1}{m} \sum_{j=1}^m A_j \rightarrow \infty$$

Les variables aléatoires A_j étant indépendantes et identiquement distribuées, on a:

$$\lim_{m \rightarrow +\infty} \frac{1}{m} \sum_{j=1}^m A_j = E(A)$$

Comme $O_j = \min(b, X_j)$, on a, d'après le théorème précédent:

$$\lim_{m \rightarrow +\infty} \frac{1}{m} \sum_{j=1}^m O_j = E[\min(b, X)]$$

où X est une variable aléatoire de distribution $(\pi(k))_{0 \leq k \leq L}$ et:

$$E[\min(b, X)] = b - \sum_{k=0}^{b-1} (b-k)\pi(k)$$

On a alors la probabilité de dépassement:

$$\Pi_{\text{over}} = 1 - \frac{b}{E(A)} + \frac{1}{E(A)} \sum_{k=0}^{b-1} (b-k)\pi(k)$$

Il reste à déterminer la distribution stationnaire $\pi(k)$.

La matrice des probabilités de transition P de la chaîne s'écrit:

	0	1	...	L-b	L-b+1	...	L-1	L
0	a_0	a_1	...	a_{L-b}	a_{L-b+1}	...	a_{L-1}	\hat{a}_L
1	a_0	a_1	...	a_{L-b}	a_{L-b+1}	...	a_{L-1}	\hat{a}_L
b-1	a_0	a_1	...	a_{L-b}	a_{L-b+1}	...	a_{L-1}	\hat{a}_L
b	a_0	a_1	...	a_{L-b}	a_{L-b+1}	...	a_{L-1}	\hat{a}_L
b+1	0	a_0	...	a_{L-b-1}	a_{L-b}	...	a_{L-2}	\hat{a}_{L-1}
L	0	0	...	a_0	a_1	...	a_{b-1}	\hat{a}_b

avec $\hat{a}_k = P[A \geq k] = 1 - (a_0 + a_1 + \dots + a_{k-1})$.

Il ne reste plus qu'à résoudre le système linéaire suivant:

$$\left\{ \begin{array}{l} \pi = \pi P \\ \sum_{k=0}^L \pi(k) = 1 \\ \pi(k) \geq 0, \quad 0 \leq k \leq L \end{array} \right. \quad (3.45)$$

Pour résoudre le système 3.45, on a utilisé la méthode de surrelaxation [Tijms 86, appendix D].

La table 3.1 donne la capacité minimale requise de la file d'attente afin que la probabilité de rejet ne dépasse pas 10^{-6} . On définit la charge normalisée $\rho_t = \rho/b$

Table 3.1

b	$\rho_t =$	0,2	0,5	0,7	0,8	0,9
1		6	11	19	29	56
2		7	12	20	30	57
5		9	14	22	32	59
10		11	18	26	36	63
25		25	30	39	49	76

En supposant une probabilité de dépassement négligeable pour un L donné, on a le nombre de moyen de clients dans la file d'attente à l'instant nT^-

$$E[X^-] = \sum_{j=0}^L j\pi(j) \approx \sum_{j=b}^L (j-b)\pi(j) + E(A) \quad (3.46a)$$

et à l'instant nT^+

$$E[X^+] = E[X^-] - E(A) \approx \sum_{j=b}^L (j-b)\pi(j) \quad (3.46b)$$

Pour obtenir le nombre moyen de clients \bar{N}_q , il faut "moyenner" $E[X^-]$ et $E[X^+]$ sur une période T. On a alors:

$$\bar{N}_q = \sum_{j=0}^L j\pi(j) - \frac{E(A)}{2} \approx \sum_{j=b}^L (j-b)\pi(j) + \frac{E(A)}{2}$$

et d'après la formule de LITTLE, le délai moyen d'attente W_q peut être déterminé par:

$$\bar{N}_q = \frac{E(A)}{T}(1-\Pi_{over})E(W_q)$$

3.2.2 Cas général: l'interrupteur ferme irrégulièrement

3.2.2.1 Description

Soit T la période composée de M slots, parmi lesquels c slots sont dédiés à une file d'attente particulière du premier étage (i.e: l'interrupteur de cette station ferme durant ces c slots). La capacité de cette file est limitée à L . En induisant M chaînes de Markov aux instants j ($j = 0, 1, \dots, M-1$), en régime stationnaire, ce système peut être décrit par l'équation 3.3. On la réécrit ci-dessous:

$$X_{j+1} = \begin{cases} \min[(X_j-1)^+ + A_{j+1}, L], & \text{si slot } j \text{ est dédié à la station} \\ \min[X_j + A_{j+1}, L], & \text{ailleurs} \end{cases} \quad (3.47)$$

En écrivant cette équation, l'existence d'une distribution limite de probabilité du nombre de clients dans la file a été anticipée, ce qui permet de supprimer l'indice n .

3.2.2.2 Résultats obtenus

La distribution des probabilités du nombre de clients dans la file se calcule en utilisant la même méthode de surrelaxation avec le programme développé dans la partie précédente (cf. §3.2.1).

3.2.2.3 Démonstrations

- Pour un slot j dédié à la file d'attente considérée, le calcul des probabilités stationnaires $\pi_j(k)$, $k = 0, 1, \dots, L$ est exactement identique à celui du cas simple avec $b=1$.
- Pour un slot j qui n'est pas dédié à la file d'attente considérée, une analyse similaire au cas simple est applicable (avec $b=0$).

En effet, il suffit d'étudier l'équation suivante:

$$X_{j+1} = \min(X_j + A_{j+1}, L).$$

Puisque les A_j sont indépendantes et identiquement distribuées, on peut enlever l'indice j et on définit:

$$a_k = P[A=k].$$

Soit p_{ij} la probabilité de transition de la chaîne de Markov $\{X_j\}$, on a:

pour $0 \leq i \leq L$,

$$p_{ij} = \begin{cases} a_{j-i}, & i \leq j < L \\ \sum_{k=L-i}^{\infty} a_k, & j = L \end{cases}$$

Enfin, pour le système global, il suffit de résoudre deux systèmes linéaires (au lieu de M en cas de capacité infinie) comme (3.45). On peut obtenir les distributions des probabilités stationnaires $\pi_j(k)$:

$$\begin{bmatrix} \pi_0(0), & \pi_0(1), & \dots, & \pi_0(L) \\ \pi_1(0), & \pi_1(1), & \dots, & \pi_1(L) \\ \vdots & & & \\ \vdots & & & \\ \pi_{M-1}(0), & \pi_{M-1}(1), & \dots, & \pi_{M-1}(L) \end{bmatrix}$$

Pour les instants $(j+nT)$, tous les résultats obtenus dans le paragraphe 3.2.1 sont applicables.

3.3 Application aux services périodiques de FIP

Nous traitons deux problèmes concrets concernant respectivement le service de messagerie périodique et le service de demande explicite périodique (cf. §2). Les comportements de ces deux services périodiques ne dépendent pas du trafic global. On note que ces deux cas utilisent deux disciplines ("queueing discipline") différentes.

Pour évaluer les performances des deux services périodiques du protocole FIP, nous nous fondons sur le délai et le taux d'utilisation ("throughput"). On rappelle que le critère de rendement du protocole est déjà donné dans le chapitre 2 (cf. §2.2.3).

D'ailleurs, on suppose que l'interrupteur ferme régulièrement pour les deux services.

3.3.1 Service de messagerie périodique

On suppose que le flux d'arrivée de messages dans une station source est un processus Poissonnien d'intensité λ . Les messages sont de taille constante donc le temps de service est 1 slot ($=1/T$). Si la fenêtre de "messagerie périodique" est suffisamment dimensionnée, $I(t) = I(t + nT)$ sera le cas simple avec $b=1$. La capacité de la file d'attente F_MSGcyc est limitée à L .

On a:

$$a_k = e^{-\lambda T} \frac{(\lambda T)^k}{k!}, \quad k = 0, 1, \dots$$

$\pi(k)$ est obtenue par l'analyse numérique (méthode de surrelaxation) . On a alors la probabilité de rejet selon (3.41):

$$\Pi_{over} = 1 - \frac{1}{\lambda T} (1 - \pi(0)) \tag{3.48}$$

NB: On note que dans le cas de capacité limitée, $\pi(0) \neq 1 - \rho$.

La table 3.2 donne la capacité minimale requise L afin que la probabilité de rejet ne dépasse pas 10^{-6} . On note que l'unité de temps pour $E(W_q)$ est T .

Table3.2

λT	0,2	0,5	0,7	0,8	0,9
L	6	11	19	29	56
X_q	0,225	0,75	1,52	2,4	4,95
\bar{N}_q	0,125	0,5	1,17	2,0	4,50
$E[W_q]$	0,625	1,0	1,671	2,5	5,0

Comme cette probabilité de rejet est très faible, on peut considérer que la probabilité pour que la file soit vide est $\pi_0 \approx 1 - \rho$. Rappelons que ce paramètre est utile pour le dimensionnement de la fenêtre de messagerie périodique (cf. §2.2.2).

Le délai de transfert (ou temps de réponse) d'un message W est composé d'un délai d'attente dans la file F_MSGcyc W_q et d'un délai de transmission W_s qui est proportionnel à la longueur du message transmis. i.e:

$$E(W) = E(W_q) + E(W_s) \quad (3.49)$$

Pour un flux d'entrée Poissonien simple d'intensité λ , en régime stationnaire, on a d'après la formule (3.43) :

$$E(W_q) = \frac{\bar{N}_q}{\lambda(1 - \Pi_{\text{over}})}$$

où \bar{N}_q est le nombre moyen de messages dans F_MSGcyc, il est donné par la formule (3.42):

$$\bar{N}_q = \sum_{j=0}^L j \pi(j) - \frac{\lambda T}{2}$$

Pour une distribution quelconque de la longueur des messages L (en nombre de bits), on suppose que $E(W_s) = E(L)/\text{débit} = m$, et la durée d'un cycle de scrutation de cette F_MSGcyc T est égale à 100 fois m , autrement dit que la capacité d'évacuation μ de cette file considérée F_MSGcyc est $\mu = 1/100m$.

Sur la figure 3.5, on donne le délai moyen de transfert d'un message normalisé: $E(W)/m$ en fonction de la charge normalisée $\rho = \lambda/\mu = \lambda T$.

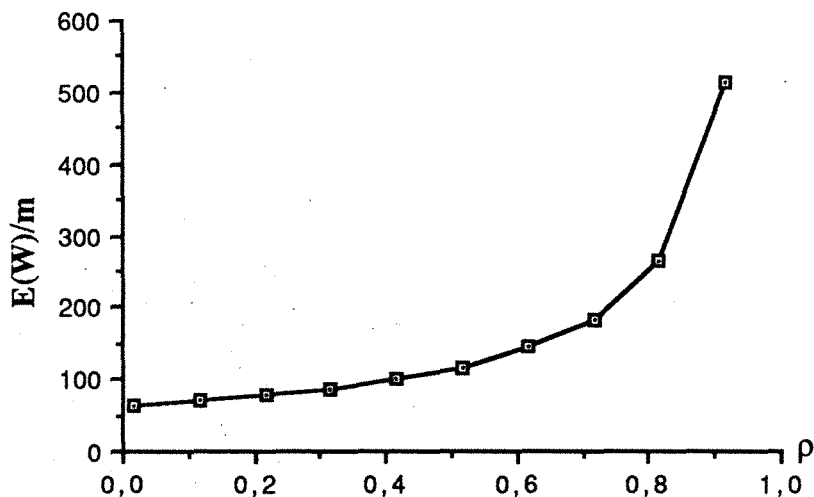


Fig. 3.6 Délai moyen de transfert vs. charge

On note que ce délai est indépendant du trafic généré par les autres F_MSGcyc. En effet, si on ne parle que du délai d'accès au médium sans compter le délai d'attente pour qu'un message se place au début de la file (on peut imaginer que la capacité de la file F_MSGcyc est limitée à $L=1$), ce délai d'accès est borné par T avec comme valeur

moyenne $T/2$. C'est donc un service "express" sur lequel on peut compter lorsqu'il y a des messages urgents à émettre, et bien sûr, à condition qu'il n'y ait pas trop de messages. La figure 3.6 montre que si un message arrive en moyenne tous les $2T$, le délai moyen de transfert ne dépassera pas T , sachant que les messages arrivent de façon aléatoire selon un processus de Poisson.

En ce qui concerne le taux d'utilisation du médium par ce service, on s'aperçoit qu'il est tout simplement le rapport entre la charge totale du réseau (en nombre de messages arrivés durant un macro cycle T_m) et la dimension (en nombre de messages équivalents) de la fenêtre messagerie périodique, i.e:

si le nombre total d'arrivées: $\rho = \sum_{i=1}^N \rho_i$ pour tous les $i=1, 2, \dots, N$ (nombre de

F_MSGcyc total dans le réseau), et la capacité d'évacuation (en nombre de messages):

$C = \frac{T_m/T_{bc} - 1}{\sum_{n=1} T_{mp}(n)/m}$, on a le taux d'utilisation:

$$\tau_{mp} = \frac{\rho t}{C} \tag{3.50}$$

On note que ce taux d'utilisation peut atteindre 100%.

3.3.2 Service de demande explicite spécifiée périodique

On suppose que le flux d'arrivée de demandes dans une station source est un processus à accroissement indépendant et stationnaire. La capacité de buffer B_REQ est limitée à $L=1$ avec discipline lifo (la nouvelle arrivée écrasera l'ancienne). Les demandes sont de taille constante: 1 slot (= $1/T$). Si la fenêtre "explicite périodique" est suffisamment dimensionnée, $I(t) = I(t + nT)$ sera le cas simple avec $b = 1$.

Puisque les résultats obtenus dans §3.1 et §3.2 ne sont pas applicables directement pour l'évaluation des performances de ce service, on subdivise ce paragraphe en deux parties: Etude analytique et Application.

3.3.2.1 Etude analytique

On constate qu'il s'agit d'un cas extrêmement particulier. On montre ce qu'il se passe dans ce système sur la figure 3.7.

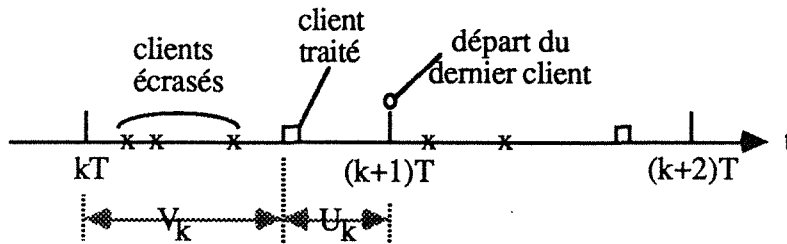


Fig.3.7 Activité des clients dans une file lifo avec $L=1$

Deux questions importantes se posent:

- Q1) *Quelle est la probabilité d'écrasement d'un client ?*
 Q2) *Quel est le temps d'attente du dernier arrivé lors d'un cycle T ?*

La réponse à Q1 est simple. On peut répondre à cette question avec deux méthodes.

Méthode 1:

La probabilité d'écrasement d'un client dans le système lifo est exactement la probabilité de rejet d'un client dans un système fifo. Selon l'étude effectuée (cf. l'équation 3.41), on a immédiatement:

$$\Pi_{\text{écrase}} = \Pi_{\text{over}} = 1 - \frac{1}{E(A)}(1 - \pi(0)) \quad (3.51)$$

où les $\pi(k)$, $k = 0, 1$ peuvent être obtenues par l'analyse numérique. A titre d'exemple, on donne dans la table 3.3 quelques probabilités d'écrasement en fonction de l'intensité du flux d'arrivée. On a supposé que le flux d'arrivée des demandes est Poissonnien d'intensité λ .

λT	0,2	0,5	0,7	0,8	0,9
$\Pi_{\text{écrase}}$	0,094	0,213	0,281	0,312	0,341

Méthode 2:

On peut déduire la probabilité d'écrasement d'une manière directe.

Soit $v(t)$ le nombre de clients arrivés durant $]0, t]$. On suppose que $(v(t), t \geq 0)$ ($v(0)=0$) est un processus à accroissements indépendants et stationnaires.

Soit X_k le nombre de clients écrasés durant le cycle k , i.e: durant $]kT, (k+1)T]$.

$$X_k = \max(0, v[(k+1)T] - v(kT) - 1), \forall k \geq 0 \quad (3.52)$$

Comme $(v(t), t \geq 0)$ est à accroissements indépendants et stationnaires, on constate que les v.a X_k pour $k \geq 0$ sont indépendantes et identiquement distribuées avec:

$$P[X_0 = n] = P[\max(0, v(T)-1) = n] = \begin{cases} P_0(T)+P_1(T), & \text{si } n = 0 \\ P_{n+1}(T), & \text{si } n \geq 1 \end{cases}$$

où:

$$P_n(T) = P[v(T) = n].$$

On en déduit la fonction génératrice de X_0 : $G_{X_0}(z) = G(z)$; $H(z) = G_{v(T)}(z)$

$$G(z) = \frac{1}{z}H(z) + P_0(T)(1 - \frac{1}{z})$$

qui fournit $\pi(0)$ et $\pi(1) = 1 - \pi(0)$.

Puis:

$$E(X_0) = E\{v(T)\} - (1 - P_0(T))$$

$$\text{Var}(X_0) = \text{Var}\{v(T)\} - 2E\{v(T)\}P_0(T) + 1 - P_0(T)$$

Lorsque $(v(t), t \geq 0)$ est un processus de Poisson d'intensité λ , on a:

$$E\{v(T)\} = \lambda T \text{ et } \text{Var}\{v(T)\} = \lambda T$$

$$E(X_0) = e^{-\lambda T} - 1 + \lambda T$$

$$\text{Var}(X_0) = 1 + \lambda T - e^{-\lambda T}(1 + 2\lambda T)$$

La probabilité d'écrasement:

$$\Pi_{\text{écrase}} = \frac{E(X_0)}{E\{v(T)\}} = \frac{e^{-\lambda T} - 1 + \lambda T}{\lambda T} \quad (3.53)$$

On a vérifié que l'équation (3.51) et l'équation (3.53) nous donnent exactement les mêmes résultats.

On en déduit aussi: $\pi_0 = P_0(T)$ et $\pi_1 = 1 - P_0(T)$.

NB: On peut aussi s'intéresser à la probabilité suivante:

$P[X_k = n \mid v[(k+1)T] - v(kT) \geq 1]$ = probabilité de perdre n clients durant le cycle $]kT, (k+1)T]$, sachant qu'il est arrivé au moins un client.

$$P[X_k=n \mid v[(k+1)T]-v(kT) \geq 1] = \frac{P[v[(k+1)T]-v(kT) = n+1]}{P[v[(k+1)T]-v(kT) \geq 1]} = \frac{P_{n+1}(T)}{1-P_0(T)}$$

Lorsque $v(t)$ est un processus Poissonnien, on trouve pour la moyenne:

$$E[X_k \mid v[(k+1)T]-v(kT) \geq 1] = \frac{e^{-\lambda T} - 1 + \lambda T}{1 - e^{-\lambda T}}$$

La réponse à Q2 n'est pas immédiate. Puisque la probabilité d'écrasement est trop importante, les formules approximatives concernant des paramètres moyens (les équations 3.42 et 3.43) ne sont plus valables. D'ailleurs, dans un système fifo, le délai d'attente est pour le client le plus ancien, tandis que dans un système lifo, le délai d'attente est toujours pour le client le plus récent.

On se place dans un intervalle fixe $]kT, (k+1)T]$, et on calcule le temps écoulé entre le dernier arrivé et l'instant de départ $(k+1)T$. Ce qui suppose que dans l'intervalle $]kT, (k+1)T]$, au moins un client soit arrivé.

On déduit la loi de V_k (cf. fig.3.7), sachant qu'il est arrivé au moins un client durant l'intervalle $]kT, (k+1)T]$.

$$P[V_k < x \mid v[(k+1)T]-v(kT) > 0] = \frac{P\{V_k < x, v[(k+1)T]-v(kT) > 0\}}{1 - P[v(T)=0]}$$

$$P\{V_k < x, v[(k+1)T]-v(kT) > 0\}$$

$$= P\{\text{aucun arrivé durant }]kT+x, (k+1)T], \text{ et un arrivé au moins durant }]kT, kT+x]\}$$

$$= P\{v[(k+1)T]-v(kT+x)=0, v(kT+x)-v(kT) > 0\}.$$

$(v(t); t \geq 0)$ étant à accroissements indépendants et stationnaires, on a pour $0 \leq x \leq T$:

$$P[V_k < x \mid v(T) > 0] = \frac{P[v[(T-x)]=0]P[v(x) > 0]}{1 - P[v(T)=0]} = \frac{P_0(T-x)(1-P_0(x))}{1 - P_0(T)}$$

Quantité qui ne dépend pas de l'intervalle choisi. On peut donc simplement considérer:

$$F(x) = P[V_0 < x \mid v(T) > 0] = \frac{P_0(T-x)(1-P_0(x))}{1 - P_0(T)} \quad ; \quad 0 \leq x \leq T \quad (3.54)$$

Connaissant la loi de V_k conditionnée par $v[(k+1)T]-v(kT) > 0$, on en déduit la loi de U_k conditionnée par $v[(k+1)T]-v(kT) > 0$.

$$P[U_k < x \mid v[(k+1)T] - v(kT) > 0] = P[V_k > T-x \mid v(T) > 0] = 1 - F(T-x) \\ = P[U_0 < x \mid v(T) > 0].$$

Finalement, on a: $G(x) = 1 - F(T-x)$, et pour $0 < x \leq T$:

$$G(x) = P[U_k < x \mid v[(k+1)T] - v(kT) > 0] = \frac{1 - P_0(T) - P_0(x) + P_0(x)P_0(T-x)}{1 - P_0(T)} \quad (3.55)$$

Dans le cas Poissonnien, il vient pour $0 < x \leq T$:

$$G(x) = \frac{1 - e^{-\lambda x}}{1 - e^{-\lambda T}}$$

et:

$$g(x) = G'(x) = \frac{\lambda e^{-\lambda x}}{1 - e^{-\lambda T}}$$

$$E[U_0 \mid v(T) > 0] = \int_0^T x g(x) dx = \frac{e^{\lambda T} - \lambda T - 1}{\lambda(e^{\lambda T} - 1)} \quad (3.56)$$

D'ailleurs, $U_k + V_k = T$.

$$E[V_0 \mid v(T) > 0] = \int_0^T x F'(x) dx = T + \frac{T}{e^{\lambda T} - 1} - \frac{1}{\lambda} \quad (3.57)$$

On a: $E[U_0 \mid v(T) > 0] + E[V_0 \mid v(T) > 0] = T$

3.3.2.2 Application

Pour un flux d'arrivée de demandes Poissonnien de paramètre $\rho = \lambda T < 1$, on a:

$$\pi_0 = P_0(T) = 1 - e^{-\rho}$$

On rappelle que ce paramètre est utile au dimensionnement de la fenêtre explicite périodique (cf. §2.2.2).

Le délai moyen de transfert d'une demande explicite spécifiée se calcule par:

$$E(W) = E(W_q) + E(W_s)$$

où le délai moyen d'attente $E(W_q)$ est donné par la formule (3.56).

Si le nombre moyen d'identifieurs d'une telle demande est n , et la durée moyenne d'un transfert d'une variable identifiée m , le délai moyen de transmission est donc:

$$E(W_s) = m * n$$

Si $T=100(m*n)$, on montre sur la figure 3.8 le délai moyen de transfert normalisé: $E(W)/m*n$ et la probabilité d'écrasement (donnée par la formule 3.53) en fonction de la charge normalisée ρ .

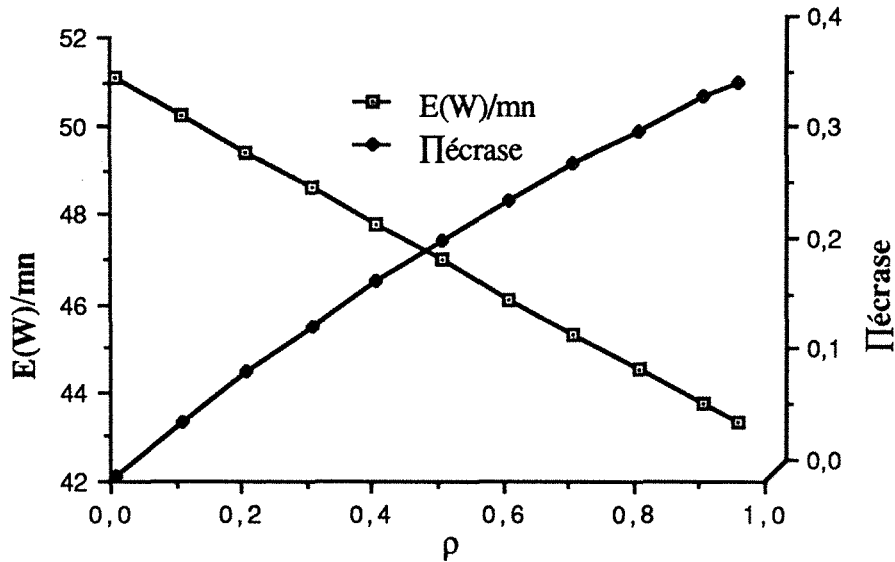


Fig. 3.8 Délai moyen d'une demande explicite spécifiée périodique et les probabilités d'écrasement vs. charge

On constate que le délai moyen de transfert diminue lorsque la charge augmente. Cela est dû à la discipline d'écrasement. En effet, le délai d'attente est borné à T puisque la capacité de buffer B_REQ est limitée à 1. La probabilité d'écrasement augmente lorsque le nombre de demandes d'arrivée dans un buffer (charge) augmente.

Pour ce service, la charge ρ peut dépasser 1. En effet, il n'y a pas de limite pour cette charge. Mais le taux d'utilisation atteint rarement 100%. La charge totale du réseau alimentée par ce service se calcule par:

$$\rho_t = \sum_{i=1}^N \rho_i (1 - \Pi_{\text{écrase}}(i))$$

et le taux d'utilisation se calcule toujours par la formule (3.50).

CHAPITRE 4

ETUDE ANALYTIQUE DU MODELE, DEUXIEME ETAGE

4. Etude analytique du modèle, deuxième étage

Dans ce chapitre, nous poursuivons l'étude analytique du modèle général établi dans le chapitre 2. Le deuxième étage est aussi modélisé par une file d'attente avec un serveur disponible de façon intermittente (cf. Fig.4.1).

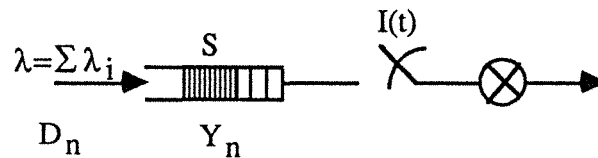


Fig. 4.1 Modèle de base du deuxième étage

La distribution des clients dans la file d'attente du deuxième étage est décrite par la figure 4.2.

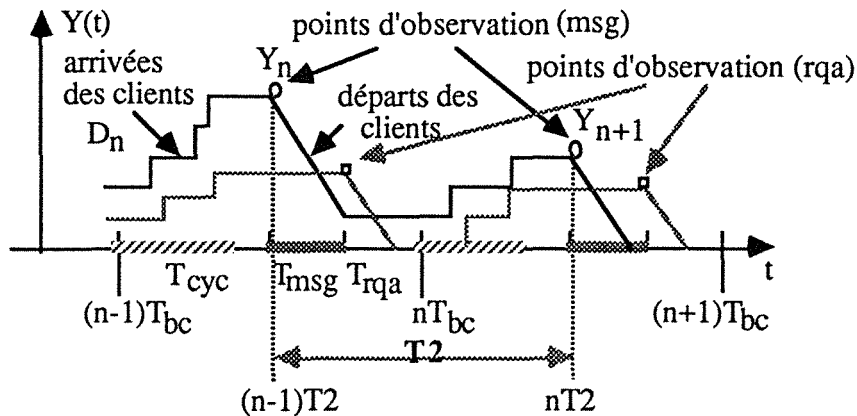


Fig. 4.2 Distribution de clients dans la file du deuxième étage

Le modèle générique est toujours:

$$Y_{n+1} = (Y_n - B_n)^+ + D_{n+1} \quad , \quad x^+ = \max(0, x) \tag{4.1}$$

où:

- $Y(t)$ est le nombre de clients dans la file à l'instant t
- Y_n est le nombre de clients dans la file à l'instant (nT_2) juste avant la fermeture de l'interrupteur $I(t)$
- T_2 est l'intervalle de temps séparant deux fermetures consécutives de l'interrupteur, on suppose dans toute notre étude que $T_2 = T_{bc}$ qui est constant
- D_n est le nombre de clients qui sont arrivés durant $[(n-1)T_2, nT_2]$ ou plus précisément durant $T_{cyc}(n) = T_{cyc}(n + T_m/T_{bc})$, car les clients entrant dans le deuxième étage ne peuvent arriver que durant T_{cyc} (cf. figure 4.2)
- B_n est la capacité d'évacuation du deuxième étage ou autrement dit, le nombre maximal de clients qui peuvent quitter la file durant la fermeture de l'interrupteur. Si la taille des clients est constante, et la durée de transmission d'un tel client est m secondes par exemple, $B_n = T_{msg}/m$ dans l'exemple de la figure 4.2

Deux paramètres font la différence entre le modèle du premier étage et celui du deuxième étage:

- le flux d'entrée (D_n) n'est plus un processus à accroissements indépendants et stationnaires connu, mais la superposition des flux de sortie de chaque file du premier étage. De plus, les clients n'arrivent qu'à des instants discrets durant T_{cyc} . Les caractéristiques de ce flux restent à déterminer.

- le nombre de clients évacués à la fois est en général $B_n > 1$ au lieu de $B_n = 1$.

La plupart du temps, l'étude est menée sous l'hypothèse que $B_n = b = \text{constant}$.

Ce chapitre comprend quatre parties:

4.1 Flux d'entrée du deuxième étage

4.1.1 Flux de sortie d'une file du premier étage

(approximation du flux de sortie permettant de supprimer la dépendance)

4.1.2 Flux d'entrée du deuxième étage

(approximation de la superposition des flux de sortie du premier étage par une loi de Poisson)

4.2 Deuxième étage

4.2.1 Capacité de la file infinie

4.2.2 Capacité de la file limitée à L

4.2.3 Divers (*Priorité, B_n variable*)

4.3 Application aux services aperiodiques de FIP

4.3.1 Service de messagerie aperiodique (msg)

4.3.2 Services de demande explicite aperiodique (rqa)

(sous les hypotheses:

- premier etage: capacite infinie et interrupteur "regulier" (cf. §3);
- deuxieme etage: capacite limitee à L et $B_n = b$)

4.4 Conclusion

4.1 Flux d'entrée du deuxième étage

L'étude analytique du deuxième étage ne peut être effectuée tant qu'on ne connaît pas les caractéristiques du flux d'entrée de cet étage, sachant que le flux d'entrée du deuxième étage durant un cycle élémentaire T_{bc} est tout simplement la somme (ou superposition) des flux de sortie des files du premier étage durant le même cycle T_{bc} , ou plus précisément, durant T_{cyc} de ce cycle élémentaire.

Dans ce paragraphe, nous étudions d'abord le flux de sortie d'une file d'attente du premier étage durant T_{cyc} et puis le flux d'entrée du deuxième étage.

4.1.1 Flux de sortie d'une station du premier étage

On suppose que la probabilité d'overflow de la file du premier étage est négligeable, ce qui nous permet de considérer une file d'attente de capacité infinie. Tous les résultats obtenus dans le chapitre précédent sont donc applicables.

Puisque le flux d'entrée du deuxième étage est la superposition des flux de sortie des files du premier étage, deux flux de sortie du premier étage différents (avec l'interrupteur "régulier" et "irrégulier") nous conduisent à étudier deux modèles.

Dans le cas où les interrupteurs du premier étage ferment régulièrement, une étude détaillée est donnée. Dans le cas où les interrupteurs du premier étage ferment irrégulièrement, la méthode d'analyse est montrée.

4.1.1.1 Cas simple où les interrupteurs du premier étage ferment régulièrement

1) Description

Soit T la période de $I(t)$. Durant T , le flux de sortie d'une file d'attente du premier étage est:

$$C_n = \begin{cases} 1, & \text{si un client quitte la station à l'instant } t_n^+ = (nT)^+ \\ 0, & \text{ailleurs} \end{cases} \quad (4.2)$$

NB: Le T dans ce paragraphe désigne l'intervalle de temps entre deux fermetures consécutives de *l'interrupteur d'une station du premier étage*. A ne pas confondre avec T2 qui est l'intervalle de temps séparant deux fermetures consécutives de *l'interrupteur du deuxième étage*. Dans la suite, on suppose que T2 est toujours égal à Tbc. Pourtant, deux cas sont possibles en réalité: $T=Tbc$ et $T>Tbc$. ($T<Tbc$ est un cas impossible).

En régime stationnaire, par la propriété de conservation (i.e: ce qui entre est égal à ce qui sort du système), la probabilité pour qu'un client quitte la file d'attente pendant $]t_n, t_{n+1}]$ est $E(A)$. En effet, notre système à étudier peut être décrit de la manière suivante:

$$\begin{cases} X_{n+1} = (X_n - 1)^+ + A_{n+1}, & X_0 = 0 \text{ et } n \geq 0 \\ C_n = \min(1, X_n), & n \geq 1 \end{cases} \quad (4.3)$$

où:

X_n est le nombre de clients dans la file à l'instant t_n^- ,

A_{n+1} est le nombre de clients arrivés pendant $]t_n, t_{n+1}]$,

et (A_n) est une séquence de variables aléatoires indépendantes et identiquement distribuées avec $E(A) = \rho = \lambda T$.

2) Les problèmes

En appliquant les résultats obtenus dans le chapitre précédent, on trouve que la distribution marginale du processus (C_n) est une variable aléatoire de Bernoulli de paramètre $\rho < 1$. Mais en général, ces variables aléatoires C_n ($n = 1, 2, \dots$) ne sont pas indépendantes, ce qui est gênant pour l'étude du deuxième étage.

3) Résultats obtenus

On trouve que la séquence (C_n) est faiblement dépendante ssi $\gamma = E[A \mathbb{1}_{[A \geq 2]}]$ est suffisamment petit. Ce qui nous permet de remplacer le flux de sortie réel par un autre flux indépendant tant que le nombre $\gamma = E[A \mathbb{1}_{[A \geq 2]}]$ est petit, avec un degré d'approximation convenable.

où:

$$\mathbb{1}_{[A \geq 2]} = \begin{cases} 1, & \text{si } A \geq 2 \\ 0, & \text{sinon} \end{cases}$$

4) Démonstrations

Selon l'étude effectuée dans le chapitre précédent, on résume les propriétés simples du flux de sortie (C_n):

- i) (C_n) n'est pas une chaîne de Markov.
- ii) En régime stationnaire, les variables aléatoires C_n sont indépendantes si et seulement si $P[A \geq 2] = 0$.
- iii) Le nombre de clients sortis durant]0, nT] est:

$$v_n = n - N_0(n) \quad (n \geq 1)$$

où:

$$N_0(n) = \sum_{k=1}^n \mathbb{1}_{[X_k=0]} \text{ est le nombre de visites de l'état 0 durant]0, nT].}$$

Il est bien connu que $\{N_0(n) \mid n \geq 1\}$ est un processus de renouvellement discret ayant les caractéristiques suivantes [Parzen 62, p266]:

Soit $T_1, T_2, \dots, T_j, \dots$ la séquence de temps d'interarrivées entre chaque visite à l'état 0, les variables aléatoires T_j ($j \geq 1$) sont indépendantes et identiquement distribuées avec la distribution:

$$P[T_j = t] = P[X_t = 0, X_{t-1} > 0, \dots, X_1 > 0 \mid X_0 = 0] \quad t \in \mathbb{N} \text{ et } t \geq 1$$

Il est évident que la distribution marginale du processus (C_n) est une variable aléatoire de Bernoulli de paramètre $\rho < 1$. Mais en général, ces variables aléatoires de C_n ($n = 1, 2, \dots$) ne sont pas indépendantes.

Heureusement, l'approximation suivante nous permet de surmonter cet obstacle.

[Approximation 1] Pour un système décrit par l'équation (4.3), la séquence (C_n) est faiblement dépendante ssi $\gamma = E[A \mathbb{1}_{[A \geq 2]}]$ est suffisamment petit [Simonot 90-2].

[Démonstration] Pour mesurer l'écart entre deux processus, on introduit la *distance en variation*: Si u et v sont des variables aléatoires prenant les valeurs dans \mathbb{N}^n , la distance en variation entre u et v est définie par:

$$\sigma(u, v) = \sup_{B \subset \mathbb{N}^n} |P[u \in B] - P[v \in B]| \tag{4.4}$$

L'idée est de trouver d'abord un autre processus en entrée qui garde la même moyenne que le processus à comparer et qui formera une sortie indépendante.

On considère deux systèmes décrits respectivement par l'équation (4.3) et l'équation suivante:

$$\begin{cases} \tilde{X}_{n+1} = (\tilde{X}_n - 1)^+ + \tilde{A}_{n+1}, & n \geq 0 \text{ et } \tilde{Y}_0=0 \\ \tilde{C}_n = \min(1, \tilde{X}_n), & n \geq 1 \end{cases} \quad (4.5)$$

Où (\tilde{A}_n) est une séquence de variables aléatoires indépendantes et identiquement distribuées qui suivent la loi de Bernoulli du même paramètre ρ que A_n .

On note d'abord que $\tilde{C}_n = \tilde{X}_n = \tilde{A}_n; \forall n \geq 1$. Le processus (\tilde{C}_n) est alors une séquence de variables aléatoires de Bernoulli qui sont indépendantes et identiquement distribuées. Supposons qu'on soit en régime stationnaire, on considère:

$$\mathbf{C}^n = (C_n, C_{n-1}, \dots, C_1),$$

$$\tilde{\mathbf{C}}^n = (\tilde{C}_n, \tilde{C}_{n-1}, \dots, \tilde{C}_1) = (\tilde{A}_n, \tilde{A}_{n-1}, \dots, \tilde{A}_1)$$

et la distance en variation (l'écart) correspondante:

$$\sigma(\mathbf{C}^n, \tilde{\mathbf{C}}^n)$$

On obtient l'inégalité suivante [Simonot 90-2]:

$$\sigma(\mathbf{C}^n, \tilde{\mathbf{C}}^n) \leq [(n+1)/(1-\rho)] E[A \mathbf{1}_{[A \geq 2]}] \quad (4.6)$$

où

A est une variable aléatoire suivant la distribution commune des A_n .

L'inégalité (4.6) donne une estimation de dépendance des variables aléatoires:

$$C_1, C_2, \dots, C_n.$$

En effet, $\forall (i_1, i_2, \dots, i_n) \in \{0,1\}^n$, on peut écrire:

$$\begin{aligned} & |P[C_n=i_n, \dots, C_1=i_1] - P[C_n=i_n] \dots P[C_1=i_1]| \\ &= |P[C_n=i_n, \dots, C_1=i_1] - P[\tilde{C}_n=i_n] \dots P[\tilde{C}_1=i_1]| \\ &= |P[C_n=i_n, \dots, C_1=i_1] - P[\tilde{C}_n=i_n, \dots, \tilde{C}_1=i_1]| \\ &\leq \sigma(\mathbf{C}^n, \tilde{\mathbf{C}}^n) \leq [(n+1)/(1-\rho)] E[A \mathbf{1}_{[A \geq 2]}] \text{ (par l'indépendance de } \tilde{C}_j). \end{aligned}$$

Si $E[A \mathbf{1}_{[A \geq 2]}]$ est suffisamment petit, la dépendance est faible. \square

En particulier, si A est une variable aléatoire de Poisson de paramètre ρ , on a:

$$E[A \mathbf{1}_{[A \geq 2]}] = \rho - e^{-\rho} \rho = \rho(1 - e^{-\rho}).$$

Par la formule de Taylor, si ρ est petit, on déduit:

$$\sigma(C^n, \tilde{C}^n) \leq (n+1)(\rho^2 + o(\rho^2)).$$

Donc, pour ρ petit, il est possible de mettre une séquence de variables aléatoires indépendantes et identiquement distribuées de Bernoulli à la place du vrai flux d'entrée. La simulation vérifie aussi cette conclusion [Simonot 90-2]. On note que dans une application FIP, la charge du premier étage est toujours faible, le facteur $E[A \mathbb{1}_{[A \geq 2]}]$ est donc petit.

4.1.1.2 Lorsque les interrupteurs du premier étage ferment irrégulièrement

Dans ce cas, l'étude est délicate. On donne simplement une description.

Soit T la période de $I(t)$. Durant la période T , c slots parmi les M sont dédiés à une file d'attente particulière du premier étage. Le flux de sortie C_n observé à l'instant nT (ou encore le nombre de clients sorties durant la période $[(n-1)T, nT]$) est donné par:

$$C_n = \begin{cases} 0, & p_0 \\ 1, & p_1 \\ \vdots & \\ \vdots & \\ c, & p_c \end{cases} \quad \text{avec} \quad \sum_{j=0}^c p_j = 1 \quad (4.7)$$

où:

$$p_k = P[C_n = k]; 0 \leq k \leq c$$

Le nombre maximal est c clients car chaque file du premier étage ne libère qu'un client à la fois. $C_n = 0$ implique qu'aucune des c files du premier étage n'a délivré de client pour le deuxième étage durant $[(n-1)T, nT]$.

Le calcul de p_0 est simple. Par contre le calcul des p_k avec $k \geq 1$ est difficile.

Théoriquement, ces probabilités p_k sont déterminées par:

- $\pi_j(s)$ ($j=0, 1, \dots, M-1$ et $s=0, 1, \dots$) qui est la distribution limite de probabilité qu'il y a s clients dans la file j . Cette probabilité peut être obtenue en appliquant les résultats du chapitre précédent,

$$- e_j = \begin{cases} 1 & \text{si l'interrupteur ferme durant le slot } j \ (0 \leq j \leq M-1) \\ 0 & \text{sinon} \end{cases}$$

qui décrit le moment où une file du premier étage peut délivrer un client.

Mais dans la pratique, il est difficile d'obtenir ces p_k surtout lorsque c est grand.

En général, les variables aléatoires (C_n) ne sont pas indépendantes les unes des autres tant qu'il reste encore des clients dans une file du premier étage après qu'elle a

délivré 1 client. Selon le théorème de "robustesse quantitative" [LeCam 60] et [Simonot 90-5], (C_n) sont quasi indépendants ssi $\gamma = E[A \mathbb{1}_{[A \geq c+1]}]$ est petit.

4.1.2 Flux d'entrée du deuxième étage

Une fois qu'on a réussi à supprimer la dépendance entre $\{C_n\}$, l'évaluation du flux d'entrée du deuxième étage est plus aisée. Il suffit de faire la somme des C_n^i ($i = 1, 2, \dots, N$, le numéro de files d'attente du premier étage) durant T_{bc} ou plus exactement durant T_{cyc} .

On distingue les deux cas suivants:

- Pour $T = T_{bc}$, le flux d'entrée D_n :

$$D_n = \sum_{i=1}^N C_n^i \quad (4.8)$$

- Pour $T > T_{bc}$, par exemple $T = kT_{bc}$ et $k \leq (T_m/T_{bc})$, le nombre de files d'attente du premier étage à scruter durant T_{bc} sera inférieur au nombre total de files: N . Si les files sont homogènes et les $I_i(t)$ sont uniformément répartis sur T , on a le flux d'entrée D_n :

$$D_n = \sum_{i=1}^{N/k} C_n^i \quad (4.9)$$

Si cela n'est pas le cas (i.e: les stations ne sont pas homogènes), on sera obligé d'entrer dans le détail pour l'obtention de $[D_n^1, D_n^2, \dots, D_n^k]$.

On note que dans les deux cas, les v. a D_n sont indépendantes et identiquement distribuées. Ceci est important pour l'étude du deuxième étage.

On remarque aussi qu'en général, la description de la distribution de probabilité de D_n n'est pas aisée. Pour surmonter cette difficulté, on peut alors adopter la deuxième approximation suivante:

[Approximation 2] Selon les travaux effectués par [LeCam 60] et [Deheuvels 86], on peut approcher D_n par une variable aléatoire de Poisson ayant la même moyenne: ρ . Où:

$$\rho = \sum_{i=1}^N E[C_n^i] = \sum_{i=1}^N \rho_i$$

[Démonstration]

On mesure maintenant l'influence de cette approximation sur la suite (Y_n) décrivant le nombre de clients dans la file du deuxième étage.

On considère les deux modèles ci-dessous:

$$(M) \quad Y_{n+1} = (Y_n - b)^+ + D_{n+1}$$

$$(MA) \quad \tilde{Y}_{n+1} = (\tilde{Y}_n - b)^+ + \tilde{D}_{n+1}$$

(M) est le modèle issu de la première approximation.

(MA) est le modèle obtenu après avoir effectué la seconde approximation.

Pour avoir la distribution stationnaire, on suppose que $E[D_n] = E[\tilde{D}_n] < b$.

Dans la suite, on exprime la distance en variation entre les variables aléatoires Y_n et \tilde{Y}_n en fonction de la distance en variation entre les variables aléatoires D_n et \tilde{D}_n .

Lemme: Soit $Z_n = Y_n - D_n$
 $\tilde{Z}_n = \tilde{Y}_n - \tilde{D}_n$

alors

$$\sigma(Y_n, \tilde{Y}_n) \leq \sigma(Z_n, \tilde{Z}_n) + \sigma(D_n, \tilde{D}_n)$$

Les séquences (Z_n) , (\tilde{Z}_n) sont des processus de Markov, par application du résultat de Kalashnikov [Kalashnikov 78, th.8, p177] suivant:

Théorème

Pour tout $\gamma > 1$, il existe un nombre L tel que:

$$\forall n \geq 1, \quad \sigma(Z_n, \tilde{Z}_n) \leq L [\sigma(D_n, \tilde{D}_n)]^{1-1/\gamma}$$

Par conséquent, ce théorème donne le corollaire suivant:

Corollaire: Pour tout $\gamma > 1$, on a:

$$\forall n \geq 1, \quad \sigma(Y_n, \tilde{Y}_n) \leq (L+1) [\sigma(D_n, \tilde{D}_n)]^{1-1/\gamma}$$

Ce résultat fournit une expression quantitative de la continuité (ou robustesse) de la séquence (Y_n) en fonction de la séquence (D_n) . \square

La quantité $\sigma(D_n, \tilde{D}_n)$ est majorée par: $2\min(9, \rho)(\max p_i)$, $1 \leq i \leq N$ [Le Cam 60]. Pour une majoration plus étroite, on peut consulter [Deheuvels 86]. Dans une application FIP, la simulation sur des exemples montre que l'approximation nous donne une majoration très étroite [Simonot 90-2].

4.2 Deuxième étage

L'étude de cet étage est enfin possible lorsque l'on connaît les caractéristiques du flux d'entrée et surtout lorsque l'on peut approcher ce flux par une suite de variables aléatoires indépendantes et identiquement distribuées.

Comme dans le chapitre 3, on distingue deux cas différents:

- la file du deuxième étage est de capacité infinie
- la file du deuxième étage est de capacité limitée à L

4.2.1 Capacité infinie

On donne d'abord la fonction génératrice de la distribution limite de probabilité du nombre de clients dans la file *aux instants d'observation* $(nT_2)^-$. Puis le nombre moyen de clients dans la file aux instants d'observation.

Ensuite, le délai d'attente d'un client issu d'une station i du premier étage est étudié. Dans un souci pratique, on a donné des bornes pour le délai moyen.

4.2.1.1 Nombre de clients dans la file

On réécrit l'équation (4.1) en mettant $B_n = b$ ci-dessous:

$$Y_{n+1} = (Y_n - b)^+ + D_{n+1}, \quad \text{avec } (x^+ = \max(0, x)) \quad (4.10)$$

On pose $P_j^n = P[Y_n = j]$

On calcule d'abord la fonction génératrice du nombre de clients dans la file d'attente considérée à l'instant $(nT_2)^-$ (juste avant le départ des clients, c.f. fig.4.2).

$$P_{n+1}(z) = E[z^{Y_{n+1}}] = E[z^{\max(Y_n - b, 0) + D_{n+1}}] = \left[\sum_{j=0}^{b-1} P_j^n + \sum_{j=b}^{\infty} P_j^n z^{j-b} \right] E[z^{D_{n+1}}]$$

On suppose que les D_n sont indépendantes et identiquement distribuées (on peut donc supprimer l'indice n), et que la distribution limite existe: $\lim_{n \rightarrow \infty} P_{n+1}(z) = P(z)$ et $\lim_{n \rightarrow \infty} P_j^n = P_j$, si bien que:

$$P(z) = D(z) \sum_{j=0}^{b-1} P_j (z^b - z^j) / [z^b - D(z)] \quad (4.11)$$

Dans l'équation 4.11, lorsque $b=1$, on retrouve l'équation 3.4.

On constate qu'il y a b inconnues: P_0, P_1, \dots, P_{b-1} dans cette équation. Ces b inconnues peuvent être obtenus en appliquant la théorème de Rouché [Hayes 84].

Le nombre moyen de clients dans la file aux points d'observation $(nT_2)^-$ peut être obtenu en dérivant deux fois l'équation (4.11) par rapport à z , puis en calculant $P'(1) = E[Y_\infty]$.

$$E[Y_\infty] = \sum_{j=0}^{b-1} \frac{P_j [b(b-1)-j(j-1)]}{2[b-D'(1)]} - \frac{b(b-1)-D''(1)}{2[b-D'(1)]} + D'(1) \sum_{j=0}^{b-1} \frac{P_j (b-j)}{b-D'(1)} \quad (4.12)$$

où:

$D'(z) |_{z=1}$ et $D''(z) |_{z=1}$ peuvent être calculés lorsqu'on connaît $D(z)$.

On note que les résultats sont valables pour tous les processus ayant l'indépendance entre les périodes successives (i.e: D_n et D_{n+1} sont indépendants).

4.2.1.2 Délai d'attente

On a remarqué que contrairement à ce qui se passe pour le système étudié dans le chapitre 3, les clients dans la file du deuxième étage sortent un par un successivement à cause d'un délai de transmission.

On note $T_2 = T_{bc}$, et $T_{msg} = b$. De plus, on suppose que $T_{bc} = T_{cyc} + T_{msg} + T_{rqa}$.

- Le délai de transmission d'un client du premier étage $h_c = T_{cyc}/N$
- Le délai de transmission d'un client du deuxième étage $h_m = T_{msg}/b$.

Le délai d'attente d'un client particulier issu de la station S_i du premier étage dans la file est montré à la Fig.4.3.

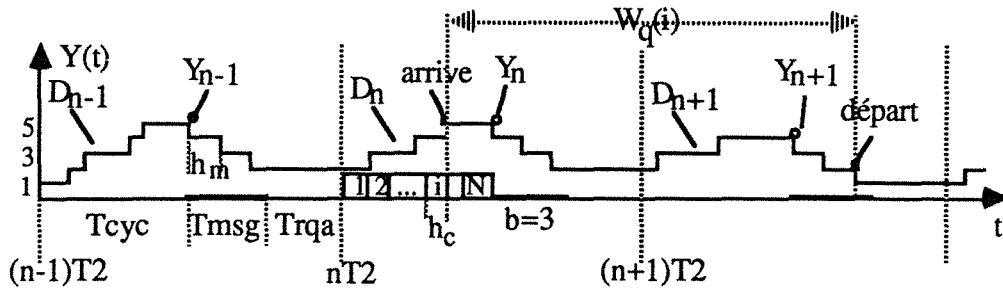


Fig. 4.3 Comportement du délai d'un client lors de la traversée du deuxième étage

On a donc:

$$W_q(i) = (N-i)h_c + Q_i T_{bc} + (R_i+1)h_m \tag{4.13}$$

où Q_i et R_i sont respectivement le quotient et le reste de la division de la variable aléatoire K_i par b .

$$K_i = bQ_i + R_i, \quad 0 \leq R_i < b$$

et

$$K_i = (Y_\infty - b)^+ + \sum_{j=1}^{i-1} C_j \quad \left(\sum_{j=1}^0 C_j = 0 \right)$$

K_i représente le nombre de clients dans la file à l'instant où le client considéré arrive.

Les lois des variables $Y_\infty, C^1, C^2, \dots, C^N$ sont connues, la loi de K_i peut, en principe, s'en déduire.

La transformée de Laplace $\mathcal{L}_i(s)$ de $W_q(i)$ peut alors s'écrire:

$$\begin{aligned} \mathcal{L}_i(s) &= E[e^{-sW_q(i)}] \\ &= e^{-s[(N-1)h_c+h_m]} \left(\sum_{q=0}^{\infty} (e^{-sT_{bc}})^q \sum_{r=0}^{b-1} (e^{-sh_m})^r \pi_{bq+r} \right) \end{aligned} \tag{4.14}$$

où $(\pi_j)_{j \geq 0}$ est la distribution de probabilité de K_i .

En prenant directement l'espérance de l'équation (4.13), on a alors le délai moyen:

$$E(W_q(i)) = (N-i)h_c + h_m + E(Q_i)T_{bc} + E(R_i)h_m$$

On peut encadrer simplement $E(W_q(i))$ afin que le délai moyen vérifie la double inégalité:

$$\begin{aligned}
 E(W_q(i)) &\leq (N-i)h_c + h_m + E(K_i)\frac{Tbc}{b} + (b-1)h_m \\
 &\leq (N-i)h_c + bh_m + E(K_i)\frac{Tbc}{b} \\
 \text{et} & \\
 E(W_q(i)) &\geq (N-i)h_c + h_m + E(K_i)\frac{Tbc}{b} + Tbc\left(\frac{1}{b} - 1\right)
 \end{aligned}
 \tag{4.15}$$

où:

$$E[K_i] = E[(Y_\infty - b)^+] + \sum_{j=1}^{i-1} E[C_j]$$

où:

$$E[C_j] = \rho_j$$

et

$$E[(Y_\infty - b)^+] = \sum_{k=b}^{\infty} (k-b)\pi(k)$$

On peut aussi utiliser une formule approchée, en remplaçant Q_i par $\frac{K_i}{b}$ et R_i par $\frac{b}{2}$.

4.2.2 Capacité limitée à L

On s'intéresse aux deux grandeurs suivantes:

- nombre de clients $Y(t)$ dans la file (§4.2.2.1)
- délai moyen d'attente d'un client W_q (§4.2.2.2)

4.2.2.1 Nombre de clients dans la file

On modifie l'équation (4.1) en ajoutant la contrainte de capacité limitée, si bien que:

$$Y_{n+1} = \min[(Y_n - b)^+ + D_{n+1}, L], \quad \text{avec } (x^+ = \max(0, x)) \tag{4.16}$$

Où les variables aléatoires D_n sont indépendantes et identiquement distribuées. D_n représente le nombre de clients arrivés durant $[nT_2, (n+1)T_2]$.

Cette équation a été étudiée dans le chapitre 3. En effet, on trouve exactement les mêmes équations en ce qui concerne la distribution limite, la probabilité de rejet et le nombre moyen de clients dans la file à des instants d'observation nT_2 . Ils figurent ci-dessous:

- La distribution limite de probabilité $\pi(k)_{0 \leq k \leq L}$ au point d'observation nT_2 peut être calculée par l'analyse numérique (méthode de surrelaxation).

- La probabilité de rejet:

$$\Pi_{\text{over}} = 1 - \frac{b}{E(D)} + \frac{1}{E(D)} \sum_{k=0}^{b-1} (b-k)\pi(k) \quad (4.17)$$

- Le nombre moyen de clients aux instants d'observation: nT_2 (cf. fig4.2):

$$E(Y_{\infty}) = \sum_{k=0}^L k\pi(k) \quad (4.18)$$

On note que le processus A_n entrant dans le premier étage (cf. §3) et le processus D_n entrant dans le deuxième étage n'ont pas la même loi en général.

Comme nous l'avons fait pour le cas de capacité infinie dans le chapitre 3, on peut envisager une approche, en remplaçant la loi de D par une loi de Poisson de même espérance, i.e:

$$\rho = \sum_{i=1}^N E[C_n^i] = \sum_{i=1}^N \rho_i$$

Dans ce cas, la matrice de transition \mathbf{P} ne dépend plus de N . Mais la solution stationnaire sera seulement approchée.

Mais on peut aussi calculer la loi exacte de D . En effet, lorsqu'on suppose que les stations du premier étage ne peuvent libérer qu'un seul client au plus à la fois, les N stations libèrent donc D clients avec:

$$D = C^1 + C^2 + \dots + C^N.$$

Si $\rho = \sum_{i=1}^N \rho_i$ est suffisamment petit, les variables aléatoires D correspondant à des instants

différents (nT_2) sont indépendants (Approximation 1).

Dans le deuxième étage, on peut alors considérer que le flux d'entrée est constitué d'une suite de variables aléatoires indépendantes et identiquement distribuées D_n distribuées selon la loi de D . Si bien que la fonction génératrice de D s'écrit:

$$G_D(z) = \prod_{i=1}^N (1 - \rho_i + \rho_i z) = \sum_{k=0}^N a_k z^k$$

par identification, on en déduit la valeur des a_k . L'algorithme est le suivant:

$$1) \quad \beta_{0,1} = \frac{1 - \rho_1}{\rho_1}$$

$$\beta_{1,1} = 1$$

2) Pour k allant de 2 à N, faire

a)
$$\alpha_k = \frac{1 - \rho_k}{\rho_k}$$

b)
$$\beta_{l,k} = \begin{cases} \beta_{0,k-1}\alpha_k, & \text{si } l=0 \\ \beta_{l-1,k-1} + \beta_{l,k-1}\alpha_k, & \text{si } 1 \leq l \leq k-1 \\ \beta_{k-1,k-1}, & \text{si } l=k \end{cases}$$

3)
$$a_l = \left(\prod_{i=1}^N \rho_i \right) \beta_{l,k} \quad 0 \leq l \leq N$$

Ensuite, on construit la matrice P, on note que P dépend de b (capacité d'évacuation), L (capacité de la file) et N (le nombre de stations du premier étage).

- Le nombre moyen de clients \bar{N}_q dans la file:

On calcule maintenant le nombre moyen de clients en moyennant l'espérance dans un intervalle de temps T2 (=Tbc). Ce paramètre n'est pas vraiment utile dans le pratique (au moins n'est pas plus utile que E[Y∞] qui est la moyenne "maximale"). Pourtant, nous l'utiliserons pour le calcul du délai moyen par application de la formule de Little.

La distribution de l'espérance du nombre de clients dans la file semble apparaît sur la figure 4.4 lorsque les taux des flux d'entrée dans chaque file du premier étage λ_i sont

égaux (avec $\sum_{i=1}^N \lambda_i = \lambda$).

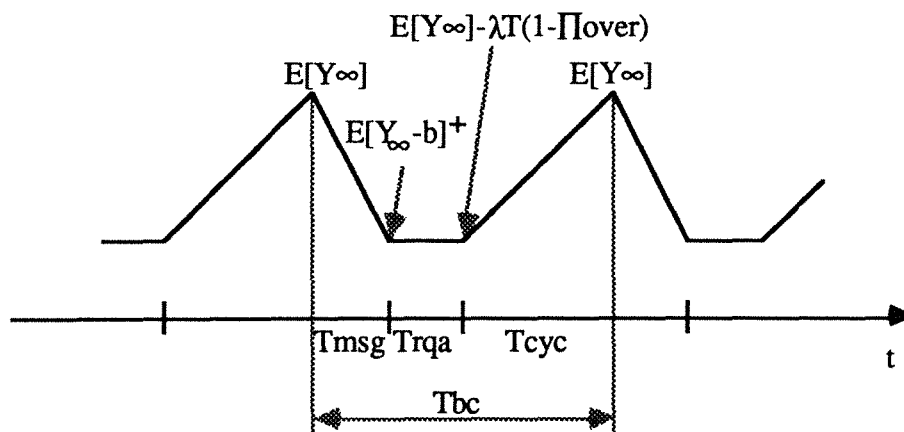


Fig.4.4 Nombre moyen de clients durant Tbc

Sur la figure 4.4, Du fait de la conservation de la "masse", on a:

$$\lambda T_{bc}(1 - \Pi_{over}) = E[Y_{\infty} - (Y_{\infty} - b)^+]$$

La partie gauche de l'équation désigne le nombre moyen d'entrées sur un cycle T_{bc} , et la partie droite désigne le nombre moyen de sorties sur un cycle T_{bc} .

Cette égalité est vérifiée dans la suite:

- Selon l'équation 4.17, on a tout de suite la partie gauche:

$$\lambda T_{bc}(1 - \Pi_{over}) = b - \sum_{k=0}^{b-1} (b-k)\pi(k)$$

- La partie droite se calcule de la façon suivante:

$$\begin{aligned} E[Y_{\infty} - (Y_{\infty} - b)^+] &= \sum_{k=0}^b k\pi(k) + bP[Y_{\infty} > b] \\ &= b + \sum_{k=0}^b k\pi(k) - b \sum_{k=0}^b \pi(k) \\ &= b - \sum_{k=0}^{b-1} (b-k)\pi(k) \end{aligned}$$

En faisant comme si $E[Y(t)]$ était linéaire durant T_{cyc} (ce qui implique que les λ_i sont égaux), le nombre moyen dans la file est alors:

$$\begin{aligned} \bar{N}_q &\equiv \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^{\tau+T} E[Y(t)] dt \\ &= \frac{1}{T_{bc}} \{ E[(Y_{\infty} - b)^+] T_{rqa} + \frac{1}{2} (E[Y_{\infty}] + E[(Y_{\infty} - b)^+]) (T_{bc} + T_{rqa}) \} \end{aligned} \quad (4.19)$$

On peut montrer que l'équation 4.19 reste valide même si les λ_i ne sont pas égaux. (En fait, le calcul de cette équation dépend de $\lambda = \sum_{i=1}^N \lambda_i$).

Remarque: Le nombre moyen de clients dans tout le deuxième étage (file + serveur):

$$\bar{n} = E[N_q] + \lambda(1 - \Pi_{over}).$$

4.2.2.2 Délai d'attente dans la file

On a introduit précédemment des $W_q(i)$ où i correspond à un client issu de la $i^{\text{ème}}$ station du premier étage. On peut introduire W_q délai d'attente d'un client quelconque, indépendamment de la station dont il est issu.

$$W_q = \begin{cases} W_q(1), & \lambda_1/\lambda \\ W_q(i), & \lambda_i/\lambda \\ W_q(N), & \lambda_N/\lambda \end{cases} \quad \text{avec} \quad \sum_{i=1}^N \lambda_i = \lambda \quad (4.20)$$

On peut évaluer

$$E[W_q] = \sum_{i=1}^N \frac{\lambda_i}{\lambda} E[W_q(i)] \quad (4.21)$$

En appliquant la formule de Little [Little 61], [Tijms 86], on a:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t Y(s) ds = \lambda(1 - \Pi_{\text{over}}) E[W_q]$$

La partie gauche est calculée de la façon suivante:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t Y(s) ds = \lim_{n \rightarrow \infty} \frac{1}{nTbc} \sum_{k=0}^{n-1} \int_{kTbc}^{(k+1)Tbc} Y(t) dt \quad \text{avec } n = \lceil \frac{t}{Tbc} \rceil$$

On applique la loi des grands nombres, d'où:

$$\frac{1}{Tbc} E[\int_0^{Tbc} Y(t) dt] = \frac{1}{Tbc} \int_0^{Tbc} E[Y(t)] dt$$

En supposant que tous les λ_i sont égaux, on a alors le délai moyen d'un client:

$$E[W_q] = \frac{1}{2}(Tbc - Trqa) + Tbc \frac{E[(Y_{\infty}-b)^+]}{E[Y_{\infty}] - E[(Y_{\infty}-b)^+]} \quad (4.22)$$

Remarques:

- $E[Y(t)]$ doit être linéaire affine si les λ_i sont égaux. En réalité $E[W_q]$ doit faire intervenir les λ_i . Il doit être possible de retrouver ce résultat en utilisant l'équation 4.21.
- L'équation 4.22 correspond au résultat trouvé pour le premier étage lorsque $b = 1$, $Trqa = 0$ et un seul flux d'entrée de clients (cf. l'équation 3.11).

— Les équations 4.19 et 4.22 sont vérifiées par la simulation (cf. §4.3.1). On verra que ces deux équations nous donnent aussi de bons résultats même si les λ_i ne sont pas égaux.

4.2.3 Divers

Les études précédentes ont été effectuées en supposant $B_n = b$. En pratique, $B_n = b(n) = T_{\text{msg}}(n)/m$ où m est la taille d'un message et $n (= 0, 1, \dots, T_m/T_{bc}-1)$ le numéro du cycle élémentaire dans un macro cycle. B_n représente le nombre maximal de clients qui peuvent sortir durant les fenêtres correspondantes (T_{msg} ou T_{rqa}). $b(n)$ est variable selon le numéro de cycle élémentaire n mais périodique avec $b(n) = b(n+kT_m/T_{bc})$.

Le modèle a été étudié en considérant une seule priorité de clients. Dans la pratique, deux priorités différentes (urgente et normale) existent pour les services de demande explicite aperiodique.

Dans la suite de ce paragraphe, une discussion a été développée autour de ces deux problèmes.

4.2.3.1 B_n variable

Lorsque $B_n = b(n)$ est variable, on n'observe plus une chaîne de Markov homogène aux points d'observation définis sur la figure 4.2.

Soit:

- Y_n^k le nombre de clients dans le système au début de $T_{\text{msg}}(n)$ du $k^{\text{ème}}$ macro cycle;
- D_n^k le nombre de clients qui sont arrivés durant [début de $T_{\text{msg}}(n-1)$, début de $T_{\text{msg}}(n)$] du $k^{\text{ème}}$ macro cycle]. $n = 0, 1, \dots, M-1$.
- $M = T_m/T_{bc}$ le nombre de cycles élémentaires T_{bc} dans un macro cycle T_m .

On peut montrer que (Y_n^k) , correspondant aux différentes valeurs de n , est une chaîne de Markov non homogène. Nous avons: $Y_M^k = Y_0^{k+1}$. Si l'on anticipe sur l'existence d'un état stationnaire, et en supprimant l'indice k , numéro de période T_m . Le système peut être décrit par l'équation suivante:

- capacité infinie:

$$Y_{n+1} = (Y_n - b(n))^+ + D_{n+1} \quad (4.23)$$

- capacité limitée à L :

$$Y_{n+1} = \min[(Y_n - b(n))^+ + D_{n+1}, L] \tag{4.24}$$

On peut montrer que la condition de stationnarité de l'équation (4.23) est $M\rho < \sum_{n=0}^{M-1} b(n)$.

Capacité infinie

On calcule la fonction génératrice de l'équation (4.23):

$$\begin{aligned} P_{n+1}(z) &= E[z^{Y_{n+1}}] = E[z^{\max(Y_n - b(n), 0) + D_{n+1}}] \\ &= E[z^{D_{n+1}}] \sum_{j=0}^{\infty} P[\max(0, Y_n - b(n)) = j] z^j \\ &= D(z) \{ P[Y_n = 0] + \dots + P[Y_n = b(n)] + \sum_{j=1}^{\infty} P[Y_n = b(n) + j] z^j \} \\ &= D(z) \left\{ \sum_{s=0}^{b(n)} P_n(s) + z^{-b(n)} \sum_{s=b(n)+1}^{\infty} P_n(s) z^s \right\} \\ &= D(z) \left\{ \sum_{s=0}^{b(n)} P_n(s) + z^{-b(n)} [P_n(z) - \sum_{s=0}^{b(n)} P_n(s) z^s] \right\} \end{aligned}$$

$$P_{n+1}(z) = D(z) z^{-b(n)} \left[P_n(z) + \sum_{s=0}^{b(n)} P_n(s) (z^{b(n)} - z^s) \right], \quad n=0, \dots, M-1 \tag{4.25}$$

L'application du théorème de Rouché [Hayes 84] permet de calculer les $M \sum_{n=0}^{M-1} b(n)$

inconnues $P_0[b(n)], P_1[b(n)], \dots, P_{M-1}[b(n)]$.

Ainsi nous pouvons obtenir les distributions limites et des valeurs moyennes.

Capacité limitée à L

Le calcul est simplifié lorsque la capacité de la file d'attente est limitée à L. En effet, ce qu'on a à étudier sera réduit à M systèmes linéaires similaires. Puisque les D_n sont indépendantes et identiquement distribuées, on peut supprimer l'indice n et on a: $a_k = P[D=k]$ qui est facile à calculer.

Les distributions limites peuvent être obtenues par une procédure analogue à celle utilisée dans les paragraphes 4.2.2.1 et 3.2.2.

4.2.3.2 Lorsque deux priorités différentes existent

La discipline adoptée par les services de demande explicite aperiodique de FIP est non-préemptive, c'est à dire qu'un client prioritaire ne chasse pas le client non prioritaire si ce dernier est en service (dans le serveur). Les comportements des clients prioritaires sont donc légèrement influencés par les clients non prioritaires.

Pour simplifier l'analyse, on approche ces services de FIP par un système préemptif.

Les clients prioritaires agissent comme si les clients non prioritaires n'existaient pas. Tous les résultats obtenus sont alors applicables.

Les clients non prioritaires ne peuvent quitter le système que durant la période d'oisiveté ("idle period") des clients prioritaires (Fig. 4.5).

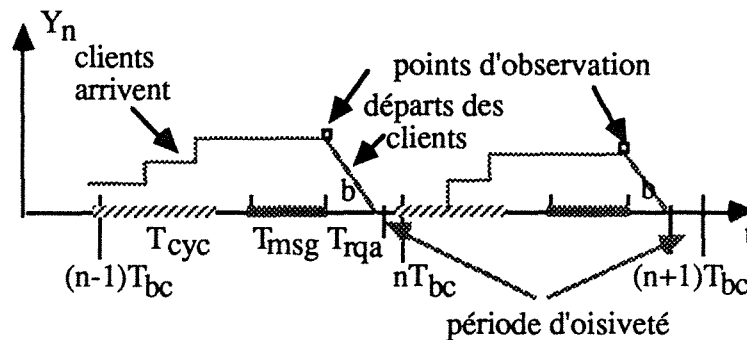


Fig. 4.5 Distribution des clients prioritaires et ses périodes d'oisiveté

Lorsque la file d'attente du deuxième étage est de capacité infinie, l'évolution du nombre de clients non prioritaires dans la file est décrite par:

$$Y'_{n+1} = (Y'_n - (b - Y'_n)^+)^+ + D'_{n+1} \quad (4.26)$$

où Y'_n est le nombre de clients non prioritaires dans la file à l'instant $(nT_2)^-$ juste avant la fermeture de l'interrupteur $I(t)$.

On peut la transcrire l'équation 4.26 en:

$$Y'_{n+1} - D'_{n+1} = (Y'_n - D'_n + D'_n - (b - Y'_n)^+)^+$$

En posant:

$$Z_n = Y'_n - D'_n$$

$$B_n = D'_n - (b - Y'_n)^+$$

on a:

$$Z_{n+1} = (Z_n + B_n)^+ \quad (4.27)$$

Z_n n'est pas une chaîne de Markov, mais la distribution limite existe [Borovkov 81]. Une étude analytique semble bien difficile à réaliser.

Lorsque la file est de capacité limitée à L , on parvient à une modélisation analogue.

4.3 Application aux services aperiodiques de FIP

L'étude analytique du modèle général proposé dans le chapitre 2 est terminée. Dans ce paragraphe, on illustre les performances, en particulier le délai de transfert (temps de réponse), de chacun des services aperiodiques fournis par la couche liaison de données du protocole FIP.

Pour le service de messagerie aperiodique, tous nos résultats sont obtenus par la méthode analytique puis sont comparés avec les résultats de simulation *du modèle*. Ce qui nous permet d'une part, de *vérifier nos formules*, et d'autre part, d'*illustrer les performances* de ce service.

Pour les services de demande explicite aperiodique, l'existence de deux priorités nous gêne beaucoup pour effectuer des calculs concrets. Nous présenterons uniquement les résultats de simulation du modèle en vue d'*illustrer les performances* de ces services.

4.3.1 Service de messagerie aperiodique

Afin de vérifier les résultats analytiques obtenus jusqu'à présent, nous nous intéressons plus particulièrement aux paramètres suivants:

- $E[W_2]$: délai moyen de transfert d'un message pour traverser le deuxième étage,
- $E[Y_\infty]$: nombre moyen de messages dans la file du deuxième étage aux instants d'observation,
- \bar{n} : nombre moyen de messages dans la file du deuxième étage.

Les résultats sont présentés sous forme de tableau.

Pour illustrer les performances de ce service, le délai de transfert total (ou encore le temps de réponse) est évalué. Les résultats sont présentés en graphique.

4.3.1.1 Configuration examinée

Le temps de réponse est défini comme la somme de délais correspondants à la traversée des deux étages:

$$E(W) = E(W_1) + E(W_2)$$

Il est évident que le délai pour traverser le deuxième étage $E(W_2)$ dépend du flux total d'entrée dans le deuxième étage, autrement dit que $E(W)$ dépend du trafic total généré par toutes les stations. Il n'est donc pas étonnant que chaque configuration particulière donne un temps de réponse particulier. En général, des configurations différentes correspondent des performances différentes.

On se propose d'étudier la configuration décrite ci-dessous:

Configuration:

Le premier étage est composé de N files d'attente homogènes de capacité suffisamment grande pour que l'on puisse les considérer comme des files de capacité infinie. La file du deuxième étage est de capacité limitée à L mais avec une probabilité de rejet négligeable.

Le réseau FIP supporte, outre le service de transfert cyclique, le service de messagerie aperiodique et les services de demande explicite (i.e: la classe de conformité 9). Ceux qu'elle ne supporte pas sont les services périodiques (i.e: messagerie périodique et demande explicite spécifiée périodique qui ont été étudiés dans §3.3).

Trois fenêtres "cyc", "msg", "rqa" réservent le médium de transmission pour chacun des trois catégories de services. On a donc $T_{bc} = T_{cyc} + T_{msg} + T_{rqa}$.

Hypothèses:

- H1) La durée de transaction d'une variable est h_c constante,
 la durée de transaction d'une demande est h_d constante,
 la durée de transaction d'un message est h_m constante.
- H2) $T_m = T_{bc}$. Les stations sont homogènes. Elles sont scrutées ainsi qu'il est montré sur la figure 4.6.

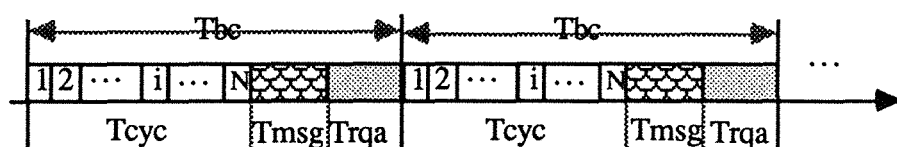


Fig.4.6 Répartition des fenêtres

Sur la figure 4.6, la fenêtre cyclique T_{cyc} est subdivisée en N slots identiques. Le slot i ($i = 1, 2, \dots, N$) est dédié à la station i .

h_c , h_m et h_d sont supposées égales à la durée d'un slot.

Donc,

durant T_{msg} , le système est capable d'évacuer $b = T_{msg}/h_m$ messages,
 durant T_{rqa} , le système est capable d'évacuer $b = T_{rqa}/h_d$ variables sur demandes
 explicites.

On suppose que le flux d'arrivée de messages suit une loi de Poisson d'intensité λ . En posant d'abord $T_{bc} = 20$, $T_{cyc} = 10$, $T_{msg} = T_{rqa} = 5 = b$, $h_c = h_m = h_d = 1$, on a donc:

- la capacité d'évacuation du deuxième étage $\mu = \frac{5}{T_{bc}}$,
- la capacité d'évacuation du premier étage $\sum \mu_i = N \frac{1}{T_{bc}}$.

Remarques:

- La capacité d'évacuation du système global est $\min(\mu, \sum \mu_i)$. Afin d'éviter le goulot d'étranglement au niveau du premier étage, on impose que $\sum \mu_i \geq \mu$.
- Pour qu'il n'y ait pas de rejet important ($\Pi_{over} < 10^{-6}$) au deuxième étage, la charge totale normalisée:

$$\rho_t = \sum_{i=1}^N \lambda_i T_{bc}/b < 1$$

doit être satisfaite (on l'appelle aussi le taux d'utilisation normalisé du médium de transmission par le trafic de messagerie aperiodique ou encore le "throughput" du trafic généré par la messagerie aperiodique).

Dans la suite, on fait varier le nombre de stations $N=5$ et 10 afin de voir l'influence de ce paramètre sur les performances globales.

Lorsque $N = 10$, on a la possibilité de faire varier le modèle de flux d'entrée. Deux modèles sont examinés:

- **Symétrique**, $\lambda_i = \lambda/N$, i.e: toutes les stations ont la même intensité de flux d'arrivée.
- **Asymétrique**, $\lambda_1 = \lambda_2 = \lambda_3 = (\lambda/2)/3$, et $\lambda_4 = \dots = \lambda_{10} = (\lambda/2)/7$, i.e: les 10 stations sont divisées en deux groupes, un groupe de trois stations délivre la moitié de charge totale et un groupe de sept stations délivre l'autre moitié.

NB: Selon la première remarque ci-dessus, le modèle du trafic asymétrique n'est pas intéressant lorsque les stations du premier étage sont peu nombreuse (e.g $N=5$), car dans ce cas, soit la condition imposée $\sum \mu_i \geq \mu$ sera violée, soit le système ne pourra être suffisamment "alimenté".

Afin que la vérification ne soit pas trop limitée, on a examiné une autre combinaison:

- le nombre de stations $N=30$,
- la période de scrutation pour chaque station $T_{bc}=50$,
- la capacité d'évacuation du deuxième étage $b = \frac{T_{msg}}{m} = 10$.

Avec deux modèles de trafic également:

Symétrique: $\lambda_i = \lambda/N$, i.e: toutes les stations ont la même intensité de flux d'arrivée.

Asymétrique: $\lambda_1 = \dots = \lambda_{10} = (\lambda/2)/10$, et $\lambda_{11} = \dots = \lambda_{30} = (\lambda/2)/20$, i.e: les 30 stations sont divisées en deux groupes, un groupe de dix stations délivre la moitié de charge totale et un groupe de vingt stations délivre l'autre moitié.

4.3.1.2 Comparaison entre les résultats analytiques et les résultats de simulation

Dans les tableaux 4.1--4.5, les résultats concernant le deuxième étage sont montrés. Ces résultats sont obtenus par l'utilisation de deux moyens différents: *simulation* et *calcul* en appliquant les formules.

Tab. 4.1 5 stations avec trafic symétrique

ρ_t	0,1998	0,3935	0,5952	0,7952	0,876	0,9424
$E[Y_\infty]$ calc.	0,999	1,968	2,976	3,976	4,380	4,712
$E[Y_\infty]$ sim.	0,998 $\pm 0,02$	1,965 $\pm 0,07$	2,972 $\pm 0,06$	3,971 $\pm 0,08$	4,376 $\pm 0,12$	4,726 $\pm 0,065$
\bar{n} calc.	0,380	0,838	1,266	1,691	1,853	2,006
\bar{n} sim.	0,371 $\pm 0,01$	0,763 $\pm 0,04$	1,220 $\pm 0,03$	1,714 $\pm 0,03$	1,916 $\pm 0,07$	2,102 $\pm 0,04$
$E[W_2]$ calc.	8,5	8,5	8,5	8,5	8,5	8,5
$E[W_2]$ sim.	7,434 $\pm 0,06$	7,758 $\pm 0,11$	8,196 $\pm 0,05$	8,622 $\pm 0,07$	8,749 $\pm 0,07$	8,886 $\pm 0,03$

On peut constater que les $E[Y_\infty]$ calculés et simulés sont très voisins. Les \bar{n} calculés et simulés, les $E[W_2]$ calculés et simulés le sont aussi tant que la charge normalisée ρ_t n'est pas trop faible. Lorsque la charge est faible, les erreurs de simulation ne sont plus négligeables.

On voit aussi que les $E[W_2]$ calculés sont constants quel que soit la charge, ce qui est tout à fait normal car durant un cycle T_{bc} , le nombre de messages pouvant arriver

dans le deuxième étage (issus du premier étage) ne dépasse jamais la capacité d'évacuation de cet étage. En effet, selon l'équation 4.22, on a:

$$E[W_2] = E[W_q] + E[W_s] = \frac{1}{2}(Tbc - Trqa) + 1$$

Lorsque le nombre de stations N=5, le cas "asymétrique" n'est pas intéressant car il y aura sûrement le phénomène de "famine" pour certaines stations.

Tab. 4.2 10 stations avec trafic symétrique

ρ_t	0,2046	0,3906	0,5868	0,7988	0,892	0,9364
E[Y _∞] calc.	1,024	1,975	3,110	4,916	7,260	11,13
E[Y _∞] sim.	1,023 ±0,02	1,962 ±0,03	3,019 ±0,09	4,751 ±0,23	7,026 ±0,88	13,39 ±6,15
\bar{n} calc.	0,364	0,754	1,276	2,541	4,810	8,245
\bar{n} sim.	0,355 ±0,01	0,731 ±0,02	1,233 ±0,05	2,413 ±0,19	4,440 ±0,84	10,68 ±6,09
E[W ₂] calc.	7,16	7,73	8,64	12,2	21,57	35,00
E[W ₂] sim.	6,932 ±0,13	7,487 ±0,13	8,407 ±0,16	12,09 ±0,72	19,89 ±3,55	45,58 ±24,6

Tab. 4.3 10 stations avec trafic asymétrique

ρ_t	0,2025	0,4060	0,6108	0,7876	0,8944	0,966
E[Y _∞] calc.	1,012	2,010	3,100	4,918	7,144	11,12
E[Y _∞] sim.	1,012 ±0,04	2,037 ±0,04	3,147 ±0,13	4,588 ±0,14	6,829 ±0,81	11,71 ±1,66
\bar{n} calc.	0,387	0,807	1,330	2,530	4,644	8,240
\bar{n} sim.	0,399 ±0,02	0,863 ±0,02	1,444 ±0,07	2,457 ±0,10	4,459 ±0,76	9,167 ±1,64
E[W ₂] calc.	7,55	8,08	8,95	13,15	21,08	35,00
E[W ₂] sim.	7,891 ±0,14	8,498 ±0,14	9,459 ±0,27	12,48 ±0,36	19,93 ±3,06	37,93 ±6,35

Tab. 4.4 30 stations avec trafic symétrique

ρ_t	0,260	0,490	0,734	0,878	0,959
E[Y _∞] calc.	2,600	4,900	7,765	11,04	20,48
E[Y _∞] sim.	2,593 ±0,07	4,892 ±0,13	7,632 ±0,25	11,15 ±1,79	21,80 ±7,12
\bar{n} calc.	0,900	1,840	3,100	5,720	15,68
\bar{n} sim.	0,916 ±0,02	1,852 ±0,07	3,236 ±0,16	5,999 ±1,63	16,30 ±7,08
E[W ₂] calc.	17,30	18,80	21,13	32,73	78,54
E[W ₂] sim.	17,62 ±0,27	18,91 ±0,35	22,03 ±0,80	34,19 ±8,00	83,06 ±35,9

Tab. 4.5 30 stations avec trafic asymétrique

ρ_t	0,254	0,505	0,753	0,884	0,967
E[Y _∞] calc.	2,540	5,060	8,006	11,30	22,53
E[Y _∞] sim.	2,532 ±0,08	5,042 ±0,16	7,896 ±0,31	10,23 ±0,49	21,36 ±4,22
\bar{n} calc.	0,877	1,850	3,200	5,510	16,73
\bar{n} sim.	1,015 ±0,04	2,195 ±0,16	3,802 ±0,23	5,512 ±0,39	16,31 ±4,17
E[W ₂] calc.	17,30	18,50	21,33	31,10	86,50
E[W ₂] sim.	19,99 ±0,28	21,73 ±0,24	25,26 ±0,96	31,16 ±1,77	84,00 ±20,1

A travers ces tableaux, on peut constater:

- les résultats de simulation correspondent bien aux résultats analytiques. Les légères différences sont principalement due à la précision limitée de la simulation (95% d'intervalle de confiance avec la méthode de réplique).
- les deux paramètres moyens E(W₂) et \bar{n} obtenus par la simulation sont liés par la formule de Little.

- le comportement du deuxième étage ne change pas quel que soit le modèle du trafic d'entrée (symétrique ou asymétrique), à condition que λ_i reste faible (i.e: $\rho_i = \frac{\lambda_i}{\mu_i} < 1$).

4.3.1.3 Le temps de réponse

Le délai moyen pour traverser le premier étage $E(W_1)$ peut être obtenu en appliquant les résultats du chapitre 3:

$$E(W_1) = E(W_q) + E(W_s) = \frac{T}{2} + \frac{\lambda_i T^2}{2(1-\rho_i)} + 1$$

où $E(W_s) = 1$ est le délai moyen de transmission (ou le temps de service).

Le délai moyen pour qu'un message traverse le deuxième étage $E(W_2)$ est donné, dans le cas de la capacité limitée, par la formule (4.22)

Afin de donner une vue tangible, on montre que le délai moyen de transfert (ou encore le temps moyen de réponse) en fonction de la charge normalisée sous forme de courbes dans Fig. 4.7 et Fig.4.8.

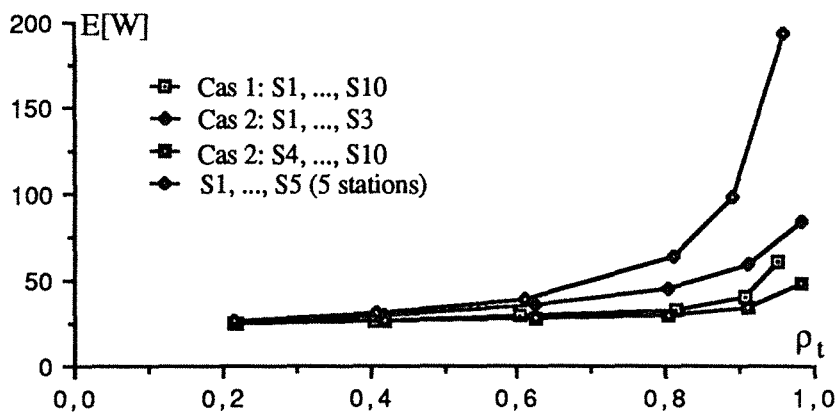


Fig. 4.7 Temps moyen de réponse vs. charge (5 et 10 stations)

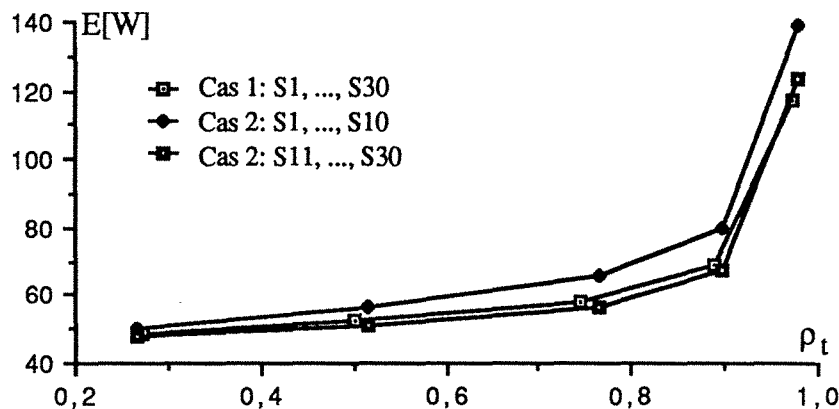


Fig. 4.8 Temps moyen de réponse vs. charge (30 stations)

Ces courbes montrent clairement que :

- le temps moyen de réponse d'un message ne s'accroît pas rapidement tant que la charge normalisée reste inférieure à 80% voire 90%.
- la figure 4.7 montre que le délai pour $N=5$ est supérieur à celui pour $N=10$. Ceci s'explique par le fait que pour $N=5$, la capacité d'évacuation du premier étage $\sum \mu_i$ est égale à la capacité d'évacuation du deuxième étage μ . Un message doit subir deux délais d'attente longs lorsque la charge normalisée est proche de 1. Il est donc préférable de concevoir le système tel que la capacité d'évacuation du premier étage soit largement supérieure à la capacité d'évacuation du deuxième étage.

On note que le délai montré est en effet celui de messages issus de la station $S_{N/2}$. Le vrai délai d'un message issu d'une station S_i doit être borné, i.e. :

$$E[W] - T_{cyc}/2 \leq E[W_i] \leq E[W] + T_{cyc}/2 .$$

4.3.2 Services de demande explicite aperiodique

On distingue trois services de demande explicite aperiodique : spécifiée, libre urgente et libre normale (rqsa, rqu, rqn).

Le but principal est de montrer les performances de ces services. On les étudie par la simulation du modèle général (approximatif pour ces services).

Les paramètres qui nous intéressent sont principalement :

- délai de chacun des trois services,
- probabilité d'écrasement du service de demande explicite spécifiée aperiodique.

4.3.2.1 Description

Le temps moyen de réponse d'une demande explicite apériodique $E(W)$ est la somme de deux délais correspondants à la traversée des deux étages du modèle. Il dépend du trafic généré par ces trois services.

Comme le service de messagerie apériodique, ces services de demande explicite apériodique se déroulent aussi dans une fenêtre qui leur est propre appelée "explicite apériodique ($Trqa$)". En pratique, on a: $Trqa + Tmsg = Tbc - Tcyc$, et on donne la taille maximale pour les deux fenêtres apériodiques, mais souvent avec $Trqa_{max} + Tmsg_{max} > Tbc - Tcyc$. Si on place la fenêtre $Tmsg$ avant la fenêtre $Trqa$, lorsque la fenêtre $Tmsg$ n'atteint pas sa dimension maximale, on peut profiter du temps laissé par $Tmsg$ pour effectuer d'avantage des transactions de demandes explicite apériodique.

Dans notre calcul, on suppose que la dimension de la fenêtre explicite apériodique $Trqa$ est constante, i.e: il n'y a pas d'influence entre $Tmsg$ et $Trqa$. De plus, puisqu'on ne connaît pas dans la pratique laquelle des deux fenêtres est placée devant, on a décidé de ne pas profiter du temps mort laissé par l'une des fenêtres apériodiques.

D'ailleurs, une transaction de demande explicite diffère de celle d'un message dans sa deuxième phase de déroulement qui correspond à la traversé du deuxième étage. En effet, une demande explicite déclenchera, en arrivant au deuxième étage, une circulation de plusieurs ($\leq 64+1$) variables identifiées. De plus, deux priorités différentes existent dans le service de demande explicite libre: urgente et normale. L'existence de deux priorités rend l'étude plus délicate. Et encore, toutes les demandes de même priorité quitteront la file d'attente du premier étage dès que cette file est scrutée. On note que la règle concernant les priorités est non-préemptive. Cette règle est adoptée tant au premier étage qu'au deuxième étage.

Différentes configurations manifestent des performances différentes. On se propose d'examiner la même configuration qu'au paragraphe précédent (cf. §4.3.1.1).

La configuration est examinée en posant $N=10$ et 30 . Dans les deux cas, on a considéré un modèle symétrique du trafic.

10 stations avec trafic symétrique:

- le nombre de stations $N=10$;
- le flux d'arrivée de demandes est Poissonnien d'intensité λ ;

- la période de scrutation pour chaque station est $T=20$, la capacité d'évacuation d'une station est donc: $\mu_i = \frac{1}{T}$;
- la durée de transaction d'une variable identifiée est $h_c = 1$;
- la durée de transaction d'une demande explicite est $h_d = 1$;
- la capacité d'évacuation du deuxième étage est $b = \frac{Trqa}{h_d \text{ (ou } h_c)} = 5$;
- le nombre de variables identifiées dans une demande est constant: n , donc une demande explicite déclenche un ensemble de transactions qui durent $(n+1)$ slots;
- $\lambda_i = \lambda/N$, i.e: toutes les stations ont la même intensité de flux d'arrivée. λ_i est subdivisé ensuite en trois parties égales pour alimenter trois "dispositifs" (i.e: B_REQ, F_REQ1 et F_REQ2, cf. §2) d'une même station.

30 stations avec trafic symétrique:

- le nombre de stations $N=30$;
- le flux d'arrivée de demande est Poissonnien d'intensité λ ;
- la période de scrutation pour chaque station est $T=50$, la capacité d'évacuation d'une station est donc: $\mu_i = \frac{1}{T}$;
- la durée de transaction d'une variable identifiée est $h_c = 1$;
- la durée de transaction d'une demande explicite est $h_d = 1$;
- la capacité d'évacuation du deuxième étage est $b = \frac{Trqa}{h_d \text{ (ou } h_c)} = 10$;
- le nombre de variables identifiées dans une demande est constant: n , donc une demande explicite déclenche un ensemble de transactions qui durent $n+1$ slots;
- $\lambda_i = \lambda/N$, i.e: toutes les stations ont la même intensité de flux d'arrivée. λ_i est subdivisé ensuite en trois parties égales pour alimenter trois "dispositifs" (i.e: B_REQ, F_REQ1 et F_REQ2) d'une même station.

Pour simplifier le calcul, des hypothèses supplémentaires sont faites:

- H1) Une demande spécifiée et une demande libre peuvent quitter la station du premier étage durant un même slot.
- H2) Les files d'attente F_REQ1 et F_REQ2 dans chaque station du premier étage sont de capacité infinie.
- H3) La file d'attente du deuxième étage est de capacité limitée à L .

Pour qu'il n'y ait pas de rejet important ($\Pi_{over} < 10^{-6}$) au deuxième étage, la charge totale normalisée doit satisfaire la condition suivante:

$$\rho_t = \sum_{i=1}^N \lambda_i T/b < 1$$

On note que ce n'est pas la charge effective (ou réelle) car la probabilité d'écrasement dans un B_REQ n'est pas négligeable et il existe donc une certaine perte de demandes explicites spécifiées.

Dans la suite, on s'intéresse d'abord à ce qui se passe au deuxième étage. Les résultats sont donnés sous forme de tableaux. Etant le critère de performances le plus important, le temps moyen de réponse de chaque service est représenté graphiquement.

4.3.2.2 Comportement du deuxième étage

On fait varier n, nombre des variables identifiées dans une demande, en vue d'observer l'influence du nombre de variables d'une demande sur les performances.

Pour N = 10, on pose n = 2, 4 et 6, i.e: le rapport $\frac{n+1}{b} = 0,6; 1,0$ et 1,4 respectivement.

On obtient les résultats de simulation sur modèle général dans les tableaux 4.6--4.8.

Tab. 4.6 10 stations avec n = 2

ρ_t	0,2029	0,3932	0,5939	0,7852	0,8918	0,9490
$\rho_t(\text{urg.})$	0,1312	0,2616	0,3966	0,5240	0,5944	0,6305
$\rho_t(\text{norm.})$	0,0717	0,1316	0,1973	0,2612	0,2974	0,3185
$E[Y_\infty]$ urg.	0,2529	0,5594	0,9342	1,420	1,787	1,992
$E[Y_\infty]$ norm.	0,1383	0,3055	0,6074	1,517	3,543	8,136
\bar{n} urg.	0,1796 $\pm 0,009$	0,4123 $\pm 0,02$	0,7141 $\pm 0,02$	1,132 $\pm 0,06$	1,465 $\pm 0,07$	1,650 $\pm 0,07$
\bar{n} norm.	0,0986 $\pm 0,006$	0,2320 $\pm 0,02$	0,4963 $\pm 0,03$	1,368 $\pm 0,18$	3,375 $\pm 0,53$	7,957 $\pm 1,68$
$E[W_2]$ urg.	16,44 $\pm 0,34$	18,91 $\pm 0,35$	21,61 $\pm 0,18$	25,93 $\pm 0,87$	29,59 $\pm 0,89$	31,40 $\pm 0,93$
$E[W_2]$ norm.	16,50 $\pm 0,37$	21,16 $\pm 1,21$	30,19 $\pm 1,36$	62,73 $\pm 7,36$	135,9 $\pm 20,0$	299,4 $\pm 59,4$
$\Pi_{\text{écrase}}$	0,0045	0,0116	0,0138	0,0201	0,0225	0,0240

Tab. 4.7 10 stations avec $n = 4$

ρ_t	0,2088	0,3934	0,5856	0,7848	0,8856	0,9364
$\rho_t(\text{urg.})$	0,1365	0,2610	0,3942	0,5242	0,5898	0,6228
$\rho_t(\text{norm.})$	0,0723	0,1324	0,1914	0,2606	0,2958	0,3136
$E[Y_\infty]$ urg.	0,1552	0,3387	0,6233	1,050	1,421	1,610
$E[Y_\infty]$ norm.	0,0821	0,1912	0,4022	1,062	2,965	5,730
\bar{n} urg.	0,1247 $\pm 0,01$	0,2804 $\pm 0,01$	0,5345 $\pm 0,04$	0,932 $\pm 0,05$	1,289 $\pm 0,12$	1,470 $\pm 0,10$
\bar{n} norm.	0,0660 $\pm 0,007$	0,1619 $\pm 0,01$	0,3587 $\pm 0,04$	1,004 $\pm 0,11$	2,898 $\pm 0,78$	5,659 $\pm 1,76$
$E[W_2]$ urg.	18,28 $\pm 0,76$	21,47 $\pm 0,47$	27,26 $\pm 0,98$	35,58 $\pm 1,47$	43,72 $\pm 2,80$	47,25 $\pm 2,37$
$E[W_2]$ norm.	18,26 $\pm 0,71$	24,43 $\pm 1,20$	37,15 $\pm 2,60$	76,81 $\pm 8,56$	194,2 $\pm 47,5$	361,0 $\pm 96,1$
$\Pi_{\text{écrase}}$	0,0034	0,0053	0,0096	0,0160	0,0153	0,0133

Tab. 4.8 10 stations avec $n = 6$

ρ_t	0,2106	0,3940	0,5919	0,7826	0,8750	0,9587
$\rho_t(\text{urg.})$	0,1364	0,2584	0,3951	0,5267	0,5790	0,6381
$\rho_t(\text{norm.})$	0,0752	0,1356	0,2068	0,2659	0,2960	0,3206
$E[Y_\infty]$ urg.	0,2079	0,4329	0,7378	1,157	1,354	1,700
$E[Y_\infty]$ norm.	0,1148	0,2591	0,5303	1,478	3,028	10,06
\bar{n} urg.	0,1729 $\pm 0,02$	0,3689 $\pm 0,01$	0,6422 $\pm 0,03$	1,034 $\pm 0,06$	1,218 $\pm 0,05$	1,552 $\pm 0,13$
\bar{n} norm.	0,0961 $\pm 0,016$	0,2252 $\pm 0,02$	0,4823 $\pm 0,05$	1,417 $\pm 0,25$	2,959 $\pm 0,64$	9,982 $\pm 4,87$
$E[W_2]$ urg.	35,51 $\pm 0,90$	39,96 $\pm 0,87$	45,51 $\pm 0,95$	55,04 $\pm 1,97$	58,91 $\pm 1,29$	68,11 $\pm 3,95$
$E[W_2]$ norm.	36,29 $\pm 1,65$	46,48 $\pm 1,97$	69,57 $\pm 5,43$	155,4 $\pm 27,6$	289,0 $\pm 70,1$	889,8 $\pm 406,8$
$\Pi_{\text{écrase}}$	0,0018	0,0062	0,0086	0,0096	0,0054	0,0104

Ces tableaux montrent clairement l'influence du nombre de variables n dans une demande explicite sur le délai moyen d'attente dans la file du deuxième étage. Surtout quand $n+1 > b$, le délai augmente brutalement par rapport aux cas où $n+1 \leq b$.

NB: Dans notre modèle approximatif, on considère que pour compléter la transaction d'une demande explicite ($RP_RQ + n*RP_DAT$), $n+1$ slots sont nécessaires.

En effet, lorsque la charge est très faible, par exemple une seule demande explicite dans le système, on a alors le délai moyen d'attente "minimal" $E[W_2]_{min}$:

Si $n+1 \leq b$,

$$8 = Tmsg + h_d + n*h_c \leq E[W_2]_{min} \leq Tcyc + Tmsg + h_d + n*h_c = 18$$

Dans le cas général, on pose $n+1 = qb + r$, où $q = \lfloor \frac{n+1}{b} \rfloor$ désigne le quotient et r le reste.

On obtient:

$$qTbc + (Tmsg+h_d+(r-1)h_c) \leq E[W_2]_{min} \leq qTbc + (Tcyc+Tmsg+h_d+(r-1)h_c).$$

On peut constater aussi que lorsque la charge normalisée ne dépasse pas 1, le taux d'utilisation de deux priorités différentes est presque une fonction linéaire de ρ_t car la probabilité d'écrasement est petite. Le délai d'attente des demandes normales se dégrade dès que la charge dépasse 78%.

Tab. 4.9 30 stations avec $n = 4$

ρ_t	0,2558	0,4810	0,7423	0,8938	0,9450
$\rho_t(urg.)$	0,1692	0,3195	0,4920	0,5973	0,6303
$\rho_t(norm.)$	0,0866	0,1615	0,2503	0,2965	0,3147
$E[Y_\infty] urg.$	0,3482	0,7014	1,265	1,876	2,034
$E[Y_\infty] norm.$	0,1828	0,4076	1,184	4,012	8,686
$\bar{n} urg.$	0,2215 $\pm 0,01$	0,4743 $\pm 0,024$	0,9288 $\pm 0,05$	1,477 $\pm 0,10$	1,615 $\pm 0,13$
$\bar{n} norm.$	0,1183 $\pm 0,008$	0,2914 $\pm 0,017$	1,005 $\pm 0,14$	3,799 $\pm 0,71$	8,460 $\pm 4,33$
$E[W_2] urg.$	32,72 $\pm 0,60$	37,13 $\pm 0,77$	47,20 $\pm 1,78$	61,81 $\pm 3,35$	64,09 $\pm 3,49$
$E[W_2] norm.$	34,20 $\pm 1,23$	45,13 $\pm 2,46$	101,2 $\pm 12,5$	322,3 $\pm 52,5$	674,5 ± 339
$\Pi_{écrase}$	0,0035	0,0048	0,0083	0,009	0,009

Tab. 4.10 30 stations avec $n = 9$

ρ_t	0,2648	0,4836	0,7265	0,8785	0,9655
$\rho_t(\text{urg.})$	0,1736	0,3184	0,4916	0,5880	0,6470
$\rho_t(\text{norm.})$	0,0912	0,1652	0,2349	0,2905	0,3285
$E[Y_\infty]$ urg.	0,2038	0,4422	0,8898	1,414	1,812
$E[Y_\infty]$ norm.	0,1072	0,2646	0,6808	3,195	9,815
\bar{n} urg.	0,1536 $\pm 0,013$	0,3512 $\pm 0,018$	0,7457 $\pm 0,07$	1,247 $\pm 0,12$	1,621 $\pm 0,18$
\bar{n} norm.	0,0802 $\pm 0,008$	0,2169 $\pm 0,029$	0,6117 $\pm 0,08$	3,108 $\pm 1,27$	9,721 $\pm 4,77$
$E[W_2]$ urg.	44,24 $\pm 2,01$	55,15 $\pm 0,86$	75,94 $\pm 3,95$	106,0 $\pm 6,54$	125,2 $\pm 9,88$
$E[W_2]$ norm.	43,91 $\pm 2,02$	65,63 $\pm 5,26$	129,3 $\pm 23,0$	522,6 $\pm 225,5$	1514 ± 726
$\Pi_{\text{écrase}}$	0,0040	0,0044	0,0048	0,005	0,005

Tab. 4.11 30 stations avec $n = 14$

ρ_t	0,2538	0,5145	0,7215	0,8820	0,9465
$\rho_t(\text{urg.})$	0,1695	0,3390	0,4875	0,5850	0,6330
$\rho_t(\text{norm.})$	0,0843	0,1755	0,2345	0,2970	0,3135
$E[Y_\infty]$ urg.	0,2482	0,5436	0,9790	1,355	1,678
$E[Y_\infty]$ norm.	0,1256	0,3636	0,8532	2,271	6,925
\bar{n} urg.	0,2060 $\pm 0,04$	0,4637 $\pm 0,019$	0,8646 $\pm 0,07$	1,224 $\pm 0,15$	1,535 $\pm 0,16$
\bar{n} norm.	0,1054 $\pm 0,02$	0,3219 $\pm 0,072$	0,7969 $\pm 0,12$	2,202 $\pm 0,73$	6,855 $\pm 2,53$
$E[W_2]$ urg.	91,13 $\pm 4,21$	102,6 $\pm 2,82$	133,1 $\pm 4,24$	157,6 $\pm 8,29$	182,1 $\pm 14,9$
$E[W_2]$ norm.	94,09 $\pm 1,98$	137,2 $\pm 14,5$	256,1 $\pm 31,2$	560,9 ± 196	1706 ± 655
$\Pi_{\text{écrase}}$	0,0000	0,0010	0,0012	0,0018	0,0031

Dans ces trois derniers tableaux, on peut constater que les résultats sont analogues à ceux correspondant à $N = 10$. Ceci signifie que nos constatations ne se limitent pas à un seul cas.

On note que le cas où $N=5$ n'est pas examiné, car pour les services de demande explicite, on n'aura pratiquement jamais la situation où la capacité d'évacuation du premier étage $\sum \mu_i \leq \mu$ qui est la capacité d'évacuation du deuxième étage. N'oublions pas qu'une demande explicite déclenche plusieurs transactions du transfert de variables identifiées, ce qui occupe largement le médium de transmission.

4.3.2.3 Délai de transfert (ou temps de réponse)

Le temps de réponse de chaque type de demande explicite est montré sur les figures 4.9 -- 4.14 en fonction de la charge et avec les différents n .

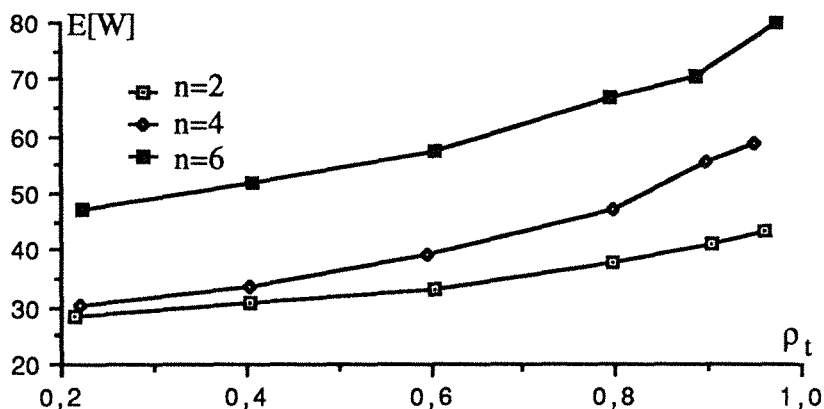


Fig. 4.9 Délai moyen de transfert du service r_{qsa} vs. charge (10 stations)

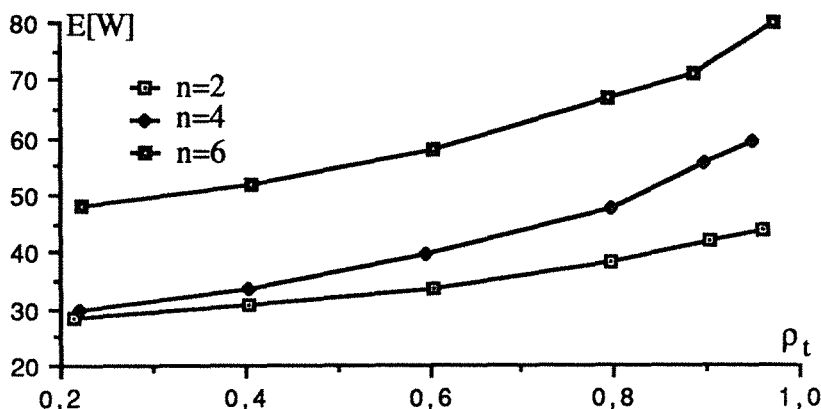


Fig. 4.10 Délai moyen de transfert du service r_{qu} vs. charge (10 stations)

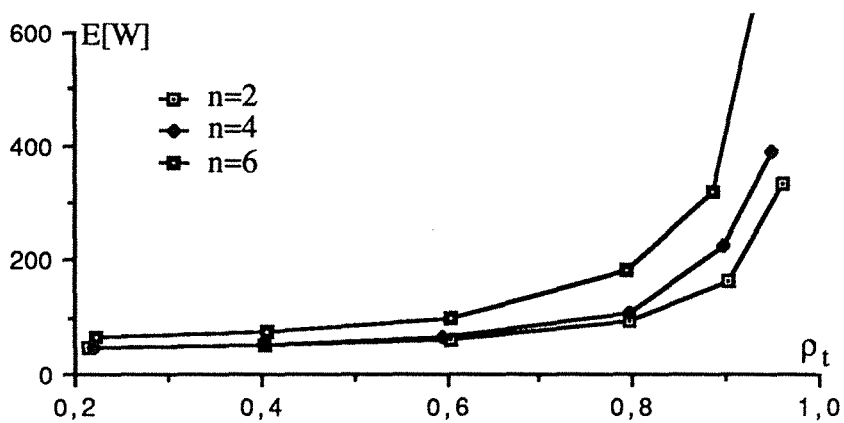


Fig. 4.11 Délai moyen de transfert du service rqn vs. charge (10 stations)

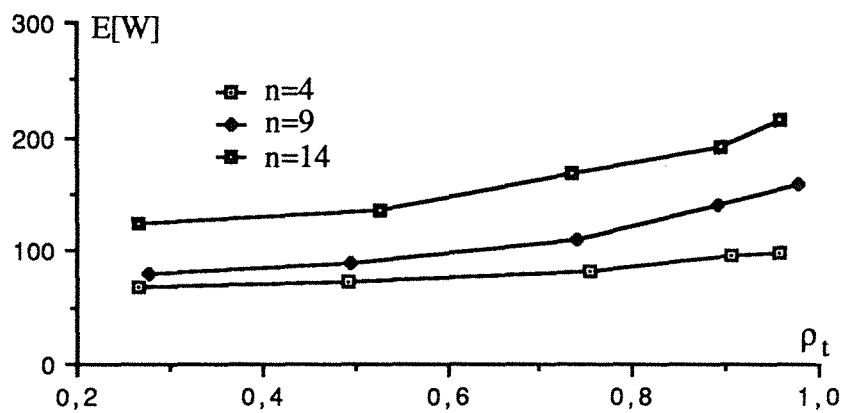


Fig. 4.12 Délai moyen de transfert du service rqa vs. charge (30 stations)

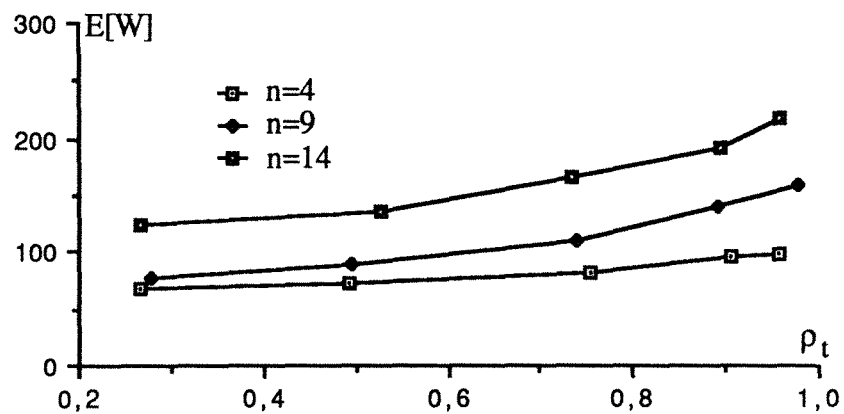


Fig. 4.13 Délai moyen de transfert du service rqu vs. charge (30 stations)

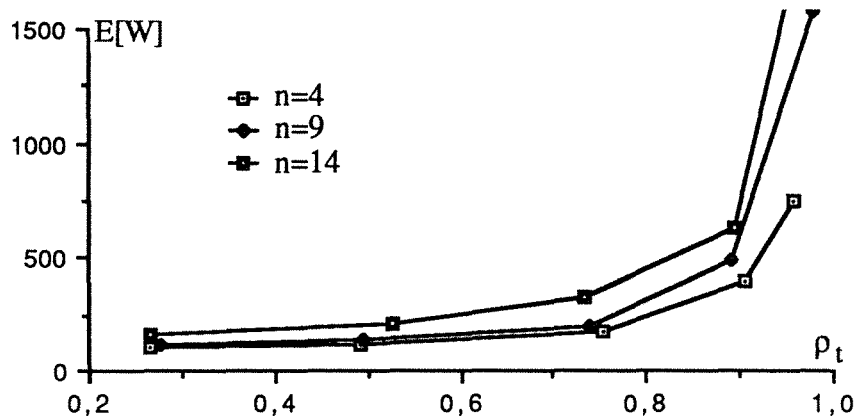


Fig. 4.14 Délai moyen de transfert du service rqn vs. charge (30 stations)

Ces courbes montrent que le service de demande spécifiée aperiodique et le service de demande explicite libre urgente (ces deux services ont la même priorité) fournissent quasiment la même qualité de service. Par contre, le temps moyen de réponse du service de demande explicite libre normale est beaucoup plus long, surtout lorsque la charge dépasse 80%.

4.4 Conclusion

On conclut ici le chapitre 3 et le chapitre 4.

Le modèle proposé dans le chapitre 2 est utilisé pour évaluer le nombre de clients (messages ou demandes explicites) dans chaque file d'attente, ainsi que les délais d'attente, les probabilités de rejet ou d'écrasement lorsque la capacité des files est limitée.

Outre ces résultats classiques, deux autres résultats concernant la stabilité (ou encore la robustesse) sont obtenus.

Résultat 1: Si les flux d'entrée dans le premier étage (qui sont des processus à accroissements indépendants et stationnaires) ne sont pas trop élevés, on peut considérer que le flux d'entrée de clients dans le deuxième étage (obtenu comme la superposition des flux de sorties du premier étage) est constitué d'une suite de variables aléatoires indépendantes et identiquement distribuées.

Résultat 2: Le résultat 1 peut être simplifié: on montre que sous les mêmes hypothèses qu'auparavant, le flux d'entrée du deuxième étage peut être approché de façon satisfaisante par une suite de variables aléatoires distribuées selon une loi de Poisson.

Conséquence: Muni de ces deux résultats, l'étude du modèle général du réseau FIP se réduit à l'étude d'un processus aléatoire (X_n) qui est au coeur de très nombreux problèmes de communication et de stockage et qui est défini par:

$$X_{n+1} = (X_n - B_n)^+ + A_{n+1}$$

avec (A_n) suite de variables aléatoires indépendantes et identiquement distribuées, et X_n représente le nombre de clients à des instants t_n convenablement choisis.

On dispose de peu de résultats lorsque B_n est aléatoire, car X_n n'est plus une chaîne de Markov homogène. Quand $B_n = \text{constante}$, la plupart des résultats concernant une file du type M/G/1 sont applicables pour des points d'observation t_n bien choisis.

Pour le premier étage du modèle, $B_n = 1$. L'étude de l'équation générique ci-dessus est détaillée. On a calculé la distribution limite des probabilités du nombre de clients dans la file et la distribution du délai d'attente. Des paramètres pratiques tels que le nombre moyen de clients et le délai moyen sont aussi déduits.

Pour le deuxième étage du modèle, $B_n > 1$. L'étude est effectuée en supposant $B_n = b$ (constant) et $B_n = b(n) = b(n + k \frac{T_m}{T_{bc}})$ (variable périodique).

Les résultats analytiques sont appliqués pour l'évaluation des performances de chacun des services aléatoires fournis par la couche liaison de données de FIP, ce qui nous permet d'une part, d'illustrer les performances de ces services, et d'autre part, de vérifier nos résultats analytiques.

Lorsque la capacité de la file d'attente est limitée, le calcul se simplifie beaucoup, car la matrice des probabilités de transition est finie.

Il convient de signaler que toutes les simulations ont été effectuées sur le modèle général établi à la fin du chapitre 2. Une simulation du vrai protocole FIP est donnée au chapitre 5 en utilisant "SimulFIP".

On ne dispose pas des résultats analytiques concernant les services de demande explicite aperiodique. Les résultats sont obtenus par simulation. Il serait intéressant d'effectuer une étude analytique approfondie.

CHAPITRE 5

PERFORMANCES DES SERVICES APERIODIQUES
OBTENUES PAR "SimulFIP"

5. Performances des services aperiodiques obtenues par "SimulFIP"

"SimulFIP" est un logiciel qui permet l'évaluation des performances des services fournis par la couche liaison de données d'un réseau FIP de configuration quelconque (cf. annexe 2). Dans ce chapitre, on simule la configuration déjà étudiée précédemment dans le chapitre 4. Afin de valider le modèle général, les résultats obtenus sont ensuite comparés aux résultats issus du modèle général qui est approximatif pour ces services aperiodiques.

5.1 Description de la configuration à simuler

Il est clair que chaque configuration particulière donnera des performances particulières.

On se propose d'étudier la configuration ci-dessous:

Configuration:

N stations homogènes sont raccordées à un bus FIP. Chaque station S_i supporte, outre le service de transfert cyclique, le service de messagerie aperiodique et les services de demande explicite (i.e: la classe de conformité 9). Ceux qu'elle ne supporte pas sont les services periodiques (i.e: messagerie periodique et demande explicite spécifiée periodique).

Chaque station S_i dispose alors des ressources suivantes:

- pour le service de transfert cyclique, un ensemble de buffers $B_DATprod$.
- pour le service de messagerie aperiodique, une file d'attente $F_MSGaper(i)$ de capacité limitée à $L(i)$, et un ensemble d'identifieurs configurés pour ce service (transporter des requêtes de messages) dont le nombre est $n_id(i)$.
- pour le service de demande explicite spécifiée aperiodique, un ensemble de buffers $B_REQ(i,j)$ (avec $j = 1, 2, \dots, N_i$ représentant le nombre de buffers dans la station S_i) de capacité limitée à une demande, soit 64 variables maximum et un nombre correspondant d'identifieurs associé à chaque buffer $n_idsp(i,j)$. On note que $n_idsp(i,j) = 1$, et on suppose que $N_i = 1$, c'est à dire que chaque station possède un seul buffer B_REQ , on peut alors ignorer l'indice j .
- pour le service de demande explicite libre (urgente ou normale), deux files d'attente $F_REQ1(i)$ et $F_REQ2(i)$ de capacité limitée à 64 variables, et un ensemble d'identifieurs configurés pour ce service dont le nombre est $n_idlib(i)$.

La station maître "arbitre de bus" dispose des ressources suivantes:

- trois fenêtres "cyc", "msg", "rqa" qui réservent le médium de transmission pour chacun des trois catégories de services.
- la table de scrutation pour le transfert cyclique.
- pour la scrutation dynamique des F_MSGaper, une file d'attente F_IDMSG dont la capacité est calculée de la façon suivante:

$$\text{Capacité de F_IDMSG} = \min \left\{ \begin{array}{l} \text{nombre total de places des F_MSGaper}(i), \\ \text{nombre total d'identifieurs configurés pour ce service} \end{array} \right\}.$$

- pour la scrutation dynamique des F_REQ1(i), F_REQ2(i) et des B_REQ(i,j), une file d'attente F_IDRQ dont la capacité est calculée de la façon suivante:

$$\text{Capacité de F_IDRQ} = \text{nombre total de } \{B_REQ(i,j) + F_REQ1(i) + F_REQ2(i)\}$$

et une file d'attente de réponse F_RPRQ de capacité limitée à 64.

Hypothèses:

- H1) La durée de transaction d'un transfert cyclique est h_c constante, la durée de transaction d'une demande et celle de variable est h_d constante, la durée de transaction d'un message est h_m constante.
- H2) $T_m = T_{bc}$ et les stations sont identiques. Elles sont scrutées ainsi qu'il est montré sur la figure 5.1.

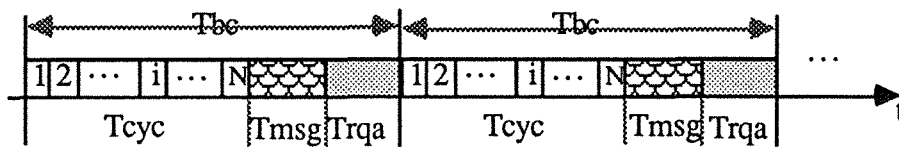


Fig.5.1 Répartition des fenêtres

Sur la figure 5.1, la fenêtre cyclique T_{cyc} est subdivisée en N slots identiques. Le slot i ($i = 1, 2, \dots, N$) est dédié à la station i .

h_c , h_m et h_d sont supposés égaux à la durée d'un slot dans notre simulation vis à vis du service considéré.

Donc,

durant T_{msg} , le système est capable d'évacuer $b = T_{msg}/h_m$ messages,

durant $Trqa$, le système est capable d'évacuer $b=Trqa/h_d$ variables sur demande explicite.

H3) Du point de vue d'un service aperiodique d'une station S_i , la fréquence que la station est scrutée est toujours $1/Tbc$ quel que soit le nombre d'identifieurs n_idxx . Si $n_idxx = 1$, cet identifieur est scruté une fois par Tbc ; si $n_idxx = k$, ces k identifieurs sont scrutés chacun une fois durant $kTbc$, et dans ce dernier cas, la période de scrutation de chacun des k identifieurs est $kTbc$ mais avec un décalage de Tbc entre eux.

Modèle de simulation:

Le modèle décrit exactement la configuration ci-dessus qui peut être schématisée sur la figure 5.2.

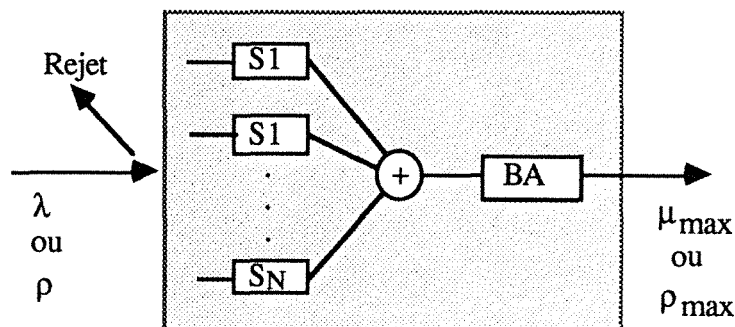


Fig.5.2 Modèle de simulation

Pour les deux catégories de services aperiodiques, en faisant entrer un flux d'arrivée d'intensité λ (ou $\rho = \lambda Tbc$) correspondant à chacune des deux catégories, on obtient les résultats sur le comportement du système. On note que ce système a une capacité maximale d'évacuation du flux d'informations (messages ou demandes explicites) μ_{\max} (ou $\rho_{\max} = \lambda Tbc$). Lorsque le flux d'entrée est supérieur à la capacité maximale d'évacuation (i.e: $\lambda > \mu_{\max}$), ou lorsque le système est mal paramétré, une partie de flux est rejetée.

Notations:

On résume ici les symboles utilisés. Il sont de deux types: *paramètres qui influencent les performances* et *paramètres qui mesurent les performances (résultats issus de simulation)*.

Paramètres globaux:

- N: Nombre de stations.
- Tbc: Durée d'un cycle élémentaire.
- Tcyc: Durée de la fenêtre de scrutation cyclique.
- Tmsg: Durée de la fenêtre de messagerie aperiodique.
- Trqa: Durée de la fenêtre de demande explicite aperiodique.

Paramètres concernant le service de messagerie aperiodique:

- L(i): Nombre de places dans la file d'attente F_MSGaper(i) de la station S_i.
- n_id(i): Nombre d'identifieurs configurés pour supporter le service de messagerie aperiodique dans une station prod/cons S_i.
- μ¹(i): Capacité d'évacuation de messages d'une station, on a supposé que μ¹(i) = 1/Tbc.
- μ²: Capacité d'évacuation de messages durant la fenêtre Tmsg, avec μ² = Tmsg/(h_mTbc).
- μ_{max}: Capacité d'évacuation de messages du système total, avec
$$\mu_{\max} = \min\left(\sum_{i=1}^N \mu^1(i), \mu^2\right)$$
 et ρ_{max} = μ_{max}Tbc qui représente le nombre maximal de messages que le système peut évacuer durant un Tbc.
- λ: Intensité du flux d'arrivée de messages, on a aussi ρ = λTbc.
- ρ_t: Charge totale normalisée de messages, avec ρ_t = ρ/ρ_{max} = λ/μ_{max}.

Paramètres des performances:

- E[W]: Temps moyen de réponse d'un message.
- ρ_{réel}: Charge réelle normalisée (ou "throughput")
- Π_{rejet}: Probabilité de rejet de messages.

Paramètres concernant les services de demande explicite aperiodique:

- n_liste: Nombre de variables dans la liste d'une demande explicite.
- n_idsp(i): Nombre d'identifieurs configurés pour supporter le service de demande explicite spécifiée aperiodique, avec n_idsp(i) = 1.
- n_idlib(i): Nombre d'identifieurs configurés pour supporter le service de demande explicite libre dans S_i, avec n_idlib ≥ 2 (cf. §5.3 pour l'explication).
- μ¹(i): Capacité d'évacuation de demandes explicites (rqs, rqu, rqn) d'une station S_i, avec μ¹(i) = 1/Tbc.
- μ²: Capacité d'évacuation de variables déclenchées par des demandes explicites durant la fenêtre Trqa, avec μ² = Trqa/(h_dTbc).
- μ_{max}: Capacité d'évacuation de variables déclenchées du système total,

avec $\mu_{\max} = \min(E[n_liste+1] \sum_{i=1}^N \mu^1(i), \mu^2)$ et $\rho_{\max} = \mu_{\max} Tbc$.	
- λ :	Intensité du flux d'arrivées de demandes explicites, on a aussi $\rho = \lambda Tbc$.
- ρ_t :	Charge totale normalisée de demandes explicites, avec $\rho_t = (\rho E[n_liste+1]) / \rho_{\max} = (\lambda E[n_liste+1]) / \mu_{\max}$.
Paramètres des performances:	
- $E[W]$:	Temps moyen de réponse d'une demande explicite (rqs, rqu, rqn).
- $\rho_{\text{réel}}$:	Charge réelle normalisée (ou "throughput") de chacun des trois services.
- Π_{rejet} :	Probabilité de rejet de demandes explicites libres (rqu, rqn).
- $\Pi_{\text{écrase}}$:	Probabilité d'écrasement de demandes explicites spécifiées aperiodiques.

Dans la suite, afin de pouvoir comparer les résultats de simulation avec les résultats analytiques obtenus précédemment, on suppose que les durées des deux fenêtres $Tmsg$ et $Trqa$ sont constantes, ce qui nous permet de simuler les deux catégories de services séparément.

5.2 Service de messagerie aperiodique

Dans ce paragraphe, on examine deux combinaisons suivantes:

- Cas 1: $Tbc = 20$, $Tcyc = 10$ et $Tmsg = Trqa = 5$.

La capacité d'évacuation du deuxième étage vaut donc: $b = \frac{Tmsg}{h_m} = 5$.

- Cas 2: $Tbc = 50$, $Tcyc = 30$ et $Tmsg = Trqa = 10$.

La capacité d'évacuation du deuxième étage vaut $b = \frac{Tmsg}{h_m} = 10$.

Le flux d'arrivée des messages est supposé de Poisson dans les deux cas.

Puisque les stations sont identiques, afin de simplifier l'écriture, on ignore la plupart du temps l'indice i , ce qui donne: $n_id(i) = n_id$, $L(i) = L$, ...etc.

5.2.1 Résultats du cas 1

Afin de mesurer l'influence de nombre de stations, on simule le service avec $N=5$ et $N=10$. Pour chaque N , on examine deux cas: $n_id = 1$ et $n_id=5$ afin de montrer son influence sur la précision du modèle approximatif. Le modèle du trafic est supposé

"symétrique", c'est à dire que $\lambda_i = \lambda/N$, toutes les stations ont la même intensité d flux d'arrivée. La capacité des files $F_MSGaper(i)$ est constante $L = 10$.

Lorsque le nombre de stations $N=5$

Dans ce cas, on note que le rapport entre la capacité d'évacuation du premier étage et celle du deuxième étage $\sum_{i=1}^N \mu^1(i)/\mu^2 = 1$.

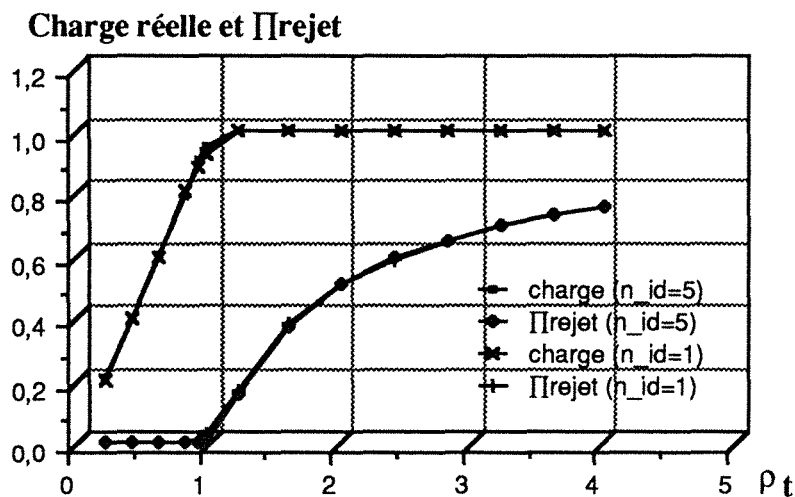


Fig. 5.3 Charge réelle et probabilité de rejet vs. charge ($N=5$)

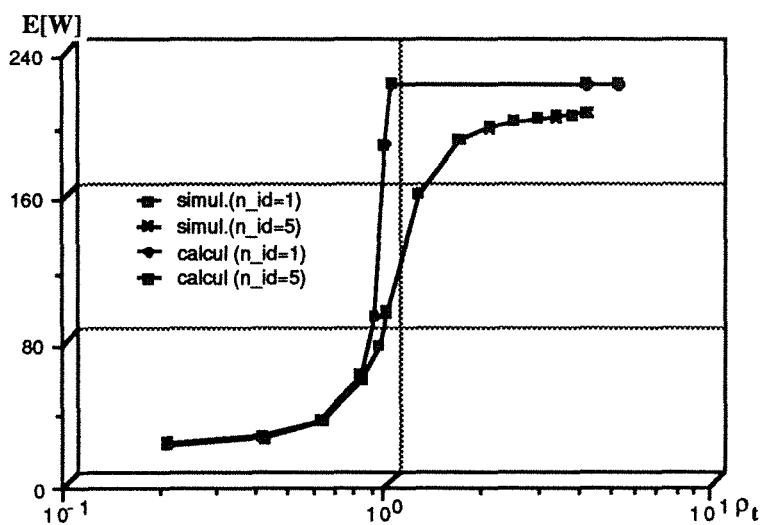


Fig. 5.4 Temps de réponse vs. charge ($N=5$)

La figure 5.3 montre qu'il y a une très légère différence entre $n_{id} = 1$ et $n_{id} = 5$ pour la charge réelle et la probabilité de rejet. La figure 5.4 nous dit que l'augmentation du nombre d'identifieurs n_{id} de 1 à 5 n'améliore pas la précision de notre approximation. En effet, dans ce cas particulier, les identifieurs des cinq stations du premier étage sont évacués par le deuxième étage durant le cycle élémentaire courant, ils ne restent donc jamais bloqués dans la file F_{IDMSG} .

Lorsque le nombre de stations $N=10$

Dans ce cas, le rapport entre la capacité d'évacuation du premier étage et celle du deuxième étage est égale à 2.

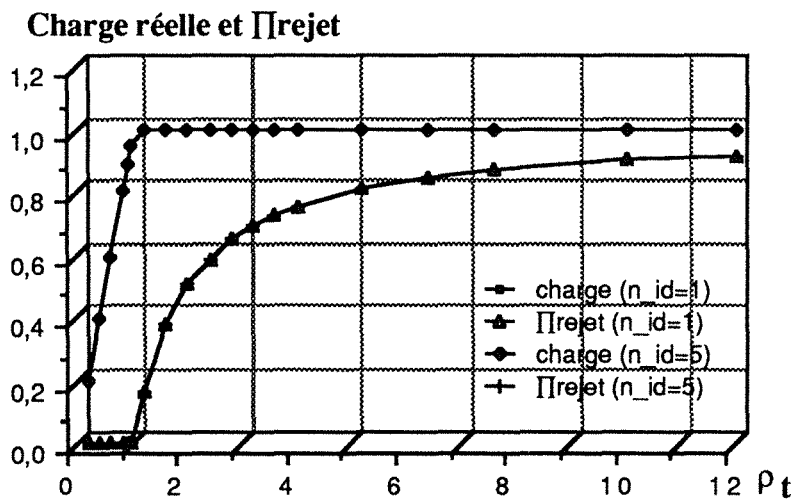


Fig. 5.5 Charge réelle et probabilité de rejet vs. charge (N=10)

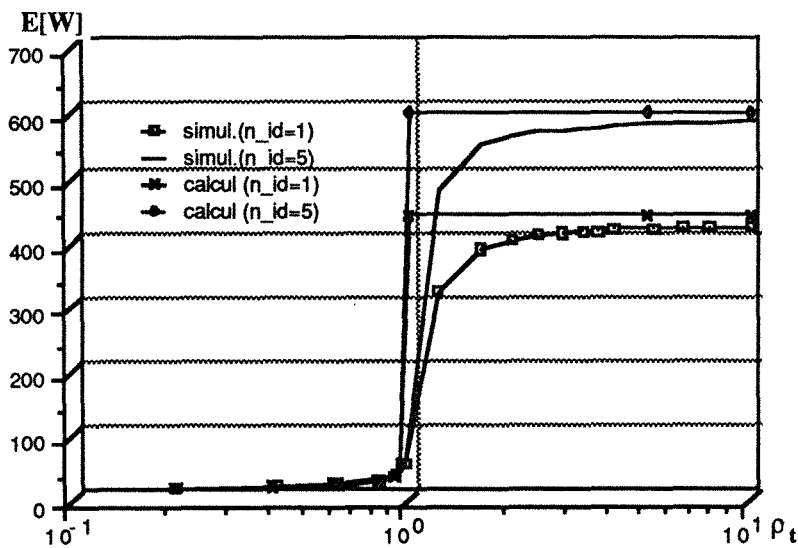


Fig. 5.6 Temps de réponse vs. charge (N=10)

Selon la figure 5.5, n_{id} n'a pas d'influence sur le charge réelle et la probabilité de rejet. Par contre, la figure 5.6 montre que l'augmentation de n_{id} augmente le délai maximal, ce qui n'est pas le cas pour $N=5$.

5.2.2 Résultats du cas 2

Le nombre de stations du premier étage vaut $N=30$. La capacité des files $F_{MSGaper(i)}$ est constante $L = 10$. Le rapport entre la capacité d'évacuation du premier étage et celle du deuxième étage est égale à 3.

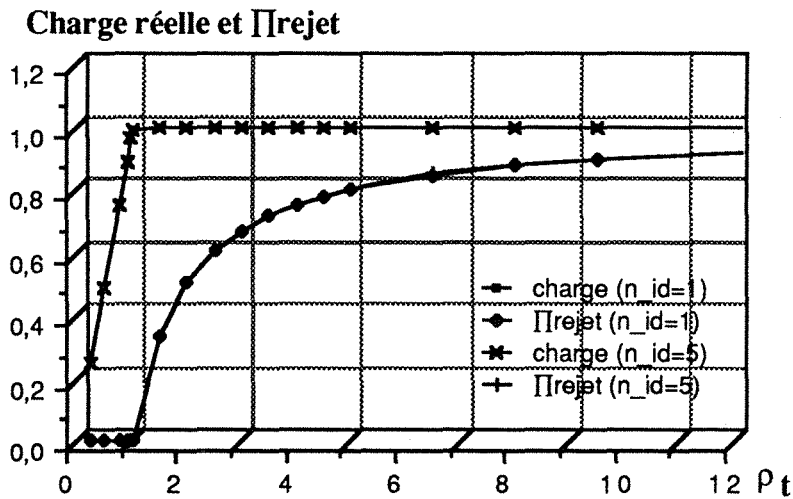


Fig. 5.7 Charge réelle et probabilité de rejet vs. charge (N=30)

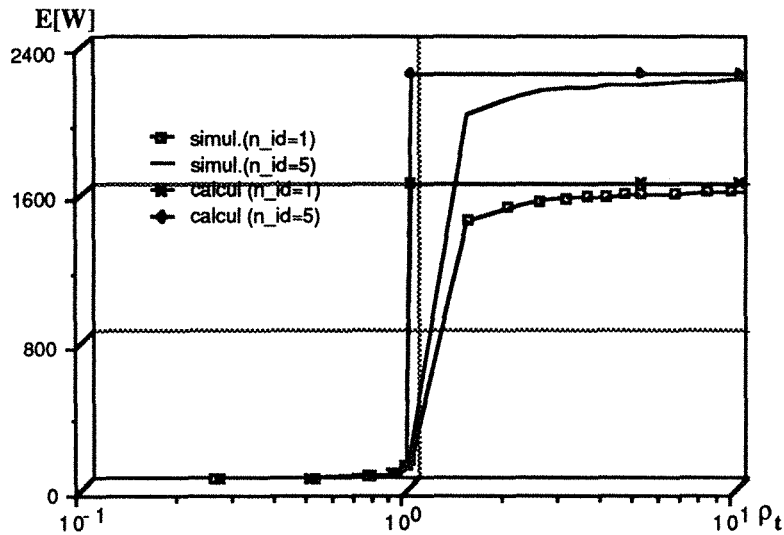


Fig. 5.8 Temps de réponse vs. charge (N=30)

On a examiné les cas où $n_{id} = 1$ et $n_{id}=5$ afin de montrer leur influence sur la précision du modèle approximatif. Le modèle du trafic est supposé "symétrique", i.e: $\lambda_i = \lambda/N$, toutes les stations ont la même intensité de flux d'arrivée.

En examinant Fig.5.7 et Fig.5.8, on peut constater que les résultats sont analogues au cas $N=10$.

On note qu'avec $n_{id} = 5$, le modèle général approche mieux la zone de transition entre $\rho < 1$ et $\rho > 1$, qu'avec $n_{id} = 1$.

5.2.3 Calcul du temps de réponse maximal d'un message

Lorsque le trafic est faible, i.e: pour pratiquement tout $\rho_t < 1$, on a constaté que le résultat analytique est en accord avec le résultat de simulation.

Lorsque le trafic est fort, i.e: pour $\rho_t \geq 1$, un délai limite existe. On évalue ce délai maximal de la manière suivante en distinguant deux cas.

- Le rapport entre la capacité d'évacuation du premier étage et celle du deuxième étage est supérieur à 1

Le nombre maximal de messages dans la file F_{IDMSG} est: $M = \min\left(\sum_{i=1}^N n_{id}(i), \sum_{i=1}^N L(i)\right)$.

Le délai maximal pour qu'un message traverse cette file est alors: $\frac{M}{\mu^2} = \frac{M \cdot Tbc}{b}$.

Le délai maximal pour qu'un message issu de la station S_i traverse la file $F_{MSGaper}(i)$ est: $\frac{L(i)}{n_{id}(i)} \frac{M \cdot Tbc}{b}$.

Le temps de réponse maximal $W_{max}(i)$ d'un message issu de la station S_i est donc:

$$W_{max}(i) = \left(\frac{L(i)}{n_{id}(i)} + 1\right) \frac{M}{\mu^2} = \left(\frac{L(i)}{n_{id}(i)} + 1\right) \frac{M \cdot Tbc}{b}$$

- Le rapport entre la capacité d'évacuation du premier étage et celle du deuxième étage est égal à 1

Le nombre maximal de messages dans la file F_{IDMSG} est: $M = Tbc \sum_{i=1}^N \mu^1(i)$.

Et le délai maximal dans cette file ne dépasse jamais Tbc , on a alors le temps de réponse maximal $W_{max}(i)$ d'un message issu de la station S_i :

$$W_{max}(i) = (L(i) + 1)Tbc.$$

On note que ce dernier cas apparaît rarement dans un système correctement conçu.

5.3 Services de demande explicite aperiodique

On suppose que la fenêtre de demande explicite aperiodique $Trqa$ est de longueur constante. Les deux combinaisons suivantes sont examinées:

- Cas 1: $Tbc = 20$, $Tcyc = 10$ et $Tmsg = Trqa = 5$.

La capacité d'évacuation du deuxième étage vaut $b = \frac{Trqa}{h_d} = \frac{Trqa}{h_c} = 5$.

- Cas 2: $Tbc = 50$, $Tcyc = 30$ et $Tmsg = Trqa = 10$.

La capacité d'évacuation du deuxième étage vaut $b = \frac{Trqa}{h_d} = \frac{Trqa}{h_c} = 10$.

Le flux d'arrivée des demandes explicites est supposé de Poisson dans les deux cas.

On note que le cas où $N=5$ n'est pas examiné car une demande explicite déclenche en général plusieurs transactions de variables identifiées, on peut donc charger le deuxième étage, et le cas où la capacité d'évacuation du premier étage est inférieure à celle du deuxième étage est donc un cas rare.

Les résultats sont présentés graphiquement. Pour chacun des trois services (rqs , rqu , rqn), ils sont de trois types: le Charge réelle $\rho_{réel}$, la probabilité de rejet Π_{rejet} ou d'écrasement $\Pi_{écrase}$ selon le service, et le temps moyen de réponse $E[W]$. On trace aussi $E[W]$ obtenu dans le chapitre 4 à l'issue du modèle général (cf. §4.3.2) afin de faciliter la comparaison.

Résultats du cas 1:

$N = 10$ avec trafic symétrique.

On pose d'abord $n_{idsp} = 1$ et $n_{idlib} = 1$ avec $n_{liste} = 2$. On obtient les résultats montrés sur les figures 5.9 -- 5.11.

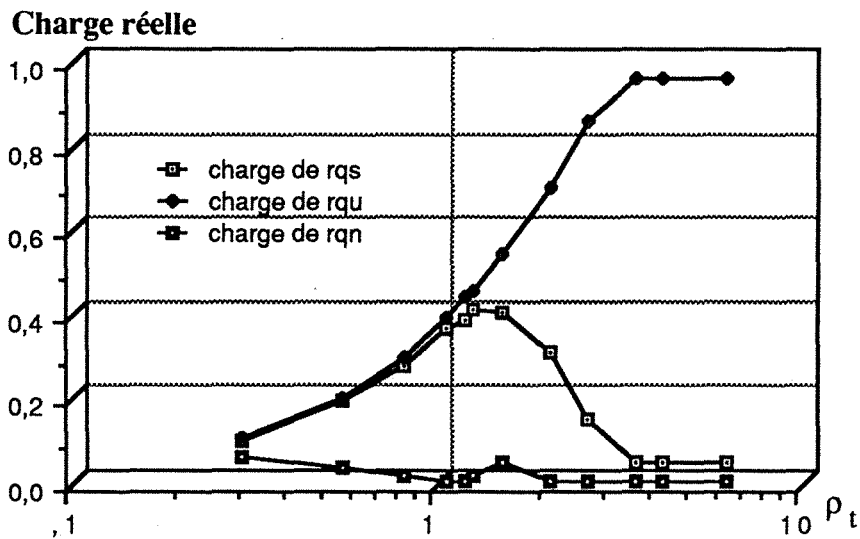


Fig. 5.9 Charge réelle des services de demande explicite aperiodique vs. charge ($N=10, n_{idlib}=1$ et $n_{liste}=2$)

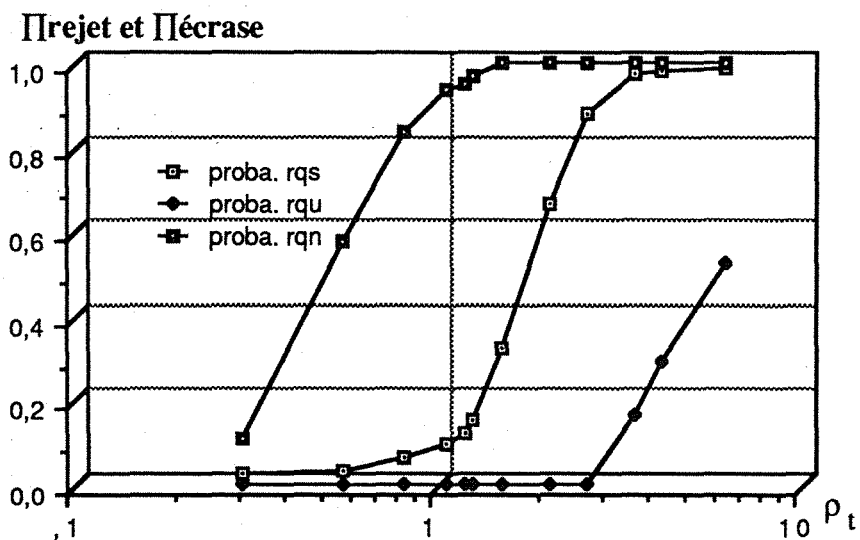


Fig. 5.10 Probabilité de rejet et d'écrasement des services de demande explicite aperiodique vs. charge ($N=10, n_{idlib}=1$ et $n_{liste}=2$)

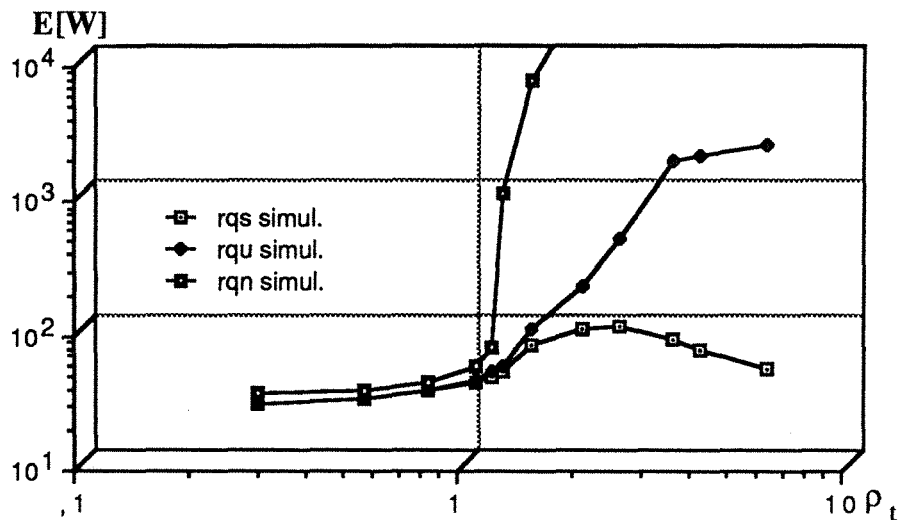


Fig. 5.11 Temps de réponse des services de demande explicite aperiodique vs. charge ($N=10$, $n_{idlib}=1$ et $n_{liste}=2$)

Constatations:

Un canal de transmission est partagé par trois types de trafics. Les performances de chacun des trois services sont illustrées clairement sur les figures. On note ici des constatations non évidentes.

- Le temps de réponse du rqu est borné.
- Quand la charge est forte, la proportion d'occupation entre rqs et rqu est $(n_{liste}+1)/(64+1)$.
- La probabilité de rejet du rqn est importante même quand la charge est faible. Cela est dû au fait qu'un seul identifieur ($n_{idlib}=1$) est partagé par deux files F_{REQ1} et F_{REQ2} , et la file F_{REQ2} a peu de chance de "capturer" cet identifieur. De plus, si $n_{idlib} = 1$, lorsque la charge devient forte, le service rqu risque de ne plus fonctionner si l'identifieur (qui est partagé par rqu et rqn) est "capturé" d'abord par rqn, et puisque la file F_{IDRQi} est fortement chargée, cet identifieur n'aura jamais l'occasion d'en sortir. Les files F_{REQ2} et F_{REQ1} restent toujours bloquées. Il est donc important de noter: **$n_{idlib} = 1$ est un cas à éviter.**

On pose maintenant $n_{idlib} = 2$. On fait varier le nombre de variables dans une demande explicite $n_{liste} = 2, 4, 6$.

On note que le nombre d'identifieurs configurés pour transporter les requêtes libres $n_{idlib} > 2$ n'est pas nécessaire. En fait, avec $n_{idlib} > 2$, on obtient presque les mêmes

performances que lorsque $n_idlib = 2$. En plus dans le cas où $n_idlib \geq 2$, il y a peu d'effet du à la priorité dans la station.

A l'issue de la simulation, on obtient les résultats montrés sur les figures 5.12 -- 5.20.

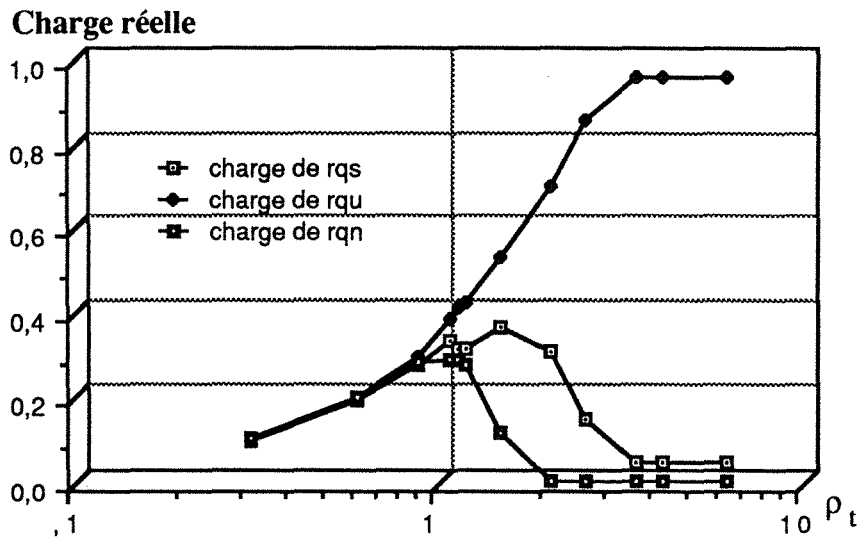


Fig. 5.12 Charge réelle des services de demande explicite aperiodique vs. charge ($N=10, n_idlib=2$ et $n_liste=2$)

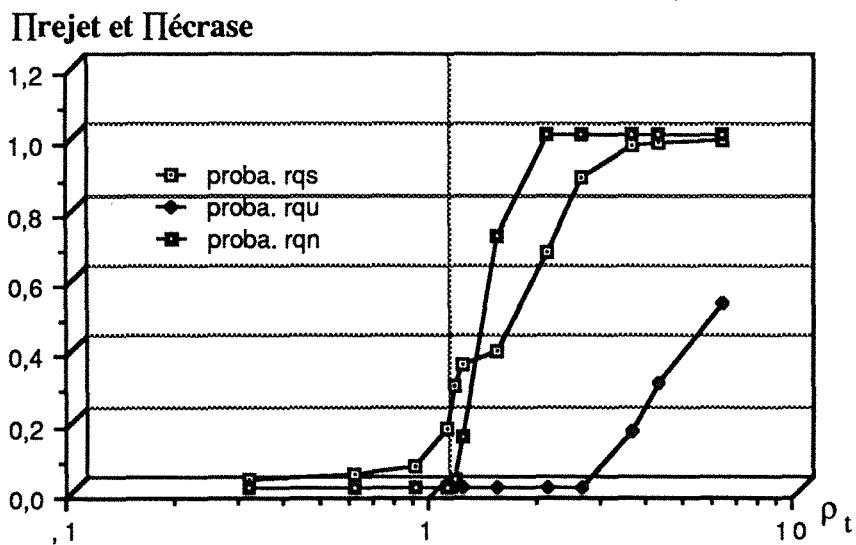


Fig. 5.13 Probabilité de rejet et d'écrasement des services de demande explicite aperiodique vs. charge ($N=10, n_idlib=2$ et $n_liste=2$)

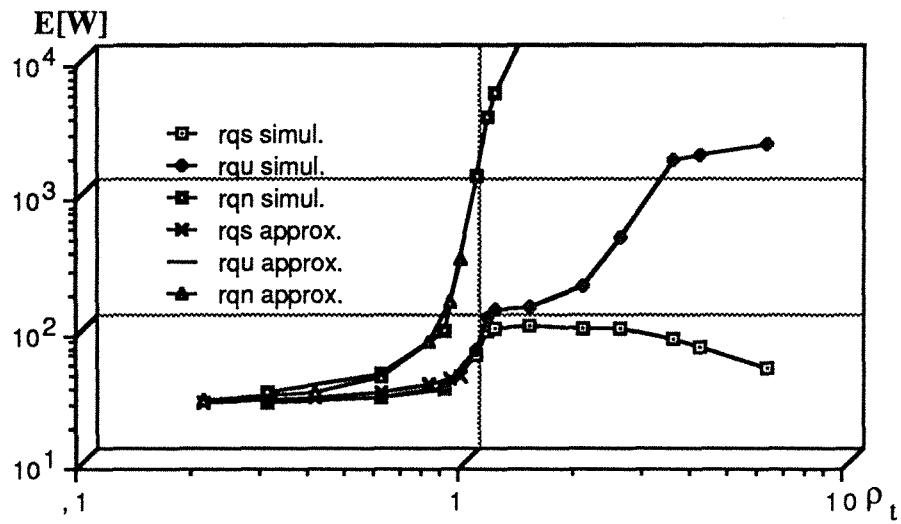


Fig. 5.14 Temps de réponse des services de demande explicite aperiodique vs. charge ($N=10$, $n_{idlib}=2$ et $n_{liste}=2$)

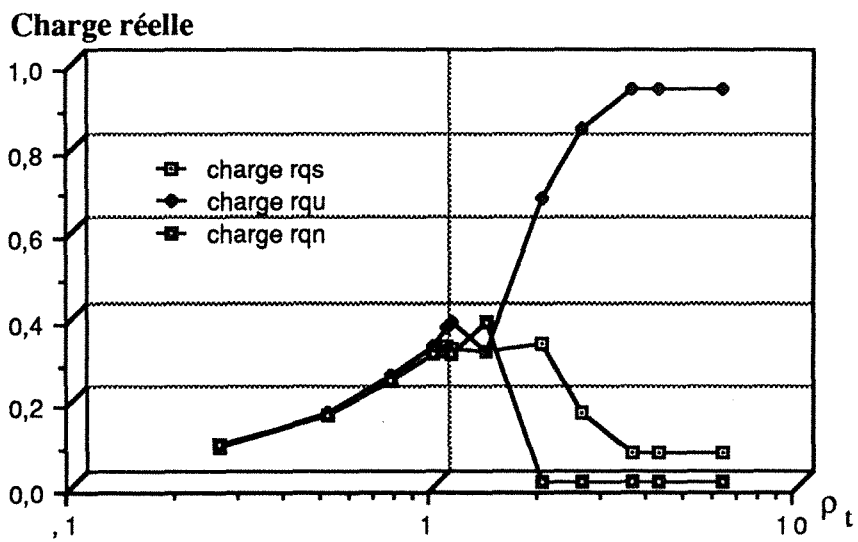


Fig. 5.15 Charge réelle des services de demande explicite aperiodique vs. charge ($N=10$, $n_{idlib}=2$ et $n_{liste}=4$)

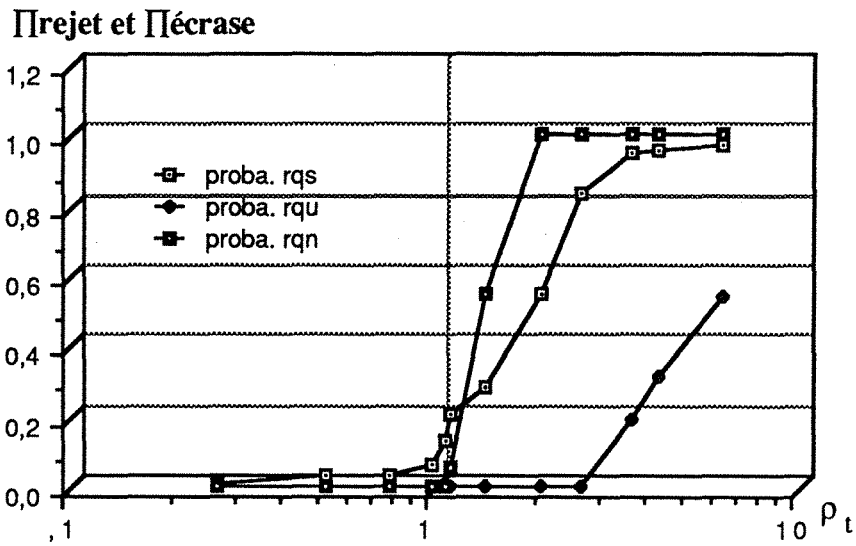


Fig. 5.16 Probabilité de rejet et d'écrasement des services de demande explicite aperiodique vs. charge ($N=10, n_{idlib}=2$ et $n_{liste}=4$)

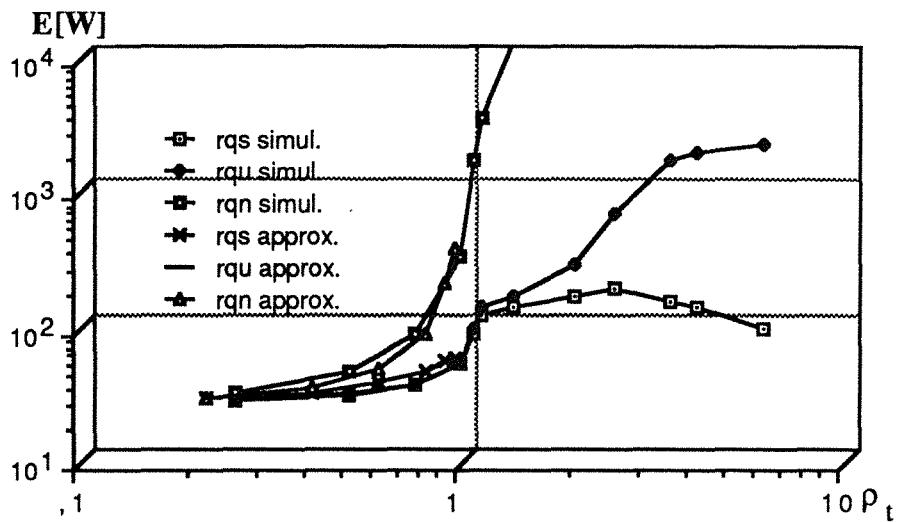


Fig. 5.17 Temps de réponse des services de demande explicite aperiodique vs. charge ($N=10, n_{idlib}=2$ et $n_{liste}=4$)

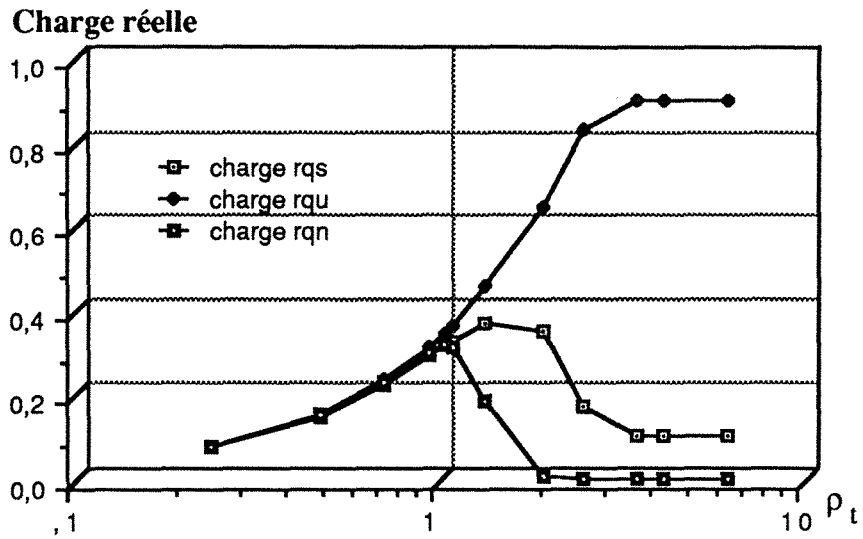


Fig. 5.18 Charge réelle des services de demande explicite aperiodique vs. charge ($N=10$, $n_{idlib}=2$ et $n_{liste}=6$)

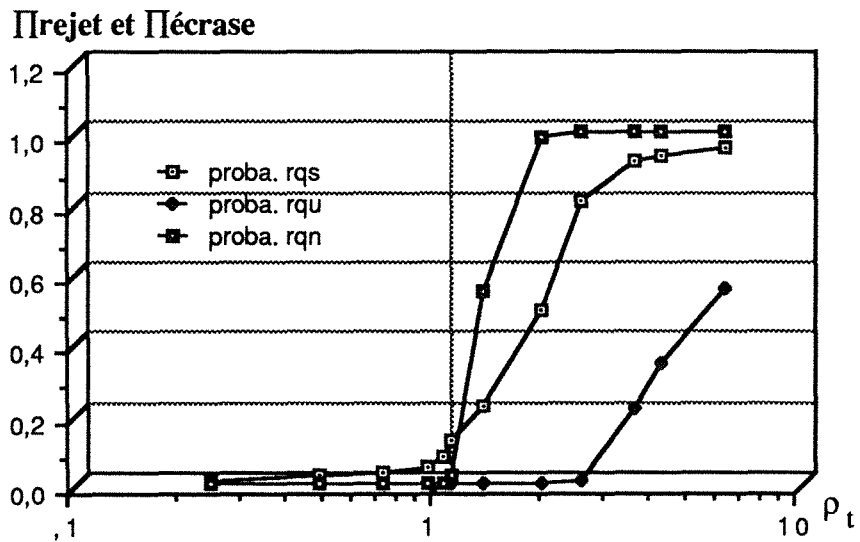


Fig. 5.19 Probabilité de rejet et d'écrasement des services de demande explicite aperiodique vs. charge ($N=10$, $n_{idlib}=2$ et $n_{liste}=6$)

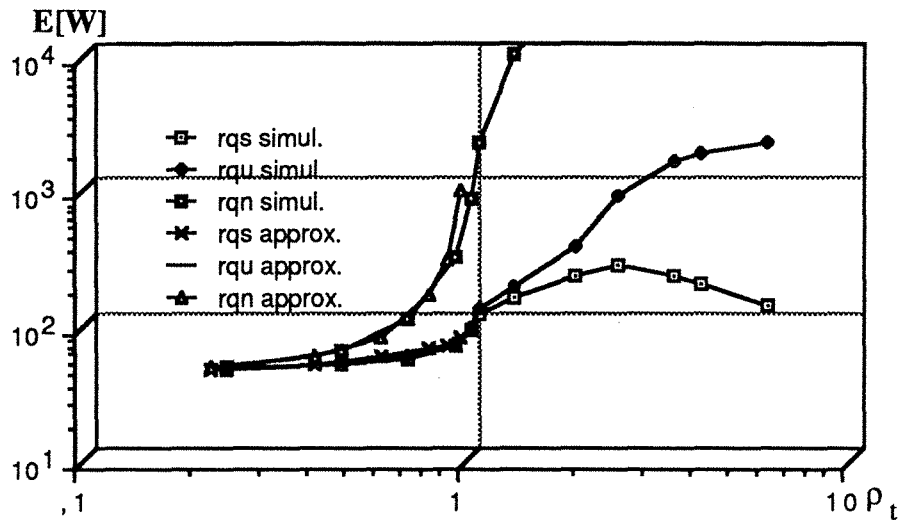


Fig. 5.20 Temps de réponse des services de demande explicite aperiodique vs. charge ($N=10, n_idlib=2$ et $n_liste=6$)

On constate tout de suite qu'avec $n_idlib = 2$, quand la charge est faible, la probabilité de rejet du rqn est beaucoup diminuée par rapport au cas où $n_idlib = 1$. Et par conséquent, le Charge réelle de ce service s'améliore. De plus, il n'y a plus de risque de "deadlock" lorsqu'un identifieur est "capturé" d'abord par F_REQ2.

Les résultats issus du modèle général (obtenus dans le chapitre 4) sont aussi illustrés. En observant les courbes $E[W]—\rho_t$, on s'aperçoit que les résultats approximatifs sont en accord avec les résultats de simulation, surtout lorsque $\rho_t < 0,8$.

Lorsque le réseau devient chargé.

- Le service de demande explicite libre de priorité normale rqn ne fonctionne plus.
- Pour le service de demande explicite spécifiée aperiodique rqs, le temps de réponse est borné. En effet:

- Une requête spécifiée n'attend pas plus longtemps que T_{bc} dans B_REQ .
- Un identifieur demandé par ce type de requête reste dans F_IDRQ au maximum $[N(n_liste+1) + N(64+1)] \frac{T_{bc}}{b}$.

- Pour le service de demande explicite libre urgente rqu, le temps de réponse est aussi borné. En négligeant le petit délai d'attente d'un identifieur, cette borne se calcule approximativement de la façon suivante: $[N(n_liste+1) + N(64+1)] \frac{T_{bc}}{b}$.

La simulation confirme cette borne.

Les résultats du cas 2 ($N = 30$ avec le trafic symétrique, et $n_{\text{liste}} = 4, 9, 14$) ne sont pas illustrés sous forme graphique compte tenu du nombre important de figures. On en tire tout de même des conclusions analogues au cas 1.

5.4 Conclusion

La simulation nous a permis d'évaluer les performances de ces deux types de services apériodiques.

Les résultats analytiques obtenus à travers des approximations et ceux de simulation sont montrés. Lorsque la charge totale est faible ($\rho_t < 1$), la différence entre la courbe analytique et celle de simulation n'est pas décelable; quand la charge totale est forte ($\rho_t > 1$), le calcul nous donne une borne supérieure pour le temps d'attente.

Le modèle général qui décrit de façon approximative ces deux services apériodiques est ainsi vérifié par la simulation.

Il convient de noter que l'étude analytique effectuée ne nous donne pas d'informations précises lorsque la charge totale est voisine de 1 ($\rho_t = 1$). Dans ce cas, l'utilisation des approximations de diffusion [Kleinrock 76, Louchard 91] reste un outil puissant.

D'ailleurs, la simulation sur "SimulFIP" nous permet d'observer de nombreux paramètres. Par exemple, on peut faire varier la capacité d'une file d'attente L , le modèle de trafic (symétrique ou asymétrique), la longueur des fenêtres T_{msg} et T_{rq} variables, ...etc.

CONCLUSION GENERALE ET PERSPECTIVES

Conclusion générale et perspectives

En conclusion de ce travail nous voudrions souligner l'apport original de notre contribution, les résultats obtenus et les perspectives.

Le temps réel ou le temps critique est source de préoccupations permanentes depuis longtemps. L'apparition des réseaux a conduit à une recrudescence de travaux sur ce thème, en particulier l'étude de systèmes de communication adaptés au temps critique.

FIP est une proposition sérieuse à la normalisation d'un tel réseau. Le besoin d'évaluation des performances était important pour valider les idées.

Pour mener à bien ce travail il a fallu:

- formaliser les problèmes concernant les performances,
- modéliser les trois types de services fournis par la couche liaison de données de FIP,
- trouver la solution du modèle établi en développant des techniques d'approximation originales,
- valider les résultats analytiques,
- valider le modèle établi.

Le travail a été effectué en deux phases parallèles; étude analytique et simulation.

- Etude analytique Après une analyse de tous les services offerts par la couche liaison de données de FIP, un modèle de files d'attente en deux étages parallèle-série représentant tous les services aléatoires a été établi. Ce modèle est exact pour certains services et approximatif pour d'autres. L'étude analytique de ce modèle est réalisée en distinguant deux cas: **faible charge** et **forte charge**, ce qui a conduit à des résultats numériques pour diverses configurations.

Dans le cas du trafic faible, pour qu'il n'y ait pas de goulot d'étranglement à la sortie du premier étage, la capacité d'évacuation du flux (trafic aléatoire) du premier étage est toujours surdimensionnée lors de la configuration. Pour que le modèle soit stable, le flux d'entrée dans le modèle doit être inférieur à la capacité d'évacuation du deuxième étage. Ce qui implique que les files d'attente du premier étage sont toujours faiblement chargées. On peut alors supprimer la dépendance du flux de sortie issu du premier étage par une approximation. L'étude du modèle est donc réduite à l'étude d'un système autonome (ou une file d'attente avec serveur intermittent) exprimé par l'équation: $X_{n+1} = (X_n - B_n)^+ + A_{n+1}$ dans laquelle nous pouvons choisir A_n et B_n selon les besoins.

Dans le cas du trafic fort, si l'on considère que les files d'attente sont de capacité limitée, le temps de réponse maximal est obtenu.

- Simulation Un logiciel "SimulFIP" est développé pour simuler une implémentation de FIP d'une configuration quelconque. Il est composé d'une part d'un simulateur de FIP décrivant complètement la couche liaison de données en langage de simulation QNAP2, et d'un configurateur du réseau (table de scrutation, tables de variables produites) qui est écrit en langage C d'autre part.

En appliquant les résultats issus de l'étude analytique aux services de la couche liaison de données, on obtient les résultats pratiques suivants:

- Temps de réponse d'une trame d'un service en fonction de la charge du réseau.
- Probabilité de rejet ou d'écrasement d'une demande de transmission (message ou demande explicite).
- Taux d'utilisation du médium par service.
- Dimension des files d'attente.

Le simulateur "SimulFIP" nous permet non seulement de paramétrer une implémentation concrète d'un réseau FIP, mais aussi de vérifier/valider notre étude analytique.

Les résultats analytiques sont comparés avec ceux de la simulation. Le résultat de la comparaison confirme que l'étude analytique nous fournit des connaissances assez précises du système.

Par ailleurs, les problèmes suivants ont été également résolus:

- La construction de la table de scrutation.
- Le dimensionnement des différentes fenêtres d'un cycle élémentaire.
- Le rendement du protocole FIP qui met en évidence l'efficacité de la structure des trames de FIP.
- La fiabilité et l'efficacité du service d'accès à une liste de variables (service de la couche application).

Les résultats précis peuvent être retrouvés dans le texte de ce mémoire selon le plan d'index suivant.

Tab. 6.1 Plan d'index des résultats de chacun des services

SERVICES: Transfert cyclique Messagerie Demande explicite

services critères	cyc	mp	rqp	msg	rqs	rqu	rqn
fenêtres de déroulement	Tcyc statique §2.2.1	Tmp §2.2.2	Trqp §2.2.2	Tmsg §4.3.1	Trqa §4.3.2		
rendement	§2.2.3	§2.2.3	§2.2.3	§2.2.3	§2.2.3		
délai	garanti	§3.3.1	§3.3.2	§4.3.1 §5.2	§4.3.2 §5.3		
taux d'utilisation	§2.2.4	§3.3.1	§3.3.2	§4.3.1 §5.2	§4.3.2 §5.3		
proba. de rejet ou d'écrasement	nulle	§3.3.1	§3.3.2	§5.2	§4.3.2 §5.3		

A l'issue de cette étude, on peut tirer les conclusions suivantes:

- FIP est un protocole conçu pour être dimensionné sur mesure. Tous les aspects temps critiques peuvent être satisfaits en utilisant FIP.
- De plus, les services de demande explicite et de messagerie fonctionnent bien tant que le trafic est faible. Lorsque le trafic devient fort, le temps de réponse maximal est tout de même borné pour la plupart de services. La connaissance des caractéristiques d'une application permet grâce à notre modélisation de dimensionner un système.
- Contrairement aux protocoles habituels, FIP réagit comme un système autonome. L'aspect de paramétrisation et configuration devient une phase primordiale pour le bon déroulement d'un réseau FIP. Le concepteur d'un réseau FIP a donc une tâche relativement lourde à assurer. Le travail réalisé dans cette thèse fournit non seulement un guide pour l'implémentation d'un réseau FIP, mais aussi un outil pratique pour configurer, paramétrer un réseau FIP.

Les perspectives se situent sur deux phases:

Comme Prolongements de ce travail, ils peuvent s'effectuer dans deux directions:

— D'un point de vue théorique:

- Lorsque la charge totale ρ_t est voisine de 1, l'utilisation des approximations de diffusion devrait permettre de préciser notre étude,
- L'analyse de la dépendance du *flux de sortie* du premier étage pourrait être raffinée en utilisant d'autres modèles de dépendance; ex: Markovien ou mélangeant.

— D'un point de vue pratique, il serait souhaitable d'intégrer la couche application, ce qui nécessite:

- Une étude statistique des flux d'entrée,
- Une optimisation de la table de scrutation et des fenêtres temporelles.

Comme Extensions au sens plus large, on peut considérer:

— Une étude comparative de FIP à d'autres protocoles (déjà amorcée dans l'annexe 3) serait intéressante.

— Une exploitation des méthodes et techniques utilisées afin de résoudre des problèmes voisins.

BIBLIOGRAPHIE

Bibliographie

- [Alijani 90] G.S. Alijani, R.L. Morrison,
"An evaluation of IEEE802 protocols and FDDI in real-time distributed systems",
Proc. of 15th conf. on local computer networks, pp334-342, Minneapolis, sept. 1990.
- [Allen 78] A.O. Allen,
Probability, statistics, and queueing theory with computer science applications,
Academic Press, 1978.
- [ANSI 86] ANSI,
FDDI token ring media access control (MAC),
Draft proposed America national standard, ASC X3T9.5, revision 10, Feb. 1986.
- [Asmussen 87] Asmussen,
Applied probability and queues,
John wiley, 1987.
- [Banks 84] J. Banks, J.S. Carson,
Discrete-event system simulation,
Prentice-Hall, 1984.
- [Bhat 84] U.N. Bhat,
Elements of applied stochastic processes (second edition),
John wiley, 1984.
- [Borovkov 76] A.A. Borovkov,
Stochastic processes in queueing theory,
Springer Verlag, 1976.
- [Borovkov 84] A.A. Borovkov,
Asymptotic methods in queueing theory,
J Wiley, 1984.
- [Bosch 87] Bosch, Klöckner-Moeller, Siemens,
Profibus: process field bus,
Draft standard 3.0, May 20, 1987.
- [Bux 84] W. Bux,
"Performance issues in local-area networks",
IBM systems journal, pp351-374, Vol. 23, N° 4, 1984.
- [Bux 89] W. Bux,
"Token-ring local-area networks and their performance",

Proc. of IEEE, Vol. 77, N°2 (february), pp238-256.

[Centre 91] Centre de compétence FIP,
FIP-Toolbox manuel de référence (version 2,0),
Club FIP, Avril 1991.

[Ciminiera 88-1] L. Ciminiera, C. Demartini, A. Valenzano,
"Performance analysis of type 3 LLC in industrial 802.5 networks",
IEEE Network, pp90-96, Vol.2, N°3, May 1988.

[Ciminiera 88-2] L. Ciminiera, A. Valenzano,
"Acknowledgement and priority mechanisms in IEEE 802.4 Token bus",
IEEE Trans. on Industrial Electronics, Vol.IE-35, N°2, May 1988.

[Costrell 73] L. Costrell,
"CAMAC instrumentation system - Introduction and general description",
IEEE Trans. Nuclear Science, Vol. NS-20, N°2, pp3-8, April 1973.

[Creusot 91] D. Creusot, J.P. Elloy, S. Kachkachi, J.P. Thomesse,
"Interconnexion de réseaux FIP-FIP et FIP-MAP",
Proceedings du colloque sur les actions de recherche en robotique & productique, pp59-79, Paris, Janv. 1991.

[Crow 82] R.M. Grow,
"A timed-token protocol for local area networks",
Proc. Electro'82, paper 17/3, May 1982.

[Decotignie 88] J.D. Decotignie, D. Desmons,
"L'intelligence dans la machine: misez tout sur le bus",
Commande numérique de machines, EPFL, pp21-28, Lausanne, 1988.

[Deheuvels 86] Deheuvels P. and Pfeifer D.,
"A semi-group approach to Poisson approximation",
Ann Prob. 14, pp663-676, 1986.

[Delcuvellerie 89] J.L. Delcuvellerie,
Ingénierie des systèmes d'automatisation de production,
Thèse doctorat de l'INPL, Mars 1989.

[Dykeman 88] Dykeman D., and Bux W.,
"Analysis and tuning of the FDDI media access control protocole",
IEEE Journal on selected areas in communications, pp997-1010, Vol. 6, N°6 , 1988.

[Eckberg 79] Adrian E. Eckberg, J R.,
"The single server queue with periodic arrival process and deterministic service times",
IEEE Trans. on Commun. vol. COM-27, No.3, March 1979.

-
- [EMUG 89] European Map Users Group (M. Elliott),
User requirements for communications in time critical applications,
Feb. 1989.
- [Fdida 86] S. Fdida, G. Pujolle, D. Mailles,
"Réseaux des files d'attente avec sémaphores",
Techniques et Sciences Informatiques (TSI), Vol.5, N°3, pp187-196, 1986.
- [Fdida 89] S. Fdida, G. Pujolle,
Modèles de systèmes et de réseaux, Tome 1: Performance,
Eyrolles, 1989.
- [Foxboro 89] Foxboro company,
Extended data link layer - A compromise proposal that provides both central and distributed access,
Draft, April 1989.
- [Galara 84] D. Galara, J.P. Thomesse,
Proposition d'un système de transmission série multiplexée pour les échanges d'informations entre des capteurs, des actionneurs et des automates réflexes,
Ministère de l'Industrie et de la Recherche, 1984.
- [Gelenbe 82] E. Gelenbe, G. Pujolle,
Introduction aux réseaux de file d'attente,
Eyrolles, 1982.
- [General 87] General Motors,
MAP 3.0 implementation release,
1987.
- [Gross 85] D. Gross and C.M. Harris,
Fundamentals of queueing theory (2nd edition),
John Wiley, 1985.
- [Hayes 84] J.F. Hayes,
Modelling and analysis of computer communications networks,
Plenum Press, 1984.
- [He 90] J.Y. He, Z. Mammeri, J.P. Thomesse,
"Clock synchronization in real-time distributed system based on FIP field bus",
2nd IEEE workshop on future trends in distributed computing systems, pp135-141,
Cairo Egypt, 1990.
- [Haverty 86] N. Haverty,
"MIL-STD 1553, A standard for data communications",

Communication & Broadcasting, Vol. 10, N°1, pp29-33, Jan. 1986.

[IEC 79] IEC/65A (secretariat) 18/WG6,

"Process data hiway (PROWAY) for distributed process control systems: part 2, functional requirements",

report No. 79/23908, March 1979.

[IEEE802.4 85] ANSI/IEEE std 802.4 - 1985 (ISO/DIS 8802/4),

Token-Passing bus access method and physical layer specifications.

[IEEE802.5 85] IEEE std 802.5 - 1985 (ISO/DP 8802/5),

Token Ring access method and physical layer specifications.

[Intel 84] Intel corp.,

The bitbus interconnect serial control bus specifications in distributed control modules,
1984.

[ISA 85] Instrument society of america,

"Proway LAN industrial data highway",

ISA-S72.01- 1985.

[ISO 7498] ISO 7498,

"Open System Interconnexion-reference model",

Norme référencée NF EN27498 en France.

[ISO 90] ISO/TC 184/SC5/WG2-TCCA,

"report of the meeting of the rapporters group on the time critical communications architecture",

Beijing-China, April 1990.

[Johnson 87] Johnson M.J.,

"Proof that timing requirements of the FDDI token ring protocol are satisfied",

IEEE Trans. on Comm, pp620-625, Vol. COM-35, N°6 (June), 1987.

[Kalashnikov 87] V.V.Kalashnikov,

"Estimations of convergence rate and stability for regenerative and renovative processes",
Colloquia Mathematica Societatis Janos Bolyai, Keszthely (HUNGARY), 1978.

[Kleinrock 75] L. Kleinrock,

Queueing systems, Vol. 1: Theory,

John Wiley, 1975.

[Kleinrock 76] L. Kleinrock,

Queueing systems, Vol. 2: Computer application,

John Wiley, 1976.

- [Lainé 91] T. Lainé,
Modélisation des application réparties pour la configuration automatique d'un bus de terrain,
Thèse doctorat de l'INPL, Mai 1991.
- [Law 82] A.M. Law, W.D. Kelton,
Simulation modelling and analysis,
McGraw-Hill, 1982.
- [LeCam 60] LeCam. L,
"An approximation theorem for the Poisson binomial distribution",
Pacific J. Math. 10, pp1181-1197, 1960.
- [Lepage 88] F. Lepage,
Les réseaux locaux industriels,
Hermès, 1988.
- [Little 61] J.D.C. Little,
"A proof for the queueing formula $L = \lambda W$ ",
oper. res. 9, pp383-387, 1961.
- [Louchard 91] G. Louchard, C. Kenyon, R. Schott,
"Data structures maxima",
F.C.T'91 (Fundamental of Computation Theory), Berlin, Lect. notes in computer science, 9-13 Sept. 1991.
- [Maglaris 88] B. Maglaris, D. Anastassion, P. Sen, G. Karlsson, J.D. Robbins,
"Performance models of statistical multiplexing in packet video communications",
IEEE trans. on comm. pp834-, Vol36, No.7, July 1988.
- [Mailles 87] D. Mailles,
Files d'attente descriptives pour la modélisation de la synchronisation dans les systèmes informatiques,
Thèse d'état de l'université de Paris 6, sept. 1987.
- [Mammeri 91] Z. Mammeri, J.P. Thomesse,
Réseaux Locaux,
TEKNEA édition communication, 1991.
- [Mark 57] C. Mark, T. Murphy, N.L. Webb,
"The efficiency of N machines uni-directionally patrolled by one operative when walking time and repair times are constants",
J. of the Royal statistical society, pp166-172, series B, Vol. 19, N°1, 1957.

- [Nussbaumer 87-1] H. Nussbaumer,
Téléinformatique I,
Presses polytechniques romandes 1987.
- [Nussbaumer 87-2] H. Nussbaumer,
Téléinformatique II,
Presses polytechniques romandes 1987.
- [Pang 89] Joseph W.M. Pang, F.A. Tobagi,
"Throughput analysis of a timer controlled token passing protocol under heavy load",
IEEE Trans. on Comm, pp694-702, Vol. 37, N° 7, July 1989.
- [Parzen 62] E. Parzen,
Stochastic processes,
Holden-Day, 1962.
- [Perros 89] H.G. Perros, T. Altiok (Editors),
Queueing networks with blocking,
North-Holland, 1989.
- [Pleinvaux 88] P. Pleinvaux, J.D. Decotignie,
"Time critical communication networks: Field buses",
IEEE network, vol.2 N°3, pp55-63, May 1988.
- [Pujolle 89] G. Pujolle, S. Fdida,
Modèles de systèmes et de réseaux, Tome 2: File d'attente,
Eyrolles, 1989.
- [Rolin 88] P. Rolin,
Réseaux locaux, normes et protocoles,
Hermès 1988.
- [Rosemount 88] Rosemount Inc.,
Proposed field bus solution to be demonstrated,
1988.
- [Sevcik 87] K.C. Sevcik, Johnson M.J.,
"Cycle time properties of the FDDI Token ring protocol",
IEEE Trans. Software Eng., pp376-385, Vol. SE-13, N°3 (March), 1987.
- [Simens 88] Simens,
Profibus proposal to ISA SP50 for field bus standard,
Feb. 1988.
- [Simonot 90-1] F. Simonot, Y.Q. Song, J.P. Thomesse,
"Approximate method for performance evaluation of message exchange in fieldbus FIP",

4th international conf. on data communication system and their performance, pp358-374, Barcelona, June 1990.

[Simonot 90-2] F. Simonot, Y.Q. Song, J.P. Thomesse,

"Queueing analysis of message exchange in fielbus FIP",

IMACS-IFAC international symposium on Mathematical and intelligent models in system simulation MIM-S²90, ppVI.B.1-1 -- 1-7, Brussels, September 1990.

[Simonot 90-3] F. Simonot, Y.Q. Song, J.P. Thomesse,

"Modelling and analysis of message exchange in fielbus FIP",

2nd IEEE workshop on future trends of distributed computing systems, pp142-148, Cairo, Egypt, October 1990.

[Simonot 90-4] F. Simonot, Y.Q. Song, J.P. Thomesse,

"Reliability of time critical multicast communication"

IECON'90, pp517-521, Pacific grove USA, Nov. 1990.

[Simonot 90-5] F. Simonot,

"Evaluation des distances de Wasserstein et Lévy-Prohorov entre deux processus de Lindley",

Rapport interne de l'université de Nancy 1, 1990.

[Simulog 91] Simulog,

QNAP2 (Version 7,0),

1991.

[Sloman 81] M.S. Sloman, S. Prince,

"Local network architecture for process control",

Local networks for computer communications, edited by A.West and P.Janson, pp407-427, North-Holland, 1981.

[Song 88] Y.Q. Song,

"Modélisation et analyse des performances d'un bus de terrain - FIP",

rapport de DEA de Université Paris 6, 1988.

[Song 89] Y.Q. Song, F. Simonot, J.P. Thomesse,

"Modélisation et simulation d'un bus de terrain: FIP",

proceedings of 17th IASTED international symposium Simulation and Modelling, pp5-9, June 1989.

[Stallings 88] W. Stallings,

Handbook of computer-communications standards: local network standard, Vol.2,

Howard W. sams & Company, 1988.

- [Strosnider 88] J.K. Strosnider, T. Marchok, J. Lehoczky,
"Advanced real-time scheduling using 802.5 token ring",
Proc. Real-time sys. symp. December 1988.
- [Summers 86] C.F. Summers, A.C. Weaver,
"Performance of IEEE 802.4 - the basis of MAP",
IECON'86, pp151-157.
- [Takagi 88] H. Takagi,
"Queueing analysis of polling models",
ACM Computing Surveys, pp5-28, Vol. 20, March, 1988.
- [Takagi 90] H. Takagi,
"Effets of Target Token Rotation Time on the performance of a Timed-Token protocol",
Performance '90, pp363-370, North-Holland, 1990.
- [Thomesse 78-1] J.P. Thomesse, J.C. Derniame,
"Flux de données et synchronisation",
IRIA/IGOD Workshop Aix en Provence, mai 1978.
- [Thomesse 78-2] J.P. Thomesse, A. Cochet-Muchy, P. Nonn,
"Data flow analysis for the description and management of synchronisation in real time applications",
IFAC/IFIP Workshop Real Time Programming, Mariehamn/Aland, Finlande, June 1978.
- [Thomesse 80] J.P. Thomesse,
SYGARE : une structuration pour la conception d'applications en temps réel et réparties,
Thèse d'Etat, INPL-ENSEM, 9 mai 1980.
- [Thomesse 87] J.P. Thomesse,
Les réseaux locaux industriels,
Collection Novotique, 1987.
- [Thomesse 90] J.P. Thomesse,
"Les réseaux locaux à temps critique",
Colloque le temps réel, pp29-40, Nantes, oct. 1990.
- [Tijms 86] H.C. Tijms,
Stochastic modelling and analysis: A computational approach,
John Wiley, 1986.
- [Tobagi 86] F.A. Tobagi,
"Multiaccess protocols and real-time communications in local area networks",
Advanced seminar on real-time local area networks, Bandol, France, pp1-22, April 1986.

-
- [Ulm 82] J.N. Ulm,
"A timed token ring local area network and its performance characteristics",
Proc. 7th Conf. on local computer networks, pp50-56, Feb. 1982.
- [UTE-601 90] UTE C46-601,
Architecture et présentation générale du bus de terrain FIP.
- [UTE-602 90] UTE C46-602,
Couche application - services périodiques et apériodiques.
- [UTE-603 90] UTE C46-603,
Couche liaison de données.
- [UTE-604 90] UTE C46-604,
Couche physique sur paire torsadée blindée.
- [UTE-605 90] UTE C46-605,
Gestion du bus de terrain FIP.
- [UTE-606 90] UTE C46-606,
Couche application - services de messagerie.
- [UTE-607 90] UTE C46-607,
Couche physique en bande de base sur fibre optique.
- [Valenzano 90] A. Valenzano,
"Time and throughput characteristics of timed token protocols",
Collana: GLI standard per l'elettronica l'informatica E l'automazione industriale, Vol.6,
1990.
- [Verlinde 89-1] C. Verlinde,
Contribution à l'étude des architectures de systèmes automatisés,
Thèse doctorat de l'INPL, Fev. 1989.
- [Verlinde 89-2] C. Verlinde, E. Georgel, J.P. Thomesse,
"A service oriented hierarchical model for the design of control system",
CCCT89, IFAC AFCET Symposium, sept. 1989.
- [Verlinde 89-3] C. Verlinde, E. Georgel, J.P. Thomesse,
"Hierarchical and functional architecture for control system",
IECON'89, Nov. 1989.
- [Wood 79] G. Wood,
"Review of common practice and accepted hardware standards in process control",
Real time data'79 conf., North Holland, pp203-208, 1979.
- [Zimmermann 80] H. Zimmermann, OSI reference model,
"The ISO model of architecture for open system interconnection",

IEEE Trans. on Comm., COM-28 N°4, pp425-432, April 1980.

[Zolotarev 83] V.M. Zolotarev,

"Probability metrics",

Theory Prob. Applications 28, pp278-302, 1983.

ANNEXES

- A1. DEFINITIONS
- A2. SIMULATEUR: "SIMULFIP"
- A3. PROTOCOLES D'ACCES AU MEDIUM
A TEMPS CRITIQUE

Annexe 1. Définitions

Afin de faciliter la lecture, nous rappelons les termes spécifiques utilisés dans ce mémoire.

⇒ **7 services sont offerts par la couche liaison de données de FIP et 5 fenêtres correspondantes:** Pour faciliter l'étude, on a redéfini les sept services qui sont déduits à partir de trois types de services définis dans NFC46-603 et 5 fenêtres correspondantes dans lesquelles se déroulent ces services. Ces sept services sont nommés:

- **cyc:** transfert cyclique des variables identifiées qui se déroule dans la fenêtre **Fcyc** de durée **Tcyc**. Ce service génère un **trafic déterminite**.
- **mp:** messagerie périodique qui se déroule dans la fenêtre **Fmp** de durée **Tmp**. Il génère un **trafic aléatoire**.
- **rqp:** demande explicite spécifiée périodique qui se déroule dans la fenêtre **Frqp** de durée **Trqp**. Il génère un **trafic aléatoire**.
- **msg:** messagerie apériodique qui se déroule dans la fenêtre **Fmsg** de durée **Tmsg**. Il génère un **trafic aléatoire**.
- **rqs:** demande explicite spécifiée apériodique qui se déroule dans la fenêtre **Frqa** de durée **Trqa**. Il génère un **trafic aléatoire**.
- **rqu:** demande explicite libre urgente apériodique qui se déroule dans la fenêtre **Frqa** de durée **Trqa**. Il génère un **trafic aléatoire**.
- **rqn:** demande explicite libre normale apériodique qui se déroule dans la fenêtre **Frqa** de durée **Trqa**. Il génère un **trafic aléatoire**.

⇒ **Arbitre de bus:** Entité d'un système FIP qui contrôle le droit d'accès au médium de chaque producteur d'information.

⇒ **Bus de terrain (field bus):** Réseau local de topologie bus pour échange d'informations entre transmetteurs, actionneurs et automates.

⇒ **Capacité d'évacuation μ_{\max} :** Nombre maximal de bits qui peuvent quitter le système par unité de temps (valeur maximale du débit).

- ⇒ **Charge totale normalisée ρ_t** : Elle est définie comme le rapport entre le flux d'arrivée et la capacité d'évacuation.
- ⇒ **Cycle élémentaire (T_{bc})**: Comprend la scrutation par l'arbitre de bus
- d'un ensemble d'identifieurs de variables, de requêtes et de messages cycliques de l'application; cet ensemble caractérise le cycle élémentaire,
 - de la fenêtre prévue pour les échanges apériodiques
 - de la fenêtre prévue pour la messagerie
 - de la fenêtre prévue pour la synchronisation.
- ⇒ **Délai de transfert (ou temps de réponse)**: Le "délai de transfert" (ou temps de réponse) qui exprime le temps d'attente, pour une station donnée, entre l'instant d'arrivée d'une trame dans la station (souvent mis en file d'attente) et la réception de celle-ci par la station destinataire. Ce critère est souvent exprimé en terme de moyenne ou de maximum.
- ⇒ **Demande explicite**: Un des trois types de services permettant de déclencher un ou plusieurs transferts sur demande de l'utilisateur. Quatre modes différents existent: spécifiée périodique **rqp**, spécifiée apériodique **rqs**, libre urgente **rqu**, libre normale **rqn**.
- ⇒ **FIP**: Factory Instrumentation Protocol or Flux Information Process est un bus de terrain. Le protocole FIP est une norme française.
- ⇒ **Flux d'arrivée λ (ou $\rho = \lambda T$)**: Nombre de bits entrant dans le système par unité de temps.
- ⇒ **Identifieur**: C'est un mot de 16 bits associé à une variable du système et qui caractérise de façon unique la variable dans un système FIP.
- ⇒ **Interrupteur**: Une fonction (ou un dispositif) adoptée par ce mémoire et utilisée à la description du modèle général afin de caractériser les moments de scrutation des variables d'une station prod/cons.

⇒ **Macro cycle (T_m):** Correspond à la juxtaposition de cycles élémentaires jusqu'à obtenir la scrutation de l'ensemble des identifiants cycliques.

⇒ **Messagerie:** Un des trois types de services permettant un échange de messages sans acquittement ou avec acquittement. Deux modes existent: périodique **mp** et apériodique **msg**.

⇒ **Modèle général:** Le modèle de files d'attente à deux étages en "parallèle-série" établi dans §2 et étudié dans §3 et §4. Il caractérise les six services aléatoires de manière exacte ou approximative.

- Une station du **premier étage** est un ensemble de files d'attente et buffers, elle représente une station prod/cons de FIP.

- **Deuxième étage** représente l'arbitre de bus et le support de transmission.

⇒ **Rendement:** Le "rendement" du protocole s'exprime par le rapport entre la quantité d'information utile, c.à.d sans tenir compte des trames de service (gestion ou "overhead"), sur la quantité totale d'information véhiculée par le médium. Ce critère est généralement utilisé pour estimer l'efficacité de la structure des trames d'un protocole d'accès.

⇒ **Services apériodiques:** Une autre façon de classer les services. Ils désignent les services qui se déroulent dans la partie dynamique de la table de scrutation tels que messagerie apériodique **msg** et demande explicite (spécifiée et libre) apériodique **rqs**, **rqu**, **rqn**.

⇒ **Services périodiques:** Une autre façon de classer les services. Ils désignent les services qui se déroulent dans la partie statique de la table de scrutation tels que messagerie périodique **mp** et demande explicite spécifiée périodique **rqp**.

⇒ **Station prod/cons:** Station esclave correspondant à un point de raccordement de FIP et qui regroupe une ou plusieurs variables produites et/ou consommées (< 100). Elle possède également des ressources (files d'attente, buffers) adaptées aux services qu'elle supporte.

⇒ **Taux d'utilisation:** Le "taux d'utilisation du médium" (ou "throughput") qui s'exprime par le rapport entre la quantité d'information véhiculée par le médium (par seconde) sur le débit du réseau. Ce critère est essentiellement utilisé pour évaluer globalement les performances. Un taux d'utilisation faible implique soit que le médium de transmission est peu sollicité par les stations, soit que la technique d'accès utilise "mal" la capacité du médium.

⇒ **Temps critique:** Le terme temps critique ("time-critical") est utilisé pour représenter la présence d'une fenêtre temporelle ("time-window"), durant laquelle une ou plusieurs actions spécifiées requises doivent être complétées avec un certain niveau de sûreté. Si ces actions spécifiées ne sont pas réalisées durant la fenêtre temporelle, cela risque de produire des défauts dans les applications ayant demandé ces actions, pouvant provoquer des perturbations dans les équipements ou l'usine.

⇒ **Trafic faible:** Lorsque la charge totale normalisée $\rho_t < 1$.

⇒ **Trafic fort:** Lorsque la charge totale normalisée $\rho_t > 1$.

⇒ **Transfert de variables identifiées:** Un des trois types de services qui désigne le transfert des valeurs des variables identifiées. Deux modes existent: transfert cyclique et transfert sur demande.

⇒ **Variable identifiée:** C'est une entité du système FIP pour laquelle on a défini un identifieur associé.

Annexe 2. Simulateur "SimulFIP"

Le simulateur "SimulFIP" est un logiciel qui permet d'évaluer les performances de la couche liaison de données d'un réseau FIP de configuration quelconque. Ce logiciel est développé sur une station de travail SUN/unix. Dans la suite, nous donnons une description de ce logiciel, suivie par une discussion des possibilités d'extension.

A2.1 Description du simulateur "SimulFIP"

1) Structure du logiciel

Le logiciel "SimulFIP" est développé en utilisant le langage de simulation à événements discrets: QNAP2, avec interface utilisateur en C.

Voici le diagramme qui montre les différents modules de ce logiciel:

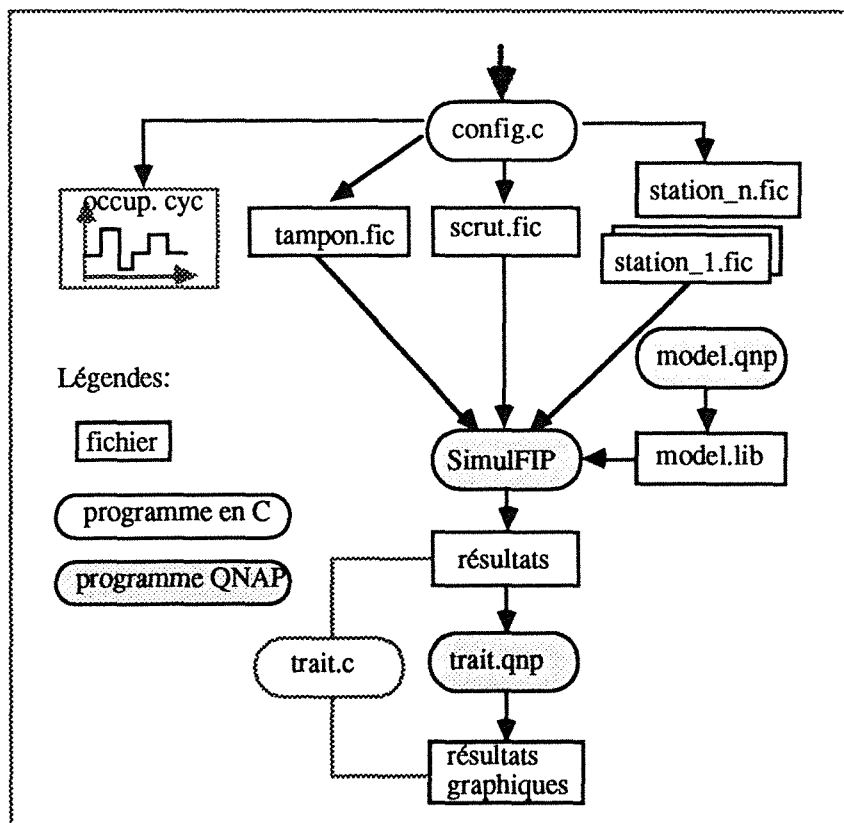


Fig. A2.1 Structure du logiciel

Sur la figure A2.1, il y a deux parties essentielles:

- simulateur en QNAP2
- configurateur en C.

Il est décrit module par module.

1) **config.c**

Ce programme permet à l'utilisateur de saisir tous les paramètres nécessaires pour la configuration du système à simuler. Il convient de signaler que le configurateur ne tient pas compte des contraintes imposées par la couche application. Pour lancer le programme, il suffit de taper **config**. La saisie est guidée et contrôlée.

Les fichiers de sortie de ce programme sont:

- **scrut.fic**
- **station_i.fic** (i=1, 2, ..., nbs, où nbs est le nombre de stations raccordées)
- **tempo.fic**

Il est aussi possible de sortir la table de scrutation sous forme graphique afin d'en avoir une vue tangible.

2) **station_i.fic** (table locale produite)

C'est un fichier séquentiel contenant les cinq champs suivants:

INTEGER	numéro_ID	[1, ..., 2 ¹⁶]
STRING	type_ressource	[B_DATprod, B_REQ, F_MSGcyc]
INTEGER	num_ressource	[1, ..., max(nbal, nbuf)]
INTEGER	size_trame	[ID_DAT seulement 5+n (n≤128) octets]
STRING	support_de	[rien, free-msg, spec-msg]

plus deux champs utilisés seulement pour la génération de la table de la scrutation:

STRING	type_trame	[ID_DAT, ID_RQ1_p, ID_MSG_p]
INTEGER	période_scrut	[en milliseconde]

Les informations permettant de générer les tables locales produites sont saisies par l'utilisateur en utilisant le programme **config.c**. Le programme génère également un fichier intermédiaire **tempo.fic** qui contient le nombre de ressources périodiques (statiques) comme F_MSGcyc, B_REQ, et des informations générales telles que le nombre des stations à simuler nbs, la conformité supportée par chaque station, etc...

2*) **table locale consommée**

INTEGER	numéro_ID	[1, ..., 2 ¹⁶]
---------	-----------	----------------------------

* Le simulateur ne tient pas compte de la consommation des informations car la réception n'influence pas les performances. Elle est donc éliminée afin d'économiser la mémoire de machine. Bien entendu, si nécessaire (par exemple pour simuler un réseau FIP multi BA relié par un "pont", cf. A2.2), l'ajout est toujours possible.

3) scrut.fic (table de scrutation)

L'élément du fichier est composé de deux champs:

INTEGER	numéro_ID	[1, ..., 2 ¹⁶]
STRING	type_trame	[ID_DAT, ID_RQ1_p, ID_MSG_p, \$]

Un caractère spécial "\$" est utilisé pour marquer la fin de la fenêtre statique de chaque Tbc.

Cette table de scrutation est générée en utilisant les informations suivantes:

- les tables locales produites **station_i.fic**
- le débit de transmission "Débit"
- la durée d'une transaction du transfert cyclique:
[ID_DAT + T0 + RP_DAT_* + T0]
- la taille minimale de la fenêtre **rqp**:
[ID_RQ1_p + T0 + RP_RQ1 + T0 + 64(ID_DAT + T0 + RP_DAT + T0)]
- la taille minimale de la fenêtre **mp**:
[ID_MSG_p + T0 + RP_MSG_NOACK + T0 + RP_FIN + T0]
où
[ID_MSG_p+T0+RP_MSG_ACK+T0+RP_ACK+T0+RP_FIN+T0]

4) model.qnp

C'est le noyau du "SimulFIP". En effet, le fonctionnement de la couche liaison de données de FIP est décrit en langage de simulation QNAP2.

Voici le modèle fonctionnel qui montre la circulation d'informations dans le bus de terrain FIP (FigA2.2).

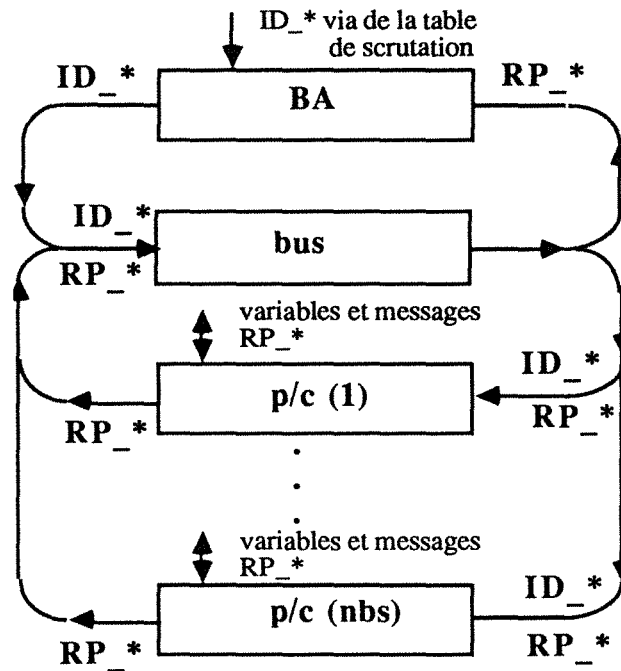


Fig. A2.2 Modèle fonctionnel de FIP

Sur la figure A2.2, BA est l'entité d'arbitre de bus, p/c(i), $i=1, 2, \dots, nbs$ représentent les stations productrice/consommatrice d'informations. ID_* et RP_* sont des trames d'identifieur et de réponse. bus représente le câble et les comportements de transmission.

Ce modèle peut être réalisé d'une façon modulaire pour faciliter la modification éventuelle (figure A2.3).

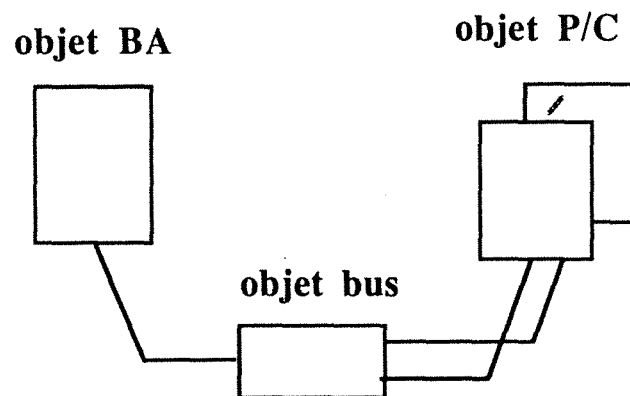


Fig. A2.3 Structure du modèle de simulation

L'objet **bus** n'est qu'une file d'attente simple qui introduit un délai de transmission proportionnel à la taille de trame (CST(taille/Débit)), une probabilité d'erreur de transmission qui conduira à la perte des trames ou à la génération des trames erronées peut être lui affectée. Il assure également la diffusion générale des trames et *le routage de certaines types de trames (cette fonction n'apparaît pas à la norme. Elle permet de diminuer l'occupation de la mémoire machine et le temps d'exécution lors de la simulation)*. Nous signalons que cette file n'a qu'un serveur qui représente la ressource commune partagée par les stations raccordées au bus. Elle permet d'observer l'occupation du canal de transmission qui est un paramètre important de performances.

L'objet **BA** et **p/c(i)** sont un ensemble de files d'attente, sémaphores et drapeaux.

Le programme **model.qnp** est compilé, et le fichier en code objet est ensuite stocké dans la bibliothèque "model.lib". Nous signalons que le nombre maximal de stations est 40, car normalement, c'est le nombre maximal de stations qu'un segment FIP peut supporter.

5) SimulFIP

Lorsque la table de scrutation **scrut.fic** ainsi que les tables locales produites **station_i.fic** ($i=1, \dots, nbs$) sont obtenues, la configuration est terminée. Il suffit d'appeler le protocole programmé en QNAP2 qui est stocké dans la bibliothèque "model.lib". On entre alors dans la phase de simulation.

SimulFIP est un module exécutable. Avant de lancer la simulation, il saisit des paramètres d'initialisation qui lui sont propres. En principe, le simulateur a besoin de paramètres d'entrées suivants:

- le fichier intermédiaire "tempo.fic" qui contient:

◇ $nbs \leq 40$;

◇ telle station possède tel nombre de ressources statiques comme $F_MSGcyc(i,j)$, $B_REQ(i,j)$, etc...

◇ si une station supporte encore les services aperiodiques, les type de ressources dynamique comme $F_MSGaper$, F_REQi , et B_REQ (si $RQ_INHIBE=FAUX$);

- la longueur des files d'attente L_*

- les intensités des flux d'entrées des trafics aléatoires lbd_* (de lbd_*_min à lbd_*_max avec l'échelle lbd_*_echel) pour toutes les stations.

Nous signalons ici que les flux d'entrées sont supposés poissonniens dans le modèle de simulation. Puisqu'il nous manque pour l'instant des informations concernant les caractéristiques de ces flux, une telle supposition semble raisonnable, au moins pour les messageries. En fait, dans FIP, les messages arrivent d'une façon aléatoire avec une taille qui ne dépasse pas 256 octets. Si la taille des messages est plus importante, une segmentation doit être réalisée dans la couche supérieure et dans ce cas, un flux de Poisson composé semble mieux adapté. Enfin, l'utilisateur a toujours la possibilité de collecter des données de flux d'entrées, puis de réaliser un tableau d'entrée.

On peut lancer la simulation en tapant **simulFIP**. Les résultats numériques seront mis dans un répertoire `"/resultats"`.

6) **trait.qnp**

Si les résultats sous forme graphique sont préférés, il suffit de lancer le programme exécutable **trait**, puis de suivre les indications. Nous signalons que ce programme est pour l'instant écrit en QNAP2. Si on veut que la présentation des résultats soit portable (sur des machines hétérogènes), il vaut mieux de l'écrire en C avec des outils graphique adaptés (**trait.c**).

2) Présentation des résultats

Dans le répertoire `"/resultats"`, on trouve les fichiers suivants:

- **monit.rst**: occupation du médium de transmission par différents trafics (cyc, mp, rqp, msg, rqa) "moyennée" sur l'intervalle d'un macro cycle.
- **occ.rst**: occupation du médium de transmission par différents trafics (cyc, mp, rqp, msg, rqa) "moyennée" sur toute la durée de la simulation.
- **dmaI.rst** (I=1,..., nbs): paramètres concernant la file F_MSGaper et les messages aperiodiques. Ce sont: flux d'entrée, délai moyen, nombre moyen dans la file, nombre maximal et probabilité de rejet.
- **dmpJ_I.rst** (I=1,..., nbs, J=1,..., nombre de F_MSGcyc): paramètres concernant la file F_MSGcyc et les messages périodiques. Ce sont: flux d'entrée, délai moyen, nombre moyen dans la file, nombre maximal et probabilité de rejet.
- **drlnI.rst** (I=1,...,nbs): paramètres concernant le service de demande explicite libre normale. Ce sont : flux d'entrée, délai moyen, nombre moyen dans la file, nombre maximal et probabilité de rejet.

-
- **drluI.rst** (I=1,...nbs): paramètres concernant le service de demande explicite libre urgente. Ce sont : flux d'entrée, délai moyen, nombre moyen dans la file, nombre maximal et probabilité de rejet.
 - **drsJ_I.rst** (I=1,...nbs, J=1,...,nombre de B_REQ): paramètres concernant le service de demande explicite spécifiée. Ce sont : flux d'entrée, délai moyen, probabilité d'écrasement.
 - **idm.rst** : paramètres concernant la file F_IDMSG. Ce sont : flux d'entrée, nombre moyen dans la file, nombre maximal et probabilité de rejet.
 - **idr1.rst** : paramètres concernant la file F_IDRQ1. Ce sont : flux d'entrée, nombre moyen dans la file, nombre maximal et probabilité de rejet.
 - **idr2.rst** : paramètres concernant la file F_IDRQ2. Ce sont : flux d'entrée, nombre moyen dans la file, nombre maximal et probabilité de rejet.
 - **pcnmaI.rst** (I=1,...nbs): paramètres concernant la file F_MSGaper. Ce sont : flux d'entrée et la distribution de probabilité marginale.
 - **pcnmpJ_I.rst** (I=1,...nbs, J=1,...,nombre de F_MSGcyc): paramètres concernant la file F_MSGcyc. Ce sont : flux d'entrée et la distribution de probabilité marginale.
 - **pcnrJ_I.rst** (I=1,...nbs, J=1,...,nombre de F_MSGcyc): paramètres concernant le buffer B_REQ. Ce sont : flux d'entrée et la distribution de probabilité marginale.
 - **pcnidm.rst** : paramètres concernant la file F_IDMSG. Ce sont : flux d'entrée et la distribution de probabilité marginale.
 - **pcnidr1.rst** : paramètres concernant la file F_IDRQ1. Ce sont : flux d'entrée et la distribution de probabilité marginale.
 - **pcnidr2.rst** : paramètres concernant la file F_IDRQ2. Ce sont : flux d'entrée et la distribution de probabilité marginale.

3) Résumé

Une fois les résultats obtenus, les performances du système simulé sont connues. Si ces performances ne correspondent pas à notre attente, il est possible de modifier certains paramètres et de relancer la simulation. C'est la phase dite alternative. Grâce aux résultats obtenus, l'utilisateur peut ainsi voir si les files d'attente sont sur-dimensionnées ou sous-dimensionnées et ainsi prendre la décision de modification. Notons qu'il ne s'agit pas d'une démarche simple. Une connaissance approfondie du système à simuler et une expérience d'évaluation des performances sont nécessaires.

Le SimulFIP présenté dans cette annexe est un outil de pré-implémentation. Il faut signaler qu'il s'agit d'une simulation d'un modèle FIP "simplifié" mais suffisamment détaillé pour nos besoins. Quant à l'interface utilisateur, il est satisfaisant pour l'utilisation en laboratoire. Un interface graphique plus convivial est envisageable si le besoin s'en fait sentir. Ce logiciel peut donc avoir un débouché industriel direct.

Comme on le voit, les outils de dimensionnement sont dans une voie de simplification et d'adaptation à une plus grande masse d'utilisateurs. S'il est vrai que ces outils sont aujourd'hui toujours difficiles d'utilisation, ils n'en sont pas moins une nécessité industrielle qui ne fait que devenir plus draconienne étant donnée l'importance des coût en jeu. Si un dimensionnement et une analyse qualitative sont correctement menés sur un système à installer, le coût de maintenance de ce système peut être réduit.

L'étude des performances n'est plus un luxe et devient une partie nécessaire à la connaissance et à la compréhension des systèmes à des coût raisonnables.

A2.2 Possibilités d'extension

SimulFIP nous a permis d'effectuer l'évaluation des performances au niveau de la couche liaison de données sur un seul segment de bus FIP (i.e. mono BA), mais il est possible de l'étendre en modifiant le simulateur, celui-ci étant réalisé de façon modulaire. Deux possibilités d'extension sont données sans entrer dans le détail.

1) Vers la couche Application

Dans le simulateur, les flux d'informations provenant de la couche Application sont modélisés comme des flux d'entrées Poissonniens. Ces flux peuvent être également générés en utilisant FIP-Toolbox [Centre 90] qui est un outil de configuration au niveau de la couche application. Le "SimulFIP" peut remplacer la carte FULLFIP, sachant que cette dernière n'intègre qu'une partie (cyclique) du protocole FIP.

2) Vers un réseau FIP multi BA

Il est également possible de simuler un réseau FIP en raccordant plusieurs segments FIP par des "ponts" [Creusot 91] et ainsi former un réseau FIP-FIP multi BA. Ce qu'il reste à faire est de réaliser un objet "pont".

3) Etablissement d'une bibliothèque des réseaux de terrain

Il est peu probable qu'une normalisation internationale de bus de terrain voit le jour dans les cinq ans à venir. Une comparaison entre différentes propositions est nécessaire. Il est possible d'établir une bibliothèque des réseaux de terrain existant aujourd'hui pour pouvoir comparer les performances des différentes propositions afin de choisir la solution la mieux adaptée à l'application.

Annexe 3. Techniques d'accès au médium à temps critique

Dans ce paragraphe, on aborde l'ingénierie de protocoles.

Un réseau local à temps critique diffère d'un réseau local ordinaire dans l'aspect contrainte du temps. Dans un réseau à temps critique, deux types de trafics coexistent: le trafic à temps critique et le trafic à temps non-critique. Le trafic à temps critique a un caractère déterministe, les trames à transférer arrivent dans la station émettrice d'une façon cyclique. Le trafic à temps non-critique a un caractère aléatoire, plus précisément, les trames à transférer arrivent dans la station émettrice d'une façon aléatoire. Une technique d'accès au médium dit "à temps critique" est nécessaire de façon à assurer tout d'abord un délai de transfert du trafic à temps critique borné dans une fenêtre de temps fixe "Twin" tout en conservant des performances du trafic à temps non-critique acceptable. Au delà de "Twin", les trames à temps critique arrivées aux destinataires ne sont plus valables.

La technique d'accès au médium utilisé dans FIP est conçue sur mesure, c'est à dire en prenant en compte des caractéristiques cycliques du trafic à temps critique, FIP adopte une technique d'accès basé sur le principe de réservation (procédure "tour de table") pour le trafic à temps critique. Le trafic à temps non-critique circule durant le temps non occupé (temps mort) du trafic à temps critique.

La technique d'accès Jeton temporisé ("Timed Token") utilisé dans MAP [MAP3.0 87] et FDDI [ANSI 86] peut satisfaire aux exigences du temps réel. Il a été proposé récemment comme technique d'accès à temps critique pour satisfaire les exigences des applications à temps critique (plus strict que temps réel). Deux priorités sont accordées à deux types de trafics: synchrone et asynchrone qui correspondent respectivement aux trafics à temps critique et non-critique. Une durée limitée est associée à la possession du jeton d'une station pour chaque priorité particulière et le temps maximal de rotation du jeton est prédéfini. On note que MAP utilise la technique d'accès normalisé "Bus à jeton" [IEEE802.4 85] tandis que FDDI utilise une technique d'accès qui ressemble à "l'Anneau à jeton" [IEEE802.5 85]. Mais on peut prouver que le mécanisme de priorité de FDDI est identique à celui du bus à jeton bien que la topologie de FDDI soit de type anneau.

En effet, les techniques d'accès au médium FIP ou le Jeton temporisé sont tous deux des cas particuliers d'un système de scrutation ("polling system"). En négligeant la différence concernant la topologie de connexion des stations, et en considérant d'une façon abstraite le système, on constate qu'un système de scrutation n'est qu'un ensemble de files d'attente (ou stations) servies par un seul serveur (ou médium de transmission) de manière cyclique.

Le système de scrutation fut d'abord étudié par [Mark 57] au titre de réparation de machines (connu comme modèle de "repair man"). Depuis cette époque, ce modèle est appliqué au domaine des télécommunications (ex: "token bus" et "token ring"), et des recherches considérables ont été effectuées. Un panorama de ce problème, accompagné d'une liste très complète de références a été présenté par Takagi [Takagi 88].

Dans la suite, on développera une simple comparaison entre ces deux techniques. Mais avant tout, nous voudrions noter quelques précisions concernant les critères à comparer et les caractéristiques des trafics.

Le critère le plus important pour un protocole à temps critique est sans doute le délai de transfert noté W . Ce délai de transfert est composé d'un délai d'attente (queueing delay) W_q dans la file d'attente d'émission, un délai d'accès W_a qui correspond au temps d'attente entre le moment où un client considéré se place en tête de la file et le moment où la transmission commence, et un délai de transmission W_s (cf. Fig.A3.1).

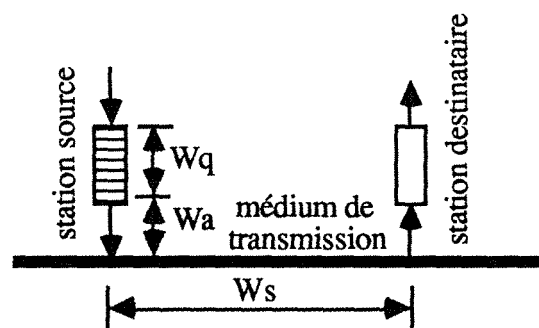


Fig. A3.1 Délai de transfert d'un client

Sur la figure A3.1, on ne tient pas compte des délais de propagation et d'attente à la réception.

On suppose que les instants d'arrivées des clients sont répartis ainsi qu'il est montré sur la figure A3.2.

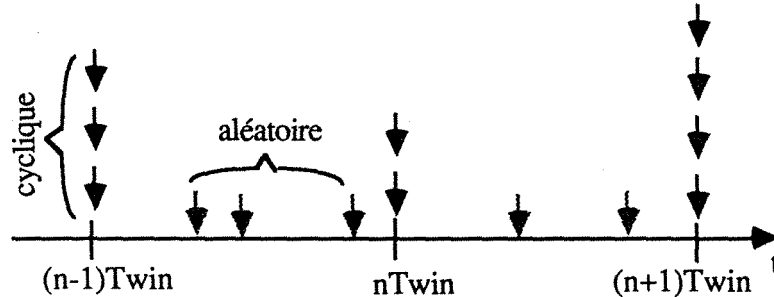


Fig. A3.2 Modèle des trafics

Les clients à temps critique arrivent dans la file d'émission de façon cyclique aux instants fixes: nT_{win} . Il doivent être transmis dans un délai T_{win} borné. Par contre, les clients à temps non-critique arrivent de façon aléatoire. Il sont moins exigeants sur le délai de transfert.

A3.1 Comparaison qualitative

On considère un réseau composé de N stations homogènes. De plus, pour FIP, on suppose qu'il n'assure que deux services: transfert cyclique et messagerie aperiodique (i.e: une implémentation de FIP de classe de conformité 6 et 7, cf. §2, tab.2.1); et pour le Jeton temporisé, on suppose qu'il ne supporte que deux priorités: synchrone et asynchrone. On a ainsi établi une correspondance entre "cyclique - synchrone" et "messagerie - asynchrone". On étudie d'abord le fonctionnement de ces deux systèmes.

A3.1.1 FIP de classe de conformité 6 et 7

La conformité 6 correspond à une implémentation de FIP avec service cyclique et messagerie sans acquittement. La conformité 7 correspond à une implémentation de FIP avec le service cyclique et messagerie avec acquittement. Nous ignorons la différence entre messagerie avec acquittement et sans acquittement en supposant une transmission fiable.

On examine la façon dont FIP traite les clients à temps critique qui arrivent de façon cyclique.

On suppose que ces clients arrivent deux par deux dans une station de période T_{win} , la table de scrutation peut être configurée afin que cette station soit scrutée deux fois consécutives par cycle élémentaire T_{bc} . FIP comprend deux options:

- Avec "option synchronisation", il suffit donc de poser $T_{win} = T_{bc}$, les deux clients à TC (temps critique) sont traités comme indiqué sur la figure A3.3:

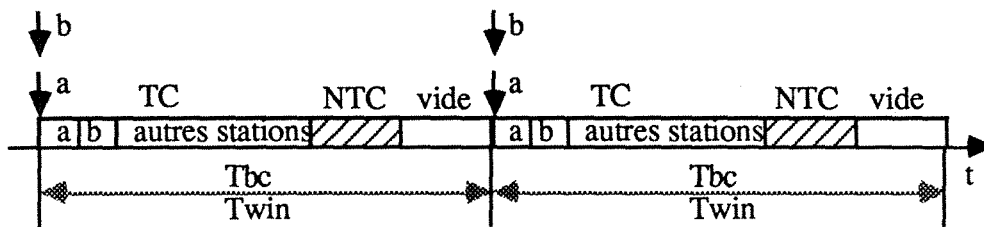


Fig. A3.3 FIP avec option synchronisation

On voit que durant la fenêtre TC, le système traite tout de suite le client a, et puis le client b dans un délai W_s correspondant au temps de transmission d'un client. Les clients à TC des autres stations sont traités de la même façon avant que la fenêtre NTC (non temps critique) soit atteinte. On note que le cycle élémentaire suivant ne commence qu'à l'expiration du temps T_{bc} . La synchronisation est assurée par l'arbitre de bus. Il pourrait donc y avoir un moment où le médium de transmission soit "vide".

- Avec "option sans synchronisation", les durées des cycles élémentaires ne sont plus constantes, mais une durée maximale $T_{bc_{max}}$ est exigée. On doit imposer, dans ce cas, $T_{win} = T_{bc_{max}}$. Les deux clients à TC (temps critique) sont traités comme indiqué sur la figure A3.4:

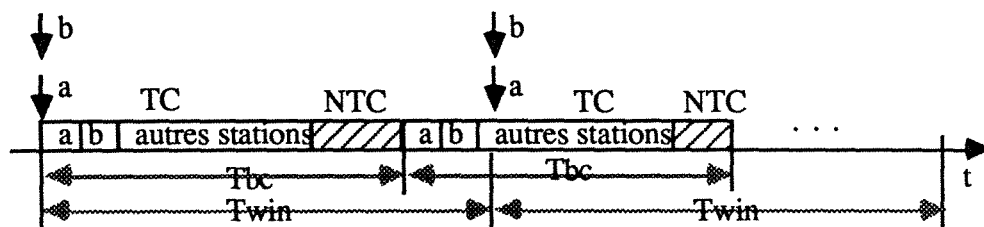


Fig. A3.4 FIP avec option sans synchronisation

La figure A3.4 montre le cas le moins favorable. Les clients a et b pourront arriver immédiatement juste après que la file d'émission correspondante ait été scrutée par

l'arbitre de bus. FIP avec option sans synchronisation peut aussi être utilisé dans des applications à TC mais avec un délai de transfert borné par $T_{bc} (\leq T_{win})$.

A3.1.2 Jeton temporisé avec deux priorités: synchrone et asynchrone

L'idée de base de la technique d'accès Jeton temporisé est présentée par Grow [Grow 82] et étudiée par Ulm [Ulm 82], en considérant deux classes de priorités: synchrone et asynchrone.

Le protocole normalisé bus à jeton (IEEE802.4) est une extension de cette idée. Chaque station supporte 4 priorités différentes appartenant à l'ensemble (6, 4, 2, 0). Cela permet de privilégier les clients urgents pour le respect des contraintes de temps liées à certaines applications en temps réel ou critique.

Le protocole FDDI utilise aussi une extension de la technique d'accès Jeton temporisé. Chaque station supporte 2 priorités ("synchrone", "asynchrone", mots spécifiques à la norme, à ne pas confondre avec l'option synchronisation et sans synchronisation de FIP) avec en option 8 sous priorités différentes de la priorité "asynchrone".

Les deux protocoles normalisés ci-dessus adoptent le même mécanisme: Jeton temporisé (Timed Token).

Technique d'accès au médium Jeton temporisé [Stallings 88]: Lorsque le réseau est initialisé, les stations reliées au médium de transmission (que ce soit bus ou anneau) décident une valeur du temps de rotation du jeton TTRT (Target Token Rotation Timer). C'est le temps maximal séparant deux réceptions successives du jeton par une station. Le temps d'accès maximal est ainsi garanti. De plus, on définit un temps maximal d'utilisation du jeton "SAi" durant lequel une station *i* peut transmettre les clients de priorité "synchrone". Ces "SAi" peuvent être différents pour des stations différentes. Mais la somme de tous les "SAi" doit être inférieure ou égale à TTRT. Chaque station dispose d'un compteur, qui mesure le temps de rotation du jeton TRT (Token Rotation Time) permettant de calculer le temps séparant deux réceptions successives du jetons. Lorsqu'une station reçoit le jeton, elle charge la valeur de TRT dans un compteur dit temps de possession du jeton THT (Token Holding Timer) puis réinitialise et déclenche, de nouveau, le compteur TRT. Ensuite, la station émet ses trames, en respectant les règles suivantes:

- 1) toute station *i* peut émettre ses trames "synchrones" jusqu'à concurrence de SAi;

2) après avoir transmis les trames "synchrones", l'incrémentation de TRT est déclenchée, la station peut émettre des trames asynchrones tant que $THT < TTRT$.

L'utilisation du médium est donc divisée en deux fenêtres, l'une pour les transferts "synchrones" et l'autre pour les transferts "asynchrones" (i.e: les trames "synchrones" et "asynchrones" sont échangées dans les fenêtres correspondantes). Le paramètre SA_i définit la durée de la fenêtre "synchrone" de la station i et la différence $(TTRT - THT)$ calculée, par la station, à la fin de la fenêtre "synchrone", lui donne la durée de sa fenêtre "asynchrone".

Maintenant, nous examinons la façon dont la technique Jeton temporisé traite les clients à TC (temps critique) qui arrivent de façon cyclique.

On suppose que les clients à TC arrivent deux par deux dans une station de période T_{win} , on peut donc définir $TTRT = T_{win}$ et $SA_i = 2$. On examine trois cas suivants:

- cas 1: pas de trafic "asynchrone" ($TTRT = \sum SA_i$).

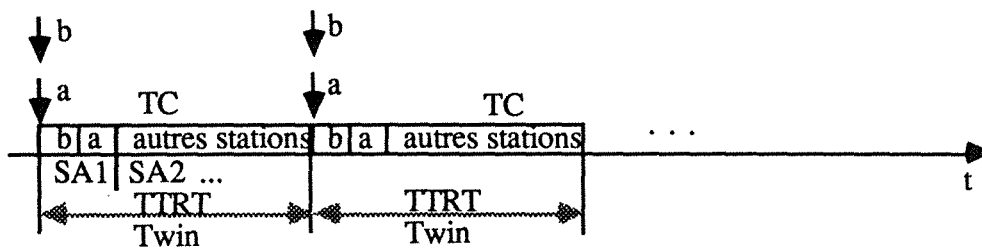


Fig.A3.5 Jeton temporisé du cas 1

Dans ce cas particulier, la technique Jeton temporisé s'adapte à la demande d'applications à TC. Il suffit de régler le débit de transmission pour que $T_{win} = TTRT = \sum SA_i$.

- cas 2: trafic asynchrone est supposé très fort ($TTRT \geq \sum SA_i + \text{durée de fenêtre asyc.}$, avec $\sum SA_i + \text{durée de fenêtre asynchrone} = \text{constante}$).

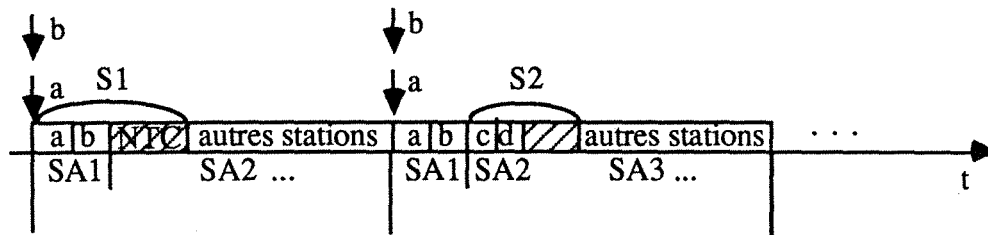


Fig. A3.6 Jeton temporisé du cas 2

Lorsque le trafic "asynchrone" est très fort, on considère qu'il y a toujours autant de clients "asynchrones" dans chaque station. Ainsi la durée de la fenêtre "asynchrone" est toujours épuisée par la première station qui trouve $THT < TTRT$. Et la durée totale d'un cycle est alors constante.

$W_a \leq TTRT - \sum SA_i$, et $W = W_a + W_s$. Dans ce cas, le Jeton temporisé s'adapte à la demande d'application à TC. Il suffit de régler le débit de transmission pour que $T_{win} = TTRT = \sum SA_i + \text{durée de fenêtre asynchrone}$.

- cas 3: cas général ($TTRT \geq \sum SA_i + \text{durée de fenêtre asynchrone}$,
et $\sum SA_i + \text{durée de fenêtre asynchrone} = \text{variable}$).

Dans ce cas général, on a toujours le délai d'accès borné $W_a \leq 2TTRT$ [Johnson 87] [Sevcik 87]. Lorsque $T_{win} \geq 2TTRT$, on peut dire que ce protocole peut être utilisé dans un environnement à TC.

En résumé, la technique Jeton temporisé s'adapte aussi au temps critique, mais avec un délai de transfert plus important que FIP.

Par ailleurs, un autre problème important est le critère du "gitter" qui est une sorte de déphasage dans le temps. Si on veut éliminer le "gitter", il ne nous reste plus que FIP avec option synchronisation.

Il semble que le protocole "Token Ring" s'adapte mieux à TC que le Jeton temporisé. En effet, "Token Ring" ressemble beaucoup à FIP avec option sans synchronisation. Si on rajoute un mécanisme de synchronisation (cela est facile à réaliser, il suffit d'armer chaque station par un "timer" absolu qui compte T_{bc}), "Token Ring" aura les mêmes performances que FIP avec option synchronisation vis à vis du TC. Un problème de synchronisation d'horloge apparaît. Dans FIP, la synchronisation d'horloge

est centralisée [He 90], tandis que dans "Token Ring", on a besoin d'une synchronisation d'horloge répartie.

A3.2 Vers une comparaison quantitative

La comparaison quantitative entre différentes techniques d'accès utilisés dans des applications à temps critique est possible. Le critère de temps "délai de transfert W" occupe une place importante. Mais ce n'est pas le seul critère de performances. On définit dans la suite une fonction de performances globales qui englobe les trois critères habituels tout en accordant une certaine importance au critère de temps (délai). Un des intérêts d'avoir une fonction de performances globales est de faciliter le choix parmi les protocoles à temps critique. En général, chaque objet a ses aspects avantageux et ses aspects désavantageux par rapport aux autres objets. Une comparaison de chacun des critères rend la prise de décision du choix de protocoles difficile.

A3.2.1 Métrique de performances globales

On définit d'abord:

- W_{TC} : délai de transfert d'un client à TC,
- W_{NTC} : délai de transfert d'un client à NTC,
- $\tau = \tau_{TC} + \tau_{NTC} \leq 1$: Taux d'utilisation du médium,
- $\eta = \frac{\eta_{TC}\tau_{TC} + \eta_{NTC}\tau_{NTC}}{\tau} \leq 1$: rendement du protocole.

La fonction de performances globales peut être donnée par:

$$F(W_{TC}, \eta, \tau) = c \frac{T_{win}}{E(W_{TC})} + (1-c)(\eta + \tau) \quad (A3.1)$$

où:

- $\frac{T_{win}}{E(W_{TC})} \geq 1$, si le délai moyen de transfert des clients à TC est borné par T_{win} ;
- $\frac{T_{win}}{E(W_{TC})} \leq 1$, si le délai moyen de transfert des clients à TC n'est pas borné par T_{win} .
- c est un coefficient du temps critique qui est défini comme:

$$c = \frac{\lambda_{TC}}{\lambda_{TC} + \lambda_{NTC}} \quad (A3.2)$$

où:

$$\lambda_{TC} = \lambda_{cyc} + p\lambda_{aléa} \quad (A3.3)$$

est l'intensité du flux d'arrivée des trames à temps critique. Il est composé du trafic cyclique λ_{cyc} et d'une partie de trafic aléatoire $p\lambda_{aléa}$ (avec $p < 1$).

Et:

$$\lambda_{NTC} = (1 - p)\lambda_{aléa} \quad (A3.4)$$

est l'intensité du flux d'arrivée des trames à temps non-critique qui est composé uniquement du trafic aléatoire.

Le trafic total généré par la station considérée est:

$$\lambda = \lambda_{TC} + \lambda_{NTC} = \lambda_{cyc} + \lambda_{aléa} \quad (A3.5)$$

Une valeur de F grande implique que les performances du protocole sont bonnes. Le coefficient c proche de 1 implique que la proportion du trafic à temps critique est importante. On accorde ainsi un poids important au critère délai de transfert ou plus précisément au terme $\frac{T_{win}}{E(W_{TC})}$.

NB 1: Le terme $\frac{T_{win}}{E(W_{TC})}$ est discutable.

Le paramètre moyen $E(W_{TC})$ ne reflète pas bien le problème. L'idéal serait de déterminer la distribution de probabilité de W_{TC} , et évaluer ensuite (analytiquement ou par simulation) le pourcentage (ou probabilité) des trames qui dépassent la fenêtre de temps limité T_{win} .

NB 2: Le délai de transfert des clients à temps non-critique W_{NTC} n'est pas inclus dans la fonction F car on a supposé que ces clients peuvent attendre longtemps. Si cela n'est pas le cas, on peut remplacer tout de même τ par le terme $\frac{\tau}{E(W_{NTC})}$ qui est le critère puissance ("Power").

Jusqu'à présent, les performances de FIP sont connues grâce à notre étude. La fonction de performances globales F peut donc être calculée.

Une courbe "fonction de performances globales vs. taux d'utilisation τ (ou charge du réseau) en changeant le coefficient $c = 0,1; 0,5; 0,9$ " pourrait bien illustrer les performances de FIP.

Par contre, on ne dispose pas de résultats pratiques (c.à.d faciles à utiliser) en ce qui concerne le Jeton temporisé. Selon [Johnson 87] et [Sevcik & Johnson 87], on peut résumer deux propriétés:

- Le temps moyen de la rotation du jetons ne dépasse pas le TTRT;
- Le temps maximal de la rotation du jeton ne dépasse pas deux fois TTRT.

En ce qui concerne les résultats analytiques, on trouve peu de choses. Avec une hypothèse de capacité de file d'attente de chaque station limitée à 1, Takagi [Takagi 90] a étudié les performances (le temps de rotation du jeton, utilisation, et délai) de son modèle.

Valenzano [Valenzano 90] a effectué une étude avec des hypothèses plus réalistes:

- 1) On ne considère que deux priorités: sync. et async.
- 2) Les files d'attente moins prioritaires sont surchargées.
- 3) Le flux d'arrivée des trames est périodique.

En revanche, on trouve beaucoup des résultats de simulation [Bux 84], [Summers 86], [Alijani 90], ...etc. Mais ces résultats sont liés aux paramètres d'un modèle concret et donc difficile à réutiliser directement pour notre but et difficile à exploiter dans un cadre général.

Une simulation du Jeton temporisé avec des paramètres correspondants à ceux utilisés pour la simulation de FIP serait très instructive et permettrait notamment de comparer les fonctions de performances globales.

TABLE DES MATIERES

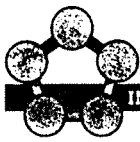
INTRODUCTION GENERALE	11
1.GENERALITES ET CONTEXTE DE TRAVAIL	17
1.1 RESEAUX LOCAUX INDUSTRIELS ET BUS DE TERRAIN	17
1.2 BUS DE TERRAIN DANS UN SYSTEME AUTOMATISE	18
1.2.1 Architecture traditionnelle	18
1.2.2 Structuration fonctionnelle	20
1.2.3 Types de trafics.....	21
<i>1.2.3.1 Trafics vertical et horizontal</i>	<i>21</i>
<i>1.2.3.2 Communication d'état et d'événement</i>	<i>23</i>
<i>1.2.3.3 Remarques sur les trafics</i>	<i>24</i>
1.2.4 Modèles Client-Serveur (CS) et Producteur-Distributeur-Consommateurs (PDC).....	25
1.2.5 Application du modèle PDC aux bus de terrain.....	27
1.3 PRESENTATION GENERALE DE FIP	29
1.3.1 Architecture de FIP	29
1.3.2 Services offerts par FIP	33
1.3.3 Technique d'accès au médium de FIP	35
1.4 METHODOLOGIE ET CRITERES DE PERFORMANCES	39
1.5 NORMALISATION	45
1.5.1 Histoire de bus de terrain.....	45
1.5.2 Normes françaises	46
1.5.3 Normalisation internationale	46
2.DESCRPTION ET MODELISATION	51
2.1 PRESENTATION DE LA COUCHE LIAISON DE DONNEES DE FIP	51
2.1.1 Vue globale	51

2.1.2	Services de transfert cyclique.....	55
2.1.3	Services de demande explicite.....	56
	2.1.3.1 <i>Demande explicite spécifiée périodique</i>	59
	2.1.3.2 <i>Demande explicite spécifiée aperiodique</i>	60
	2.1.3.3 <i>Demande explicite libre urgente</i>	61
	2.1.3.4 <i>Demande explicite libre normale</i>	62
2.1.4	Service de messagerie.....	63
	2.1.4.1 <i>Service de messagerie périodique</i>	64
	2.1.4.2 <i>Service de messagerie aperiodique</i>	65
2.1.5	Structure de trames et Temporisation.....	66
	2.1.5.1 <i>Structure de trames</i>	66
	2.1.5.2 <i>Temporisation</i>	66
2.1.6	Synthèse des questions.....	67
 2.2 ANALYSE DES PROBLEMES "SIMPLES"		68
2.2.1	Construction de la table de scrutation.....	68
2.2.2	Dimensionnement des fenêtres de messagerie périodique (T _{mp}) et de demande explicite périodique (T _{rqp}).....	74
	2.2.2.1 <i>Fenêtre de messagerie périodique</i>	74
	2.2.2.2 <i>Fenêtre de demande explicite périodique</i>	75
2.2.3	Rendement du protocole FIP.....	76
2.2.4	Taux d'utilisation du réseau FIP.....	78
2.2.5	Fiabilité de transfert cyclique de variables identifiées.....	79
 2.3 MODELISATION DES PROBLEMES COMPLEXES		82
2.3.1	Service de demande explicite spécifiée périodique.....	83
2.3.2	Service de messagerie périodique (avec ou sans acquittement).....	84
2.3.3	Service de messagerie aperiodique (avec ou sans acquittement).....	85
2.3.4	Service de demande explicite spécifiée aperiodique.....	94
2.3.5	Service de demande explicite libre (urgente ou normale).....	95
 2.4 MODELISATION GENERALE		97
2.4.1	Modèle commun de tous les services aléatoires.....	97
2.4.2	Principe d'étude du modèle.....	98
	2.4.2.1 <i>Etude du premier étage</i>	98
	2.4.2.2 <i>Etude du flux d'entrée du deuxième étage</i>	99
	2.4.2.3 <i>Etude du deuxième étage</i>	99

2.5 CONCLUSION	100
3.ETUDE ANALYTIQUE DU MODELE, PREMIER ETAGE.....	105
3.1 CAPACITE INFINIE	108
3.1.1 Cas simple: l'interrupteur ferme régulièrement.....	108
3.1.1.1 <i>Description du problème</i>	108
3.1.1.2 <i>Résultats obtenus</i>	109
3.1.1.3 <i>Démonstrations</i>	110
3.1.2 Cas général: l'interrupteur ferme irrégulièrement.....	119
3.2 CAPACITE LIMITEE A L	120
3.2.1 Cas simple: l'interrupteur ferme régulièrement.....	121
3.2.1.1 <i>Description</i>	121
3.2.1.2 <i>Résultats obtenus</i>	121
3.2.1.3 <i>Démonstrations</i>	122
3.2.2 Cas général: l'interrupteur ferme irrégulièrement.....	125
3.2.2.1 <i>Description</i>	125
3.2.2.2 <i>Résultats obtenus</i>	125
3.2.2.3 <i>Démonstrations</i>	125
3.3 Application aux services périodiques de FIP.....	126
3.3.1 Service de messagerie périodique.....	127
3.3.2 Service de demande explicite spécifiée périodique.....	129
3.3.2.1 <i>Etude analytique</i>	129
3.3.2.2 <i>Application</i>	133
4.ETUDE ANALYTIQUE DU MODELE, DEUXIEME ETAGE.....	137
4.1 FLUX D'ENTREE DU DEUXIEME ETAGE	139
4.1.1 Flux de sortie d'une station du premier étage.....	139
4.1.1.1 <i>Cas simple où les interrupteurs du premier étage ferment régulièrement.</i>	139
4.1.1.2 <i>Lorsque les interrupteurs du premier étage ferment irrégulièrement.</i>	143
4.1.2 Flux d'entrée du deuxième étage.....	144

4.2 DEUXIEME ETAGE	146
4.2.1 Capacité infinie	146
4.2.1.1 Nombre de clients dans la file.....	146
4.2.1.2 Délai d'attente	147
4.2.2 Capacité limitée à L	149
4.2.2.1 Nombre de clients dans la file.....	149
4.2.2.2 Délai d'attente dans la file.....	153
4.2.3 Divers.....	154
4.2.3.1 B_n variable.....	154
4.2.3.2 Lorsque deux priorités différentes existent	156
4.3 APPLICATION AUX SERVICES APERIODIQUES DE FIP	157
4.3.1 Service de messagerie apériodique.....	157
4.3.1.1 Configuration examinée.....	157
4.3.1.2 Comparaison entre les résultats analytiques et les résultats de simulation	160
4.3.1.3 Le temps de réponse.....	163
4.3.2 Services de demande explicite apériodique.....	164
4.3.2.1 Description	165
4.3.2.2 Comportement du deuxième étage.....	167
4.3.2.3 Délai de transfert (ou temps de réponse).....	171
4.4 CONCLUSION	173
5.PERFORMANCES DES SERVICES APERIODIQUES OBTENUES PAR "SIMULFIP"	177
5.1 DESCRIPTION DE LA CONFIGURATION A SIMULER	177
5.2 SERVICE DE MESSAGERIE APERIODIQUE	181
5.2.1 Résultats du cas 1	181
5.2.2 Résultats du cas 2	184
5.2.3 Calcul du temps de réponse maximal d'un message	185
5.3 SERVICES DE DEMANDE EXPLICITE APERIODIQUE	186

5.4 CONCLUSION	194
CONCLUSION GENERALE ET PERSPECTIVES	197
BIBLIOGRAPHIE	203
ANNEXE 1. DEFINITIONS	215
ANNEXE 2. SIMULATEUR "SIMULFIP"	219
A2.1 DESCRIPTION DU SIMULATEUR "SIMULFIP"	219
1) Structure du logiciel.....	219
2) Présentation des résultats.....	224
3) Résumé.....	225
A2.2 POSSIBILITES D'EXTENSION	226
1) Vers la couche Application.....	226
2) Vers un réseau FIP multi BA	226
3) Etablissement d'une bibliothèque des réseaux de terrain	227
ANNEXE 3. TECHNIQUES D'ACCES AU MEDIUM A TEMPS CRITIQUE.....	228
A3.1 COMPARAISON QUALITATIVE	230
A3.1.1 FIP de classe de conformité 6 et 7	230
A3.1.2 Jeton temporisé avec deux priorités: synchrone et asynchrone	232
A3.2 VERS UNE COMPARAISON QUANTITATIVE	235
A3.2.1 Métrique de performances globales	235



INPL

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

AUTORISATION DE SOUTENANCE DE THESE DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

VU LES RAPPORTS ETABLIS PAR :

Monsieur PUJOLLE, Professeur, Université Pierre & Marie Curie Versailles,
Monsieur SCHOTT, Professeur, Université de Nancy I.

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur SONG Ye-Qiong

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse
intitulée :

"Etude de performances de FIP, aide au dimensionnement d'applications"

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, 26 Juin 1991

Le Président de l'INPL

M. GANTOIS
DE
LORRAINE
NANCY



2, avenue de la Forêt de Haye - B.P. 3 - 54501 VANDOEUVRE CEDEX

Téléphone : 83. 59. 59. 59 - Télex : 961 715 F - Télécopie : 83. 59. 59. 55

Résumé:

Ce travail est une contribution à l'évaluation des performances du bus de terrain FIP (Factory Instrumentation Protocol).

A partir d'une analyse des architectures de systèmes automatisés, nous montrons le besoin de communications à temps critique et non critique entre des équipements qui se trouvent connectés à un tel réseau. Ensuite, le protocole FIP est présenté comme une solution adaptée, encore faut-il que l'évaluation des performances d'un tel protocole le prouve. Notre étude s'est concentrée sur la couche liaison de données de FIP en utilisant deux moyens: l'étude analytique et la simulation.

Pour étudier analytiquement le bus de terrain FIP, les services offerts par sa couche liaison de données sont modélisés par un système de files d'attente composé de deux étages "parallèle-série" avec des serveurs intermittents.

Une méthode d'approximation originale est développée afin que ce modèle puisse être étudié en utilisant la théorie des files d'attente.

Parallèlement à l'étude analytique, un simulateur appelé "SimulFIP", qui décrit la couche liaison de données de FIP en langage de simulation QNAP2, a été réalisé. Il est capable de simuler un bus de terrain FIP de configuration quelconque. A l'issue d'une telle simulation, la paramétrisation optimale du réseau à installer peut être réalisée.

Enfin, la pertinence des résultats analytiques est mise en évidence par la simulation, ce qui permet de justifier la méthode analytique utilisée.

Mots clés: Bus de terrain, Protocole d'accès au médium, Temps critique, Evaluation des performances, File d'attente, Simulation

Abstract:

This work is a contribution to the performance evaluation of the field bus FIP (Factory Instrumentation Protocol).

Through an analysis of the industrial process control systems architectures, we show the need of the time critical and non-critical communications among the devices connected to such network. Then, the FIP protocol is presented as an adapted solution, and the performance evaluation of such a protocol appears necessary. Our study is focussed on the data link layer of FIP by using two approaches: analytic study and simulation.

For the analytic study, the services of the data link layer of FIP are modelled by a queueing system made of two stages in "parallel-series" with the intermittent servers.

An original approximate method was made so that this model can be studied by using the queueing theory.

In parallel with the analytic study, a simulator called "SimulFIP", which describes the data link layer of FIP in the simulation language QNAP2, has been developed. It is able to simulate a FIP field bus of whatever configuration. The results obtained from such a simulation can be used to achieve the optimal parameterization of the network to be installed.

Finally, a comparison between the analytic results and the simulation ones provides a justification of the approximations used throughout our analytic study.

Key words: Field bus, MAC protocol, Time critical, Performance evaluation, Queueing analysis, Simulation