N° ordre: 2340 Année 1996

THESE

Présentée au

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTEMES DU CNRS

en vue de l'obtention du titre de

DOCTEUR DE L'UNIVERSITE PAUL SABATIER DE TOULOUSE

Spécialité: Informatique Industrielle

par

Francisco VASQUES de CARVALHO

SUR L'INTEGRATION DE MECANISMES D'ORDONNANCEMENT ET DE COMMUNICATION DANS LA SOUS-COUCHE MAC DE RESEAUX LOCAUX TEMPS-REEL

Soutenue le 25 Juin 1996, devant le jury:

F.	COTTET	Rapporteur
J.-D.	DECOTIGNIE	Rapporteur
M.	DIAZ	Président du jury
G.	JUANOLE	Directeur de Thèse
C.	KAISER	Examinateur
Z.	MAMMERI	Rapporteur
M.	SAMAAN	Examinateur

Rapport LAAS N° 96229

Cette thèse a été préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

7, avenue du Colonel Roche 31077 TOULOUSE Cedex

1024 a

N° ordre: 2340 Année 1996

THESE

Présentée au

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTEMES DU CNRS

en vue de l'obtention du titre de

DOCTEUR DE l'UNIVERSITE PAUL SABATIER DE TOULOUSE

Spécialité: Informatique Industrielle

par

Francisco VASQUES de CARVALHO

SUR L'INTEGRATION DE MECANISMES D'ORDONNANCEMENT ET DE COMMUNICATION DANS LA SOUS-COUCHE MAC DE RESEAUX LOCAUX TEMPS-REEL

Soutenue le 25 Juin 1996, devant le jury:

F.	COTTET	Rapporteur
J.-D.	DECOTIGNIE	Rapporteur
M.	DIAZ	Président du jury
G.	JUANOLE	Directeur de Thèse
C.	KAISER	Examinateur
Z.	MAMMERI	Rapporteur
M.	SAMAAN	Examinateur

Rapport LAAS N° 96229

Cette thèse a été préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

7, avenue du Colonel Roche 31077 TOULOUSE Cedex

GBI.3(043)/CARL/SUR
UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA
N.º 50899
CDU
Data 13/06/2000
Cat

Aos meus filhos, pelos longos meses que estiveram sem o pai
e especialmente à Ana, pela sua dedicação e paciência
sem as quais este trabalho nunca teria terminado

Avant Propos

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique, dirigé par Monsieur le Professeur A. COSTES, que je remercie pour son accueil.

J'exprime ma gratitude à Monsieur M. DIAZ et à l'ensemble des permanents du Groupe OLC, qui m'ont accueilli dans leur groupe, pour tout leur soutien.

Je tiens à exprimer mes remerciements à Monsieur G. JUANOLE, qui a accepté de diriger mon travail de thèse, pour toute sa disponibilité et tous les conseils qu'il m'a apporté tout au long de ce travail. Je lui serais toujours reconnaissant pour toute la confiance et l'autonomie de recherche qu'il m'a accordée, qui m'ont permis de suivre une direction novatrice au sein de son équipe de recherche et dont les principaux résultats sont présentés dans ce mémoire.

J'exprime mes plus vifs remerciements à:

Monsieur F. COTTET Professeur à l'ENSMA,

Monsieur J.-D. DECOTIGNIE, professeur à l'EPFL

Monsieur M. DIAZ, Directeur de Recherche au CNRS,

Monsieur G. JUANOLE, Professeur à l'UPS,

Monsieur C. KAISER, Professeur au CNAM

Monsieur Z. MAMMERI, Maître de Conférence à l'ENSEM,

Monsieur M. SAMAAN, ingénieur à EDF

pour l'honneur qu'il me font en participant à la Commission d'Examen et particulièrement à Messieurs F. COTTET, J.-D. DECOTIGNIE et Z. MAMMERI pour avoir accepté la lourde charge de rapporteur de ce mémoire.

J'adresse un remerciement tout particulier à Rosa CARMO et à Pedro ROSA pour leur collaboration aux travaux qui ont amené à une partie significative du chapitre IV de ce mémoire.

Mes remerciements s'étendent à l'ensemble de mes collègues, en particulier à Jean-Luc ALBACETE, Nathalie BERGE, Murilo CAMARGO, Luiz et Rosa CARMO, Luiz MARTINS, Adelardo MEDEIROS, Pedro ROSA, João Manoel SILVA et plus spécialement à Laurent GALLON et Roberto WILLRICH pour toute leur amitié.

Toute ma reconnaissance à l'ensemble du personnel technique/administratif du LAAS pour leur soutien dans la réalisation de mon travail de thèse, en particulier à tous ceux qui intègrent le service de documentation pour le magnifique service qu'ils ont toujours mis à ma disposition, et à Jean-Michel PONS du service II pour tout sa disponibilité tout au long de ces années.

Finalement, je tiens à remercier l'Université de Porto, la JNICT et le groupe OLC pour leur support financier qui m'ont permis de me consacrer totalement à ce travail de thèse.

Table de matières

Introduction

L'évolution technologique de ces derniers années, en particulier, dans les réseaux locaux de communication, a induit le grand développement de systèmes distribués temps-réel dans de nombreux domaines du secteur industriel comme la production manufacturière ou continue, les équipements embarqués, la domotique, ... Les applications réalisés sur ces systèmes sont des applications soumises le plus souvent à de fortes contraintes temporelles et donc une propriété essentielle que doit assurer le fonctionnement de ces systèmes est la *justesse temporelle* de la fourniture de résultats, de démarrage d'actions, de cessation d'actions, ... Dans ce contexte de contraintes temporelles et de distribution géographique, l'ordonnancement des interactions entre les différentes tâches distantes, qui coopèrent, est une activité essentielle, dont le bon déroulement est conditionné par le bon fonctionnement du réseau local sous-jacent.

Dans le réseau local, c'est plus précisément la sous-couche MAC, qui gère l'accès à la ressource de transmission partagée entre les tâches sur les différents sites, qui est la fonction clef pour garantir *l'attribut "temps-réel"* au fonctionnement du système. Il est important de bien noter la différence en termes de performance attendues entre un réseau local classique (sans contraintes temporelles) et un réseau local temps-réel: dans un réseau local classique, les performances s'expriment en termes de temps de réponse moyen (par exemple, temps moyen écoulé entre l'instant d'occurrence d'une requête et l'instant d'occurrence de l'indication conséquente); par contre, la notion de temps moyen est une notion qui peut être dangereuse dans un contexte réseau local temps-réel où il s'agit avant tout de *garantir des temps de réponse bornés supérieurement dans le pire cas.*

A l'instant de notre réflexion sur les principaux protocoles MAC dans les réseaux locaux temps-réel, on peut faire le double constat suivant:

- en ce qui concerne la considération des deux types de trafic temps-réel (périodique et apériodique): d'une part, on a des réseaux qui définissent des mécanismes principalement pour le trafic périodique (le trafic apériodique n'étant traité qu'en arrière plan) comme par exemple le réseau FIP [NFa, 90]; d'autre part, on a des réseaux qui définissent globalement des mécanismes pour le trafic temps-réel sans distinguer le trafic périodique ou apériodique, comme par exemple la technique du jeton temporisé [Gro, 82] et ses dérivés [ACZD, 94];
- en ce qui concerne la présentation des protocoles MAC, les présentations traditionnelles se focalisent le plus souvent sur les techniques d'accès à la ressource de transmission et passent complètement sous silence l'aspect ordonnancement du transfert des flux de

messages qui est un aspect essentiel et nécessaire pour spécifier et concevoir rationnellement des protocoles MAC temps-réel.

C'est ce double constat qui nous a amené à développer des travaux sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC des réseaux locaux temps-réel, afin de définir et de proposer une architecture de communication qui, d'une part, ordonnance et gère de manière explicite les échanges relatifs à chaque type de trafic temps-réel (périodique et apériodique) et, d'autre part, fait le meilleur effort pour le trafic non-temps-réel.

En outre, comme les architectures de communication sont des systèmes faisant intervenir des mécanismes complexes, il est absolument nécessaire de supporter toute proposition par une représentation formelle afin de procéder à des vérifications et des évaluations. C'est ce que nous avons fait sur la base du modèle "Réseau de Petri Temporisé Stochastique" qui, outre le pouvoir d'expression bien connu des Réseaux de Petri (parallélisme, synchronisation, choix) offre la possibilité de représenter des contraintes temporelles et de les analyser.

Ce travail, qui a été effectué au sein du groupe Outils et Logiciels pour la Communication (OLC) du LAAS du CNRS, s'est déroulé en partie dans le cadre d'un contrat d'étude entre le groupe ACSAR (Analyse Comportementale des Systèmes et d'Automatismes de Réseaux) de la DER (Direction des Études et Recherches) d'EDF sur la proposition de la norme ISA-SP-50 / IEC-65C pour la couche liaison de données des bus de terrain.

Le travail réalisé est présenté dans ce mémoire suivant cinq chapitres:

- le premier chapitre présente le contexte de l'étude effectuée, à la fois, en termes de la problématique des systèmes distribués temps-réel et des réseaux temps-réel et en termes de formalisme "Réseau de Petri Temporisé Stochastique", c'est-à-dire ses pouvoirs d'expression et d'analyse;
- le deuxième chapitre a pour premier objectif de présenter les principales techniques d'ordonnancement monoprocesseur de tâches temps-réel (en effet, l'ordonnancement des messages temps-réel reprend ces techniques); un deuxième objectif est d'insister sur les techniques d'ordonnancement non-préemptif car le caractère non-préemptif est une différence fondamentale entre la transmission de messages et l'exécution de tâches;
- le troisième chapitre propose une classification des principaux protocoles MAC existants, qui met en exergue l'aspect ordonnancement des messages;
- le quatrième chapitre présente un ensemble de contributions que nous pensons avoir apporté aux différentes classes de protocoles MAC, à la fois, en termes d'algorithmes d'ordonnancement, de mécanismes de protocole et de conditions d'ordonnançabilité de normes existantes;
- le cinquième chapitre présente notre proposition d'architecture pour une sous-couche MAC temps-réel, la modélisation de cette proposition au moyen du modèle "Réseau de Petri Temporisé Stochastique" et son analyse en termes d'ordonnançabilité.

Chapitre I

Contexte de l'étude

1. Introduction

L'objectif de ce chapitre est de situer le travail présenté dans ce mémoire, à la fois, au niveau des mécanismes des systèmes temps-réel plus particulièrement étudiés et, au niveau du formalisme considéré pour spécifier formellement et évaluer ces mécanismes.

Ce chapitre comprend quatre paragraphes:

- Le premier paragraphe présente des généralités sur les systèmes temps-réel;
- Les deuxième et troisième paragraphes présentent les problématiques de l'ordonnancement temps-réel des tâches et des réseaux de communication temps-réel;
- Le quatrième paragraphe est consacré à la présentation du modèle Réseaux de Petri Temporisés Stochastiques qui est le formalisme utilisé.

2. Sur les systèmes temps-réel

2.1. Présentation générale

Les systèmes temps-réel sont des systèmes qui permettent la réalisation d'applications temps-réel. Une application temps-réel est une application qui "met en oeuvre un système informatique dont le fonctionnement est assujetti à l'évolution dynamique de l'état d'un environnement (procédé) qui lui est connecté et dont il doit contrôler le comportement" [CNRS 88]. Ce système informatique est appelé le système de contrôle. L'interface entre le procédé et le système de contrôle est constitué par un ensemble de capteurs (qui fournissent, au système de contrôle, les images du procédé) et d'actionneurs (qui fournissent, au procédé, les commandes du système de contrôle).

L'exactitude d'une commande est conditionnée par deux attributs: la justesse de sa valeur et la justesse de sa date d'application au procédé. Ce deuxième attribut, c'est-à-dire le respect des contraintes (échéances) temporelles, est un aspect fondamental et spécifique aux systèmes de contrôle dans les systèmes temps-réel.

D'une manière générale, compte tenu de l'aspect "respect des contraintes (échéances) temporelles", on distingue trois grands types de systèmes temps-réel:

- Les Systèmes Temps-Réel à Contraintes Strictes (STRCS) pour lesquels le non respect des contraintes temporelles ne peut pas être toléré;
- Les Systèmes Temps-Réel à Contraintes Relatives (STRCR) qui tolèrent certains dépassements d'échéances temporelles;
- Les Systèmes Temps-Réel à Contraintes Mixtes (STRCM) qui contiennent des contraintes temporelles strictes et relatives.

Le système de contrôle consiste en un ensemble de logiciels d'application qui représentent les fonctions nécessaires à la prise en compte de l'état du procédé et à la commande du procédé.

Les logiciels d'application sont codés sous la forme d'un ensemble de tâches, dont l'exécution sur une architecture physique s'appuie sur un exécutif temps-réel.

La structure physique du système de contrôle peut être basée sur une architecture centralisée (monoprocesseur ou multiprocesseur), décentralisée (ensemble des structures centralisés autonomes, reliées par un réseau de communication) ou distribuée (ensemble d'unités de traitement reliées par un réseau de communication et qui peuvent prendre en compte n'importe quelle fonction de l'application).

Un exécutif temps-réel doit fournir, en particulier, les services suivants: ordonnancement des tâches, communication et synchronisation entre les tâches, gestion des ressources nécessaires à l'exécution des tâches.

Une vue générale d'un système temps-réel est représentée sur la figure 1.1:

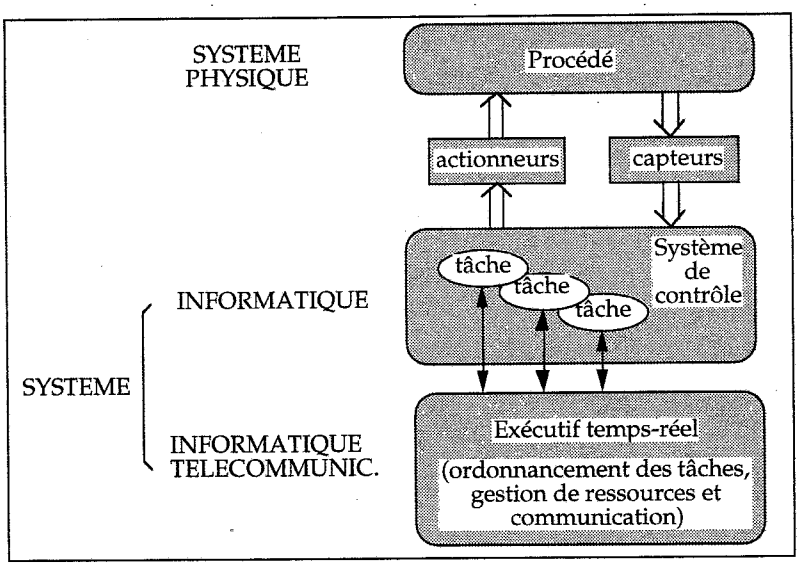

Figure 1.1: Système temps-réel

2.2. Sur les exécutifs temps-réel

Des problématiques essentielles dans les exécutifs temps-réel, en considérant des architectures de systèmes de contrôle décentralisés ou distribués, sont la mise en oeuvre, d'une part, de l'ordonnancement temps-réel des tâches et, d'autre part, de services de communication adaptés à des besoins d'échanges d'information entre tâches distantes, temporellement contraintes, qui nécessite un ordonnancement temps-réel des messages et une gestion temps-réel des communications (nous appelons "gestion des communications" tout ce qui concerne la gestion des connexions et les contrôles de flux et d'erreur).

L'aspect ordonnancement temps-réel de tâches est présent dans toutes les structures de contrôle des systèmes temps-réel. Les deux autres aspects (ordonnancement temps-réel des messages et gestion temps-réel des communications) sont, par contre, spécifiques aux deux architectures (décentralisées et distribuées) des systèmes de contrôle et représentent deux problématiques de base des réseaux de communication temps-réel. **Notre travail se situe précisément dans ces deux derniers aspects.**

3. L'ordonnancement temps-réel de tâches

3.1. Généralités

La problématique de l'ordonnancement temps-réel est la construction d'une séquence temporelle de l'ensemble des tâches à exécuter. Cette construction nécessite, d'une part, la modélisation des tâches, et d'autre part, la définition d'un algorithme d'ordonnancement. De nombreuses études ont été faites sur ces sujets: [AD, 92], [CM, 94], [Del, 94] et [Mar, 94].

Des éléments importants pour la modélisation des tâches sont les éléments suivants:

- Les tâches sont elles liées (contraintes de précédence) ou indépendantes?
- Quelles sont les caractéristiques temporelles des tâches?
- Les tâches utilisent-elles des ressources partagées?

Ici, nous ne considérons pas les contraintes de précédence et nous nous focalisons plus particulièrement sur les caractéristiques temporelles des tâches. La fonction d'un algorithme d'ordonnancement est d'utiliser la connaissance sur les tâches afin de définir la séquence temporelle d'exécution des travaux.

3.2. Les caractéristiques temporelles des tâches

Les caractéristiques temporelles des tâches s'expriment en termes de type d'activation dans la vie du système temps-réel et de type de flexibilité par rapport à leurs contraintes temporelles (échéances).

Une tâche peut être activée à intervalles réguliers (tâche périodique) ou de manière aléatoire (tâche apériodique).Une tâche peut ne pas avoir le droit de violer son échéance sous peine de mettre en péril le procédé contrôlé (tâche à contrainte stricte ou tâche critique), ou peut de temps en temps manquer son échéance sans que le procédé soit affecté (tâche à contrainte relative).

Par la suite, nous ne considérons que les tâches avec des contraintes strictes, dont nous présentons ici les modèles:

Modèle des tâches

Considérons un ensemble de n tâches, $\{\tau_1, \tau_2, \ldots \tau_i, \ldots \tau_n\}$, chaque tâche τ_i étant caractérisée par un tuplet $\{C_i, d_i, P_i, I_i\}$, $1 \le i \le n$, où:

- C_i représente la durée maximale d'exécution d'un travail de τ_i;
- d_i est l'échéance de τ_i, c'est-à-dire c'est l'intervalle maximum admissible, entre l'instant d'arrivée d'une requête d'exécution d'un travail de la tâche τ_i et l'accomplissement de ce travail

- P_i représente, soit la périodicité (pour des tâches périodiques), soit l'intervalle minimum, entre des requêtes d'exécution des travaux de τ_i (pour des tâches apériodiques);
- I_i représente la différence de phase entre la première requête d'exécution de la tâche τ_i et une référence temporelle commune à l'ensemble de tâches;

Considérons le cas des tâches périodiques: chaque tâche τ_i génère une séquence infinie de travaux. La requête du $j^{ème}$ travail est exécutée à l'instant $I_i + (j-1)P_i$ avec une échéance associée qui expire à l'instant $I_i + (j-1)P_i + d_i$. Si l'échéance d'une requête correspond à la date de réveil de la requête suivante ($d_i = P_i$) alors la tâche est dite à **échéance sur requête**; dans le cas contraire, la tâche est dite à **échéance avant requête.**

La figure 1.2 illustre le modèle des tâches.

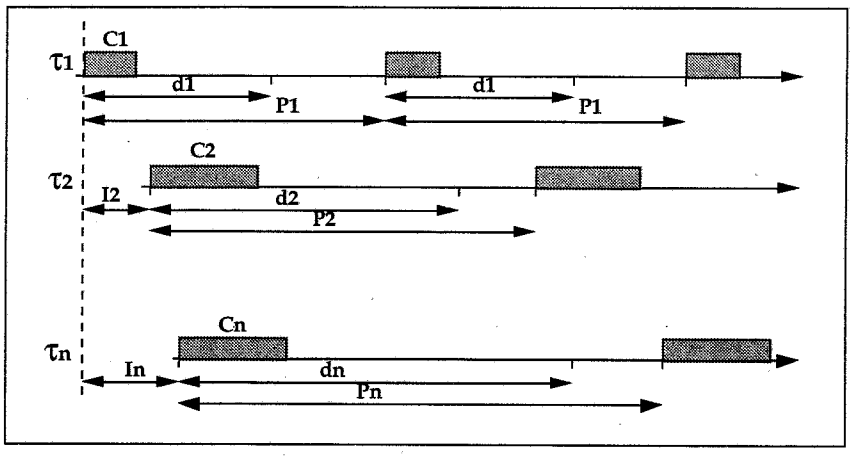

Figure 1.2: Modèle des tâches

3.3. Les principaux types d'algorithmes

Nous considérons ici les algorithmes à priorité, c'est-à-dire où une priorité qui dépend des caractéristiques temporelles est associée à chaque tâche. Une priorité peut être *statique* (elle ne varie pas au cours de la vie de la tâche) ou *dynamique* (elle évolue au cours de la vie de la tâche).

Un algorithme d'ordonnancement doit, à un instant donné, choisir la tâche la plus prioritaire à exécuter par le processeur. En ce qui concerne ce choix, on distingue deux grandes classes d'algorithmes qui diffèrent sur la connaissance que l'ordonnanceur possède sur les tâches avant de commencer l'application:

- Dans les *algorithmes hors-ligne*, toutes les tâches et leurs paramètres temporels sont supposés connus avant le démarrage de l'application. Il est alors possible de construire une séquence d'exécution de travaux et de vérifier hors-ligne que toutes les contraintes temporelles sont satisfaites. Les travaux sont exécutés dans l'ordre préétabli sans que cet ordre puisse être modifié. Cette approche ne permet donc pas de tenir en compte de l'éventuelle occurrence en-ligne de tâches inconnues au préalable.
- Dans les *algorithmes en-ligne*, seuls les paramètres temporels des tâches prêtes à un instant donné suffisent pour générer une séquence à appliquer aux instants suivants. Une tâche apériodique peut donc être prise en compte et exécutée dès son occurrence.

  Par opposition aux algorithmes hors-ligne, les algorithmes en-ligne construisent une séquence dynamiquement. Dans un contexte STRCS, où toutes les contraintes doivent être

impérativement satisfaites, il ne peut y avoir de démarrage de l'application sans certitude sur le respect de toutes les échéances. Il apparaît donc nécessaire de développer des *tests d'ordonnançabilité* pour tout algorithme en-ligne. Ces tests portent sur les caractéristiques temporelles des tâches et permettent, lorsqu'ils sont satisfaits, de garantir le respect de toutes les contraintes temporelles avant le démarrage de l'application.

En ce qui concerne le mode d'exécution des tâches, on distingue deux classes qui différent sur la notion d'interruptibilité:

- Une tâche est dite *préemptible* si, au cours de son exécution, elle peut être interrompue par une autre tâche, et reprise ultérieurement au point d'exécution précédent.
- Une tâche est dite *non-préemptible* si son exécution ne peut pas être interrompue par aucune autre tâche.

4. Sur les réseaux de communication temps-réel

4.1. Architecture

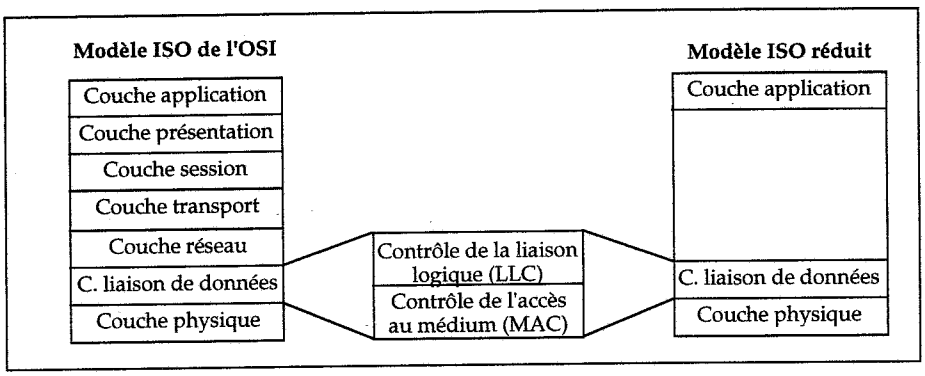

Figure 1.3: Le modèle de référence ISO de l'OSI

L'architecture la plus générale d'un réseau de communication est une architecture à sept couches basée sur le modèle ISO de l'OSI [ISO, 84]. Cependant, dans le contexte des applications industrielles, on n'a pas besoin des sept couches, et on trouve de nombreux réseaux, comme les réseaux de terrain [DP, 93] qui ont une architecture à trois couches (figure 1.3). C'est cette architecture ISO réduite que nous considérons.

La sous-couche MAC ("Medium Access Control"), qui gère l'accès à la ressource de transmission (ressource commune à toutes les stations, et donc à toutes les tâches), c'est-à-dire qui réalise à la fois l'ordonnancement des messages et le protocole d'accès (protocole MAC), est la fonction essentielle pour permettre le déroulement temps-réel des applications.

Nous indiquons maintenant quelques notions de base relativement à l'ordonnancement temps-réel des messages.

4.2. Ordonnancement temps-réel de messages

Les messages temps-réel sont conséquents de l'exécution des tâches temps-réel et donc la présentation de leur ordonnancement peut être faite en s'inspirant de la présentation de l'ordonnancement des tâches. Nous ne considérons pas non plus les aspects de précédence de messages.

Nous considérons donc des messages indépendants:

1- qui ont des caractéristiques temporelles (messages périodiques et apériodiques à contraintes strictes;

2- qui se partagent la ressource de transmission.

Compte tenu de la caractéristique 1, nous pouvons, comme pour les tâches, définir des algorithmes en-ligne ou hors-ligne avec des priorités statiques ou dynamiques.

En ce qui concerne la caractéristique 2, elle amène, d'une part, à faire apparaître la différence fondamentale entre l'ordonnancement des tâches et l'ordonnancement des messages et, d'autre part, à situer la problématique de l'ordonnancement des messages dans le contexte des études sur l'ordonnancement des tâches:

- Différence fondamentale: alors que l'exécution d'une tâche peut être habituellement interrompue au bénéfice d'une tâche plus prioritaire, la transmission d'un message ne peut pas être interrompue (un message est transmis entièrement ou pas du tout). Donc l'ordonnancement des messages doit être basé sur des algorithmes *non-préemptifs*.
- Situation de la problématique de l'ordonnancement des messages: comme toutes les stations et donc toutes les tâches utilisent la même ressource de transmission (une seule ressource), l'ordonnancement des messages à travers cette ressource, s'assimile à un *ordonnancement monoprocesseur* [Mam, 95].

4.3. Modélisation de messages au niveau du service MAC

La présentation et l'étude des protocoles MAC temps-réel nécessite de spécifier les caractéristiques temporelles des unités de données de service (SDUs), que nous appelons messages, du service MAC.

D'une manière générale, les messages venant des applications et soumis au service MAC peuvent être classés soit comme des messages temps-réel (contraintes strictes associées au transfert), soit comme des messages non-temps-réel (pas de contraintes associées au transfert). En effet, concernant cette dernière catégorie, on peut toujours admettre qu'un réseau, qui doit principalement transmettre du trafic temps-réel, a également à transmettre du trafic non contraint temporellement.

Nous modélisons donc les messages au niveau du service MAC d'un réseau, qui comporte m stations de la manière suivante:

- Pour les messages temps-réel nous distinguons deux cas:
  - Soit nous divisons les messages temps-réel en deux sous-ensembles: un sous-ensemble $Mp = \{Mp_1, \ldots Mp_i, \ldots Mp_p\}$, comportant p flux de messages **p**ériodiques et un sous-ensemble $Ma = \{Ma_1, \ldots Ma_j, \ldots Ma_s\}$, comportant s flux de messages **a**périodiques à contraintes strictes.
  - Soit, nous ne considérons qu'un seul ensemble de messages temps-réel, qui englobe à la fois les messages périodiques et les messages apériodiques à contraintes strictes: $M = \{M_1, \ldots M_i, \ldots M_n\}$; ce groupement peut être fait dans la mesure où on définit pour les flux apériodiques un temps minimal entre deux arrivées consécutives (ce temps minimal peut être vu comme une période).
- Pour les messages non-temps-réel, nous définissons un ensemble N comportant m flux de messages: $N = \{N_1, \ldots N_k, \ldots N_m\}$.

De plus, nous définissons $nœud_k$ comme l'ensemble de flux de messages qui appartiennent à une station k.

Il nous parait intéressant de distinguer le nombre de flux de messages temps-réel, qu'ils soient périodiques (p) ou apériodiques (s) du nombre des stations (m). En effet, on peut avoir plusieurs flux par station avec contraintes temporelles différentes et donc on doit particulariser l'accès au réseau de chaque flux. Par contre, les messages non-temps-réel n'ayant pas (par définition) des contraintes temporelles, nous les regroupons dans un seul flux par station.

Nous présentons maintenant les caractéristiques importantes des messages temps-réel vues du côté de l'ordonnanceur:

- P_i (P_j) est la plus petite valeur de temps séparant deux requêtes consécutives de messages d'un flux temps-réel Mp_i (Ma_j) pour leur ordonnancement. Dans le cas particulier des messages périodiques, P_i est la période;
- C_i (C_j) est la durée maximale de l'utilisation de la ressource de transmission pour transférer un message d'un flux temps-réel Mp_i (Ma_j).
- d_i (d_j) est l'échéance, c'est-à-dire l'intervalle de temps maximum entre l'instant d'arrivée d'une requête d'un message d'un flux temps-réel Mp_i (Ma_j) et l'instant où l'utilisation de la ressource de transmission pour transférer ce message doit être terminée; on peut avoir une *échéance avant requête*, c'est-à-dire $d_i < P_i$ ($d_j < P_j$) ou *sur requête*, c'est-à-dire $d_i = P_i$ ($d_j = P_j$).
- I_i (I_j) est la différence de phase entre la première arrivée d'une requête d'un message du flux Mp_i (Ma_j) et une référence temporelle commune à l'ensemble des flux de messages.

5. Sur les Réseaux de Petri Temporisés Stochastiques

Le modèle "Réseaux de Petri Temporisés Stochastiques" (RdPTS) [ACJV, 94] [Atm, 94] est un modèle temps réel basé sur le concept de temps physique quantifié.

L'objectif du modèle RdPTS est, d'une part, de modéliser les mécanismes fondamentaux des systèmes distribués temps critique (parallélisme, synchronisation, contraintes temporelles) et, d'autre part, de fournir un cadre unique pour permettre des analyses qualitatives (logique des mécanismes) et quantitatives (évaluation des performances fonctionnelles et de sûreté de fonctionnement).

3.1. Définition formelle

Un RdPTS est un triplet $\langle PN, IO, FO \rangle$ où:

- PN: réseau de Petri sous-jacent (RdP place-transition, avec des arcs inhibiteurs);
- IO est la fonction intervalle de tir initial. A chaque transition t_i est associé un intervalle $[\theta_{mi}, \theta_{Mi}]$. Nous avons: $0 \leq \theta_{mi} \leq \theta_{Mi} \leq \infty$; θ_{mi} étant la date de tir au plus tôt et θ_{Mi} étant la date de tir au plus tard. θ_{mi} et θ_{Mi} sont relatifs à l'instant de sensibilisation de la transition. Le tir est instantané.
- FO est la fonction de densité de probabilité de tir initiale. A chaque transition t_i est associée une fonction de densité de probabilité $f_i(x)$ définie sur l'intervalle de tir de la transition. Nous avons: $\int_{\theta_{mi}}^{\theta_{Mi}} f_i(x)dx = 1$.

Une fonction de densité de probabilité peut être: continue (uniforme, exponentielle), discrète (impulsion de Dirac) ou mixte (uniforme et discrète). Les densités de probabilités discrètes, uniformes et mixtes permettent la représentation de contraintes temporelles.

3.2. Analyse

L'analyse du comportement dynamique des réseaux de Petri temporisés stochastiques est effectuée par une analyse d'accessibilité basée sur l'objet "graphe d'états probabilisé". Ce graphe permet à la fois des analyses qualitatives et quantitatives.

3.2.1. Le graphe d'états probabilisé.

Ce graphe est composé d'états et de transitions entre états :

- un état est un triplet $\langle M, I, F \rangle$ où M est le marquage, I l'ensemble des intervalles de tir des transitions sensibilisées par le marquage M et F l'ensemble des fonctions de densité de probabilité associées aux intervalles de tir.
- une transition entre deux états est un triplet $< t_i, p_i, \theta_i >$ où t_i est la transition du réseau de Petri sous-jacent dont le tir provoque le changement d'état, p_i la probabilité de branchement associée, et θ_i le temps moyen de tir ou temps conditionnel de séjour dans l'état de départ de la transition. Les valeurs p_i et θ_i dépendent de l'intervalle de tir, de la fonction de densité de probabilité de la transition t_i, et des transitions sensibilisées simultanément (s'il y en a).

Les conditions de tir d'une transition t_i à un instant θ_i avec une probabilité p_i sont :

- t_i sensibilisée dans le réseau de Petri sous-jacent ;
- θ_i compris entre la borne minimale de l'intervalle de tir associé à la transition t_i et la plus petite valeur des bornes maximales des intervalles de tir de toutes les transitions sensibilisées;
- la probabilité de tir p_i entre la borne minimale de l'intervalle de tir de t_i et la plus petite valeur des bornes maximales des intervalles de tir de toutes les transitions sensibilisées, est non nulle.

Notons que, compte tenu de la règle de tir des transitions à un temps moyen, si on considère les différents types de densité de probabilités, on obtient un graphe qui ne représente que de manière approchée, dans le cas le plus général, le comportement du système modélisé [Atm, 94]. On a, par contre, les cas particuliers suivants:

- avec uniquement des distributions déterministes, on a une représentation exhaustive (qualitativement et quantitativement);
- avec uniquement des distributions exponentielles, on a une représentation exhaustive du point de vue qualitatif (du point de vue quantitatif on a des valeurs moyennes);
- avec des distributions déterministes et exponentielles, on a une représentation exhaustive du point de vue qualitatif (du point de vue quantitatif on a là aussi, des valeurs moyennes pour les activités décrites avec des distributions exponentielles).

Dans le cadre de l'analyse effectuée dans ce travail, nous ne considérons que des transitions déterministes.

3.2.2. Exploitation du graphe d'états probabilisé

L'exploitation peut être faite à deux niveaux complémentaires:

PREMIER NIVEAU: on ne considère pas les probabilités p_i et les temps θ_i associés aux transitions mais on considère la sémantique associée aux transitions (type d'action, envoi de message, réception de message, ...); on peut à l'aide de ce graphe faire une analyse qualitative du comportement du système étudié comme on le fait avec les systèmes de transitions étiquetés classiques (seulement ici, il y a une différence très importante, la structure du graphe obtenu dépend des contraintes temporelles); L'analyse qualitative permet d'analyser deux types de propriétés:

- propriétés générales (caractère borné, vivant, présence de cycles, ...);
- propriétés spécifiques, c'est-à-dire qui dépendent de la mission du système modélisé; ces propriétés sont obtenues à partir d'une interprétation de la sémantique des places et des transitions ; en particulier, on peut :
  - identifier des séquences d'événements et établir des relations entre deux événements sous la forme de formules de logique temporelle: il est possible ..., il est inévitable...;
  - obtenir des vues abstraites au moyen d'un automate quotient qui représente le comportement du système par rapport à un sous-ensemble d'événements appelés événements observables (on peut utiliser, en particulier, les techniques de projection basées sur la relation d'équivalence observationnelle [Mil, 80]).

DEUXIÈME NIVEAU: on considère maintenant les valeurs des probabilités p_i et des temps θ_i ce qui nous permet de faire une analyse quantitative. L'analyse quantitative peut être une analyse en régime transitoire et/ou en régime permanent. L'analyse en régime transitoire concerne, en particulier, l'évaluation de la probabilité de chemins de séquences d'événements. Si le graphe d'états probabilisé a une composante terminale fortement connexe, l'analyse en régime permanent peut être effectuée à partir des deux matrices P (matrice des probabilités de transitions) et Θ (matrice des temps de transitions):

- le vecteur des probabilités à l'équilibre γ est obtenu à partir de la matrice P:

$$\gamma \cdot P = \gamma, \text{ avec } \sum \gamma_i = 1 \tag{1.1}$$

- les temps inconditionnels de séjour (η_i) sont obtenus en combinant les matrices P et Θ;
- les probabilités en régime permanent π_i des états sont calculées à partir du vecteur des probabilités à l'équilibre γ et des temps inconditionnels de séjour η_i :

$$\pi_i = \frac{\eta_i \gamma_i}{\sum_i \eta_i \gamma_i}$$

3.3. L'outil RdPTS

L'outil RdPTS [Atm, 93] permet d'une part, l'édition du modèle RdPTS et d'autre part, la construction du graphe d'états probabilisé et l'obtention de vues abstraites quantitatives pour effectuer des analyses qualitatives et quantitatives. Il a été développé en langage C, au LAAS et tourne sous SunOS 4 et Solaris 2.

Nous utilisons cet outil pour effectuer au chapitre V la modélisation et l'analyse d'une architecture de communication temps-réel.

Conclusion

Trois points nous paraissent devoir être soulignés:

- nous avons identifié, d'une part, les problématiques de base des systèmes distribués temps-réel, à savoir l'ordonnancement de l'exécution des tâches, l'ordonnancement du transfert de messages et la gestion des connexions, et, d'autre part, une différence fondamentale entre l'ordonnancement de l'exécution de tâches et du transfert des messages (ce dernier n'est pas préemptif);
- nous avons montré la nécessite de disposer d'un modèle temporel des tâches et des messages; ce modèle est essentiel pour exprimer les contraintes temporelles;
- nous avons montré l'intérêt du modèle "Réseau de Petri Temporisé Stochastique", à la fois, en termes de pouvoir d'expression (pour des systèmes distribués avec des contraintes temporelle) et de pouvoir d'analyse.

Références

[ACJV, 94] Y. Atamna, R. Carmo, G. Juanole, F. Vasques; "Le Modèle "Réseaux de Petri Temporisés Stochastiques" (RdPTS) pour l'Analyse Qualitative et/ou Quantitative des Systèmes Distribués Temps-Réel", in Proc. of Real-Time Systems Conference, Paris, 12-15 Janvier, 1993.

[ACZD, 94] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol; in IEEE Transactions on Computers, vol. 43, no 3, pages 327-339, March 1994;

[AD, 92] M. Alabau, T. Dechaize; Ordonnancement temps-réel par échéance; TSI, n° 11(3), 1992, pp. 59-123.

[Atm, 93] Y. Atamna "RdPTS, a tool for Stocastic Timed Petri Nets", in Proc. of Workshop on Petri Nets and Performance Models, Tools Exhibition (PNPM'93), Toulouse, Francs, October 1993.

[Atm, 94] Y. Atamna "Réseaux de Petri temporisés stochastiques classiques et bien formés: Définition, analyse et application aux systèmes distribués temps réel.", Doctorat de l'Université Paul Sabatier de Toulouse, Octobre 1994.

[CM, 94] C. Cardeira, Z. Mammeri; Ordonnancement des tâches dans les systèmes temps-réel et répartis: Algorithmes et critères de classification; APII n° 28(4), 1994, pp 353-384.

[CNRS, 88] CNRS: Groupe de refléxition temps-réel du CNRS, "Le temps-réel"; TSI, n° 7(5), 1988, pp 493-500

[DP, 93] J.-D. Decotignie, P. Pleineveaux; A Survey on Industrial Communication Networks; Ann. Télécommunications, 48 (9-10), 1993

[Del, 94] J. Delacroix; Un contrôleur d'ordonnancement temps-réel pour la stabilité de earliest deadline en surcharge: le régisseur; Thèse de Doctorat, CNAM, Paris, 1994.

[Gro, 82] R. Grow; A Timed Token Protocol for Local Area Networks; IEEE Electro'82, Token Access Protocols, 17/3, 1982

[ISO, 84] ISO 7498; Information Processing Systems, Open Systems Interconnection, Basic Reference Model, 1984.

[Mam, 95] Z. Mammeri; Gestion des Contraintes Temporelles dans les Systèmes Temps-Réel et Répartis; Thèse de Habilitation à Diriger des Recherches, Institut National Politechnique de Lorraine, 1995

[Mar, 94] P. Martineau; Ordonnancement en-ligne dans les systèmes informatiques temps-réel; Thèse de Doctorat (n° ED 82-93), Ecole Centrale de Nantes, 21 Octobre 1994.

[Mil, 80] R. Milner; A Calculus of Communicating Systems; in Lecture Notes of Computer Science, n° 92, Springer-Verlag.

[NFa, 90] FIP Bus for Exchange of Information between Transmitters, Actuators and Programmable Controllers, Data Link Layer; NF C 46-603, June 1990

Chapitre II

Ordonnancement monoprocesseur de tâches temps-réel: état de l'art et contributions

1. Introduction

Le premier objectif de ce chapitre est, tout d'abord de présenter des algorithmes qui garantissent le respect des contraintes temporelles au moyen d'une fonction de garantie ou *test d'ordonnançabilité*. Le test d'ordonnançabilité doit indiquer, par un calcul si possible simple et rapide, si l'ensemble des tâches peut être ordonnancé par l'algorithme et ce sans avoir besoin d'exécuter l'algorithme.

Il est plus particulièrement essentiel:

- d'une part, de rappeler les principaux algorithmes préemptifs optimaux au sens de la garantie des contraintes temporelles (un algorithme est dit *optimal*, pour des hypothèses données, si aucun autre algorithme (pour les mêmes hypothèses) n'est capable d'ordonnancer une configuration de tâches qui ne soit pas ordonnançable par cet algorithme); ces algorithmes sont les algorithmes les plus utilisés et sont la base de toutes les études actuelles sur l'ordonnancement.
- d'autre part, de présenter les principaux travaux existants, à notre connaissance, sur les algorithmes non-préemptifs. Dans le contexte de notre travail sur l'ordonnancement de messages temps-réel à travers une ressource de transmission (la transmission est non-préemptive par définition), ces algorithmes sont à considérer très particulièrement; notons que les algorithmes non-préemptifs présentés sont des versions non-préemptives des algorithmes préemptifs (d'où l'intérêt d'avoir présenté au préalable les algorithmes préemptifs).

Un second objectif de ce chapitre est précisément de présenter les travaux que nous avons faits sur les algorithmes non-préemptifs.

En conséquence, ce chapitre comprend trois paragraphes:

- dans un premier paragraphe, nous résumons l'état-de-l'art sur les principaux algorithmes préemptifs, c'est-à-dire, l'ordonnancement de tâches périodiques, l'ordonnancement conjoint de tâches périodiques et apériodiques, ainsi que la

problématique du partage de ressources (le problème des surcharges de fonctionnement n'est pas considéré);

- dans un deuxième paragraphe, nous présentons les principaux travaux sur les algorithmes non-préemptifs;
- dans un troisième paragraphe, nous présentons notre contribution au domaine des algorithmes non-préemptifs;

2. Algorithmes préemptifs

2.1. Tâches périodiques

Les algorithmes présentés sont basés sur la considération du cas pire, c'est-à-dire, ils évaluent l'ordonnançabilité d'un ensemble de tâches périodiques à un *instant critique*, un instant critique étant un instant où le temps de réponse de l'ordonnanceur à la demande d'exécution de toutes les tâches est le plus long.

L'instant critique d'une tâche [LL, 73] est l'instant où la demande d'exécution d'une tâche arrive simultanément avec la demande d'exécution de toutes les tâches de priorité supérieure. Le cas pire est donc considéré si on suppose que les déphasages entre les tâches sont nuls: il suffit alors de vérifier si la première demande de chaque tâche est exécutée avant son échéance.

Nous résumons successivement les cas des algorithmes (c.f. chapitre I) à échéance sur requête ($d_i = P_i$) et à échéance avant requête ($d_i < P_i$).

2.1.1. Échéance sur requête

Les travaux fondamentaux sur les algorithmes à échéance sur requête ont été faits par Liu et Layland [LL, 73] et concernent les algorithmes "Rate Monotonic" (RM) et "Earliest Deadline" (ED) dont nous présentons maintenant les éléments principaux.

1. Algorithme RM

C'est un algorithme qui est basé sur des priorités statiques (une tâche a une priorité fixe qui est inversement proportionnelle à sa période) et qui est optimal.

Test d'ordonnançabilité

- **Condition suffisante [LL, 73]:**

  L'ordonnancement RM d'un ensemble de n tâches périodiques $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$ est faisable si le facteur d'utilisation $U = \sum_{i=1}^{n} C_i/P_i$ vérifie la relation suivante:

  $$U \leq n \cdot \left(\sqrt[n]{2} - 1\right) \tag{2.1}$$

  Compte tenu que cette condition n'est que suffisante, il ne faut pas écarter des ensembles de tâches qui ne satisfont pas ce test. Des travaux [LSD, 89] ont présenté une condition nécessaire et suffisante d'ordonnançabilité.

- **Condition nécessaire et suffisante [JP, 86]:**

  Considérons un ensemble de n tâches périodiques $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$ ordonnancé par RM; sachant que le temps de réponse d'une tâche τ_i est borné supérieurement par

$RT_i = \left(\sum_{j=1}^{i-1} \lceil RT_i/P_j \rceil \cdot C_j\right) + C_i$, l'ordonnancement de l'ensemble τ est faisable si et seulement si:

$$\forall_i,\ 1 \le i \le n,\ RT_i \le P_i \tag{2.2}$$

Exemple

Considérons un ensemble de 3 tâches $\tau = \{\tau_1, \tau_2, \tau_3\}$, avec $(C_i, P_i) = \{(1,5),(3,6),(5,100)\}$, à ordonnancer par l'algorithme RM. Comme la condition suffisante d'ordonnançabilité est satisfaite $\left(U = 0.75 < 3 \cdot \left(\sqrt[3]{2} - 1\right) = 0.78\right)$, l'ensemble de tâches est ordonnançable. La séquence temporelle d'exécution des tâches après l'instant critique est représentée sur la figure 2.1.

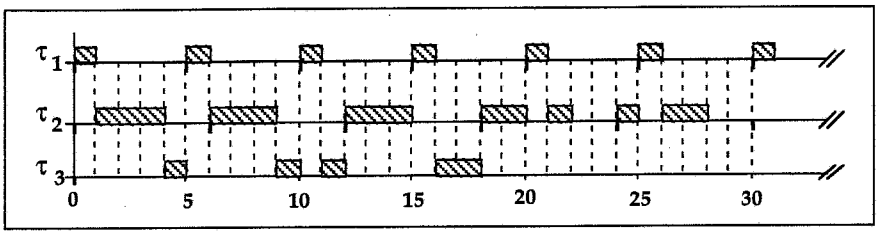

Figure 2.1: Exemple d'ordonnancement RM

La tâche τ_1 est toujours exécutée au début de la période, car cette tâche est toujours la tâche la plus prioritaire (algorithme à priorité statique). La tâche moins prioritaire (τ_3) est toujours préemptée pour permettre l'exécution d'une tâche plus prioritaire (τ_1, τ_2).

2. Algorithme ED

C'est un algorithme qui est basé sur des priorités dynamiques (à toute instant, la priorité est inversement proportionnelle à la date de l'échéance) et qui est optimal. L'algorithme ED permet d'accepter un plus grand nombre de configurations de tâches que l'algorithme RM.

Test d'ordonnançabilité (condition nécessaire et suffisante)

L'ordonnancement ED d'un ensemble de n tâches périodiques $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$ est faisable si et seulement si le facteur d'utilisation $U = \sum_{i=1}^{n} C_i/P_i$ vérifie la relation suivante:

$$U \le 1 \tag{2.3}$$

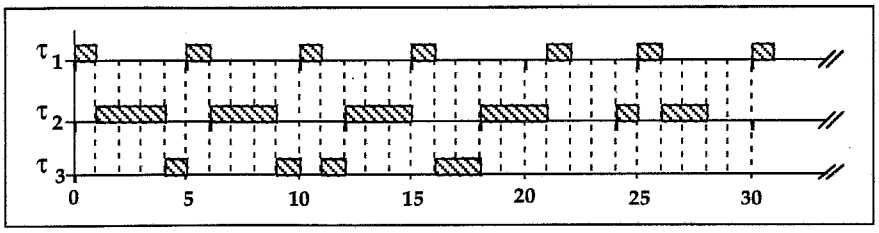

Figure 2.2: Exemple d'ordonnancement ED

Exemple

Considérons le même ensemble de *3* tâches périodiques: $(C_i, P_i) = \{(1,5),(3,6),(5,100)\}$, dans le cas d'un ordonnancement ED. Comme le test d'ordonnançabilité est vérifié, l'ensemble des tâches est ordonnançable ($U = 0.75 \leq 1$). La séquence temporelle d'exécution des tâches après l'instant critique est représentée dans la figure 2.2.

Dans ce cas, comme l'algorithme ED est un algorithme à priorité dynamique, la tâche τ_1 n'est pas toujours la plus prioritaire, et donc ses travaux ne sont plus systématiquement exécutés au début de la période. A chaque instant, le travail en attente à exécuter sera celui dont la date d'échéance est la plus proche.

2.1.2. Échéance avant requête

Un des inconvénients de l'algorithme RM est celui de considérer les tâches à échéance sur requête, c'est-à-dire, $d_i = P_i$.

Il s'avère intéressant de considérer aussi le cas des tâches avec des échéances inférieures à la période: $d_i < P_i$. Supposons par exemple un système avec une tâche déclenchée toutes les 10s qui doit se terminer dans les 10ms qui suivent. Un ordonnancement qui utilise l'algorithme RM ou ED est complètement inefficace, parce que l'accomplissement de l'exécution des travaux de la tâche n'est garanti qu'avant la fin de sa période, c'est-à-dire 10s.

1. Algorithme "Deadline Monotonic" (DM)

C'est un algorithme qui est basé sur des priorités statiques (une tâche a une priorité fixe qui est inversement proportionnelle à son échéance). Cet algorithme est équivalent à l'algorithme RM dans le cas de tâches à échéance sur requête.

Cet algorithme n'a vraiment été utilisé que depuis qu'une condition suffisante d'ordonnançabilité a été présentée dans [AB, 90].

Test d'ordonnançabilité (condition suffisante)

> L'ordonnancement DM d'un ensemble de n tâches périodiques $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$ est faisable si:
>
> $$\forall i: 1 \leq i \leq n: \left\{ C_i + \sum_{j=1}^{i-1} \lceil d_i / P_j \rceil \cdot C_j \leq d_i \right\} \qquad (2.4)$$

Dans ce test d'ordonnançabilité, $\left\{ \sum_{j=1}^{i-1} \lceil d_i / P_j \rceil \cdot C_j \right\}$ est une mesure de l'interférence des processus de plus haute priorité dans l'exécution de τ_i. Ce test énonce que pour une tâche τ_i, la somme de la durée de son travail, plus l'interférence dans son exécution qui lui est imposée par les travaux des tâches de priorité supérieures, ne doit jamais dépasser son échéance.

La condition énoncée est suffisante mais pas nécessaire, parce que la mesure de l'interférence effectuée est pessimiste dans le sens qu'elle ne prend pas en compte les instants de déclenchement des différentes tâches.

2. Algorithme "Earliest Deadline" (ED)

Il a été montré [LM, 80] qu'une configuration arbitraire de tâches périodiques, avec échéance avant requête, est ordonnançable par l'algorithme ED, si:

$$\sum_{i=1}^{n} C_i / d_i \leq 1 \qquad (2.5)$$

Cette condition est une condition suffisante (mais non nécéssaire) d'ordonnançabilité.

2.1.3. Ordonnancement des tâches apériodiques

On dispose d'un ensemble de tâches périodiques qui ont leurs échéances garanties (c.f.: II.2.1.1 et II.2.1.2). Le problème est d'ordonnancer des tâches apériodiques sans remettre en cause l'ordonnançabilité des tâches périodiques:

- pour les tâches apériodiques à contraintes relatives il s'agit de minimiser le temps de réponse;
- pour les tâches apériodiques à contraintes strictes il s'agit de maximiser le nombre de tâches qui peuvent être acceptées;

Nous résumons les principales techniques [Mar, 94] [Del, 94]:

1. Tâches apériodiques à contraintes relatives

Les techniques les plus importantes sont la technique du *serveur de tâches* et la technique de *l'ordonnancement conjoint*. Les tâches périodiques sont ordonnancées avec l'algorithme RM, quand on utilise la technique du serveur des tâches, et avec l'algorithme ED quand on utilise la technique de l'ordonnancement conjoint.

Le serveur des tâches

Le serveur est une tâche périodique, généralement de la plus haute priorité, qui est créé pour ordonnancer les tâches apériodiques.

On a plusieurs types de serveurs qui diffèrent par la manière dont ils répondent aux arrivées des travaux: traitement par scrutation, serveur ajournable, serveur sporadique (on peut se reporter aux références [LSS, 87] et [SSL, 89])

L'ordonnancement conjoint

L'idée est de transformer les tâches apériodiques à contraintes relatives en tâches apériodiques à contraintes strictes, en ajoutant à leurs paramètres temporels une échéance fictive. Cette échéance fictive est calculée chaque fois qu'un travail d'une tâche apériodique survient et correspond à l'échéance minimale que l'on doit associer à la tâche pour qu'elle s'exécute avec un temps de réponse minimal.

L'échéance fictive est la première date pour laquelle le temps total d'inactivité du processeur, qui doit ordonnancer, dans le respect de leurs échéances, les travaux des tâches périodiques et tâches apériodiques précédemment déclenchés et non encore achevés, est égal au temps d'exécution de la tâche [CC, 89].

2. Tâches apériodiques à contraintes strictes

On a deux techniques principales: la technique de la *pseudo-période* et la technique du *test de garantie.*

Pseudo-période

On associe aux tâches apériodiques une pseudo-période qui représente l'intervalle minimal entre deux occurrences successives d'un travail d'une tâche apériodique. On ne travaille donc qu'avec un seul modèle de tâches (tâches périodiques) que l'on sait bien traiter avec les algorithmes RM ou ED. Cette technique induit cependant une sous utilisation du processeur (du fait du choix de l'intervalle minimal comme pseudo-période).

Test de garantie

Cette technique, contrairement à la précédente, ne fait aucune supposition sur la fréquence d'arrivée des tâches qui peuvent survenir à tout instant. Plusieurs modalités peuvent être envisagées. Citons, en particulier, la modalité où on teste si la nouvelle requête apériodique peut être placée dans le temps creux d'une configuration périodique ordonnancée selon l'algorithme ED et dans le respect des échéances des tâches périodiques et des tâches apériodiques précédemment acceptées [CC, 89].

2.1.4. Partage de ressources

Le partage de ressources en mutuelle exclusion peut engendrer des inversions de priorités (une tâche utilisant une ressource bloque momentanément une tâche plus prioritaire qui voudrait utiliser la ressource) ou des interblocages. Ceci se produit lorsque l'ordonnancement des tâches pour l'accès au processeur est explicitement distinct de l'ordonnancement pour l'accès aux ressources. Afin de limiter les deux phénomènes précités, des techniques qui considèrent de façon globale l'ordonnancement pour l'accès au processeur et aux ressources ont été introduites: les techniques par héritage [SRL, 90]. On distingue principalement la technique à priorité hérité et la technique à priorité plafond.

1. La priorité héritée

Principe

Lorsqu'une tâche de faible priorité τ_{k+1} bloque, pour l'utilisation d'une ressource, des tâches de plus forte priorité $\tau_1,\dots\tau_k$, alors, le travail de la tâche τ_{k+1} s'exécute avec une priorité égale à $max\{p(\tau_1), p(\tau_2),\dots p(\tau_k)\}$, où $p(\tau_i)$ représente la priorité de τ_i. Cette technique limite la durée des inversions de priorité mais n'interdit pas les interblocages.

Test d'ordonnançabilité

En considérant l'algorithme RM, la prise en compte du partage de ressources entraîne des modifications par rapport au test d'ordonnançabilité de cet algorithme (paragraphe 2.1.1) et on ne peut plus trouver de condition nécessaire et suffisante. Ce test peut être étendu de deux façons:

- La première méthode consiste à vérifier que chaque tâche peut respecter son échéance même si elle est bloquée par des tâches de priorité plus faible. Dans ce cas, l'ordonnancement d'un ensemble de n tâches périodiques $\tau=\{\tau_1,\dots\tau_i,\dots\tau_n\}$ en utilisant la technique à priorité héritée, est faisable si:

$$\forall i,\ 1\le i\le n,\ \sum_{j=1}^{i} C_j/P_j + B_i/P_i \le i\cdot\left(\sqrt[i]{2}-1\right) \qquad (2.6)$$

  où B_i est la durée maximale de blocage de la tâche τ_i par les tâches de priorité inférieure ($\tau_{i+1},\dots\tau_n$). Le calcul de B_i est présenté dans [SRL, 90].

- La seconde méthode est plus simple à mettre en oeuvre, mais plus restrictive. Dans ce cas, l'ordonnancement d'un ensemble de n tâches périodiques $\tau=\{\tau_1,\dots\tau_i,\dots\tau_n\}$ en utilisant la technique à priorité héritée, est faisable si:

$$\sum_{i=1}^{n} C_i/P_i + \max_{1\le i\le n}\{B_i/P_i\} \le n\cdot\left(\sqrt[n]{2}-1\right) \qquad (2.7)$$

  Le calcul de B_i est le même que précédemment.

2. La priorité plafond

Principe

La technique à priorité héritée ne limite pas suffisamment la durée des inversions de priorité (toute tâche peut être bloquée par toutes les tâches de priorité inférieure et par le verrouillage de tous les sémaphores auxquels elle accède) et n'empêche pas les interblocages. C'est pourquoi la technique à priorité plafond a été définie: dans cette technique, une tâche, pour entrer en section critique, doit avoir une priorité strictement supérieure aux "valeurs plafonds" des sémaphores verrouillés par les autre tâches. Une "valeur plafond" est la "valeur maximale" parmi les priorités des tâches qui pourraient utiliser la ressource.

En conséquence, une tâche τ peut préempter une tâche τ' en section critique si et seulement si sa priorité est supérieure aux "valeurs plafonds" (ce qui veut dire que cette tâche τ accède à des ressources auxquelles la tâche τ' n'accède pas et que sa priorité est supérieure aux valeurs plafonds des sémaphores verrouillés par la tâche τ').

Test d'ordonnançabilité

- En considérant l'algorithme RM, les tests présentés pour la technique à priorité héritée peuvent être considérés (le calcul de B_i peut être affiné [Mar, 94]);
- En considérant l'algorithme ED, une condition suffisante d'ordonnançabilité est la suivante [Bak, 91]:

$$\forall_{1 \le k \le n} \left(\sum_{i=1}^{n} \frac{C_i}{P_i} \right) + \frac{B_k}{P_k} \le 1 \qquad (2.8)$$

3. Algorithmes non-préemptifs: l'algorithme ED non-préemptif

Peu de travaux ont été faits sur les algorithmes non-préemptifs. En effet, dans un contexte monoprocesseur, même si un ordonnancement non-préemptif est plus simple à mettre en oeuvre qu'un ordonnancement préemptif (la préemption induit de fréquents changements de contexte d'exécution), on n'atteint pas cependant les mêmes taux d'ordonnançabilité à cause des inversions de priorités.

Un travail intéressant sur l'algorithme ED non-préemptif a cependant été fait. C'est le résumé de ce travail que nous présentons maintenant.

Certains chercheurs [JSM, 91] ont abordé le problème de l'ordonnancement non-préemptif de tâches périodiques et apériodiques à contraintes strictes. Ils ont tout d'abord, d'une part, défini des conditions nécessaires d'ordonnançabilité des tâches périodiques et, d'autre part, montré que ces conditions nécessaires suffisent également à l'ordonnançabilité de tâches apériodiques à contraintes strictes. Ils ont enfin démontré que si une tâche ne respecte pas son échéance, une des conditions nécessaires d'ordonnançabilité a été violée.

Nous présentons maintenant les résultats d'une analyse qui permet l'obtention des conditions nécessaires d'ordonnançabilité définies dans [JSM, 91]. Considérons un ensemble de tâches périodiques $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$, avec $\tau_i = \{C_i, P_i\}$ et $P_i \ge P_j$ si $i > j$, à échéance sur requête, et supposons un temps discret pour l'occurrence des requêtes des tâches.

L'obtention des conditions d'ordonnançabilité est basée:

- d'une part, sur la définition du pire cas de déphasage d'une tâche τ_i;

- d'autre part, dans l'hypothèse de ce pire cas de déphasage d'une tâche τ_i, sur l'évaluation de la charge de travail du processeur.

3.1. Pire cas de déphasage d'une tâche τ_i

Définition: On est dans le pire cas de déphasage d'une tâche τ_i (figure 2.3) lorsque la requête de la tâche τ_i se produit une unité de temps avant l'arrivée des requêtes des tâches plus prioritaires ($\tau_1, \ldots \tau_{i-1}$).

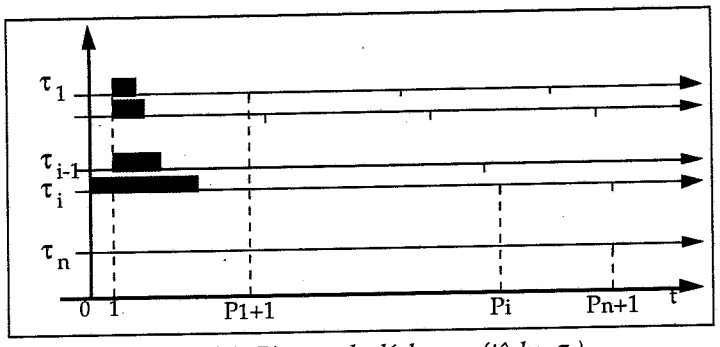

Figure 2.3: Pire cas de déphasage (tâche τ_i)

Le pire cas de déphasage de la tâche τ_i induit donc une inversion de priorité (τ_i est traité avant les tâches $\tau_1, \ldots \tau_{i-1}$). Les tâches $\tau_1, \ldots \tau_{i-1}$ subissent donc un retard et ne pourront être considérées qu'à la fin de l'exécution de la tâche τ_i. Il est donc essentiel de vérifier dans quelles conditions toutes les tâches $\tau_1, \ldots \tau_{i-1}$ de priorités supérieures à τ_i pourront satisfaire leurs échéances ($P_1, \ldots P_{i-1}$). Pour ce faire, il faut, sur l'intervalle de temps, que nous appelons *intervalle de temps critique lié à la tâche* τ_i (cet intervalle débute à l'invocation de la tâche τ_i (instant *0*) et finit avant l'échéance de τ_i (instant P_i)), évaluer l'utilisation minimale demandée au processeur, compte tenu que toutes les invocations des tâches τ_j de priorité supérieure à τ_i (c'est-à-dire, qui ont des périodes inférieures) doivent avoir leurs échéances respectées. Nous disons utilisation minimale parce que les tâches τ_k de priorité inférieure à τ_i, c'est-à-dire qui ont des périodes supérieures, ont leur échéance après l'instant P_i et donc il n'est pas nécessaire qu'elles utilisent le processeur avant l'instant P_i.

L'intervalle de temps critique associé à la tâche τ_i est noté: $[0, L]$, avec $P_1 + 1 \leq L < P_i$, soit encore $P_1 < L < P_i$.

3.2. Évaluation de l'utilisation minimale demandé au processeur sur l'intervalle de temps critique associé à la tâche τ_i

Appelons $\{d_{0,L}\}_i$ cette utilisation minimale. On a:

$$\{d_{0,L}\}_i = C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{L-1}{P_j} \right\rfloor C_j \quad , P_1 < L < P_i \tag{2.9}$$

Cette expression comprend:

- d'une part, le temps d'utilisation relatif à l'exécution de la tâche τ_i: C_i;
- d'autre part, la somme des temps d'utilisations des différentes tâches τ_j $(j < i)$ plus prioritaires que la tâche τ_i; le facteur $\lfloor (L-1)/P_j \rfloor$ exprime le fait que, pour toute tâche τ_j, seules les invocations qui débutent et finissent dans l'intervalle $L-1$ doivent être considérées. Les tâches τ_k $(k > i)$ moins prioritaires que la tâche τ_i ne sont pas considérées puisque leurs échéances sont postérieures à l'instant P_i.

3.3. Test d'ordonnançabilité

L'ordonnancement de l'ensemble de n tâches: $\tau = \{\tau_1, \ldots \tau_i, \ldots \tau_n\}$ en utilisant un algorithme ED non-préemptif est faisable si:

$$\bullet \sum_{i=1}^{n} C_i/P_i \leq 1 \tag{2.10}$$

$$\bullet C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{L-1}{P_j} \right\rfloor C_j \leq L \quad , \begin{cases} 1 < i \leq n \\ P_1 < L < P_i \end{cases} \tag{2.11}$$

La première condition, équivalente à celle de l'algorithme ED, représente une garantie de non surcharge du système. Informellement, nous pouvons dire que la somme cumulative de la charge induite par chaque tâche, ne doit pas dépasser l'unité. La deuxième condition impose que la demande faite au processeur dans l'intervalle de temps critique $[0, L]$ associé à une tâche τ_i $(\forall_i)$ doit être toujours inférieure à la longueur de cet intervalle.

3.3. Intérêt des algorithmes non-préemptifs

Sachant que l'exécution des tâches n'est jamais interrompue, l'implémentation d'un ordonnanceur non-préemptif est toujours plus simple que celle d'un ordonnanceur préemptif, car les changements de contextes ne se produisent qu'à la fin d'exécution des tâches (prévisibilité de fonctionnement). En plus, tant qu'on reste dans le cas monoprocesseur, on n'a plus besoin de mécanismes d'exclusion mutuelle (rendez-vous, sémaphores, ...) pour protéger des données ou des ressources partagées. En conséquence, la charge d'exécution due au fonctionnement d'un ordonnanceur non-préemptif est généralement plus réduite.

Cependant, il faut noter que la préemption donne plus de liberté à l'ordonnanceur pour choisir une solution d'ordonnancement [CM, 94].

4. Contributions

Nous présentons maintenant des travaux que nous avons réalisés sur les algorithmes non-préemptifs et qui constituent, à notre avis, des contributions intéressantes dans le domaine des algorithmes non-préemptifs. Le travail le plus important est relatif à l'algorithme ED non-préemptif. Une réflexion sur l'algorithme RM non-préemptif est également présentée.

4.1. Algorithme ED non-préemptif

4.1.1. Justification et présentation de l'étude effectuée

Le travail sur l'algorithme ED non-préemptif effectué dans [JSM, 91] est intéressant, car c'est la première fois que des conditions nécéssaires et suffisantes d'ordonnançabilité pour un ordonnancement en-ligne non-préemptif (c.f. II.3) ont été énoncées. Cependant, la deuxième condition d'ordonnançabilité (condition spécifique de l'attribut "non-préemptif") énoncée est exprimée dans une échelle temporelle discrète, ce qui:

a) implique, d'une part, que toutes les périodes et durées des tâches soient des multiples de l'unité de temps choisie et, d'autre part, que le pire cas de déphasage soit égal à cette unité de temps,

b) induit une complexité de calcul incompatible avec l'attribut en-ligne de l'ordonnancement (car pour chaque tâche τ_i, on doit évaluer l'utilisation minimale

$\{d_{0,L}\}_i$ demandée au processeur pour tous les L appartenant à l'ensemble $\{(P_1+1),(P_1+2),\ldots,(P_i-1)\}$).

Ce sont précisément ces deux constatations a) et b) qui ont motivé notre travail sur l'algorithme ED non-préemptif et qui nous ont donc amené:

- à considérer une échelle temporelle continue (les périodes et durées des tâches peuvent donc être quelconques et le déphasage initial peut être infiniment petit; la complexité des calculs est moins grande)
- et donc à reformuler la deuxième condition d'ordonnançabilité, dans le cadre de cette échelle temporelle continue, et à proposer une méthodologie d'évaluation de cette deuxième condition d'ordonnançabilité facile à mettre en oeuvre (et par conséquent, compatible avec l'attribut "en-ligne" de l'ordonnancement).

4.1.2. Considération d'une échelle temporelle continue

1. Expression de la deuxième condition d'ordonnançabilité

Reprenons l'expression de l'utilisation minimale demandé au processeur pendant l'intervalle de temps critique lié à une tâche i $\{d_{0,L}\}_i$ et remplaçons la variable temporelle discrète L et l'intervalle de temps unitaire par, respectivement, les variables temporelles continues t et α.

Le pire cas de déphasage de la tâche i appartenant à l'ensemble de tâches $T=\{\tau_1,\tau_2,\ldots\tau_i,\ldots\tau_n\}$ est obtenu lorsque $(\alpha\to 0^+)$ (figure 2.4).

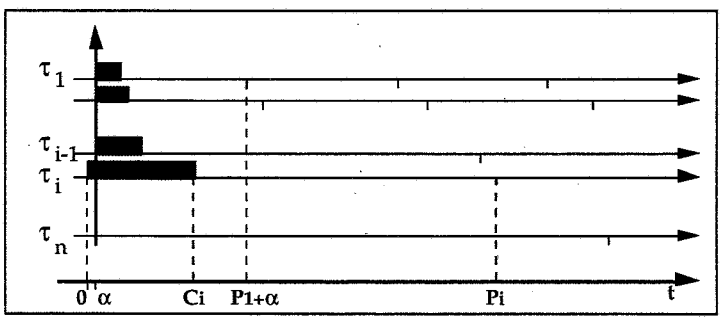

Figure 2.4: Pire cas de déphasage

L'utilisation minimale demandée au processeur pendant l'intervalle de temps critique lié à la tâche τ_i (intervalle $[0,t]$ avec $P_1+\alpha\le t<P_i$ et $(\alpha\to 0^+)$) est maintenant notée $\{d_{0,t}\}_i$ et s'exprime de la manière suivante:

$$\{d_{0,t}\}_i = C_i + \sum_{j=1}^{i-1}\left\lfloor\frac{t-\alpha}{P_j}\right\rfloor C_j \qquad (2.12)$$

La deuxième condition d'ordonnançabilité s'exprime maintenant sous la forme suivante:

$$\{d_{0,t}\}_i \le t \ , \quad \forall_{P_1+\alpha\le t<P_i} \ , \ \forall i_{1<i\le n}. \qquad (2.13)$$

L'évaluation de cette deuxième condition d'ordonnançabilité nécessite:

- tout d'abord de calculer, $\forall i,\ 1<i\le n$, la fonction $\{d_{0,t}\}_i$ qui est une fonction en escalier croissante,

- dont les discontinuités se produisent à tous les instants t, tels que $(t-\alpha)$ est un multiple de P_j ($\forall j,\ 1 \le j \le i-1$), c'est-à-dire, $t-\alpha = k \cdot P_j$, soit $t = k \cdot P_j + \alpha$ ou encore comme $\left(\alpha \to 0^+\right)$, $t = k \cdot P_j^+$; ces instants représentent les instants d'arrivées des requêtes des tâches τ_j;
- dont la valeur sur le palier à l'instant $t = k \cdot P_j^+$ est la durée de l'utilisation demandée au processeur sur l'intervalle $\left[0, k \cdot P_j^+\right]$; nous appelons $u_i\left(\left[0, k \cdot P_j^+\right]\right)$ cette durée,

- et ensuite, sachant que $\left\{d_{0,t}\right\}_i$ est une fonction en escalier croissante, il suffit de vérifier qu'à chaque instant $t = k \cdot P_j^+$, $\forall i,\ 1 < i \le n$, la durée de l'utilisation demandée au processeur $u_i\left(\left[0, k \cdot P_j^+\right]\right)$ est strictement inférieure à $k \cdot P_j^+$.

2. Reformulation de la deuxième condition d'ordonnançabilité

Nous introduisons la notion de demande normalisé $W_i(t) = \dfrac{\left\{d_{0,t}\right\}_i}{t}$, ce qui nous amène à reformuler la deuxième condition d'ordonnançabilité sous la forme suivante:

$$W_i(t) \le 1 \qquad \forall t:\ P_1 \le t < P_i\ ,\ \forall i:\ 1 < i \le n\ . \tag{2.14}$$

Notons que comme $\left\{d_{0,t}\right\}_i$ est une fonction en escalier croissante, dont les discontinuités se produisent aux instants $k \cdot P_j^+$ (fonction ouverte à gauche et fermée à droite des instants de discontinuité), la demande normalisée est une fonction qui présente des maxima aux instants $k \cdot P_j^+$ et qui est décroissante entre les instants $k \cdot P_j^+$.

La vérification de la deuxième condition d'ordonnançabilité demande à vérifier, $\forall i,\ 1 < i \le n$, qu'à chaque instant $k \cdot P_j^+$, la demande normalisée $W_i\left(k \cdot P_j^+\right)$ qui est définie par le rapport $\dfrac{u_i\left(\left[0, k \cdot P_j^+\right]\right)}{k \cdot P_j^+}$ est strictement inférieure à 1. La notion "strictement inférieure à 1" découle de la définition du pire cas de déphasage (α toujours positif).

Il nous est paru intéressant d'introduire cette valeur de demande normalisée qui permet de traiter sous la forme d'un rapport, c'est-à-dire de manière (à notre avis) plus parlante, la deuxième condition d'ordonnançabilité.

4.1.3. Méthode d'évaluation de la deuxième condition d'ordonnançabilité

La présentation précédente a montrée l'importance des instants $t = k \cdot P_j^+$ pour évaluer la deuxième condition d'ordonnançabilité. Nous appelons ces instants $t = k \cdot P_j^+$ les points d'ordonnancement.

La méthode proposée est la suivante:

1- pour tout i, $1 < i \le n$:

a) définir l'ensemble S_i des points d'ordonnancement:

$$S_i = \bigcup_{j=1}^{i-1} \left\{ \bigcup_{k=1}^{\lceil P_i/P_j - 1 \rceil} \left(k \cdot P_j^+\right) \right\} \tag{2.15}$$

b) calculer les maxima de la demande normalisée $W_i(t)$, c'est-à-dire on calcule la valeur de la demande normalisée en chaque point d'ordonnancement:

on a: $\forall_i$, $\forall_{k\cdot P_j^+ \in S_i}$, $W_i\left(k\cdot P_j^+\right) = \dfrac{u_i\left(\left[0, k\cdot P_j^+\right[\right)}{k\cdot P_j^+} = \dfrac{C_i + \sum_{l=1}^{i-1}\left\lfloor \left(k\cdot P_j^+\right)/P_l \right\rfloor \cdot C_l}{k\cdot P_j^+}$ (2.16)

Appelons W_i l'ensemble de ces valeurs.

c) déterminer la valeur maximale sur l'ensemble des valeurs $W_i\left(k\cdot P_j^+\right)$; appelons $W_{i_{max}}$ cette valeur maximale:

$$W_{i_{max}} = max\, W_i\left(k\cdot P_j^+\right), \quad \forall_{k\cdot P_j^+ \in S_i} \tag{2.17}$$

2- la phase précédente permet d'obtenir l'ensemble des valeurs W_i relatives à chaque tâche i $(1 < i \le n)$. On détermine la valeur maximale de ces valeurs $W_{i_{max}}$, que l'on appelle W:

$$W = \max_{1<i\le n}\left\{W_{i_{max}}\right\} \tag{2.18}$$

3- la deuxième condition d'ordonnançabilité est vérifiée si $W < 1$

4.1.4. Exemple d'application

Considérons l'ensemble de tâches $T = \{\tau_1, \tau_2, \tau_3, \tau_4\}$, tel que: $\{(C_i, P_i)\} = \{(2,6), (3,10), (4,20), (4,30)\}$, pour lequel la première condition d'ordonnançabilité est vérifiée $\left(\sum_{i=1}^{4} C_i/P_i = 0.9667 \le 1\right)$.

Appliquons la méthodologie que nous venons de présenter pour le calcul de la deuxième condition d'ordonnançabilité:

1- Pour tout i, $2 \le i \le 4$:

a) les ensembles S_i des points d'ordonnancement sont:

$$S_2 = \left\{6^+\right\},\ S_3 = \left\{6^+, 10^+, 12^+, 18^+\right\},\ S_4 = \left\{6^+, 10^+, 12^+, 18^+, 20^+, 24^+\right\}$$

b) les ensembles W_i des maxima des demandes normalisées sont:

$$W_2 = \left\{\frac{5}{6^+}\right\},\ W_3 = \left\{\frac{6}{6^+}, \frac{9}{10^+}, \frac{11}{12^+}, \frac{13}{18^+}\right\},\ W_4 = \left\{\frac{6}{6^+}, \frac{9}{10^+}, \frac{11}{12^+}, \frac{13}{18^+}, \frac{20}{20^+}, \frac{22}{24^+}\right\};$$

c) la valeur maximale des demandes normalisées: $W_{i_{max}}$, est:

$$W_{2max} = \frac{5}{6^+},\ W_{3max} = \frac{6}{6^+} \text{ et } W_{4max} = \frac{6}{6^+} \tag{2.19}$$

Nous utilisons encore la notation suivante: $W_{2max} = 0.833^-$, $W_{3max} = 1^-$ et $W_{4max} = 1^-$ qui veut visualiser des relations de type "strictement inférieur".

2- on obtient: $W = \max_{1<i\le 4}\left\{W_{i_{max}}\right\} = 1^-$

3- Donc la deuxième condition d'ordonnançabilité est vérifiée ($W < 1$).

Nous représentons sur les figures 2.5 et 2.6 les séquences temporelles des tâches (en haut des figures) et les demandes normalisées (en bas des figures); les courbes relatives aux demandes normalisées visualisent à la fois les points d'ordonnancement et les maxima des demandes normalisées en ces points.

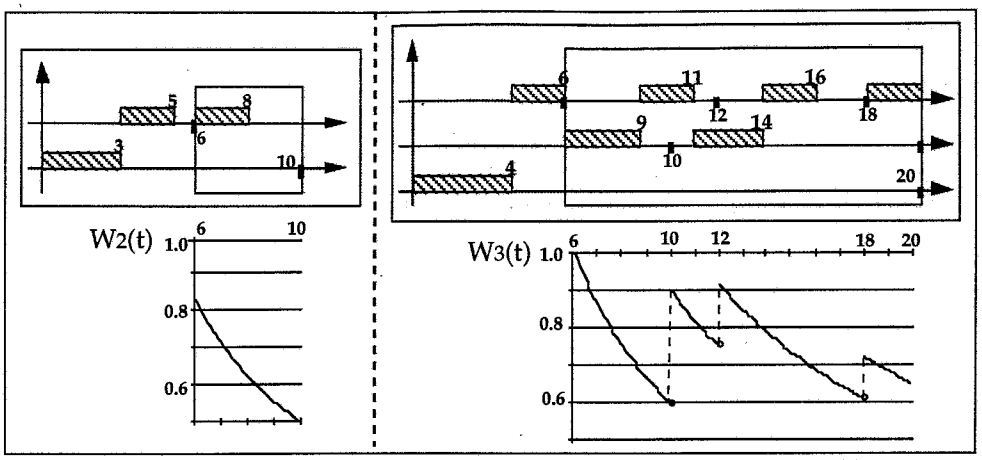

Figure 2.5: Séquences temporelles et demande normalisée $W_2(t)$ et $W_3(t)$

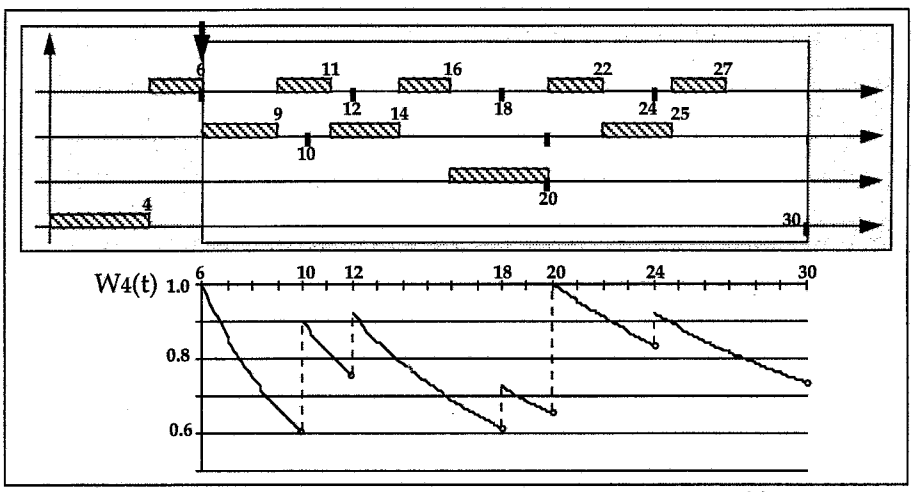

Figure 2.6: Séquence temporelle et demande normalisée $W_4(t)$

4.2. Algorithme RM non-préemptif

4.2.1. Proposition

A notre connaissance, il n'y a jamais eu de travaux sur les algorithmes RM non-préemptifs. Cependant, en utilisant les résultats de l'étude sur le partage de ressources avec les algorithmes préemptifs, et en particulier, sur les techniques à priorité héritée [SRL, 90], on peut définir une condition suffisante d'ordonnançabilité pour un algorithme RM non-préemptif.

Considérons les deux expressions présentées au paragraphe 2.1.4 et qui représentent deux tests d'ordonnançabilité de l'algorithme RM préemptif:

- premier test: $$\forall i,\ 1 \le i \le n,\ \sum_{j=1}^{i} \frac{C_j}{P_j} + \frac{B_i}{P_i} \le i \cdot \left(\sqrt[i]{2} - 1\right) \qquad (2.30)$$

  où B_i représente la durée maximale du blocage d'un travail de la tâche τ_i par un travail d'une tâche de priorité inférieure $\tau_{I+1}, \ldots \tau_n$.

- deuxième test: $$\forall i,\ 1 \le i \le n,\ \sum_{i=1}^{n} \frac{C_i}{P_i} + \max_{1<i\le n}\left\{\frac{B_i}{P_i}\right\} \le n \cdot \left(\sqrt[n]{2} - 1\right) \qquad (2.31)$$

où B_i a la même signification que précédemment.

Ces deux formules peuvent être utilisées pour définir deux tests d'ordonnançabilité pour l'algorithme RM non-préemptif, en considérant que le travail de la tâche τ_i est maintenant bloqué pendant "toute la durée d'exécution" d'un travail d'une tâche de priorité inférieure $\tau_{I+1}, \ldots \tau_n$ (attribut de non-préemption).

Nous définissons donc B_i comme étant la durée maximale des tâches de priorité inférieure à la tâche τ_i, c'est-à-dire:

$$B_i = \max_{i+1 \le j \le n} \{C_j\} \tag{2.32}$$

Notons que les deux tests sont pessimistes (comme le test de l'algorithme RM préemptif) comme nous le montrons sur l'exemple suivant.

4.2.2. Exemple

Nous considérons un exemple où nous appliquons, à la fois, le premier et le deuxième tests d'ordonnançabilité.

Soit l'ensemble de tâches $\tau = \{\tau_1, \tau_2, \tau_3\}$ avec $\{C_i, P_i\} = \{(2,8),(2,10),(5,20)\}$, ordonnancé par l'algorithme RM non-préemptif.

Nous représentons sur les figures 2.8a et 2.8b l'intervalle de temps critique associés, respectivement, à la tâche τ_2 et à la tâche τ_3. On voit donc que l'ordonnancement est possible quel que soit le pire cas de déphasage considéré.

Cependant, les tests d'ordonnançabilité ne sont pas satisfaits quand on vérifie l'ordonnançabilité de τ_3, ce qui montre que ces tests d'ordonnançabilité sont pessimiste.

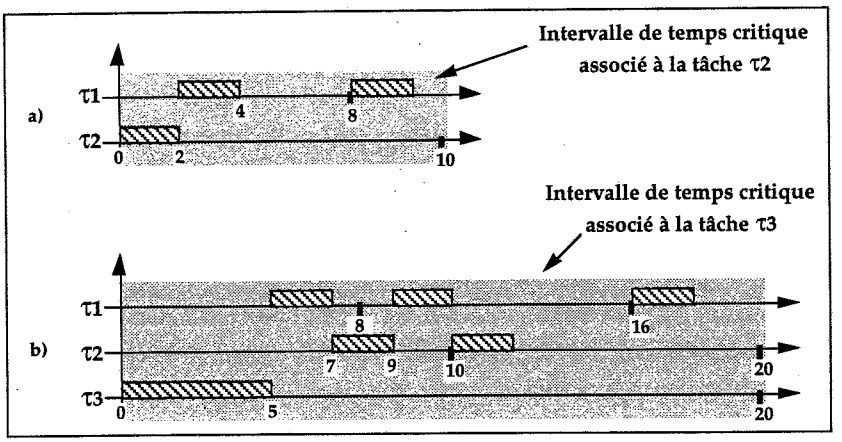

Figure 2.8:

5. Conclusion

Outre la présentation des principaux algorithmes pour des tâches indépendantes contraintes temporellement (algorithmes RM, DM, ED) et de la problématique du partage de ressources (techniques par héritage), nous voulons insister sur le travail présenté relativement aux algorithmes non-préemptifs (l'attribut "non-préemptif" est important pour l'ordonnancement du transfert de messages):

- algorithme ED non-préemptif (deuxième condition d'ordonnançabilité [JSM, 91], qui est la condition spécifique de l'attribut "non-préemptif" et qui a été exprimée dans une

échelle temporelle discrète): nous avons, d'une part, développé une extension de cette deuxième condition à une échelle temporelle continue (ce qui permet de considérer des périodes et des durées quelconques pour les tâches, ainsi que des déphasages initiaux infiniment petits), et, d'autre part, proposé une méthode de calcul relativement simple et facilement implantable en-ligne;

- algorithme RM non-préemptif: ne connaissant pas de travaux dans ce domaine, nous avons suggéré une condition d'ordonnançabilité qui exploite le résultat sur les techniques à priorité héritée pour le partage de ressources.

6. Références

[AB, 90] N. Audsley, A. Burns; Real-Time Systems Scheduling; YCS 134, Departement of Computer Science, University of York, 1990

[Bak, 91] T. Baker; Stack-Based Scheduling of Realtime Processes; Journal of Real-Time Systems, 3, pages 67-99, 1991

[CC, 89] H. Chetto, M. Chetto; Some Results of the Earliest Deadline Scheduling Algorithm; IEEE Tr. on Software Engineering, 15(10), pages 1261-1269, 1989

[CM, 94] C. Cardeira, Z. Mammeri; Ordonnancement de Tâches dans les Systèmes Temps Réel et Répartis: Algorithmes et Critères de Classification; APII 28(4), pp 353-384, 1994

[Del, 94] J. Delacroix; Un contrôleur d'ordonnancement temps-réel pour la stabilité de earliest deadline en surcharge: le régisseur; Thèse de Doctorat, CNAM, Paris, 1994.

[JP, 86] M. Joseph, P. Pandaya; Finding Response Times in a Real-Time System; The Computer Journal, 29(5), pp. 390-395, 1986.

[JSM, 91] K. Jeffay, D. Stanat, C. Martel; On Non-Preemptive Scheduling of Periodic and Sporadic Tasks, in Proc. of RTSS'91 - IEEE Real-Time Systems Symposium, San Antonio, Texas , December 1991

[LL, 73] C. Liu and J. Layland, Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, Journal of ACM, vol. 29, n° 1, 1973, pp 46-61

[LM, 80] J. Leung, M. Merril; A Note on Preemptive Scheduling of Periodic Real-Time Tasks; Information Processing Letters, 11(3), pp 115-118, November 1980.

[LSD, 89] J. Lehoczky, L. Sha, Y. Ding; The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behaviour; IEEE RTSS'89, pages 166-171, December 1989

[LSS, 87] J. Lehoczky, L. Sha, J. Strosnider; Enhanced Aperiodic Responsiveness in Hard Real-Time Environments; IEEE RTSS'87, pages 261-270, 1987

[Mar, 94] P. Martineau; Ordonnancement en-ligne dans les systèmes informatiques temps-réel; Thèse de Doctorat (n° ED 82-93), Ecole Centrale de Nantes, 21 Octobre 1994.

[SRL, 90] L. Sha, R. Rajkumar, J. Lehoczky; Priority Inheritance Protocols: An Approach to Real-Time Synchronization; IEEE Tr. on Computers, 39(9), September 1990

[SSL, 89] B. Sprunt, J. Lehoczky, L. Sha; Aperiodic Task Scheduling for Hard Real-Time Systems; Journal of Real-Time Systems, 1, pages 27-60, 1989

Chapitre III

Réseaux de communication temps-réel: les principaux protocoles MAC

1. Introduction

Le but de ce chapitre est de présenter les principaux protocoles MAC qui peuvent être utilisés dans un contexte temps-réel (c'est-à-dire, qui permettent la mise en oeuvre d'un ordonnancement de transfert de flux de messages contraints temporellement).

En ce qui concerne ces protocoles, les présentations traditionnelles se focalisent principalement sur les "techniques d'accès à la ressource de transmission" et passent généralement sous silence l'aspect "ordonnancement du transfert de flux de messages à travers la ressource de communication". Ce dernier aspect est absolument essentiel, et plus particulièrement dans un contexte temps-réel, si on veut spécifier et concevoir de manière rationnelle un protocole MAC. C'est ce souci qui va nous guider dans la présentation proposée, afin de bien mettre en évidence les différences entre les protocoles existants.

Ce chapitre comprend cinq parties:

- La première partie concerne une classification de protocoles MAC temps-réel en deux grandes classes: classe 1 et classe 2;
- La deuxième partie présente les principaux protocoles de classe 1;
- La troisième partie présente les principaux protocoles de classe 2;
- La quatrième partie concerne la présentation d'un protocole intégrant les caractéristiques de la classe 1 et de la classe 2;
- La cinquième partie présente une comparaison des protocoles de classe 1 et de classe 2;

2. Classification des protocoles MAC temps-réel

2.1. . La problématique de l'ordonnancement du transfert de flux de messages temps-réel

Une classification des protocoles MAC temps-réel a été proposée dans [MZ, 95]. Dans ce travail les auteurs considèrent d'abord que l'ordonnancement du transfert des messages doit être vu sous l'angle d'une séquence de deux processus:

- un processus *d'arbitrage d'accès* qui détermine *quand* la station a le droit d'utiliser la ressource de transmission;
- un processus de *contrôle de transmission* qui détermine *combien de temps* la station a le droit d'utiliser la ressource de transmission;

Ensuite, une classification des protocoles MAC temps-réel est proposée selon la prépondérance de chacun de ces processus sur l'ordonnancement des messages; deux approches sont définies:

- une approche basée sur l'arbitrage d'accès, où l'accent est mis sur le processus d'arbitrage d'accès;
- une approche basée sur le contrôle de transmission, où l'accent est mis sur le processus de contrôle de transmission.

Un problème évident associé à cette classification réside dans le cas où aucun des deux processus est prépondérant vis-à-vis de l'autre (c'est le cas du protocole IEEE 802.3 DCR (c.f. III.2.4.2)).

À notre avis, la prépondérance relative de ces deux processus ne doit pas être le seul paramètre choisi pour effectuer une classification de protocoles de MAC temps-réel.

Nous proposons une autre classification des protocoles MAC temps-réel basée non sur la prépondérance relative de ces deux processus, mais sur la *mise en oeuvre* de l'ordonnancement du transfert des messages.

Nous élargissons d'abord la définition de ces deux processus, en considérant séparément les notions de flux de messages et de station:

Définition

- *Le processus d'arbitrage d'accès détermine quand le flux de messages ou la station a le droit d'utiliser la ressource de transmission.*
- *Le processus de contrôle de la durée de transmission détermine combien de temps le flux de messages ou la station a le droit d'utiliser la ressource de transmission*

Si on observe les systèmes existants, on constate que les protocoles MAC temps-réel mettent en oeuvre des mécanismes qui travaillent, soit sur l'accès des flux de messages, soit sur l'accès des stations (plus précisément, qui garantissent un temps d'accès borné aux stations).

En conséquence, nous proposons la définition de deux grandes classes de protocoles MAC temps-réel (figure 3.1), selon que l'ordonnancement est mis en oeuvre sur les flux de messages ou sur les stations:

- La classe 1 qui est relative aux protocoles réalisant un *ordonnancement basé sur une assignation de priorité aux flux de messages* (priorité traduisant les contraintes temporelles); les protocoles MAC de la classe 1 travaillent à partir des contraintes

temporelles des flux de messages, qu'ils interprètent et qu'ils utilisent pour réaliser l'ordonnancement (ordonnancement global) et les transferts.

- La classe 2 qui est relative aux protocoles réalisant un *ordonnancement basé sur la notion d'une garantie d'un temps d'accès borné aux entités MAC (donc aux stations)*; dans ce cas, le protocole MAC offre simplement un service d'accès, en temps borné et en exclusion mutuelle, à la ressource de transmission; ce service doit être utilisé par le(s) niveau(x) supérieur(s) au niveau MAC, qui doit (doivent) mettre en oeuvre l'ordonnancement des flux de messages pendant le temps d'accès (ordonnancement local).

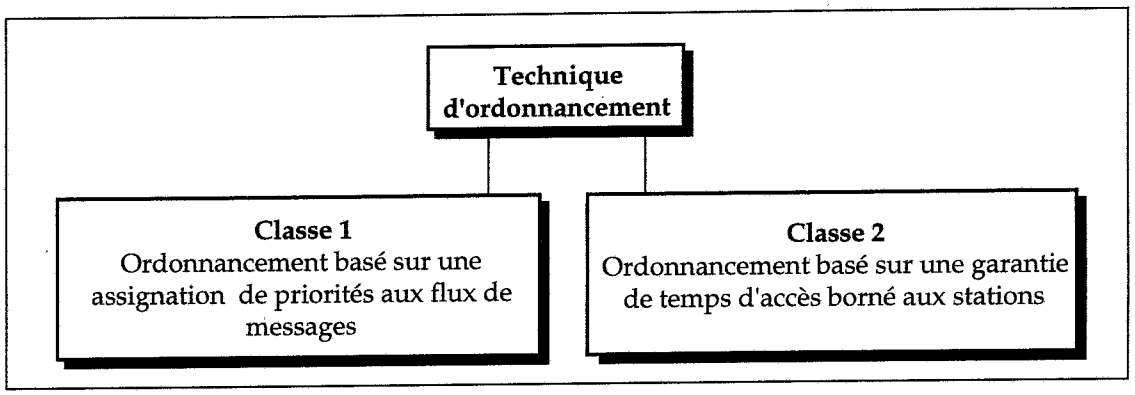

Figure 3.1: Deux classes de protocoles

La classe 1 est, à notre avis, vraiment représentative de ce que les protocoles MAC dits "temps-réel" doivent être. De toute manière, quelle que soit la classe de protocole que l'on considère comme point de départ, c'est-à-dire l'ordonnancement des flux de messages ou l'ordonnancement des stations, on doit considérer cet ordonnancement comme faisant intervenir les deux processus suivants:

- *L'arbitrage d'accès* (accès de flux de messages pour la classe 1; accès de stations pour la classe 2);
- *Le contrôle de la durée de transmission* (pour un flux de messages dans la classe 1; pour une station dans la classe 2).

2.2. L'arbitrage d'accès et le contrôle de la durée de transmission

2.2.1. Classe 1

- L'arbitrage d'accès consiste en deux mécanismes:
  - Un mécanisme *d'assignation de priorités* à chaque flux de messages (assignation statique) ou à chaque message individuel (assignation dynamique).
  - Un mécanisme global *d'arbitrage de priorités* qui détermine, en-ligne ou hors-ligne, le flux de message ou le message qui accède à la ressource de transmission.
- Le contrôle de la durée de transmission est, en général, très simple: on transfère un message par accès arbitré.

L'opération générale de l'arbitrage d'accès dans les protocoles de "classe 1" est équivalente à celle de l'opération d'un ordonnanceur de tâches dans un système monoprocesseur. L'équivalence se traduit par l'existence d'une seule ressource à partager: *processeur/réseau*, parmi plusieurs utilisateurs: *tâches/flux de messages*. Compte tenu de la similarité de ces deux systèmes et du grand nombre de travaux faits sur l'ordonnancement monoprocesseur temps-réel, on a assisté ces dernières années, à un intérêt croissant pour l'adaptation des

algorithmes classiques d'ordonnancement de tâches à l'ordonnancement temps-réel des messages dans les réseaux de communication.

Dans le cas de l'arbitrage d'accès en-ligne, on a simultanément la mise en oeuvre de l'arbitrage d'accès et la mise en oeuvre effective du transfert. De plus, la mise en oeuvre du transfert est transparente aux évolutions des caractéristiques des flux de messages à ordonnancer (c'est-à-dire des modes de fonctionnement).

Dans le cas de l'arbitrage d'accès hors-ligne, on doit distinguer deux activités au sein d'un mode de fonctionnement:

- Une première activité qui, à partir de la connaissance des priorités, évalue la séquence des flux de messages à transférer;
- Une deuxième activité qui consiste en la mise en oeuvre effective du transfert de la séquence des flux de messages.

La stratégie hors-ligne n'est pas transparente aux changements de mode, et il faut donc prévoir, en-ligne, un mécanisme de changement de mode de fonctionnement.

En ce qui concerne le trafic périodique, l'ordonnancement hors-ligne est toujours plus compétitif que celui effectué en-ligne [B_all, 92]. Nous pouvons citer les avantages suivants par rapport à un ordonnancement en-ligne: taux d'utilisation élevé; prédiction de la séquence temporelle du trafic; garantie de non surcharge du réseau; coût de fonctionnement réduit; propagation de fautes temporelles limitée.

En ce qui concerne le trafic apériodique, comme l'ordonnanceur hors-ligne n'a qu'une connaissance très limitée sur le futur, sa performance sera toujours inférieure à celle d'un ordonnanceur en-ligne [B_all, 92]. Donc, ce type de trafic ne peut être pris en compte qu'en utilisant des moyens de réservation de bande passante peu efficaces. De plus, la mise en oeuvre de moyens de réservation de bande passante, requiert des mécanismes d'arbitrage supplémentaires, ce qui réduit encore leur efficacité.

2.2.2. Classe 2

- L'arbitrage d'accès a pour but de permettre l'accès de manière équitable aux différentes stations.
- Le contrôle de la durée de transmission peut être très élémentaire (un message transmis par station), mais peut être également très sophistiqué: en effet, toute station doit pouvoir disposer d'un temps de transmission suffisant pour transmettre les flux de messages temps-réel venant du niveau supérieur, et donc il est important d'utiliser des techniques d'allocation de bande passante qui permettent de garantir le respect des contraintes temporelles associées aux flux de messages des différentes stations. Cette allocation de bande est effectuée par l'intermédiaire de deux mécanismes:
  - Un mécanisme de gestion globale qui gère le compromis de la répartition d'une ressource potentiellement insuffisante, c'est-à-dire le temps d'utilisation du réseau, parmi un ensemble de stations avec des besoins temps-réel différents;
  - Un mécanisme de gestion locale qui évalue d'abord les besoins particuliers de chaque station pour en informer la gestion globale, et ensuite gère la répartition de l'allocation de bande parmi les flux de messages appartenants à la station, de façon à garantir leurs contraintes temporelles.

2.3. Mise en oeuvre de l'arbitrage d'accès et du contrôle de la durée de transmission: les techniques d'accès

La présentation des techniques d'accès est la manière traditionnelle de parler des protocoles MAC (sans faire référence aux aspects d'ordonnancement qu'ils réalisent).

Nous indiquons sur la figure 3.2, les principales techniques d'accès:

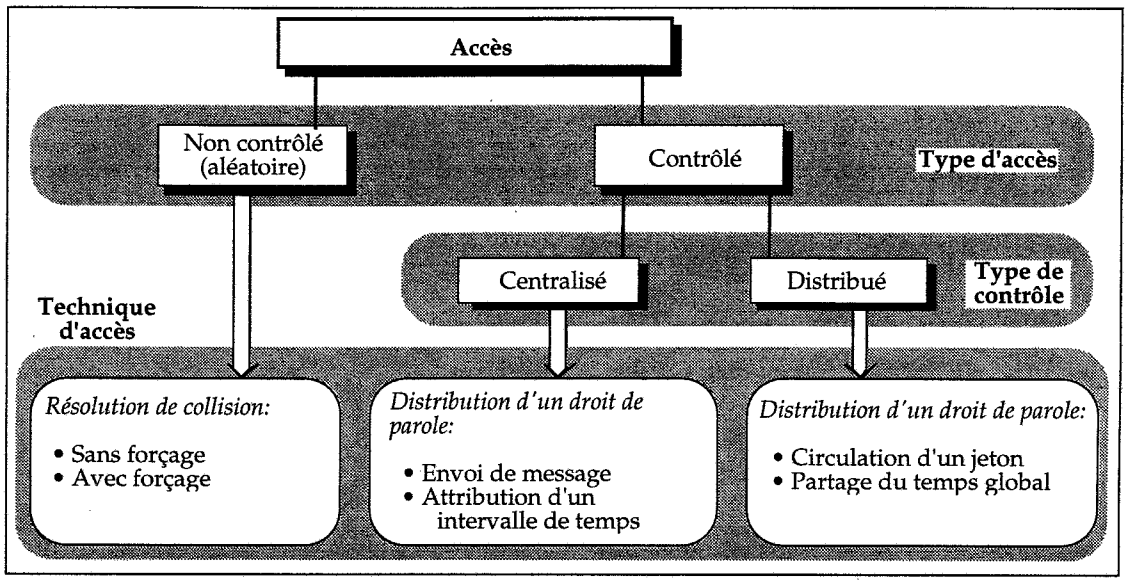

Figure 3.2: Les techniques d'accès

- L'accès non-contrôlé (aléatoire) est une technique dite de compétition qui génère, par définition, des collisions.

  Deux variantes, à des finalités temps-réel, de cette technique ont été définies:

  - la résolution de collision sans forçage.
  - la résolution de collision avec forçage.

- L'accès à contrôle centralisé est basé sur l'existence d'une station de contrôle qui distribue un droit de parole aux différentes stations. On distingue deux variantes:

  - soit la station de contrôle envoie à chaque station un message qui lui donne le droit d'utiliser le réseau;
  - soit la station de contrôle joue le rôle d'un horloge qui définit des intervalles de temps que les stations peuvent utiliser.

- L'accès à contrôle distribué est basé sur une coopération entre toutes les stations afin de définir laquelle a le droit de parole, c'est-à-dire, le droit d'utiliser le réseau. On distingue également deux variantes:

  - la circulation d'un jeton (technique de jeton circulant) que les stations se transmettent; le jeton est un mécanisme de coopération explicite;
  - la technique de partage du temps global, qui est basée sur l'hypothèse que chaque station a la connaissance du temps global et des intervalles de temps où elle peut utiliser le réseau; c'est un mécanisme de coopération implicite.

3. Protocoles de la "classe 1"

3.1. Principales normes

Les principales normes de "classe 1" existantes sont représentées sur la figure 3.3.

Ces normes sont le plus généralement spécifiées en décrivant la mise en oeuvre de la technique d'accès, mais en passant sous silence l'aspect ordonnancement et, par voie de conséquence, le lien que l'on doit faire entre la technique de l'ordonnancement et la technique d'accès.

Nous présentons maintenant les éléments essentiels de l'arbitrage d'accès en mettant précisément en exergue l'aspect ordonnancement (avec les principaux travaux) et le lien avec la technique d'accès.

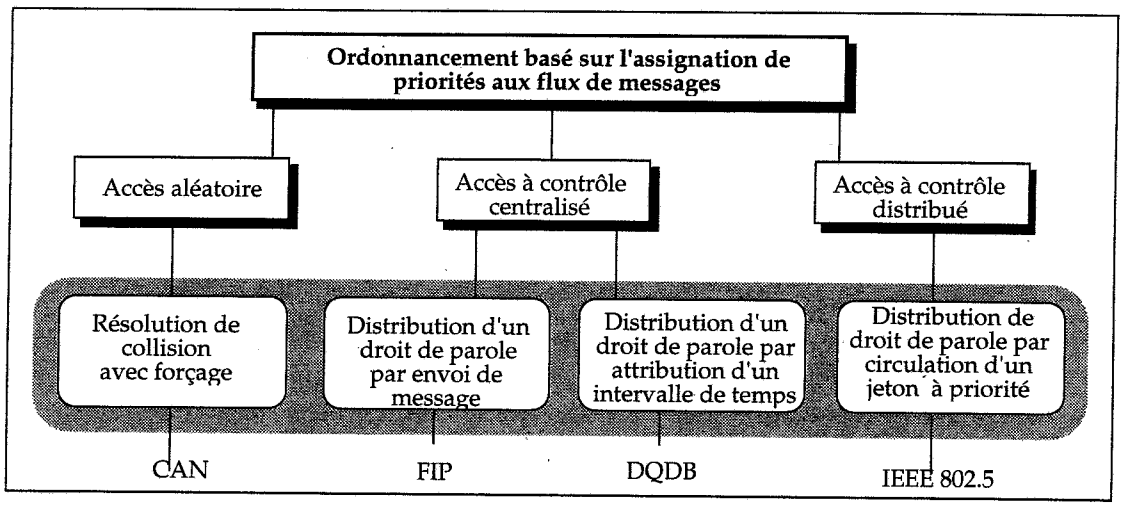

Figure 3.3: Les protocoles "classe 1" existants

3.2. Le protocole CAN

1. Généralités

Le protocole CAN [SAE, 92] a été proposé par Bosch dans le cadre des applications automobiles embarquées. Une caractéristique très intéressante de la norme CAN est que l'on spécifie un système de flux de messages indépendants (2^{11} ou 2^{29} flux différents, selon la version de la norme), chacun possédant une priorité unique dans le système. Ceci offre un cadre pour pouvoir définir des protocoles qui implantent le contrôle de l'accès à la ressource de transmission sur la base de l'interprétation de ce schéma de priorités.

2. Assignation de priorités

Un schéma d'assignation de priorités à des flux de messages périodiques, basé sur une adaptation de l'algorithme DM, a été proposé dans [TWH, 94]. Ce schéma de priorité, au lieu d'être borné sur les échéances, est basé sur des fenêtres de transmission qui prennent en compte la gigue maximale qui peut affecter l'arrivée dans les files d'attente de la sous-couche MAC des messages des flux périodiques.

Sur la figure 3.4, nous représentons la fenêtre de transmission des messages d'un flux M_i de période P_i, compte tenu que la gigue maximale est J_i et l'échéance associée aux messages est d_i. La fenêtre de transmission est $(d_i - J_i)$.

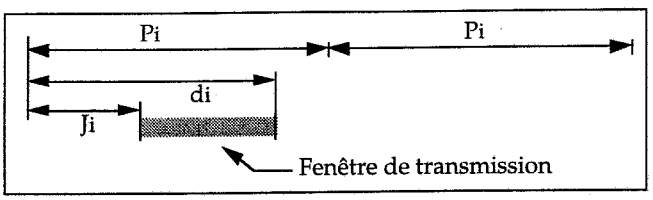

Figure 3.4: Fenêtre de transmission des messages

Dans leurs travaux, les auteurs proposent une assignation de priorités statiques inversement proportionnelle aux durées des fenêtres de transmission et déterminent la borne supérieure t_i du délai d'accès d'un message du flux M_i. La borne t_i est [TWH, 94]:

$$t_i = \max_{i+1 \le j \le n} C_j + \sum_{1 \le j \le i-1} \left\lceil \frac{t_i + J_i + \tau_{bit}}{P_j} \right\rceil \cdot C_j \qquad (3.1)$$

où le premier terme traduit l'aspect non-préemptif de la transmission et le deuxième terme traduit l'interférence des messages des flux M_j de plus haute priorité (compte tenu de la gigue J_j de ces flux, et de la durée d'un bit τ_{bit} qui est l'unité de mesure de temps).Une condition suffisante d'ordonnançabilité est:

$$\forall i, \quad t_i \le d_i - J_i \qquad (3.2)$$

3. Arbitrage de priorités

La technique de résolution de collisions avec forçage, qui gère l'arbitrage d'accès, est basée, d'une part, sur des structures des en-têtes des trames qui reflètent le schéma de priorité des messages et, d'autre part, sur une durée du bit sur le bus qui est nécessairement supérieure au double du temps de propagation maximum. Compte tenu de cette propriété de la durée du bit, une trame sur le bus est alors le ET-logique des trames émises. En conséquence, toute station qui ne reconnaît pas son en-tête détecte une collision et se retire; la station qui ne détecte pas de collision (c'est celle dont l'en-tête compte la plus longue séquence de zéros) continue sa transmission (il y a forçage).

4. Remarques

Un inconvénient majeur de cette technique d'arbitrage d'accès est que chaque station ne peut émettre, au plus, qu'un bit par intervalle de temps (intervalle égal à quatre fois le délai maximum de propagation sur le canal). Donc, la longueur maximale du réseau est fortement dépendante du taux de transfert choisi. Dans le cas de ce protocole, les longueurs maximales définies sont assez réduites: 50m à 1 Mbit/s ou 100m à 500 Kbit/s.

En outre, pour garantir une bonne *efficacité temps-réel* du protocole, un compromis doit être établi entre la longueur maximale des messages et la longueur de l'en-tête nécessaire pour l'arbitrage des priorités:

- L'augmentation de la durée des messages augmente la durée des inversions de priorité dans le bus, ce qui peut empêcher l'ordonnançabilité de l'ensemble des flux de messages.
- La diminution de la durée des messages augmente proportionnellement le coût de l'arbitrage des priorités, ce qui peut réduire considérablement la largeur de bande utile pour la transmission de messages.

Dans le cas du protocole CAN, on a choisi de réduire la longueur des messages, ce qui amène à une efficacité discutable du protocole (en-tête de 66 bits pour des messages dont le champ utile des données ne dépasse pas les 64 bits). Par contre, la durée maximale des inversions de priorité dans le bus est assez réduite: 130µs à 1 Mbit/s.

3.3. Le protocole FIP

1. Généralités

Le protocole FIP [NFa, 90] considère deux services de transmission: la transmission de variables et la transmission de messages. Les variables représentent les données échangées pour le contrôle d'un procédé industriel (on a des variables périodiques et apériodiques). Les messages permettent de gérer la configuration du système de contrôle et sont plutôt de nature apériodique. L'ordonnancement de ces échanges de variables et de messages est défini et mis en oeuvre dans un contrôleur central appelé *Arbitre de Bus*.

L'arbitre de bus réalise les fonctions suivantes:

1- La scrutation périodique de variables périodiques (c'est-à-dire, la mise en oeuvre des échanges de variables périodiques);

2- La scrutation périodique de variables apériodiques et de messages (c'est-à-dire, l'envoi des requêtes périodiques à des stations, pour demander s'il y a des variables apériodiques ou des messages en attente); cette fonction peut être vue comme une fonction *serveur* au sens des techniques d'ordonnancement de travaux apériodiques et peut être appelée *fonction de scrutation directe de variables apériodiques et de messages.*

3- La scrutation déclenchée de variables apériodiques et de messages (c'est-à-dire, la prise en compte des demandes venant des stations pour faire circuler des variables apériodiques et des messages); cette fonction peut être encore appelée *fonction de scrutation indirecte de variables apériodiques et de messages.*

Les principaux travaux sur l'arbitrage d'accès dans FIP définissent des techniques d'ordonnancement qui privilégient le trafic périodique (fonctions 1 et 2): un schéma d'assignation de priorités et d'ordonnancement associé aux variables et requêtes périodiques est défini. Il faut cependant remarquer que la notion de priorités pour l'échange de données a été introduite dans ces travaux, car elle n'est pas incluse dans la norme.

2. Assignation de priorités

Deux principaux schémas d'assignation de priorités ont été proposés: priorités associées aux messages (priorités dynamiques) [Laine, 91] et priorités associées aux flux (priorités statiques) [RN, 93].

- Dans [Laine, 91] l'assignation de priorités est effectuée en considérant chaque message individuellement (c'est-à-dire, si le flux F génère m messages pendant un intervalle de temps donné, il y aura m messages différents à considérer par l'algorithme d'ordonnancement). Un des avantages de cette assignation c'est la prise en compte des contraintes de précédence parmi les messages. Cependant, comme le nombre de messages pendant l'intervalle de temps peut être significatif (par rapport au nombre de flux de messages), ce schéma d'assignation de priorités peut devenir très complexe.
- Dans [RN, 93] les priorités sont assignées aux flux de messages (et non plus aux messages individuellement) et ensuite un ordonnancement basé sur des extensions aux algorithmes RM et ED [LL, 73] détermine la séquence des échanges.

3. Arbitrage de priorités

Nous indiquons maintenant le principe de la technique de la *table de scrutation* qui supporte l'ordonnancement dans le réseau FIP. La table de scrutation contient la liste de tous les identifieurs des variables et des requêtes du trafic périodique et est organisée comme un échéancier d'échanges de ces identifieurs. Cet échéancier est basé sur les notions de cycle élémentaire ou micro-cycle (plus petite fenêtre temporelle pendant laquelle chaque flux de trafic périodique peut transmettre au plus une entité d'information) et de macro-cycle (séquence de micro-cycles telle que les contraintes temporelles des flux du trafic périodique sont respectées sûrement). En considérant que l'on a p flux dans le trafic périodique avec des périodes $P_1, P_2, \ldots P_p$, les durées T_{mc} et T_{MC}, respectivement du micro-cycle et du macro-cycle, sont définies de la manière suivante:

$$T_{mc} = p\,gcd\{P_i\},\ 1 \le i \le p \tag{3.3}$$

$$T_{MC} = ppcm\{P_i\},\ 1 \le i \le p \tag{3.4}$$

Comme la durée d'un micro-cycle est généralement grande devant la durée des échanges des différents flux du trafic périodique, on a donc un espace temporel qui peut être utilisé pour le trafic lié à la fonction 3. En conséquence, le micro-cycle est décomposé en trois fenêtres: la fenêtre périodique (trafic périodique relatif aux fonctions 1 et 2), la fenêtre apériodique (fonction 3 relative aux échanges de variables apériodiques) et la fenêtre messages (fonction 3 relative aux échanges de messages). À la fin du macro-cycle, l'arbitre émet une séquence de bourrage du bus.

Un exemple de table de scrutation est donné sur la figure 3.5:

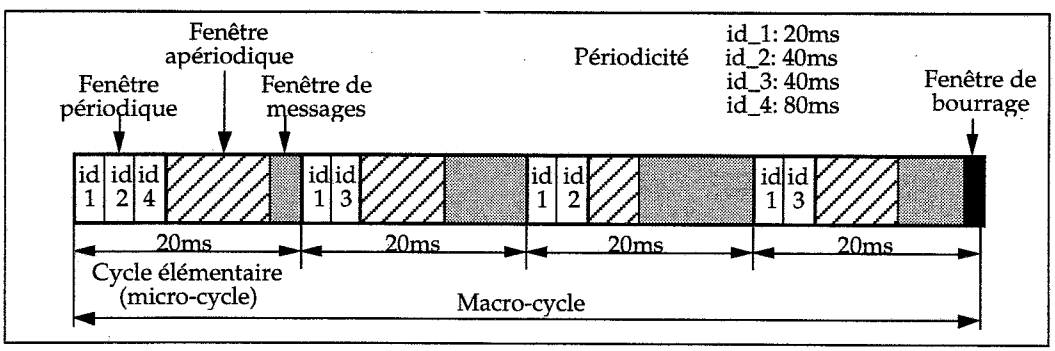

Figure 3.5: Table de scrutation dans FIP

Dans cet exemple, on considère un trafic périodique relatif à quatre variables et/ou requêtes de périodes *20ms*, *40ms*, *40ms* et *80ms* représentées, respectivement, par les identifieurs id_1, id_2, id_3 et id_4. La table de scrutation de l'arbitre a un macro-cycle de durée $T_{MC} = 80ms$ découpé en quatre micro-cycles de durée $T_{mc} = 20ms$, chacun des micro-cycles comportant, pendant la fenêtre périodique, l'émission d'un ou plusieurs identifieurs de trafic lié aux fonctions 1 et 2.

4. Mise en oeuvre de l'arbitrage d'accès dans l'arbitre

Les mécanismes de l'arbitrage d'accès sont représentés sur la figure 3.6.

L'ordonnancement durant un micro-cycle, en considérant uniquement le transfert de variables (on ne considère pas la messagerie) consiste en:

1- La mise en oeuvre de la fenêtre périodique qui consiste, successivement, en la consultation de la séquence élémentaire des identifieurs de variables périodiques et de requêtes périodiques, la mise en circulation de ces identifieurs à travers,

respectivement, les trames *id_dat* et *id_rq*, et enfin, la prise en compte des réponses à ces trames.

Les réponses aux trames *id_dat* peuvent être de deux types:

a) soit ce sont des trames *rp_dat* qui ne véhiculent que les valeurs des variables périodiques;

b) soit ce sont des trames *rp_dat_rq* qui, outre les valeurs des variables périodiques, véhiculent également des requêtes (ces requêtes sont stockées dans les files de "demandes apériodiques" qui peuvent avoir le caractère urgent ou normal); c'est ce mécanisme qui initie ce qui nous avons appelé la *scrutation indirecte de variables apériodiques* et qui sera réalisé dans la fenêtre apériodique.

Les réponses aux trames *id_rq* sont des trames *rp_rq* qui véhiculent des identifieurs de variables apériodiques (ces identifieurs sont stockés dans le buffer de reprise et vont être mis en circulation à travers des trames *id_dat* dans la fenêtre périodique); cet aspect représente ce que nous avons appelé la *scrutation directe de variables apériodiques*.

2- La mise en oeuvre de la fenêtre apériodique, c'est-à-dire:

a) la prise en compte des requêtes des files de "demandes apériodiques" et la mise en circulation de trames *id_rq* véhiculant les identifieurs de ces requêtes; les trames réponses (*rp_rq*) véhiculent des identifieurs de variables apériodiques qui sont stockés dans la file "apériodiques en cours".

b) la prise en compte des identifieurs de la file "apériodique en cours" et la mise en circulation de trames *id_dat* contenant ces identifieurs.

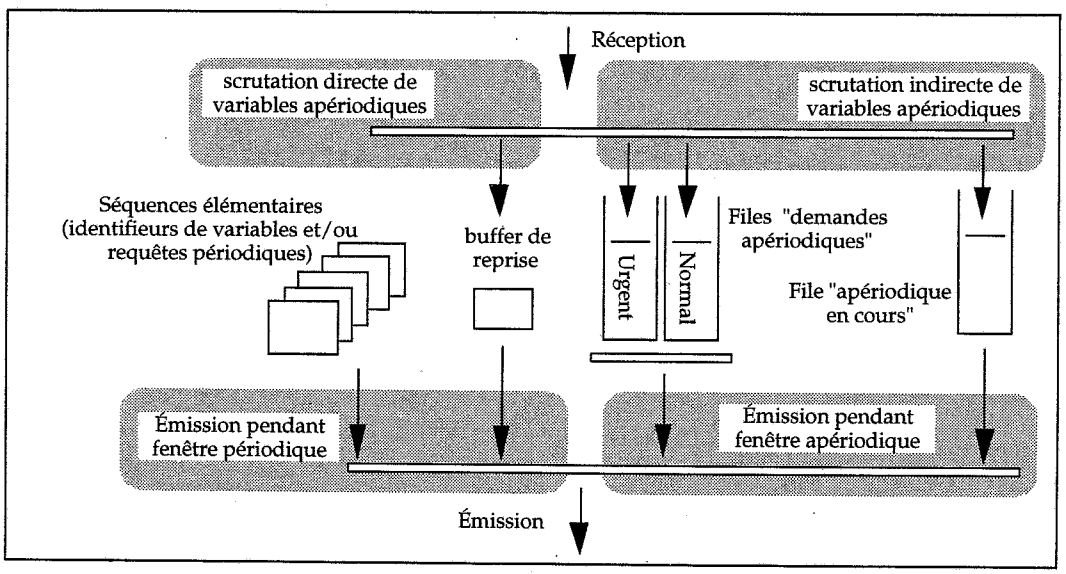

Figure 3.6: Les mécanismes de l'arbitrage d'accès dans FIP

Remarque 1: Le déroulement de la fenêtre apériodique d'un micro-cycle comporte les phases suivantes:

- tout d'abord, on traite toute la "file apériodique en cours", c'est-à-dire, on fait circuler les identifieurs de variables apériodiques que l'on n'avait pas pu faire circuler lors de la fenêtre apériodique du micro-cycle précédent;
- ensuite, lorsque la file "apériodique en cours" devient vide, on a l'exécution d'une demande stockée dans la file "demandes apériodiques" urgent, s'il y en a, ou normal

dans le cas contraire; rappelons qu'à la suite de ces exécutions, on aura de nouveaux identifieurs qui arriveront dans la file "apériodique en cours".

Remarque 2: L'ordonnancement du trafic périodique et du trafic apériodique suivant la technique de scrutation directe est garantie au préalable. Par contre, l'ordonnancement des variables apériodiques urgentes, suivant la technique de la scrutation indirecte, doit être évalué. En effet, le transfert de ce trafic dans la fenêtre apériodique est faiblement lié (du point de vue temporel) avec les besoins des transferts. Nous étudions ce problème dans le chapitre IV.

5. Scénarios d'échange de variables

Nous schématisons les principaux exemples de scénarios d'échange de variables:

1: La scrutation périodique de variables périodiques, qui est la fonction de base (figure 3.7);

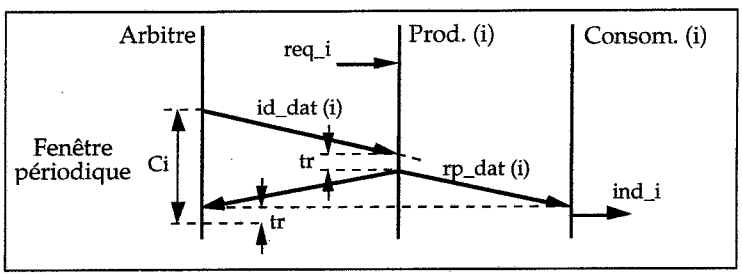

Figure 3.7: Scrutation périodique de variables périodiques

L'arbitre envoie périodiquement un PDU *id_dat* adressé au producteur de la variable du flux périodique Mp_i (variable identifiée par i et que nous appelons variable i); comme réponse (après la durée t_r fixée par le temps de retournement de la station), le producteur diffuse le PDU *rp_dat* (qui contient la valeur de la variable i), qui sera consommé par tous les consommateurs de la variable i.

La durée maximale d'utilisation du réseau C_i comprend, en particulier, la durée $2 \cdot t_r$, c'est-à-dire, outre le temps de retournement du producteur, on a également le temps de retournement de l'arbitre afin que ce dernier puisse démarrer une nouvelle scrutation.

2: La scrutation directe de variables apériodiques (figure 3.8);

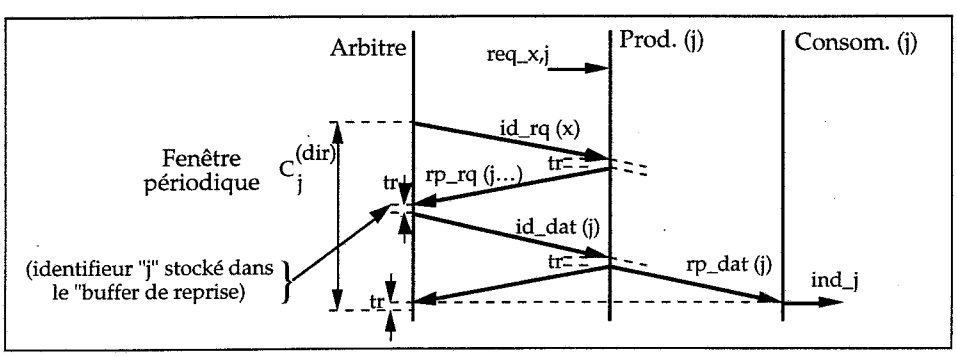

Figure 3.8: Scrutation directe de variables apériodiques

L'arbitre envoie périodiquement un PDU *id_rq* adressé à un identifieur de requête x de la station à scruter; comme réponse, la station lui renvoie (dans un PDU *rp_rq*) la liste des flux apériodiques $(j...)$ qui ont des variables en attente; cette liste est stockée dans le "buffer de reprise", et immédiatement après, l'arbitre déclenche le transfert de

toutes les variables en attente (pendant la fenêtre périodique en cours), en envoyant des PDUs *id_dat* adressés à chacune de ces variables $(j...)$.

La durée maximale d'utilisation du réseau pour le transfert d'une variable apériodique j est notée $C_j^{(dir)}$.

3: La scrutation indirecte de variables apériodiques (figure 3.9);

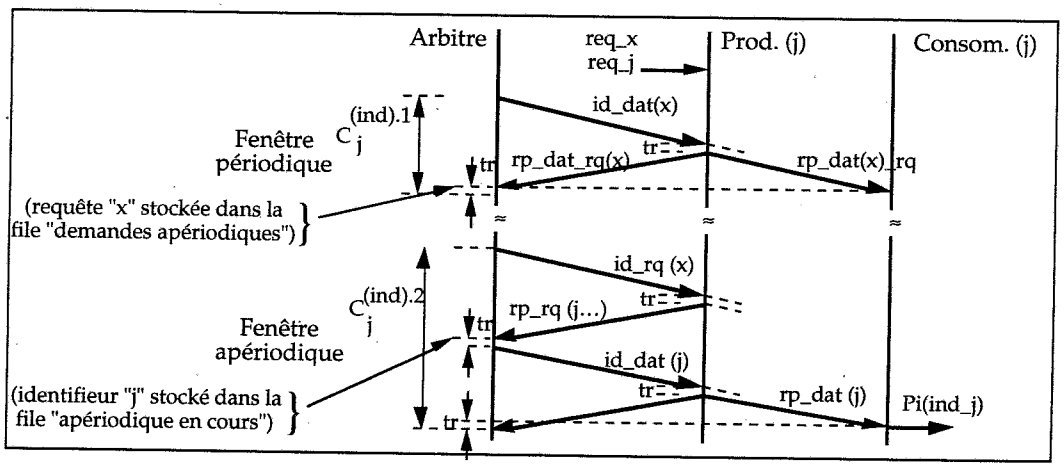

Figure 3.9: Scrutation indirecte de variables apériodiques

Pendant la scrutation périodique d'une variable x, la station scrutée peut aussi informer l'arbitre sur l'existence de trafic apériodique en attente (par l'intermédiaire d'un fanion *rq* attaché au PDU *rp_dat* qui véhicule la valeur de la variable x); cette information sera stockée par l'arbitre dans la "file de demandes apériodiques" sous la forme de l'identifieur x.

Ensuite, pendant une fenêtre apériodique, l'arbitre envoie la trame *id_rq(x)* qui suscite une trame réponse *rp_rq(j...)* spécifiant la liste des identifieurs *(j...)* de variables apériodiques qui ont des données prêtes (ces identifieurs sont stockés dans la file "demandes apériodiques en cours"). Enfin, l'arbitre fait la scrutation de ces variables *(j...)*.

La durée maximale d'utilisation du réseau pour le transfert d'une variable apériodique j est notée C_j^{ind} avec $C_j^{ind} = C_j^{(ind).1} + C_j^{(ind).2}$.

3.4. Le réseau DQDB

1. Généralités

Le réseau DQDB a été adopté par l'organisme de normalisation IEEE comme une norme (IEEE 802.6) pour les réseaux métropolitains[IEEE, 90]. L'architecture du réseau est composée d'une paire de bus unidirectionnels, d'une série de noeuds connectés aux deux bus et de deux générateurs de slots (intervalles de temps) à l'extrémité des bus (figure 3.10). Les générateurs de slots du bus A et du bus B sont appelés respectivement HOB(A) et HOB(B). Les HOB(A) et HOB(B) génèrent des slots de taille fixe.

Deux mécanismes pour contrôler l'accès au bus sont définis: "Queue Arbitrated" (QA) et "Pre-Arbitrated" (PA). Le premier est réservé au trafic asynchrone (données). Le deuxième est réservé au trafic isochrone (voix, vidéo). Les générateurs des slots des bus A et B génèrent donc deux types de slots (slots QA et slots PA).

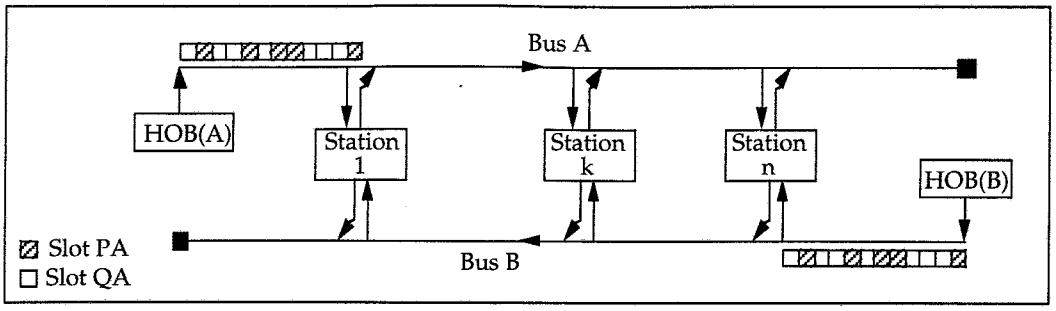

Figure 3.10: Le réseau DQDB

Le mécanisme QA (dont l'idée est de créer et de maintenir une file d'attente distribuée basée sur une stratégie FIFO) n'est pas compatible avec les exigences du trafic temps-réel [SSS, 92].

Le mécanisme PA, par contre, est prévu pour offrir des mécanismes temps-réel. Son principe est le suivant:

> Chaque générateur de slots génère sur son bus des slots PA en adéquation, en termes de distribution temporelle, avec les besoins de chaque flux de messages isochrones à échanger sur ce bus (les slots PA portent dans l'en-tête un identifieur de flux de messages; cet identifieur est appelé *numéro de circuit virtuel*); Un slot PA (avec un identifieur x) est utilisé en écriture par la station productrice du flux x et en lecture par les stations consommatrices du flux x.

La norme DQDB ne précise rien quant à l'ordonnancement du trafic PA. Peu de travaux ont été faits à ce sujet, si ce n'est les travaux de [SSMT, 93] et nos propres travaux [CVJ, 94] où nous avons proposé la mise en oeuvre dans chaque HOB d'un algorithme d'ordonnancement RM hors-ligne et un algorithme de changement de mode en-ligne (afin de prendre en compte l'évolution des besoins des échanges des flux isochrones). L'algorithme de changement de mode est une contribution originale que nous présentons dans le chapitre IV.

Nous résumons maintenant la proposition de la mise en oeuvre d'un algorithme d'ordonnancement RM, et plus précisément la méthode d'assignation de priorités.

2. Assignation de priorités

Nous proposons un schéma d'assignation de priorités aux flux de messages temps-réel selon l'algorithme RM avec un double objectif:

a) chaque flux Mp_i (avec période P_i et durée d'utilisation maximale du réseau C_i) de trafic isochrone doit recevoir par période P_i le nombre de slots PA nécessaire pour satisfaire ses échéances (en supposant que l'on a n flux, la borne supérieure de l'utilisation maximale du réseau est $n \cdot \left(2^{1/n} - 1\right)$ et a la valeur *69%* lorsque $n \to \infty$).

b) si la borne supérieure de l'utilisation maximale du réseau n'est pas atteinte, on doit essayer de répartir la distribution de slots PA sur la totalité du cycle (afin d'éviter une concentration de ces slots PA au début du cycle, ce qui pénaliserait les accès au bus pour le trafic QA); pour ceci, on définit un flux fictif de période P_{QA} qui recevra donc des slots QA, tel que l'ensemble des n flux isochrones Mp_i et de ce flux fictif atteigne la borne de l'utilisation du réseau.

Nous résumons maintenant l'évaluation du test d'ordonnançabilité de l'ensemble des n flux isochrones Mp_i et le calcul de la période P_{QA}:

- test d'ordonnançabilité de l'ensemble des n flux isochrones Mp_i (l'utilisation U_i demandée par un flux Mp_i est égale à C_i/P_i):

  Appelons P_{PA_i} la période de génération des slots PA (durée unitaire) associés au flux Mp_i: l'utilisation U_{PA_i} du bus pour ce flux est égale à $1/P_{PA\_i}$. On doit donc garantir que: $U_{PA\_i} \geq U_i$, soit $P_{PA\_i} = \lfloor P_i/C_i \rfloor$.

  Pour l'ensemble de flux isochrones Mp_i, le test d'ordonnançabilité doit être:

$$\sum_{i=1}^{n} \frac{1}{P_{PA\_i}} < n \cdot \left(2^{1/n} - 1\right) \qquad (3.5)$$

  Notons bien que l'on doit avoir une relation de type "strictement inférieur" afin de pouvoir considérer l'objectif b).

  Notons encore que si la durée des slots est l'unité de temps, l'ordonnancement effectué à l'aide d'un algorithme préemptif est valable, tant que la durée des périodes est un multiple de la durée des slots (condition nécéssaire pour établir une équivalence entre ordonnancement discret et ordonnancement continu [BRH, 90]); la durée d'un slot fixe la granularité temporelle du système.

- calcul de la période P_{QA}:

  Si le test d'ordonnançabilité pour l'ensemble des n flux isochrones Mp_i est respecté, on peut évaluer la période P_{QA} de la manière suivante:

$$\sum_{i=1}^{n} \frac{1}{P_{PA\_i}} + \frac{1}{P_{QA}} \leq (n+1)\left(2^{1/(n+1)} - 1\right) \text{ , d'où } P_{QA} = \left\lfloor 1 \Big/ \left[(n+1) \cdot \left(2^{1/(n+1)} - 1\right) - \sum_{i=1}^{n} 1/P_i \right] \right\rfloor \qquad (3.6)$$

3. Arbitrage de priorités

Cet arbitrage est effectué pour chaque bus dans la station HOB (génération de slots PA avec des identifieurs).

3.5. Le protocole IEEE 802.5

1. Généralités

Le protocole MAC IEEE 802.5 [IEEEb, 85] définit, sur une structure en anneau, un arbitrage d'accès global basé sur un mécanisme de passage de jeton. Toute station x qui capte le jeton, émet une trame de données à destination d'une station y et c'est la station x qui retire la trame de données de l'anneau avant de régénérer le jeton.

La norme IEEE 802.5 définit huit niveaux de priorités pour des messages, ce qui permet donc de prévoir la mise en oeuvre d'un ordonnancement basé sur l'assignation de priorités aux flux, car le mécanisme de passage de jeton garantit que le message le plus prioritaire en attente dans tous les flux sera celui qui aura le droit d'accès accordé. Cependant, compte tenu du nombre réduit de niveaux de priorité, on doit, dans le cas le plus général, associer à chaque niveau de priorité plusieurs flux de messages (avec des priorités algorithmiques différentes), ce qui génère obligatoirement des inversions de priorité indésirables.

Quelques travaux ont été faits sur la mise en oeuvre effective d'un ordonnancement et, plus particulièrement, sur des modalités d'assignation de priorités [SML, 88] [Ple, 92].

2. Assignation de priorités [SML, 88]

Les auteurs proposent l'utilisation de l'algorithme RM pour effectuer l'assignation de priorités. Dans leurs travaux, ils présentent également le calcul du test d'ordonnançabilité ainsi que le calcul de la durée maximale d'utilisation du réseau (C_i) pour transférer un message périodique (flux Mp_i).

Appelons T_{DATA} la durée d'une trame de données, T_{TOKEN} la durée du jeton et W_T la durée d'un tour de l'anneau. La durée d'utilisation du réseau C_i a les expressions suivantes:

- Si la durée d'un tour de l'anneau est inférieure ou égale à la durée de la trame de données [SML, 88], C_i est:

$$C_i = T_{DATA} + T_{TOKEN} + W_T \tag{3.7}$$

- Si la durée d'un tour de l'anneau est strictement supérieure à la durée de la trame de données [Ple, 92], C_i est:

$$C_i = T_{TOKEN} + 2 \cdot W_T + T_{FDS} \tag{3.8}$$

où T_{FDS} représente la durée de transmission des champs de contrôle de la trame de données.

Le test d'ordonnançabilité est une application directe de l'algorithme RM [LL, 73].

3. Arbitrage de priorités

Cet arbitrage est décrit dans la procédure suivante. Considérons d'abord les variables:

$P(jeton)$: priorité actuelle du jeton;

$P(dernier\_jeton)$: priorité du dernier jeton capturé par la station;

$P(réservation)$: priorité de réservation du jeton;

$P(message)$: priorité la plus élevée parmi les priorités des messages en attente dans la station.

$S(jeton)$: statut du jeton (libre / occupé).

Algorithme d'arbitrage de priorités du protocole IEEE 802.5

Dans chaque station, à l'arrivée du jeton, effectuer:
Tant qu'il y a des messages en attente:
    Si $P(jeton) \leq P(message)$ et $S(jeton)$ = "libre":
        $P(dernier\_jeton) \leftarrow P(jeton)$
        $S(jeton) \leftarrow$ "occupé";
        Tant que $P(message) \geq P(réservation)$:
            $P(jeton) \leftarrow P(message)$;
            Envoi message;
            Re-évaluer $P(message)$;
        $P(jeton) \leftarrow max\{P(dernier\_jeton), P(réservation)\}$;
        $S(jeton) \leftarrow$ "libre";

    Sinon $P(jeton) > P(message)$ ou $S(jeton)$ = "occupé":
        Si $P(message) > P(réservation)$:
            $P(réservation) \leftarrow P(message)$;

4. Critique

Les auteurs [SML, 88] appliquent le test d'ordonnançabilité de l'algorithme RM préemptif [LL, 73] à la transmission des messages, alors que, par définition, la transmission ne peut être préemptée au sens des tâches (un message est transmis ou n'est pas transmis) et donc on a forcément des inversions de priorité. Cependant, sachant que la condition de préemption au sein des tâches ne peut pas être respectée, ils proposent une fragmentation des messages de manière à limiter la durée des inversions de priorité (le choix du degré de fragmentation résulte d'un compromis entre l'efficacité du codage du transfert des fragments (taille des informations de contrôle par rapport à la taille des données) et la durée de l'inversion de priorité).

Il n'empêche que, malgré la fragmentation, l'application d'un test d'un algorithme préemptif au transfert de messages peut conduire à des réponses fausses: le test dit que l'ensemble de flux de message est ordonnançable et pratiquement (c'est-à-dire, en considérant la non-préemption) on ne peut pas l'ordonnancer. Donc, quel que soit le résultat du test, on ne peut jamais être sûr de l'ordonnançabilité de l'ensemble.

Considérons l'exemple suivant, où nous avons un ensemble de flux de messages temps-réel $M = \{M_1, M_2, M_3, M_4\}$, qui a les caractéristiques suivantes: $P_i = \{100, 100, 100, 500\}$ et $C_i = \{20, 20, 20, 60\}$. Le test d'ordonnançabilité de l'algorithme RM dit que l'ensemble des flux est ordonnançable ($U = 0.72 < n\left(2^{1/n} - 1\right) = 0.7568$).

Cependant, nous pouvons constater que, pour un déphasage particulier entre les flux de messages (le message de plus faible priorité M_4 démarre son transfert et juste après des messages de plus haute priorité émettent des requêtes pour être transmis, mais devront atteindre la fin de la transmission du message M_4), l'ensemble M n'est pas ordonnançable (l'échéance de M_3 n'est pas respectée); ceci est visualisé sur la figure 3.11 (la partie a) représente le cas "sans fragmentation"; la partie b) représente le cas "avec fragmentation", la longueur maximale de fragmentation étant fixée à 50 et donc seul le message de M_4 est fragmenté).

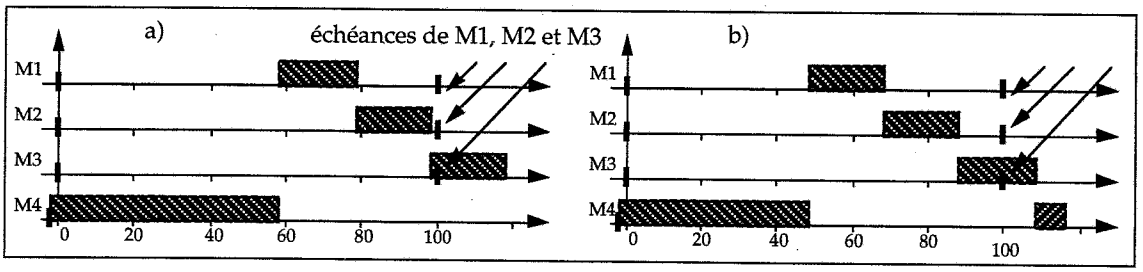

Figure 3.11: Exemple de non ordonnançabilité en utilisant l'algorithme RM

Un test d'ordonnançabilité acceptable est celui de l'algorithme RM non-préemptif que nous avons proposé au chapitre II et qui donne le résultat suivant dans le cas de l'exemple traité:

- pas de fragmentation: $U = 0.72 + \frac{60}{100} = 1.32 > 0.7568$
- avec fragmentation: $U = 0.72 + \frac{50}{100} = 1.22 > 0.7568$

4. Protocoles de la "classe 2"

4.1. Principales normes

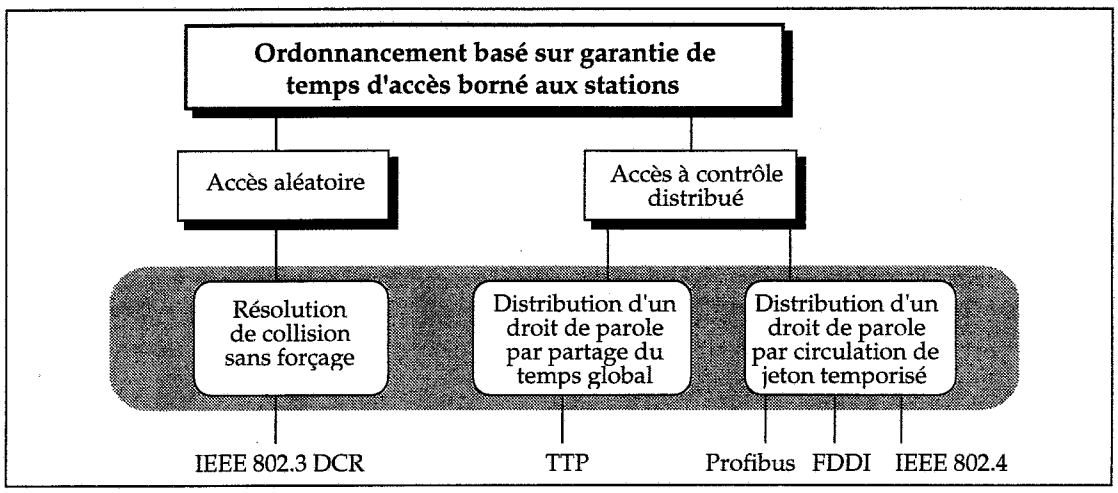

Figure 3.12: Les protocoles "classe 2" existants

Les principales normes de "classe 2" existantes sont représentées sur la figure 3.12.

Nous présentons maintenant les éléments essentiels de l'arbitrage d'accès et du contrôle de transmission dans le cas de chaque protocole.

4.2. Le protocole IEEE 802.3 DCR

1. Principe

Ce protocole, qui s'intègre dans le cadre de la norme IEEE 802.3 [IEEEc, 85], y apporte la valeur ajoutée de la garantie d'un délai d'accès borné en modifiant l'algorithme de résolution des collisions.

- L'arbitrage d'accès est élémentaire tant qu'il n'y a pas de collisions; à l'instant où il y a une collision, le protocole utilise le principe de résolution d'une *époque temporelle* au moyen d'un arbre binaire, qui associe des priorités soit aux stations [Rol, 89], soit aux flux de messages des stations [ARS, 90]. C'est ce mécanisme qui permet d'obtenir un délai d'accès borné. Nous considérons ici uniquement le mode général à entrées bloquées [Rol, 91].
- Le contrôle de transmission est élémentaire (on transmet un message par époque).

2. Comparaison entre les techniques d'association de priorités aux stations et aux flux de messages

Considérons que l'on a m stations et n flux de messages (généralement $n >> m$).

Si on associe des priorités aux stations [Rol, 89] [ARS, 90], la taille de l'arbre est minimale ($2^{\lceil log_2 m \rceil}$) mais l'ordonnancement ne tient pas compte des priorités des flux de messages, donc de leurs caractéristiques temporelles, ce qui n'est pas totalement satisfaisant dans un contexte temps-réel.

Si on associe des priorités aux flux de messages [ARS, 90], la taille de l'arbre est maximale ($2^{\lceil log_2 n \rceil}$), ce qui donne une longue époque temporelle et donc peut induire des inversions de priorité de longue durée (un flux de messages plus prioritaire que ceux qui sont impliqués dans la résolution de l'époque temporelle doit attendre la fin de cette époque pour essayer d'accéder à la ressource de transmission).

Compte tenu de ces constations, on peut définir [ARS, 90] un schéma intermédiaire de priorités associées aux messages (q niveaux de priorités avec $m < q < n$), ce qui peut donc donner une même priorité associée à des flux de messages de différentes stations). Dans ce cas, la résolution de l'époque peut se voir comme une opération à deux niveaux: tout d'abord on résout l'époque sur la base de la priorité des messages; si une collision subsiste, lorsqu'on a atteint les feuilles de l'arbre binaire, on résout la contention sur la base des adresses des stations.

De toute manière, comme la durée des inversions de priorité est proportionnelle à la taille de l'arbre, on doit donc, pour garantir une bonne efficacité temps-réel, trouver un compromis entre la taille de l'arbre et la prise en compte des caractéristiques temporelles des messages.

3. Arbitrage d'accès

Cet arbitrage est décrit dans la procédure suivante:

```
Algorithme d'arbitrage d'accès à époque temporelle

Dans chaque station, tant qu'il y a des messages en attente:
    Si le canal de communication est libre:
            L'accès est accordé immédiatement;

    Si le canal de communication est occupé:
            L'algorithme d'arbitrage d'accès est re-exécuté à l'instant où le
            canal de communication redevient libre;
    Tant que le message est en cours d'émission:
        Si une collision dans le canal de communication est détectée:
            Le transfert du message est suspendu;
            La station envoie une séquence de brouillage;
            Une époque temporelle est déclarée ouverte;
            Un algorithme de résolution de collision garantit que le message
                intervenant dans cette collision sera transmis pendant un
                intervalle de temps borné;
            L'époque temporelle est déclarée fermée;
```

4. Remarque

En considérant la modalité d'assignation de priorités aux flux de messages, on doit se poser la question du classement de ce protocole dans la classe 2 et non dans la classe 1. La raison en est que, du fait du concept d'époque temporelle en mode à entrées bloquées, on n'a pas un ordonnancement global des messages (seuls les premiers messages de chaque station sont ordonnancés suivant le schéma de priorité; on peut avoir, par exemple, la situation où le deuxième message d'une station est plus prioritaire que les premiers messages des autres stations et qui ne pourra donc être ordonnancé que dans la fenêtre temporelle suivante). Une fenêtre temporelle ne garantit qu'un délai d'accès borné aux messages de la station.

4.3. Le protocole TTP

1. Généralités

Le protocole TTP [K_all, 89] qui a été défini dans le cadre du système d'exploitation MARS, implante une méthode d'accès à contrôle distribué sur un bus. Ce protocole est une implémentation très particulière du protocole TDMA ("Time Division Multiple Access") classique, où chaque station possède une vue de l'horloge globale du système.

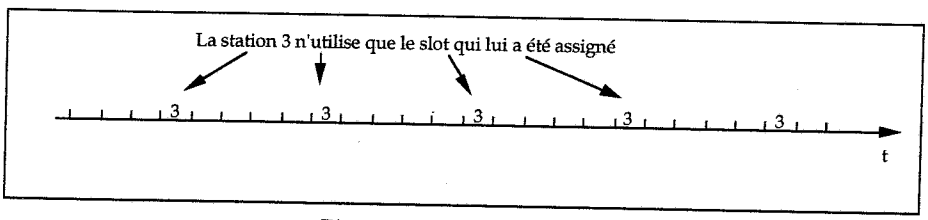

Figure 3.13: Protocole TTP

2. Arbitrage d'accès

Dans ce protocole, le "temps global" est divisé (hors-ligne) en une séquence de slots, chaque slot étant pré-assigné à une seule station; cette séquence définit la table de scrutation du protocole (figure 3.13). Dans le cas le plus simple, le nombre de slots assignés dans la table de scrutation est égal au nombre de stations, et donc le temps global est équitablement partagé; cependant, quand les besoins des stations ne sont pas identiques, un nombre différent de slots sera assigné à chaque station dans la table de scrutation.

Toutes les stations, connaissant la table de scrutation et ayant une référence temporelle commune, savent quand elles peuvent utiliser le réseau (évidement, par le principe même, un slot est perdu s'il n'y a pas de messages à transmettre).

Une particularité intéressante de ce protocole est la suivante: dans chaque station, la différence entre l'instant d'arrivée réel d'un message et l'instant d'arrivée prévu pour ce même message, donne une indication sur la différence existante entre les horloges des deux stations; cette indication est suffisante pour l'exécution d'un algorithme distribué de synchronisation d'horloges (fondamental pour le fonctionnement correct du protocole, puisque l'accès est à contrôle distribué).

3. Contrôle de transmission

Une station ne peut transmettre qu'un message par slot.

4.4. Les protocoles basés sur le jeton temporisé: IEEE 802.4, FDDI et Profibus

L'arbitrage d'accès est tout simple: on a un passage séquentiel du jeton entre les stations. Le contrôle de transmission est le mécanisme fondamental de ces protocoles car c'est lui qui peut garantir le délai d'accès borné. Ce contrôle de transmission est basé sur le protocole du "jeton temporisé" [Gro, 82], que nous allons détailler maintenant. Nous proposerons au chapitre IV une modification de ce protocole dans l'optique de meilleures performances temps-réel.

Ensuite, nous indiquons, en prenant ce protocole comme référence, les particularités des trois normes IEEE 802.4, FDDI et Profibus qui reprennent la philosophie du protocole "jeton temporisé".

4.4.1. Le protocole "jeton temporisé"

1. Principe

Ce protocole est basé sur la considération de deux types de trafic: le trafic synchrone (appelé "synchronous class") qui désigne un trafic avec des contraintes temporelles et le trafic asynchrone (appelé "asynchronous class") qui désigne un trafic non contraint temporellement.

Le trafic synchrone est considéré comme le trafic de base. L'idée fondamentale de ce protocole est:

- d'une part, d'allouer à chaque station une durée pré-définie qui représente le temps maximum pendant lequel elle peut transmettre du trafic synchrone chaque fois qu'elle reçoit le jeton;
- d'autre part, de contrôler la durée de rotation du jeton ou temps de cycle (intervalle de temps entre deux arrivées consécutives du jeton à une station), afin que les contraintes temporelles du trafic synchrone puissent être satisfaites.

Ici, nous considérerons que les trafics temps-réel (périodique ou apériodique) et non-temps-réel sont véhiculés, respectivement, par le trafic synchrone et par le trafic asynchrone.

Un paramètre fondamental, négocié dans la phase d'initialisation d'un réseau, est le paramètre T_{TRT} ("Target Token Rotation Time") qui définit le temps de rotation cible du jeton (temps vers lequel le temps de rotation doit tendre en moyenne). C'est par rapport à ce paramètre que l'on définit le temps maximum pendant lequel une station peut transmettre du trafic temps-réel chaque fois qu'elle reçoit le jeton (ce temps maximum est encore appelé la bande passante allouée à la station). Plus précisément, en supposant que l'on a m stations dans un réseau, on spécifie la relation suivante:

$$\sum_{k=1}^{m} H_k \leq T_{TRT} - \tau \tag{3.10}$$

où H_k est la bande passante allouée à la station k et τ est la partie de T_{TRT} non allouable aux stations (on a $\tau = \theta + \Delta$, θ étant la latence totale de l'anneau logique et Δ représentant le surplus qui dépend du protocole (durée du PDU "jeton", "asynchronous overrun", etc)).

Cette relation, qui peut encore s'écrire

$$\sum_{k=1}^{m} H_k \leq T_{TRT}(1-\alpha), \qquad \alpha = \tau / T_{TRT} \tag{3.11}$$

est la relation traduisant la **contrainte du protocole**, c'est-à-dire, la somme des allocations de bande passante pour le trafic temps-réel doit être inférieure à T_{TRT}.

En ce qui concerne le trafic non-temps-réel, ce protocole prévoit qu'une station peut en transmettre si le jeton lui revient au bout d'un temps inférieur à T_{TRT}.

La mise en oeuvre de ce protocole dans une station k est basé sur l'utilisation de trois compteurs: TRT_k ("Token Rotation Time") qui évalue le temps de cycle (ce compteur est initialisé à la valeur T_{TRT} et re-initialisé à cette valeur T_{TRT} soit lorsque le jeton arrive à la station, soit lorsque le temps T_{TRT} est écoulé); THT_k ("Token Holding Time") qui définit le temps pendant lequel la station k peut transmettre du trafic non-temps-réel; LC_k ("Late Counter") qui mémorise le nombre de fois où le temps T_{TRT} s'est écoulé depuis la précédente arrivée du jeton à la station k.

L'algorithme du protocole est indiqué sur la procédure suivante:

```
Algorithme "jeton temporisé" [Gro, 82]

Dans chaque station k, (k = 1, 2, ..m):
    THT_k ← 0 ;
    LC_k ← 0 ;                                   /*procédure d'initialisation */
    TRT_k ← T_TRT ;
    Démarrer TRT_k (compteur décrémental)

Tant que le réseau est en fonctionnement:
    si TRT_k = 0, faire:                                  /* attente du jeton*/
        TRT_k ← T_TRT ;
        LC_k ← LC_k + 1

    À l'arrivée du jeton, faire:                    /* transfert de données */
        cas LC_k = 0 :                               /* cas jeton en avance */
            THT_k ← TRT_k ;
            TRT_k ← T_TRT
            Démarrer compteur TRT_k (décrémental)
            transfert de messages temps-réel pendant jusqu'à H_k unités de temps
            Démarrer compteur THT_k (décrémental)
            tant que THT_k > 0 et (mgs non-temps-réel en attente)
                transfert de messages non-temps-réel
            passage du jeton à la station (k+1)
        cas LC_k = 1 :                               /* cas jeton en retard */
            LC_k ← 0
            transfert de messages temps-réel pendant jusqu'à H_k unités de temps
            passage du jeton à la station (k+1)

        cas LC_k > 1 :                                        /* cas erreur */
            procédure "récupération d'erreur"
```

Les points suivants doivent être remarqués:

- on a les notions d'arrivée du jeton en avance ($LC_k = 0$) ou en retard ($LC_k = 1$, le cas $LC_k > 1$ étant considéré comme une erreur (par exemple: perte du jeton)).
- quelle que soit la condition du jeton lorsqu'il arrive dans une station (en avance ou en retard), on transfère toujours le trafic temps-réel (s'il y en a) pendant un temps égal au maximum à H_k).
- le trafic non-temps-réel n'est pas transmis si le jeton est en retard.
- le retard du jeton est engendré par une durée trop longue d'envoi de trafic non-temps-réel par des stations.

Le phénomène de la potentialité d'un retard dans le passage du jeton est un aspect très important que nous présentons maintenant, en considérant le cas le plus dur, c'est-à-dire, qui induit la plus longue durée pour le temps de cycle du jeton (cas pire). Prenons le cas limite pour la contrainte de protocole $\left(\sum_{k=1}^{m} H_k = T_{TRT}(1-\alpha)\right)$ et considérons le scénario où d'une part toutes les stations ont du trafic temps-réel à transmettre en permanence sur toute leur bande passante et, d'autre part, la première station utilise tout le temps dont elle dispose pour transmettre du trafic non-temps-réel (elle peut en transmettre pendant un temps T_{TRT}).

Nous donnons sur la figure 3.14 le diagramme temporel représentant l'occupation du temps pour des échanges sur le réseau (on considère 3 stations).

On a: $T_{CYCLE_{max}} = \sum_{k=1}^{m} H_k + T_{TRT} + \tau$ avec $\sum_{k=1}^{m} H_k = T_{TRT} - \tau$, soit $T_{CYCLE_{max}} = 2 \cdot T_{TRT}$ et donc:

$$T_{CYCLE} \leq 2 \cdot T_{TRT} \tag{3.12}$$

Cette relation $T_{CYCLE} \leq 2 \cdot T_{TRT}$ est une **propriété temporelle** du protocole "jeton temporisé".

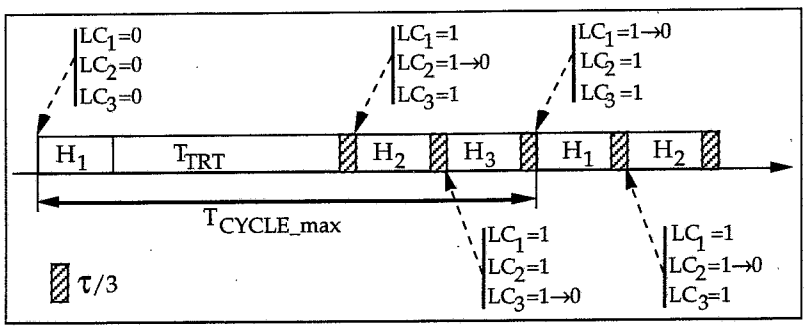

Figure 3.14 Diagramme temporel

Remarque 1: Dès que l'on vient d'entrer dans ce "cas le plus dur", on peut dire que tant que toutes les stations utilisent toute la bande passante qui leur est attribuée pour le trafic temps-réel (H_k pour la station k), on ne pourra plus faire circuler du trafic non-temps-réel. Ce n'est que si les stations n'utilisent plus la totalité de leur bande passante H_k que le jeton pourra de nouveau être en avance et donc du trafic non-temps-réel pourra être retransmis de nouveau.

Remarque 2: Si toutes les stations respectent la contrainte de protocole, on peut voir que l'on n'aura jamais la valeur $LC = 2$, c'est-à-dire, une situation d'erreur.

2. Prise en compte des caractéristiques des flux de messages temps-réel

Dans le paragraphe précédent, nous avons exposé les caractéristiques du protocole du "jeton temporisé" en termes d'allocation de bande passante aux stations pour le trafic temps-réel, mais cette présentation a été transparente aux caractéristiques des flux de trafic à transmettre. Il faut donc maintenant préciser quelles sont les relations que l'on doit avoir entre les caractéristiques des flux temps-réel et les caractéristiques du protocole afin que les échéances des flux temps-réel soient satisfaites.

Nous considérons, dans la suite de cette présentation, que l'on a un flux périodique par station (comme dans les travaux de [ACZD, 92]) et nous considérons donc un ensemble de m flux de messages périodiques $\{M_i\}$, $1 \leq i \leq m$, avec $M_i = (P_i, C_i)$, où P_i est la période du flux M_i et C_i est la durée maximale d'utilisation du réseau nécessaire à ce flux M_i par période. La bande passante allouée à la station qui génère le flux M_i est appelée H_i. Les échéances sont sur requête.

Afin que les contraintes des échéances de tous les flux M_i soient satisfaites, il faut que:

1- tout d'abord, le jeton passe dans la station pendant la période P_i;

2- ensuite, le temps minimum X_i dont dispose la station pour transmettre le flux M_i sur la totalité des passages du jeton pendant la période P_i, soit suffisant pour assurer la durée nécessaire d'utilisation du réseau.

La condition 1 implique que la période P_i soit supérieure au temps maximum de cycle, donc:

$$\forall i,\ 1 \leq i \leq m,\ P_i \geq 2 \cdot T_{TRT} \tag{3.13}$$

Le calcul de la condition 2 nécessite d'évaluer le nombre minimum v_i de passages du jeton dans la station pendant une période P_i. Pour faire cette évaluation, il faut, d'une part, prendre en compte l'asynchronisme entre les requêtes des flux de messages et les passages du jeton et

donc se placer dans le cas pire de déphasage et, d'autre part, considérer la circulation du jeton dans le cas le plus dur de retard (figure 3.15).

Le nombre minimum v_i de passages du jeton pendant la période P_i se déduit du cas représenté sur la figure: $v_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor$, d'où on obtient: $X_i = v_i \cdot H_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor \cdot H_i$. La condition 2 implique que le temps X_i soit supérieur ou égal à C_i, soit:

$$\left\lfloor \frac{P_i}{T_{TRT}} - 1 \right\rfloor \cdot H_i \geq C_i \tag{3.14}$$

Cette condition 2 exprime ce que l'on appelle la **contrainte de l'échéance.**

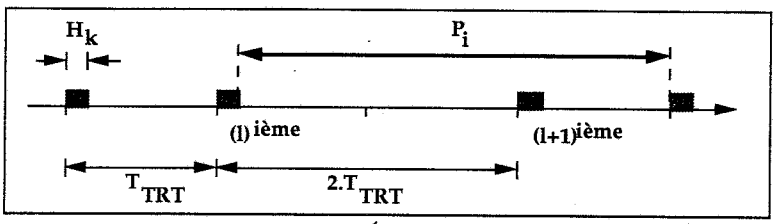

Figure 3.15: Évaluation de v_i

3. La problématique de l'allocation de la bande passante

Un algorithme d'allocation de bande passante, doit évaluer, en fonction de T_{TRT}, l'ensemble $\vec{H} = \{H_1, H_2, \ldots H_i, \ldots H_n\}$ des allocations de bande passante aux différentes stations, sachant que seules les caractéristiques temporelles (C_i, P_i) des flux de messages sont connues au préalable.

L'évaluation de cet ensemble $\vec{H}$ doit être effectuée en deux étapes:

- D'abord, une valeur doit être choisie pour T_{TRT}, de manière à ce que le jeton passe dans chaque station au moins une fois pendant la période de chaque flux M_i: donc il faut que $v_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor \geq 1$. Ceci implique le choix d'une valeur pour T_{TRT} telle que: $\forall i,\ 1 \leq i \leq m,\ P_i \geq 2 \cdot T_{TRT}$.
- Ensuite, l'utilisateur doit vérifier si l'ensemble des allocations de bande passante spécifié respecte les contraintes de protocole et de l'échéance (respectivement: $\left(\sum_{i=1}^{n} H_i \leq T_{TRT}(1-\alpha)\right)$ et $\left(\lfloor (P_i/T_{TRT}) - 1 \rfloor H_i \geq C_i\right)$), qui fournissent la garantie de respect des contraintes temporelles des flux de messages.

Nous présentons maintenant les principaux algorithmes d'allocation de bande passante.

4. Les principaux algorithmes d'allocation de bande passante

Les algorithmes proposés dans [ACZD, 92], [ACZ, 93] et [ACZD, 94] ont été définis dans le cadre d'un seul flux de messages temps-réel par station, et sont basés sur les connaissances, d'une part, des caractéristiques des flux de messages et, d'autre part, du temps T_{TRT} qui est défini à la configuration. Les principaux algorithmes ainsi définis sont:

- L'algorithme d'allocation *durée complète* qui alloue à un flux M_i une bande passante H_i égale à la durée C_i d'un message:

$$H_i = C_i \tag{3.15}$$

Avec cet algorithme, à l'arrivée d'un jeton, le message en attente, s'il y en a, est complètement transmis (il n'y a pas de fragmentation de messages).

- L'algorithme d'allocation *proportionnel* qui alloue à un flux M_i une bande passante proportionnelle à ses besoins d'utilisation du réseau. Sachant que $(T_{TRT} - \tau)$ représente le temps total maximum de cycle du jeton disponible pour le trafic temps-réel, on prend:

$$H_i = \frac{C_i}{P_i} \cdot (T_{TRT} - \tau) \tag{3.16}$$

Avec cet algorithme les messages du flux M_i sont toujours fragmentés (P_i étant toujours supérieure à $(T_{TRT} - \tau)$ puisque $P_i \geq 2 \cdot T_{TRT}$) et donc, à chaque arrivée du jeton, le flux M_i dispose d'un temps suffisant pour transmettre un seul fragment de chaque message temps-réel en attente. La durée de ce fragment est:

$$\frac{C_i}{P_i/(T_{TRT} - \tau)} \tag{3.17}$$

L'inconvénient de cet algorithme c'est qu'il essaye de diviser la durée du transfert d'un message pour un nombre non entier d'arrivées du jeton ($P_i/(T_{TRT} - \tau)$ n'est pas forcément un entier!).

- L'algorithme d'allocation *proportionnel adapté* est un algorithme d'allocation équivalent à l'algorithme proportionnel, mais qui prend en compte que le jeton n'arrive à la station pour servir le flux M_i qu'un nombre v_i entier de fois pendant l'intervalle $[t, t + P_i]$:

$$H_i = \frac{C_i}{\lfloor (P_i/T_{TRT}) - 1 \rfloor} \tag{3.18}$$

5. Mesure de performance des algorithmes d'allocation de bande passante

Les algorithmes d'allocation de bande passante sont généralement comparés au moyen de l'évaluation de la borne inférieure de l'utilisation maximale [ACZD, 92]. C'est une adaptation de l'évaluation proposée pour l'analyse de l'algorithme RM [LL,73].

La borne minimale U_x de l'utilisation maximale associée à un algorithme x d'allocation de bande passante est ainsi définie:

- si $\forall_M$, $U(M) \leq U_x$, alors M est ordonnançable par l'algorithme x.

L'intérêt d'une borne minimale U_x de l'utilisation maximale c'est que tant que on a une configuration de flux de messages temps-réel M, telle que $U(M) \leq U_x$, on sait que l'algorithme x garantit le respect des contraintes temporelles. Par contre, dans le cas contraire, il faut vérifier (donc calculer) si ces contraintes sont satisfaites.

Les algorithmes d'allocation présentés ont les bornes U_x suivantes [ACZD, 92] [ACZD, 94]:

- algorithmes *durée complète* et *proportionnel*: $U_x = 0\%$; dans ce cas, il faut donc toujours vérifier si les contraintes temporelles de la configuration M considéré sont satisfaites;
- algorithme *proportionnel adapté*: $U_x \rightarrow 33.3\%$ (α étant voisin de zéro).

4.4.2 Les particularités des protocoles FDDI, IEEE 802.4 et Profibus

Bien que le protocole du "jeton temporisé" puisse être vu comme le dénominateur commun des mécanismes des protocoles FDDI [ANSI, 90], IEEE 802.4 [IEEEa, 85] et Profibus [DIN, 91], il existe cependant des différences entre ces protocoles dont nous indiquons maintenant (sans rentrer dans le détail) les points essentiels. Nous nous focalisons sur les points suivants: la classification des messages; le concept de T_{TRT}; le trafic temps-réel; le trafic non temps-réel; le concept "temps maximal de cycle".

1. Classification des messages

Deux classes de trafic sont toujours considérées: le trafic synchrone ou haute-priorité (c'est-à-dire, celui qui peut supporter du trafic temps-réel) et le trafic asynchrone ou basse-priorité (c'est-à-dire, du trafic non-temps-réel). Dans le trafic asynchrone, plusieurs niveaux de priorité sont considérés (8 pour FDDI; 3 pour IEEE 802.4; 4 pour Profibus).

2. Le concept de T_{TRT}

FDDI et Profibus utilisent ce concept de manière similaire au protocole du "jeton temporisé", c'est-à-dire, une seule valeur pour T_{TRT} quelle que soit la classe de trafic considérée.

Par contre, IEEE 802.4 utilise le concept de T_{TRT} uniquement pour le trafic asynchrone, avec des valeurs différentes pour chaque niveau de priorité. Le temps de cycle du jeton est maintenant évalué indépendamment pour chaque niveau de priorité, et en fonction de la dernière fois que la station a pu transférer du trafic de cette priorité.

3. Le trafic temps-réel

A chaque arrivée du jeton (dans FDDI et IEEE 802.4) on dispose d'un temps pré-défini (H_k pour FDDI; *High Priority Token Holding Time* - T_{HP_THT} pour IEEE 802.4) pendant lequel on peut transférer du trafic temps-réel. En ce qui concerne Profibus, il n'y a aucun temporisateur associé spécifiquement au trafic temps-réel: à chaque arrivée du jeton, on peut toujours transférer au moins *1* message temps-réel (on ne peut transférer qu'un message temps-réel si la cible T_{TRT} est dépassée; si cette cible n'est pas dépassée, on peut transférer plusieurs messages (jusqu'à l'écoulement de la cible)).

4. Le trafic non-temps-réel

Dans FDDI et Profibus, la condition pour faire du transfert de trafic non-temps-réel est identique à celle présentée pour le protocole "jeton temporisé", c'est-à-dire, la station peut transférer ce trafic tant que la cible T_{TRT} n'est pas dépassée. En ce qui concerne les différents niveaux de priorités dans FDDI, chaque niveau dispose d'un temps maximum d'émission pré-défini (allocation de bande), tandis que, dans Profibus, on envoie tout le trafic d'un niveau avant de considérer le niveau inférieur.

Dans IEEE 802.4, la poursuite du transfert de trafic d'un niveau de priorité x (auquel est associé un T_{TRT_x}) dépend du temps de rotation effectif du jeton (T_{REJ}), c'est-à-dire, du temps écoulé depuis le précédent passage du jeton pour ce niveau de priorité: si, et tant que, $T_{REJ} \leq T_{TRT\_x}$ on peut effectuer des transferts; dans le cas contraire, on ne peut pas. Une analyse intéressante de ce fonctionnement est faite dans [PT, 89].

5. Concept de temps maximal de cycle

On a les résultats suivants:

- FDDI (identique au "jeton temporisé": $2 \cdot T_{TRT}$ [SJ, 87];
- IEEE 802.4: $\sum_{k=1}^{m} H_k + \max_x \{T_{TRT\_x}\}$ [MCV, 92];
- En ce qui concerne Profibus, à notre connaissance, il n'y a pas de résultats bien établis sur le temps maximal de cycle. Nous présentons notre propre réflexion sur ce protocole au chapitre IV.

5. Un protocole mixte: classe 1&2

5.1. Généralités

La proposition de norme ISA SP-50 / IEC-65C [IEC, 94] définit un protocole centralisé, où le contrôleur central, appelé l'*ordonnanceur actif de la ligne* ("LAS: Link Active Scheduler"), gère les accès des autres stations (appelées "DLE: Data Link Entity") au bus. De nombreuses possibilités d'adressage sont définies (CEP, SAP, groupes de CEPs et de SAPs), ce qui permet d'offrir l'accès au bus à différents types de sources d'information, que ce soit un flux d'une station, un groupe de flux ou la station globalement.

Trois PDUs sont définis, afin de permettre au LAS de distribuer trois types de droit de parole (on dit encore de déléguer un jeton (notion de jeton délégué) pour permettre à une station DLE d'utiliser le bus):

- type 1: Le PDU *CD* ("Compel Data") qui permet au LAS de forcer l'envoi immédiat d'un PDU qui est dans un DLE (l'envoi de ce PDU redonne aussitôt la main au LAS).
- type 2: Le PDU *ET* ("Execute Transaction"), qui permet au LAS de demander à un DLE d'exécuter immédiatement une transaction (séquence de deux PDUs, où l'envoi du deuxième PDU redonne aussitôt la main au LAS); notons que le premier PDU de la transaction peut être un PDU CD (ce qui permet un degré d'indirection dans le forçage d'envoi).
- type 3: Le PDU *ES* ("Execute Sequence"), qui permet au LAS de donner à un DLE un temps de droit de parole (un paramètre *ES duration* est dans ce PDU). L'envoi, par le DLE, du dernier PDU (c'est un PDU qui a un fanion particulier ou c'est un PDU EE ("End of Execution")) redonne la main au LAS.

Le LAS doit évidement posséder les informations concernant les besoins dans les DLEs, afin d'ordonnancer les échanges. Certains besoins (échanges périodiques) sont définis à la configuration et la séquence temporelle de ces échanges peut donc être connue du LAS, ce qui permet au LAS de les ordonnancer. D'autres besoins (échanges apériodiques) peuvent être connus de manière dynamique par le LAS (au moyen de PDUs qui lui redonnent la main).

Il est important de noter que les profils d'utilisation de cet ensemble de fonctionnalités ne sont pas définis dans la norme ISA SP-50 / IEC-65C. L'implantation de politiques d'ordonnancement temps-réel est encore ouverte. Cependant, cette norme ISA SP-50 / IEC-65C est riche en possibilités d'ordonnancements permis, comme nous l'indiquons maintenant. Dans le chapitre V, nous proposons précisément une architecture de communication temps-réel basée sur notre interprétation de cette norme.

5.2. Ordonnancements permis

Les droits de parole de type 1 et de type 2 définissent ce que nous appelons la technique du jeton délégué à réponse immédiate:

- le type 1 permet, par exemple, d'ordonnancer le transfert de flux de messages périodiques depuis leurs producteurs vers leurs consommateurs. Ce service est identique au service de scrutation périodique de variables de FIP.
- le type 2 permet (dans l'hypothèse où le premier PDU de la séquence est le PDU CD) d'ordonnancer des demandes périodiques de consommateurs à des producteurs.

La technique du jeton délégué à réponse immédiate peut donc être utilisée pour mettre en oeuvre un ordonnancement basé sur une assignation de priorités aux flux de messages (ordonnancement de classe 1, tel comme nous l'avons défini).

Le droit de parole de type 3 définit ce que nous appelons la technique du jeton délégué temporisé. Cette technique, appliquée successivement et de manière ordonnée à l'ensemble des stations DLE, permet d'implanter un protocole du type "jeton temporisé", c'est-à-dire, on peut donc mettre en oeuvre un ordonnancement basé sur une garantie de droit d'accès borné aux stations DLE (ordonnancement de classe 2, tel comme nous l'avons défini).

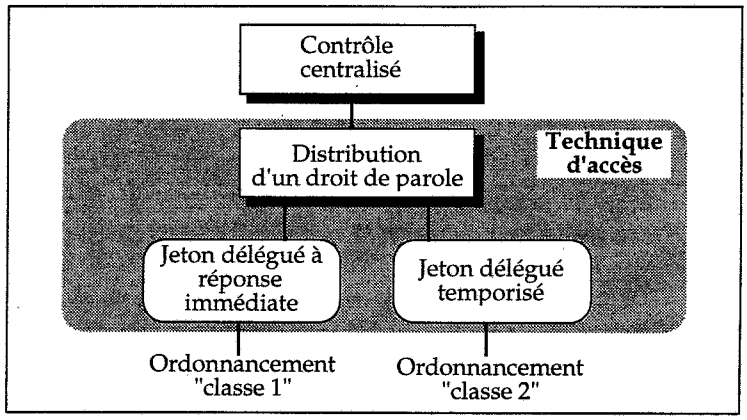

Figure 3.16: Le protocole IEC 65C

En conséquence, les mécanismes protocolaires de la norme ISA SP-50 / IEC-65C peuvent être utilisés pour mettre en oeuvre les deux classes d'ordonnancement 1 et 2 (voir figure 3.16).

6. Évaluation comparative des classes 1 et 2

6.1. Critères d'évaluation

Quatre critères sont essentiels pour caractériser des protocoles temps-réel: la robustesse (c'est-à-dire la capacité à absorber des variations de paramètres temporels sans affecter l'ordonnancement global), la borne minimale de l'utilisation maximale (c'est-à-dire la garantie de l'ordonnançabilité de la configuration des flux), la capacité de prise en compte du trafic non-temps-réel (conjointement avec du trafic temps-réel) et le temps de l'arbitrage d'accès.

6.1.1. Comparaison

Robustesse

Un protocole à ordonnancement hors-ligne quelle que soit la classe (1 ou 2) est, par définition, toujours plus robuste qu'un protocole à ordonnancement en-ligne: en effet l'ordonnancement étant effectué au préalable, la séquence de transfert des messages n'est pas perturbée par la violation de spécifications temporelles dans des flux. Par exemple, si un flux augmente sa cadence de production, ce flux ne peut pas transmettre tous les messages qu'il produit (seule une séquence correspondante à l'ordonnancement pré-défini est transmise) mais les autres flux ne s'en aperçoivent pas (seul le flux fautif est pénalisé).

Dans le cas d'un protocole à ordonnancement en-ligne, l'ensemble des flux de messages temps-réel reste ordonnançable tant que le test d'ordonnançabilité est respecté. Dans le cas

d'une violation par un flux des spécifications temporelles, il peut y avoir des échéances (d'autres flux) non respectées. Les différences entre la classe 1 et la classe 2 sont les suivantes:

- dans un protocole de classe 1, si l'algorithme d'assignation de priorités est dynamique la propagation d'une faute temporelle est imprévisible, c'est-à-dire, que les flux qui peuvent voir leurs échéances non respectées ne sont pas connus au préalable; par contre, si l'algorithme est statique, la propagation des fautes temporelles est prévisible car elle affecte d'abord les flux de messages qui ont les priorités les plus basses dans le système.
- dans un protocole de classe 2, la violation des spécifications temporelles n'affecte que les flux de la station concernée, car la durée pendant laquelle chaque station peut utiliser le réseau est limitée, soit à la bande passante allouée par cycle du jeton (protocole jeton temporisé), soit à un message par époque (protocole à époque temporelle).

Nota: nous ne considérons que les aspects concernant la robustesse aux fautes temporelles dans les caractéristiques des flux de messages. Pour effectuer une analyse approfondie sur la robustesse du protocole, il faudrait considérer aussi les fautes temporelles du protocole lui-même.

Borne minimale de l'utilisation maximale

Les protocoles de classe 1 autorisent des valeurs importants (qui peuvent avoisiner les 69% avec l'algorithme RM et les 100% avec l'algorithme ED), ce que ne permet pas l'exemple type du protocole de classe 2 qu'est le protocole du jeton temporisé (33% avec l'algorithme d'allocation de bande proportionnel adapté). Ceci résulte du fait que dans la classe 1, on a une synchronisation forte entre l'ordonnancement et les flux, alors que dans le protocole du jeton temporisé, les passages de jeton et les productions des flux sont asynchrones.

Capacité de prise en compte du trafic non-temps-réel

La classe 1 offre des potentialités intéressants grâce au mécanisme de serveur. En ce qui concerne la classe 2, les conclusions sont différentes selon que l'on a le protocole à jeton temporisé ou le protocole avec collision et résolution d'époque temporelle: la potentialité du trafic non-temps-réel d'une station dépend, avec le premier, du trafic non-temps-réel écoulé par les autres stations et, avec le deuxième, du trafic temps-réel de la station.

Temps d'arbitrage d'accès

Ce temps est évidemment réduit, par définition, dans le cas des protocoles à ordonnancement hors-ligne (que l'on soit en classe 1 ou 2). En ce qui concerne les protocoles à ordonnancement en-ligne:

- dans la classe 1, une opération d'arbitrage est effectuée chaque fois qu'un message doit être transféré et donc le coût temporel des opérations d'arbitrage est élevé (en particulier dans le cas des réseaux haut-débit, ce coût peut devenir très important et donc l'ordonnancement en-ligne n'est pas intéressant de ce point de vue là).
- dans la classe 2, le protocole à jeton temporisé présente des propriétés très intéressants (avec un seul passage de jeton dans une station, plusieurs messages peuvent être transmis); par contre, le protocole avec collision et résolution d'époque temporelle peut demander des temps d'arbitrage très longs (si la taille de l'arbre est grande).

7. Conclusion

Le travail présenté dans ce chapitre est, à notre avis, fondamental pour bien intégrer la problématique des protocoles MAC temps-réel, à la fois, en termes d'ordonnancement et de transfert de messages. Nous voulons mettre en exergue les points suivants:

- nous avons proposé une classification des protocoles MAC en deux grandes classes: la classe 1, où l'ordonnancement est basé sur une assignation de priorités aux flux de messages, et la classe 2, où l'ordonnancement est basé sur une garantie de temps d'accès borné aux stations;
- nous avons développé une analyse des protocoles MAC temps-réel existants dans les deux classes en termes de deux processus: l'arbitrage d'accès et le contrôle de la durée de transmission; les protocoles CAN, FIP, DQDB et IEEE 802.5 se situent dans la classe 1; les protocoles IEEE 802.3DCR, TTP, "jeton temporisé" et ses dérivés FDDI, IEEE 802.4 et Profibus se situent dans la classe 2;
- nous avons fait une présentation complète du protocole "jeton temporisé" en développant, en particulier, les schémas d'allocation de bande passante qu'il offre, ainsi que les taux d'utilisation permis;
- nous avons montré l'intérêt de la norme ISA SP-50 / IEC-65C qui permet, à la fois, la mise en oeuvre de mécanismes de la classe 1 et de la classe 2 et donc offre de grandes possibilités d'ordonnancement;
- nous avons effectué une brève réflexion comparative entre les protocoles de classe 1 et les protocoles de classe 2, par rapport aux attributs robustesse, borne minimale de l'utilisation maximale, potentialité de trafic non-temps-réel et temps d'arbitrage d'accès.

8. Références

[ACZD, 92] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines in High-Speed Token-Ring Networks with the Timed Token Protocol; in Proc. of 12th IEEE Dist. Computing Systems, Yokohama, pages 468-475, 1992;

[ACZ, 93] G. Agrawal, B. Chen and W. Zhao; Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the Timed Token Protocol; Proc. of INFOCOM'93

[ACZD, 94] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol; in IEEE Transactions on Computers, vol. 43, no 3, pages 327-339, March 1994;

[ANSI, 90] ANSI Standard X3T9.5/88-139, Rev. 4: FDDI MAC; 1990

[ARS, 90] K. Arvind, K. Ramamritham, J. Stankovic; Window MAC protocols for Real-Time Communication Services; COINS Technical Report 90-127, University of Massachusetts at Amherst, Jan. 1991;

[B_all, 92] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, F. Wang; On the Comptitiveness of On-Line Real-Time Task Scheduling; Journal of Real-Time Systems, 4, pages 125-144, 1992

[B_all, 92] S. Baruah, L. Rosier, R. Howell; Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic Real-Time Tasks on one Processor; Journal of Real-Time Systems, 42, pages 301-324, 1990

[CVJ, 94] R. Carmo, F. Vasques, G. Juanole; Real-Time Communication Services in a DQDB Network; IEEE RTSS'94, pages 249-258, S. Juan, 1994

[DIN, 91] DIN 19245 Part 1 & 2: Process Field Bus; April 1991

[Gro, 82] R. Grow; A Timed Token Protocol for Local Area Networks; IEEE Electro'82, Token Access Protocols, 17/3, 1982

[IEC, 94] Digital Data Communications for Measurements and Control - Field Bus for use in Industrial Control Systems; Parts 3 (105 v1.1) & 4 (106 v1.1), IEC SC65C/WG6

[IEEEa, 85] Token Bus Access Method and Physical Layer Specifications, IEEE Std 802.4 - 1985

[IEEEb, 85] Token Ring Access Method and Physical Layer Specification, IEEE Std 802.5 - 1985

[IEEEc, 85] Carrier Sense Multiple Access with Collition Detection (CSMA/CD), IEEE Std 802.3 - 1985

[IEEE, 90] Distributed Queue Dual Bus (DQDB) Subnetwork of Metropolitan Area Network, IEEE Std 802.6 - 1990

[K_all, 89] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft, R. Zainlinger; Distributed Fault-Tolerant Real-Time Systems: The MARS Approach; IEEE MICRO, Feb. 1989

[Laine, 91] T. Laine; Modélisation d'Applications Réparties pour la Configuration Automatique d'un Bus de Terrain; Thése de Doctorat, INP de Lorraine, CRIN, 29 Mai 1991

[LL, 73] C. Liu, J. Layland; Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment; Journal of the ACM 20 (1), pages 46-61, 1973;

[MCV, 92] P. Montuschi, L. Ciminiera and A. Valenzano; Time Characteristics of IEEE 802.4 Token Bus Protocol; IEE Proceedings-E, vol. n°1, January 1992.

[MZ, 95] N. Malcolm, W. Zhao; Hard Real-Time Communication in Multiple-Access Networks; The Journal of Real-Time Systems, 8(1), pages 35-77, 1995;

[NFa, 90] FIP Bus for Exchange of Information between Transmitters, Actuators and Programmable Controllers, Data Link Layer; NF C 46-603, June 1990

[Ple, 92] P. Pleineveaux; An Improved Hard-Real-Time Scheduling for the IEEE 802.5; The Journal of Real-Time Systems, 4(2), pages 99-112, 1992;

[PT, 89] J. Pang, F. Tobagi; Throughput Analysis of a Timer Controlled Token Passing Protocol Under Heavy Load; IEEE Tr. on Comm., 37(7), pages 694-702, July 1989.

[RN, 93] P. Raja, G. Noubir; Static and Dynamic Pooling Mechanisms for Fieldbus Networks; in Operating Systems Review, ACM Press, 27(3), pages 34-45, July 1993.

[Rol, 91] P. Rolin; Réseaux Locaux: normes et protocoles; 2e édition, Hermès, Paris, 1989

[SAE, 92] Controller Area Network (CAN), an In-Vehicule Serial Communication Protocol-SAE J1583; (report of the Vehicule Network for Multiplexing and Data Communication Standards Comitee); SAE Handbook, vol. II, 1992;

[SJ, 87] K. Sevcik and M. Johnson; Cycle Time Properties of the FDDI Token Ring Protocol; IEEE Trans. on Software Eng., vol SE-13, n°3, 1994, pp 376-385.

[SML, 88] J. Strosnider, T. Marchok, J. Lehoczky; Adavanced Real-Time Scheduling using the IEEE 802.5 Token Ring; IEEE RTSS'88, pages 42-52, 1988;

[SSMT, 93] D. Saha, M. Saksena, S. Mukherjee, S. Tripathi; On Guaranteed Delivery of Time-Critical Messages in DQDB. Technical Report CS-TR-3149, University of Maryland, October 1993.

[SSS, 92] L. Sha, S. Sathaye, J. Strosnider; Scheduling Real-Time Communication on Dual-Link Networks; IEEE RTSS'92, pages 188-197, 1992;

[THW, 94] K. Tindell, H. Hansson, A. Wellings; Analysing Real-Time Communications: Controller Area Network (CAN); IEEE RTSS'94, S. Juan, pages 259-263, Dec. 1994

Chapitre IV

Sur les réseaux de communication temps-réel: contributions

1. Introduction

Le but de ce chapitre est de présenter les contributions significatives, à notre avis, que nous pensons avoir apportées au domaine des protocoles MAC temps-réel de classe 1 et classe 2, à la fois, en termes conceptuels et en termes de réflexions sur des protocoles existants.

Dans de nombreux protocoles de classe 1 [NFa, 90] [IEEE, 90] l'ordonnancement du trafic périodique (ordonnancement de base) est effectué hors-ligne et mis en oeuvre de manière centralisée dans une station particulière (nous avons indiqué au chapitre III tout l'intérêt de l'ordonnancement hors-ligne pour le trafic périodique). Ce n'est qu'ensuite, lorsque la séquence temporelle du trafic périodique a été spécifiée, que l'on essaie de voir comment on peut ordonnancer, dans les intervalles de temps vides, le trafic apériodique temps-réel (nous appelons cette technique la technique de l'essai d'ordonnancement à posteriori du trafic apériodique temps-réel). Notons que dans le contexte d'un ordonnancement hors-ligne, il est essentiel de définir des stratégies de changement de mode en-ligne afin de pouvoir prendre en compte l'évolutivité du fonctionnement d'un système distribué sur un réseau.

C'est dans ce double contexte (ordonnancement hors-ligne de trafic périodique et algorithme de changement de mode en-ligne; essai d'ordonnancement à posteriori du trafic apériodique temps-réel) que se situent deux types de travaux que nous avons effectués dans le domaine des protocoles de classe 1 et qui, à notre connaissance, représentent des résultats nouveaux:

- le premier type concerne la définition d'un algorithme de changement de mode en-ligne sachant que l'on a un ordonnancement hors-ligne basé sur l'algorithme RM (ces travaux ont été effectués lors d'études pour l'ordonnancement de trafic isochrone sur le réseau DQDB au moyen du mécanisme PA [IEEE, 90];
- le deuxième type concerne l'étude de l'ordonnançabilité du trafic apériodique temps-réel dans le réseau FIP et plus précisément ce qui est appelé dans la norme [NFa, 90] la scrutation déclenchée de variables apériodiques urgente et que nous appelons la scrutation indirecte d'une variable apériodique (c.f. III.3.3.5).

Dans le domaine des protocoles de classe 2, le protocole "jeton temporisé" [Gro, 82] est un protocole qui fait référence et dont plusieurs chercheurs ont analysé les caractéristiques et les propriétés.

Cependant, à notre connaissance, il n'y a pas eu de travaux faisant une analyse critique de ce protocole. Un mécanisme qui, à notre avis, mérite une analyse critique, est le mécanisme de la gestion du temps de cycle du jeton. On a en effet un temps de cycle irrégulier qui peut dépasser la valeur de T_{TRT} pour atteindre la valeur maximale de $2 \cdot T_{TRT}$. Ce fonctionnement a des conséquences fâcheuses pour le dimensionnement du protocole par rapport au trafic temps-réel car, à une prévision de temps de cycle maximal de $2 \cdot T_{TRT}$, ne correspond qu'un temps de cycle dont la valeur moyenne est T_{TRT}. Pratiquement, on a donc des ressources surdimensionnées. C'est précisément cet aspect du protocole du jeton temporisé qui a motivé notre réflexion et nous a amené à proposer une adaptation de ce protocole que nous appelons le protocole "jeton temporisé régulier" et dans lequel le temps de cycle ne subit pas des augmentation sporadiques jusqu'à $2 \cdot T_{TRT}$. De cette façon, le surdimensionnement est diminué.

Enfin, en ce qui concerne le protocole Profibus [DIN, 91], s'il présente des similitudes avec le protocole "jeton temporisé", il a cependant une différence fondamentale qui découle de la non allocation d'une bande passante bien définie, par station, pour le trafic temps-réel et qui a donc des incidences sur les performances de ce type de trafic. C'est précisément cet aspect du protocole Profibus qui a motivé notre réflexion et qui nous a amené a définir des conditions pour l'ordonnançabilité du trafic temps-réel dans le réseau Profibus.

En conséquence ce chapitre est divisé en deux grandes parties:

- la première partie est relative à des protocoles de la classe 1 et comprend deux parties: la première concerne la présentation d'un algorithme de changement de mode en-ligne sachant que l'on a un ordonnancement RM hors-ligne; la deuxième concerne la présentation de conditions pour l'ordonnançabilité du trafic apériodique urgent (sur la base de la scrutation indirecte) dans le réseau FIP,
- la deuxième partie est relative à des protocoles de la classe 2 et comprend deux parties: la première concerne la définition du protocole "jeton temporisé régulier"; la deuxième concerne la présentation de conditions pour l'ordonnançabilité du trafic temps-réel dans le réseau Profibus.

2. Sur les protocoles de classe 1

2.1. Algorithme de changement de mode de fonctionnement dans un protocole DQDB à ordonnancement RM hors-ligne [CVJ, 94] [JCV, 94]

2.1.1. Introduction

Nous considérons un ensemble $M = \{M_1, \ldots M_i, \ldots M_n\}$ de flux de slots à ordonnancer, avec $\{C_i, P_i\} = \{1, P_i\}$ (slots PA de durée unitaire), chacun étant associé à un flux de messages isochrones (c.f. III.3.4).

L'algorithme de génération de la séquence temporelle étant basé sur l'algorithme RM, les slots sont alloués dans l'ordre inverse des périodes des flux. La séquence temporelle est évaluée sur l'intervalle t_l, $0 \le t_l \le ppcm\{P_1, P_2, \ldots P_n\}$. Nous appelons $ppcm\{P_1, P_2, \ldots P_n\}$ le cycle de la séquence temporelle.

Nous notons par la suite P_i^k, la $k^{ème}$ occurrence de la période du flux M_i dans une séquence temporelle, et Γ comme l'intervalle correspondant à cette occurrence: $k \cdot P_i \leq \Gamma \leq (k+1) \cdot P_i$.

Algorithme de génération de la séquence temporelle:

Considérer l'ensemble $M = \{M_1, \ldots M_i, \ldots M_n\}$ ordonné dans l'ordre croissant de leurs périodes: $P_1 \leq P_2 \leq \ldots P_i \leq P_{i+1} \leq \ldots P_n$

Pour $i = 1, 2 \ldots n$ faire: /* pour chaque flux M_i */

  Pour $k = 0, 1, \ldots (ppcm/P_i - 1)$ /* pour chaque P_i^k */

    Allouer au flux M_i, le premier slot libre de l'intervalle Γ

Exemple:

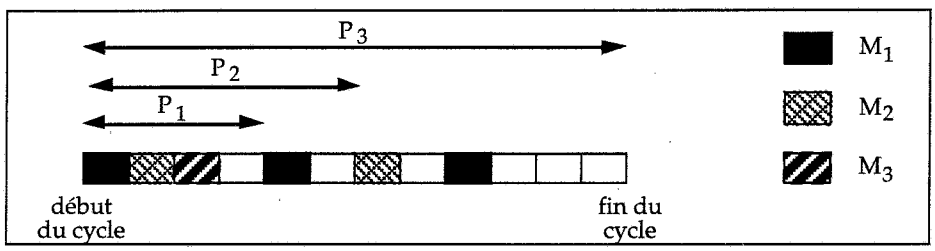

Figure 4.1: Exemple d'une séquence temporelle

Un exemple d'une séquence temporelle est présenté sur la figure 4.1. Nous avons 3 flux de slots: $M_1(P_1 = 4)$, $M_2(P_2 = 6)$ et $M_3(P_3 = 12)$. Alors $ppcm(P_i) = 12$.

2.1.2. Définitions

Nous définissons tout d'abord les ensembles de flux de slots des deux séquences temporelles: celle qui est en exécution et la nouvelle.

Définition 4.1: (Ensemble de flux de slots de la séquence temporelle en exécution)

Un ensemble de flux de slots $M_{ste} = \{M_1, \ldots M_i, \ldots M_n\}$ avec des périodes $P_1 \leq \ldots P_i \leq \ldots P_n$ est un ensemble où $\forall M_i \in M_{ste}$, M_i est un flux de slots de la séquence temporelle en exécution.

Définition 4.2 (Ensemble de flux de slots de la nouvelle séquence temporelle):

Un ensemble de flux de slots $M_{stn} = \{M_1, \ldots M_i, \ldots M_n\}$ avec des périodes $P_1 \leq \ldots P_i \leq \ldots P_n$ est un ensemble où $\forall M_i \in M_{stn}$, M_i est un flux de slots de la nouvelle séquence temporelle.

L'objectif de cet algorithme est de changer la séquence temporelle en exécution par une nouvelle séquence temporelle et d'assurer la *correction* de la génération de slots.

Définition 4.3 (Correction de la génération de slots):

La correction de la génération de slots est assurée si un et seulement un slot est attribué pour toute période P_i^k, $k = 0, 1 \ldots (ppcm/P_i - 1)$, $\forall M_i$, $i = 1, 2, \ldots n$, $\{M_i \in M_{stn} \wedge M_i \in M_{ste}\}$.

Le changement de séquence temporelle doit donc être transparent pour les flux de slots qui restent dans la séquence. Plus précisément, il faut éviter d'avoir des slots dupliqués ou perdus comme nous allons le montrer sur les deux exemples suivants.

2.1.3. Exemples

1. Exemple 1: ajout d'un flux de slots

Considérons la séquence temporelle générée par un ensemble de 6 flux de slots $M_{ste} = \{M_1, \ldots M_6\}$, avec leurs périodes respectives $P_1 = 5$, $P_2 \ldots = P_6 = 20$ et la nouvelle séquence qui est obtenue après l'inclusion du flux M_7 avec période $P_7 = 5$ (figures 4.2.a et 4.2.b). Supposons que le changement de séquence soit fait après la génération du quatrième slot de la séquence temporelle en exécution. Les slots seront générés comme illustré sur la figure 4.2.c. Nous pouvons observer qu'un slot supplémentaire est attribuée au flux M_4 pour cette occurrence de la période. **On a donc une duplication de slots.**

L'attribution d'un slot supplémentaire pour un flux M_i durant la $k^{ème}$ occurrence de la période P_i, résulte du fait que la condition du flux de slots est différente dans les séquences temporelles lors du changement. Considérant l'exemple de la figure 4.2, la condition de M_4 dans la séquence temporelle en exécution est: *le slot demandé pour P_4^1 a été déjà attribué;* dans la nouvelle séquence temporelle sa condition est: *le slot demandé pour P_4^1 n'a pas encore été attribué.*

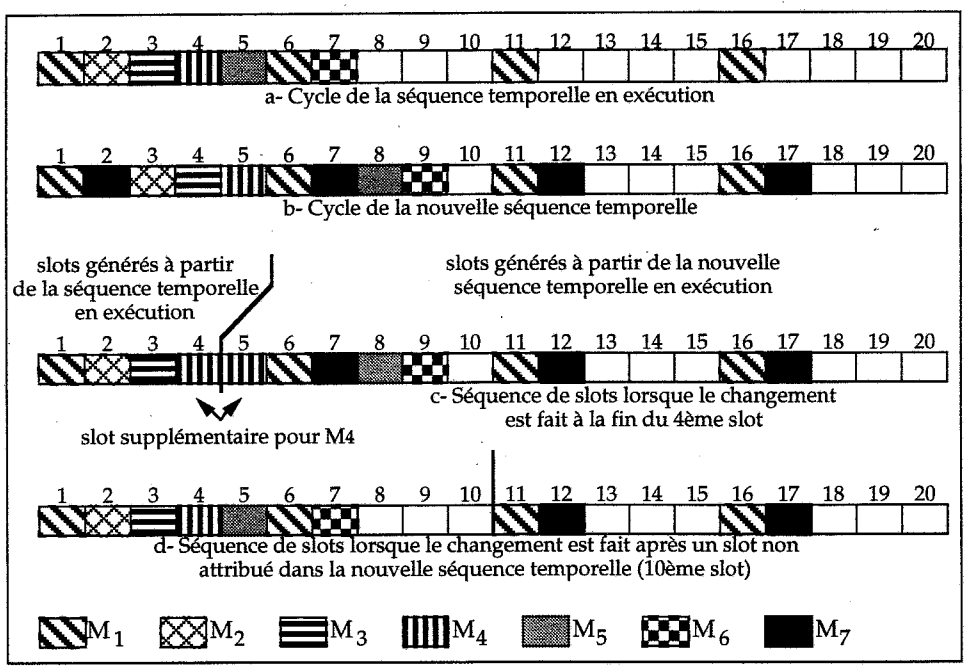

Figure 4.2: Le changement de la séquence temporelle lorsqu'un flux de messages est ajouté

2. Exemple 2: retrait d'un flux de slots

Considérons le cas contraire au précédent, avec les mêmes paramètres. La nouvelle séquence temporelle et la séquence temporelle en exécution de l'exemple précédent sont respectivement la séquence temporelle en exécution et la nouvelle séquence temporelle dans cet exemple (figures 4.3.a, 4.3.b).

Supposons que le flux de slots doit être retiré de la séquence temporelle. Si le changement est fait après la génération du quatrième slot, un slot n'est pas attribué au flux M_4 pour P_4^1 (figure 4.3.c) et **nous avons donc une perte**. Comme dans l'exemple précédent, la condition de M_4 est différent dans les séquences temporelles à l'instant du changement: *un slot n'a pas été attribué à M_4* dans la séquence temporelle en exécution; *un slot n'a pas été attribué à M_4* dans la nouvelle séquence temporelle.

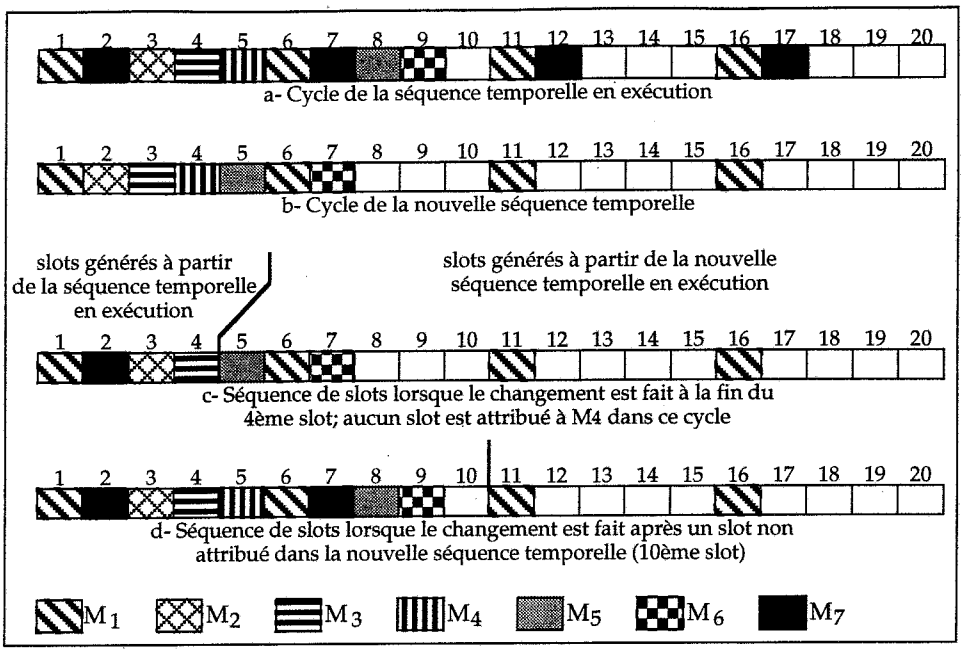

Figure 4.3: Le changement de la séquence temporelle lorsqu'un flux de messages est retiré

3. Conclusion

Les deux exemples précédents ont montré que la correction de la génération de slots ne peut pas être assurée lorsque le changement de la séquence temporelle est fait à un instant arbitraire.

2.1.4. Concepts

Nous introduisons dans cette section deux types de concepts: des concepts sur la condition des flux de messages dans une séquence temporelle et des concepts sur l'instant de comparaison de deux séquences. Le premier type de concept sert à identifier la condition qui doit être respectée à l'instant du changement. Le deuxième type établit une correspondance entre les cycles des deux séquences.

1. Concepts sur la condition des flux de messages dans une séquence temporelle

Définition 4.4 (Flux de slots en attente):

Un flux de slots M_i est en attente à l'instant t_l d'une séquence temporelle avec $k \cdot P_i \le t_l \le (k+1) \cdot P_i$, $k = 0, 1, \ldots ppcm\{P_1, P_2, \ldots P_n\}$, si aucun slot n'a été attribué pour ce flux de slots pour la $k^{ème}$ occurrence de la période.

Définition 4.5 (Ensemble de flux de slots en attente):

Un ensemble de flux de slots $M_p = \{M_1, M_2, \ldots M_n\}$ avec des périodes $P_1 \le P_2 \le \ldots P_n$ est en attente à l'instant t_l, si $\forall M_i \in M_p$, M_i est un flux de slots en attente.

Nous notons M_{pe} et M_{pn} l'ensemble de flux de slots en attente à l'instant t_l, respectivement, de la séquence temporelle en exécution et de la nouvelle séquence temporelle.

Lemme (4.1 Flux de slots en attente):

Si t_l est l'instant initial d'un slot non attribué, alors l'ensemble M_p de la séquence temporelle est vide à l'instant t_l.

Démonstration

Supposons qu'il y a un flux de slots M_i en attente à l'instant t_l, $k \cdot P_i \le t_l \le (k+1) \cdot P_i$, $k = 0, 1, \ldots ppcm\{P_1, P_2, \ldots P_n\}$, et t_l est l'instant initial d'un slot non attribué. Ceci contredit l'algorithme de génération de la séquence temporelle qui attribue le premier slot libre de l'intervalle Γ, $k \cdot P_i \le \Gamma \le (k+1) \cdot P_i$, $k = 0, 1, \ldots \left((ppcm/P_i) - 1\right)$ au flux M_i.

Fin démonstration

2. Concepts sur le changement entre deux séquences temporelles

Les séquences temporelles peuvent avoir des durées différentes. Pour effectuer le changement entre deux séquences temporelles, nous devons, premièrement, définir le cycle à partir duquel les deux séquences doivent être comparées.

Considérons un instant t_{lc} arbitraire de la séquence temporelle en exécution. Nous identifions le cycle *correspondant* dans la nouvelle séquence temporelle à partir duquel les deux séquences peuvent être comparées. Notons les *ppcm* de la séquence temporelle en exécution et de la nouvelle séquence temporelle respectivement $ppcm_e$ et $ppcm_n$. Il faut considérer deux cas:

- $ppcm_e \le ppcm_n$: t_{lc} appartient au premier cycle de la nouvelle séquence temporelle.
- $ppcm_e > ppcm_n$: Notons $X = (ppcm_e / ppcm_n)$. Nous avons X cycles de la nouvelle séquence temporelle durant le cycle courant de la séquence temporelle en exécution. De façon générale, t_{lc} est telle que $(b-1) \cdot ppcm_n \le t_{lc} \le b \cdot ppcm_n$, avec $b = 1, 2, \ldots X$. L'instant t_{lc} appartient donc au $b^{ème}$ cycle de la nouvelle séquence temporelle.

Définition 4.6 (Instant de changement de la séquence temporelle):

L'instant de changement de la séquence temporelle doit assurer la correction de la génération de slots.

Théorème 4.2:

Si le changement de la séquence temporelle est fait à l'instant t_{lc} et si à cet instant les ensembles C_{pe} et C_{pn} sont vides, C_{pn} étant l'ensemble de flux de slots en attente de la $b^{ème}$ occurrence du cycle de la nouvelle séquence temporelle, alors la correction de la génération des slots est assurée.

Démonstration (théorème):

Puisqu'à l'instant t_{lc} il n'y a pas de flux de slots en attente dans les deux séquences temporelles, cela signifie que les slots ont été déjà attribuées pour le P_i^k courant des deux séquences temporelles, $\forall M_i$, $i = 1, 2, \ldots n$. Alors, même si la séquence temporelle est changée à cet instant, comme il n'y a pas de flux de slots en attente dans les deux séquences temporelles, un et seulement un slot sera attribué pour tout P_i^{k+1} de la nouvelle séquence temporelle, $\forall M_i$, $i = 1, 2, \ldots n$. Par conséquence, la correction est assurée.

Fin démonstration (théorème)

Remarquons que l'approche la plus simple pour changer la séquence temporelle et garantir la correction de la génération de slots, serait d'effectuer le changement à la fin du cycle de la séquence temporelle en exécution. Cependant, cette approche présente un inconvénient: elle entraîne un temps de réponse important aux demandes des utilisateurs dans le cas d'un cycle de longue durée.

2.1.5. Ajout de flux de slots

Supposons une séquence temporelle nouvelle générée après l'addition d'un nombre arbitraire de flux de slots à la séquence en exécution. Nous avons donc $\#(M_{stn}) > \#(M_{ste})$. Nous définissons une condition suffisante mais pas nécessaire pour identifier le premier instant t_{lc}, où les ensembles M_{pe} et M_{pn} sont vides.

Lemme 4.2 (Flux de slots en attente):

$\forall t_{lc}$, si M_{pn} est vide, alors M_{pe} est aussi vide.

Démonstration

Considérons $\#(M_{stn}) - \#(M_{ste}) = z$. Nous montrons que pour $z = 1$ la proposition est vraie. Considérons M_a le flux de slots ajouté à l'ensemble M_{pe} et P_a sa période. Nous considérons premièrement le cas où l'addition d'un nouveau flux ne modifie pas le cycle de la séquence temporelle. Après, nous généralisons la démonstration.

- *Cas 1: $ppcm_e = ppcm_n$.* Deux cas doivent être considérés:
  - *Cas 1.1: Le flux de slots M_a a une période P_a telle que $P_1 \ldots \leq P_n = P_a$;*

Considérons la construction de la nouvelle séquence temporelle. Les slots sont attribués en premier aux flux $M_1, M_2, \ldots M_n$ et un slot sera attribuée en dernier à chaque P_a^k du flux M_a. L'addition du flux de slots M_a ne va pas modifier la position des flux $M_1, M_2, \ldots M_n$ par rapport à leurs positions dans la séquence temporelle en exécution: ils ont la même position dans les deux séquences temporelles. Par conséquent, tous les slots non attribués dans la nouvelle séquence temporelle sont aussi non attribués dans la séquence temporelle en exécution. Il découle du lemme 4.1 que $\forall t_{lc}$, si l'ensemble M_{pn} est vide à l'instant t_{lc}, alors M_{pe} est aussi vide.

  - *Cas 1.2: Le flux de slots M_a a une période P_a telle que $P_1 \ldots \leq P_a \ldots \leq P_n$;*

Établissons une correspondance entre les positions occupées par les flux de slots dans les deux séquences temporelles. Selon l'algorithme de génération de la séquence temporelle, la position d'un flux de slots dans la nouvelle séquence temporelle sera la même, ou décalée à droite de m trames par rapport à sa position dans la séquence temporelle en exécution. La correspondance entre les deux positions dans les deux séquences est donc:

- Un flux de slots M_i, avec $1 \leq i < a$, occupe la même position dans les deux séquences.
- La position d'un flux de slots M_i, avec $a < i \leq n$, dans la séquence temporelle en exécution est occupé par un flux de slots M_j avec $a \leq j \leq i$ dans la nouvelle séquence.

Tous les slots occupés dans la séquence temporelle en exécution sont donc occupés dans la nouvelle. Alors, tous les slots non attribués dans la nouvelle séquence temporelle sont aussi non attribués dans la séquence en exécution. $\forall t_{lc}$, si t_{lc} est l'instant initial d'un slot non attribué dans la nouvelle séquence temporelle, il en est de même aussi dans la séquence temporelle en

exécution. Il découle du lemme 4.1 que $\forall t_{lc}$, si l'ensemble M_{pn} est vide à l'instant t_{lc}, alors M_{pe} est aussi vide.

- *Cas 2:* $ppcm_e < ppcm_n$. Notons $X = (ppcm_n / ppcm_e)$. Deux cas doivent être considérés:
  - *Cas 2.1: Le flux de slots* M_a *a une période* P_a *telle que* $P_1 \ldots \leq P_n < P_a$;

La démonstration est la même que dans le cas 1.1; la différence étant le nombre d'occurrences des périodes des slots dans les deux séquences temporelles. Dans le cas 1.1, le nombre d'occurrences est le même. Dans le cas présent le nombre d'occurrences dans la nouvelle séquence temporelle est X fois le nombre d'occurrences dans la séquence en exécution.

  - *Cas 2.2: Le flux de slots* M_a *a une période* P_a *telle que* $P_1 \ldots \leq P_a \ldots \leq P_n$;

La démonstration est la même que dans le cas 1.2; la différence étant le nombre d'occurrences des périodes des flux de slots dans les deux séquences temporelles. Dans le cas 1.2, le nombre d'occurrences est le même. Dans le cas présent le nombre d'occurrences dans la nouvelle séquence temporelle est X fois le nombre d'occurrences dans la séquence en exécution.

Hypothèse inductive: Nous supposons que la proposition est vraie pour $z = x$ et nous montrons qu'elle est vraie pour $z = x + 1$.

Considérons l'ensemble $M_{stn'}$ avec $\#(M_{stn'}) = x$ et l'ensemble M_{stn} avec $\#(M_{stn}) = x + 1$. Nous supposons donc que $\forall t_{lc}$, si $M_{stn'}$ est vide à l'instant t_{lc}, alors M_{pe} est aussi vide. Il a été montré dans le cas de base que $\forall t_{lc}$, si M_{stn} est vide à l'instant t_{lc}, alors $M_{pn'}$ est aussi vide. Alors, $\forall t_{lc}$, si M_{pn} est vide à l'instant t_{lc}, alors M_{pe} est aussi vide.

Fin démonstration

Finalement, nous définissons *l'instant de changement* t_{lt} qui préserve la correction de la génération de slots.

Théorème 4.3:

L'instant de changement t_{lt} *est celui qui correspond à l'instant initial du premier slot non attribué dans la nouvelle séquence temporelle.*

Démonstration:

Il découle du lemme 4.1 que l'ensemble M_{pn} est vide à l'instant t_{lt}. D'après le lemme 4.2, l'ensemble M_{pe} est aussi vide. Alors, d'après le théorème 4.2, la correction de la génération des slots est assurée.

Fin démonstration

Nous représentons dans la figure 4.2.d, le changement de séquence temporelle qui assure la correction de la génération de slots.

2.1.6. Retrait de flux de slots

Nous avons montré (théorème 4.2) que lorsque les ensembles M_{pe} et M_{pn} sont vides, le changement peut être fait et la correction de la génération de slots est assurée. Analysons donc le cas où plusieurs flux de slots sont retirés.

Considérons le lemme 4.2. Nous pouvons facilement voir qu'il peut être adapté à travers une permutation entre la séquence temporelle en exécution et la nouvelle séquence.

Lemme 4.3:

$\forall t_{lc}$, si M_{pe} est vide à l'instant t_{lc}, alors M_{pn} est aussi vide.

Nous pouvons donc définir l'instant de changement t_{lt}.

Théorème 4.4:

L'instant de changement t_{lt} est celui qui correspond à l'instant initial du premier slot non attribué dans la séquence temporelle en exécution.

Démonstration

Il découle du lemme 4.1 que l'ensemble M_{pe} est vide à l'instant t_{lt}. D'après le lemme 4.3, l'ensemble M_{pn} est aussi vide. Alors, d'après le théorème 4.2, la correction de la génération de slots est assurée.

Fin démonstration

Nous représentons dans la figure 4.3.d, le changement de séquence temporelle qui assure la correction de la génération de slots.

2.1.7. Temps maximum pour le changement de la séquence temporelle

Le temps maximum pour remplacer la séquence temporelle en exécution par une nouvelle séquence temporelle, dépend du nombre maximum de slots consécutifs attribués aux flux de messages. Nous définissons par la suite la borne supérieure de ce nombre.

Nous allons, tout d'abord, caractériser l'ensemble de flux de slots périodiques qui entraîne le taux maximum d'utilisation du réseau. Pour ceci nous utilisons un résultat du domaine de l'ordonnancement des tâches périodiques sur un processeur.

Dans [LL, 73] (théorèmes 4 et 5, pages 181-183), les auteurs ont montré que la borne supérieure minimale obtenue pour le taux d'utilisation d'un processeur $\left(U_x = n \cdot \left(\sqrt[n]{2} - 1\right)\right)$ peut être atteinte si et seulement si $C_i = P_{i+1} - P_i$ et $P_n < 2 \cdot P_i$. Ces deux relations caractérisent l'ensemble de tâches qui entraîne le taux maximum d'utilisation du processeur lorsqu'il est ordonnancé par l'algorithme RM.

Dans le contexte de notre étude, les slots sont de taille unitaire ($C_i = 1$). L'ensemble de flux de slots périodiques M_M qui entraîne le taux maximum d'utilisation du réseau $\left(n \cdot \left(\sqrt[n]{2} - 1\right)\right)$ est donc caractérisé par: $P_{i+1} = P_i + 1$ et $P_n < 2 \cdot P_i$. Si nous résolvons la première équation, nous obtenons $P_n = P_1 + (n-1)$. En remplaçant ce résultat dans la deuxième équation, nous avons $P_1 > (n-1)$. L'ensemble M_M est donc caractérisé par n flux de slots périodiques avec des périodes consécutives et $P_1 > (n-1)$.

Considérons la séquence temporelle ST_M générée par l'ensemble M_M avec l'algorithme d'ordonnancement hors-ligne proposé. Nous montrons qu'il y a toujours un slot non attribué durant l'intervalle critique de cette séquence.

Lemme 4.4

Il y a toujours un slot non attribué durant l'intervalle critique de la séquence ST_M générée par l'ensemble M_M

Démonstration

Nous montrons premièrement que, si $P_i = n$ le test d'ordonnançabilité est faux; et deuxièmement que, si $P_1 \geq (n+1)$ il y a toujours un slot non attribué durant l'intervalle critique de la séquence ST_M.

- Cas 1: $P_1 = n$

  L'ensemble M_M a les caractéristiques suivantes: $P_1 = n$, $P_{i+1} = P_i + 1$ et $P_n < 2 \cdot P_i$; Le taux d'utilisation du réseau est donné par $U = \sum_{k=n}^{2n-1} 1/k$ qui est toujours supérieur à $n\left(2^{1/n} - 1\right)$.

- Cas 2: $P_1 \geq (n+1)$

  Dans l'intervalle critique, un slot sera attribué au flux M_n, au plus tard, dans l'intervalle $[n-1, n[$. Puisque le prochain slot pour le flux M_1 sera attribué au plus tôt à l'instant $(n+1)$, il y a au moins un slot non attribué dans l'intervalle $[n, n+1[$.

Théorème 4.5

Le nombre maximum de slots consécutifs attribués dans la séquence temporelle générée avec l'algorithme d'ordonnancement hors-ligne est borné supérieurement par $(P_n - 1)$

Démonstration

Il découle du lemme 4.4 qu'il y a toujours un slot libre dans l'instant critique de la séquence temporelle. Supposons que ce slot est le dernier de l'instant critique. Il y a donc au maximum $(P_n - 1)$ slots consécutifs attribués dans la séquence temporelle.

2.2. Conditions d'ordonnançabilité du trafic apériodique temps-réel dans le réseau FIP [VJ, 93] [VJ, 94]

2.2.1. Cadre de l'analyse

Nous présentons tout d'abord la contrainte de protocole, c'est-à-dire, la condition pour que la charge maximale du trafic apériodique urgent pendant un macro-cycle puisse être écoulée pendant ce macro-cycle (garantie de non-surcharge). Dans le cas contraire, on aurait une accumulation de charge sur les macro-cycles suivants ce qui traduirait une instabilité du protocole. Nous évaluons ensuite la contrainte de l'échéance.

Considérons les notations suivantes:

- t_r est le temps de retournement d'un modem;
- $C_{\{id\_dat\}}$, $C_{\{rp\_dat\}}$, $C_{\{id\_rq\}}$ et $C_{\{rp\_rq\}}$ sont les durées maximales d'utilisation du réseau associées respectivement aux PDUs *id_dat*, *rp_dat*, *id_rq* et *rp_rq*; nous ne distinguons pas ici les durées associées aux PDUs *rp_dat* et *rp_dat_rq* (nous parlons toujours de $C_{\{rp\_dat\}}$).

2.2.2. Contrainte de protocole

Afin d'évaluer cette contrainte, on doit tout d'abord calculer la durée maximale associée au transfert d'un message dans chaque type de transfert (c.f. III.3.3: figures 3.7, 3.8 et 3.9):

- pour un message d'un flux périodique (scrutation périodique de variables périodiques):

$$C_i = C_{\{id\_dat\}} + C_{\{rp\_dat\}} + 2 \cdot t_r \tag{4.1}$$

- pour une variable d'un flux apériodique, dans le cas du transfert par scrutation directe (à condition qu'une seule variable soit demandée):

$$C_j^{(dir)} = C_{\{id\_rq\}} + C_{\{rp\_rq\}} + C_{\{id\_dat\}} + C_{\{rp\_dat\}} + 4 \cdot t_r \tag{4.2}$$

- pour une variable d'un flux apériodique, dans le cas où le transfert est effectué par scrutation indirecte:

$$C_j^{(indir)} = 2 \cdot \left(C_{\{id\_dat\}} + C_{\{rp\_dat\}} \right) + C_{\{id\_rq\}} + C_{\{rp\_rq\}} + 6 \cdot t_r \tag{4.3}$$

La contrainte du protocole (afin que l'on ait une garantie de non-surcharge) est:

$$\sum_{i=1}^{n} C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \underbrace{\sum_{j=1}^{s} C_j^{(dir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil}_{\substack{Ma_j \text{ scruté} \\ \text{directement}}} + \underbrace{\sum_{j=1}^{s} C_j^{(indir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil}_{\substack{Ma_j \text{ scruté} \\ \text{indirectement}}} \leq T_{MC} \tag{4.4}$$

2.2.3. Contrainte de l'échéance

Le calcul de la contrainte de l'échéance nécessite d'introduire deux notions (figure 4.4):

- le *retard d'avertissement* Δa_j de l'arbitre qui représente le temps écoulé entre l'instant d'occurrence d'une requête d'une variable apériodique d'un flux Ma_j dans une station et l'instant où l'arbitre est informé de cette requête; l'arbitre en sera informé dans la fenêtre périodique suite à une scrutation périodique de variables périodiques Mp_i de cette station (fin de $C_i^{(ind).1}$ (paragraphe III.3.3.3: figure 3.9)).
- le *retard d'ordonnancement* Δo_j, qui représente le temps écoulé entre l'instant où l'arbitre est informé de la demande de transfert et l'instant où l'arbitre a fini l'ordonnancement du transfert de cette variable (fin de $C_i^{(ind).2}$ (paragraphe III.3.3.3: figure 3.9)).

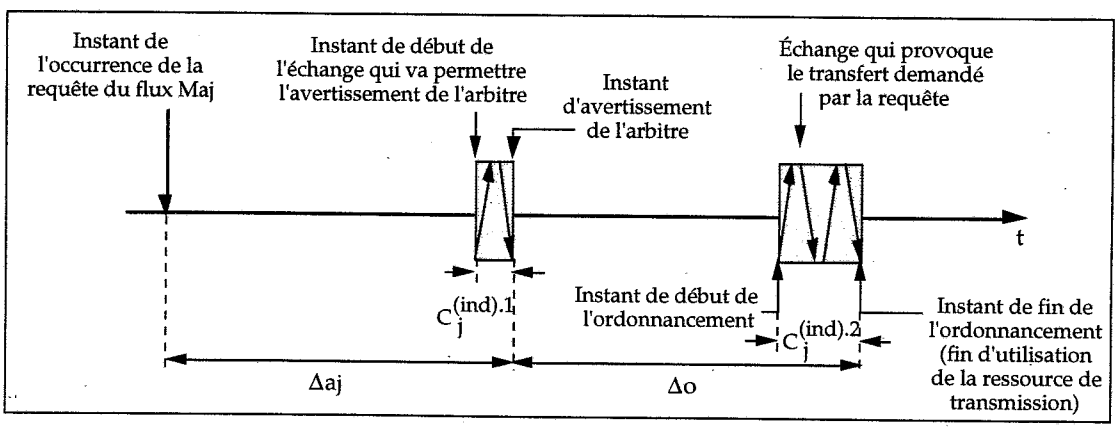

Figure 4.4: Retards d'ordonnancement et d'avertissement dans la scrutation indirecte d'une variable apériodique

Ces deux notions sont visualisées sur la figure 4.4. Pour bien comprendre cette figure, il est important de se reporter au scénario de la technique de la scrutation indirecte de variables apériodiques donnée au chapitre III (paragraphe III.3.5; figure 3.9).

Le retard d'avertissement Δa_j, est conditionné par les périodes P_i des variables des flux périodiques Mp_i de la station k à laquelle appartient le flux apériodique Ma_j. On a donc la relation suivante:

$$\Delta a_j \leq \min_{Mp_i \in nœud_k} \{P_i\} + \left(C_{\{rp\_dat\}} + t_r\right) \tag{4.5}$$

où $nœud_k$ représente l'ensemble des flux temps-réel de la station k et $\left(C_{\{rp\_dat\}} + t_r\right)$ représente le temps écoulé entre l'instant de début de l'envoi de la trame rp_dat_rq (que nous notons rp_dat; c.f. début du paragraphe) par la station k qui va permettre l'avertissement de l'arbitre et l'instant où l'arbitre est informé. Notons que, lorsqu'il est informé, l'arbitre enregistre cette requête dans la "file des demandes apériodiques urgentes".

Le retard d'ordonnancement dépend de l'état de la "file de demandes apériodiques urgentes" (file FIFO) à l'instant où l'arbitre est averti de cette requête. Nous considérons le pire cas, c'est-à-dire l'arbitre n'est averti de cette requête que juste après avoir été averti de toutes les autres requêtes possibles. En conséquence de la considération du pire cas, le retard d'ordonnancement Δo_j est le même quel que soit le flux Ma_j considéré; on le note donc Δo.

Comme l'ordonnancement du trafic apériodique par scrutation indirecte n'est effectué qu'après l'ordonnancement du trafic périodique, et compte tenu de la satisfaction de la contrainte du protocole, on a donc la relation suivante:

$$\Delta o \leq \sum_{i=1}^{n} C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \underbrace{\sum_{j=1}^{s} C_j^{(dir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil}_{\substack{Ma_j \text{ scruté} \\ \text{directement}}} + \underbrace{\sum_{j=1}^{s} C_j^{(indir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil}_{\substack{Ma_j \text{ scruté} \\ \text{indirectement}}} \leq T_{MC} \tag{4.6}$$

Contrainte de l'échéance: Pour garantir le respect des échéances du trafic apériodique urgent, il faut garantir que, quel que soit le flux apériodique Ma_j, la somme des retards d'avertissement et d'ordonnancement soit inférieur à l'échéance d_j, c'est-à-dire:

$$\Delta a_j + \Delta o \leq d_j \,,\ 1 \leq j \leq s \tag{4.7}$$

2.2.4. Exemple

Considérons un système avec 6 flux de messages périodiques $Mp = \{Mp_1 \ldots Mp_6\}$, caractérisés par $\{C_i, P_i\} = \{(4,40),(4,40),(4,80),(4,80),(4,80),(4,160)\}$, et 6 flux de messages apériodiques urgents identiques $Ma = \{Ma_1 \ldots Ma_6\}$, scrutés indirectement et caractérisés par $\{C_i, P_i\} = \{(4,155),\ldots(4,155)\}$.

Les flux de messages sont divisés parmi 3 stations de la façon suivante: $nœud_1 = \{Mp_1, Mp_2, Ma_1, Ma_2\}$, $nœud_2 = \{Mp_3, Mp_4, Ma_3, Ma_4\}$ et $nœud_3 = \{Mp_5, Mp_6, Ma_5, Ma_6\}$

Les micro et macro cycles sont respectivement:

$$T_{mc} = pgcd\{40,80,160\} = 40 \quad \text{et} \quad T_{MC} = ppcm\{40,80,160\} = 160$$

La contrainte du protocole impose que:

$$\sum_{i=1}^{n} C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \sum_{j=1}^{s} C_j \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil = 60 + 48 \leq T_{MC} = 160 \qquad \text{(condition vraie)}$$

Le retard d'avertissement dépend des durées $C_{\{rp\_dat\}}$ et t_r qui ne sont pas connues; néanmoins, sachant que $C_i > C_{\{rp\_dat\}} + t_r$, ce retard est borné par:

$$\Delta a_j = \{44, 44, 84, 84, 84, 84\}$$

Le retard d'ordonnancement est:

$$\Delta o = 108$$

La contrainte de l'échéance impose que: $\Delta a_j + \Delta o \leq d_j$, $1 \leq j \leq s$, ce qui n'est vrai que pour les flux apériodiques appartenant à la station *1* ($152 < 155$); pour les flux apériodiques appartenant aux stations *2* et *3*, la contrainte de l'échéance n'est pas respectée ($192 < 155$).

On voit que, quoique le taux d'utilisation du réseau dû au trafic périodique ne dépasse pas *37.5%*, nous ne pouvons ordonnancer que des flux apériodiques supplémentaires appartenant à la première station; il n'est pas possible d'ordonnancer des flux à faible taux d'utilisation du réseau (*2.6%*) dans les stations *2* et *3*.

2.2.5. Conclusion

Les expressions relatives à l'ordonnançabilité du trafic apériodique urgent (par la technique de scrutation indirecte) montrent que l'efficacité de ce service, dans l'hypothèse du pire cas, est réduite, car:

- on a besoin d'échanges de PDUs supplémentaires pour informer l'arbitre des requêtes en attente dans les stations (séquence d'échanges *id_dat / rp_dat_rq* et *id_rq / rp_rq*);
- le délai d'avertissement est élevé, à cause de l'asynchronisme entre la requête apériodique urgente et l'échange de PDUs *id_dat / rp_dat_rq* ;
- le caractère FIFO de la file de demandes apériodiques urgentes génère des inversions de priorité non contrôlées et donc le délai d'ordonnancement doit être évalué avec la file FIFO pleine à l'instant de l'échange de PDUs *id_rq / rp_rq*

La considération du pire cas induit évidement un test d'ordonnancement pessimiste.

3. Sur les protocoles de classe 2

3.1. Le protocole "jeton temporisé régulier"

3.1.1. Principe

L'idée est de considérer T_{TRT} non comme le temps cible pour le cycle du jeton mais comme le temps maximal.

Les principes de base du protocole sont:

1 on alloue à une station k une bande passante H_k pendant laquelle la station k transfère son trafic temps-réel (tant qu'elle en a) et ensuite, tant que l'allocation de bande passante H_k n'est pas terminée, transfère du trafic non-temps-réel (si elle en a); notons la différence fondamentale avec la définition de H_k dans le protocole jeton temporisé.

2 on contraint les allocations de bande passante aux m stations d'un réseau par la relation suivante qui définit la contrainte de protocole:

$$\sum_{k=1}^{m} H_k \leq T_{TRT} - \tau \qquad (4.8)$$

Le temps de cycle est donc compris entre τ et $\left(\tau + \sum_{k=1}^{m} H_k\right)$ qui est borné par T_{TRT}.

Compte tenu du principe 1, on peut dire que ce protocole ne favorise pas le trafic non-temps-réel: d'une part, une station k ne peut en transmettre que si le trafic temps-réel de cette station

n'a pas utilisé toute la bande H_k et, d'autre part, on n'a plus le mécanisme défini dans le protocole "jeton temporisé" (si le jeton arrive dans une station avant la fin du T_{TRT}, c'est-à-dire si le jeton est en avance, ce qui veut dire que les autres stations n'ont pas utilisée leur bande passante, la station peut transférer du trafic non-temps-réel).

La mise en oeuvre de ce protocole dans une station k est basée, d'une part, sur l'utilisation de deux compteurs (TRT_k et THT_k qui ont la même sémantique que dans le protocole "jeton temporisé") et, d'autre part, sur le temps de référence T_{TRT}.

L'algorithme du protocole est indiqué sur la procédure suivante:

```
Algorithme "jeton temporisé régulier"

Dans chaque station k, k = 1,2,...m :
    THT_k ← 0 ; TRT_k ← T_TRT ;                    /*procédure d'initialisation */
    Démarrer TRT_k (compteur décrémental)

Dans chaque station k, k = 1,2,...m :
    si TRT_k = 0 : procédure "récupération de l'anneau"          /* erreur*/

    À l'arrivée du jeton, faire:                  /* transfert de données */
        THT_k ← H_k ;
        TRT_k ← T_TRT ;
        Démarrer compteurs (décrémentaux) TRT_k et THT_k
        tant que THT_k > 0 et (messages temps-réel en attente):
            envoi messages temps-réel
        tant que THT_k > 0 et (messages non-temps-réel en attente):
            envoi messages non-temps-réel
        passage du jeton à la station (k+1) (modulo k)
```

Notons que ici le rôle du compteur TRT_k est limité uniquement à un rôle de vérification que la contrainte de protocole est assurée (dans le cas contraire, on déclenche une procédure de "récupération de l'anneau").

3.1.2. Prise en compte des caractéristiques des flux de messages temps-réel

La garantie d'un délai d'accès borné à une station est une condition nécessaire mais non suffisante pour garantir le respect des échéances associées aux flux de messages de la station. Afin de garantir ce respect pour tout flux temps-réel M_i (période P_i et durée d'utilisation maximale du réseau C_i), il faut que les deux contraintes suivantes (exprimées déjà pour le protocole "jeton temporisé") soient satisfaites:

1 le jeton passe dans la station pendant la période P_i;

2 le temps minimum X_i dont dispose la station pour transmettre le message du flux M_i sur la totalité des passages du jeton pendant la période P_i, est suffisant pour assurer la durée nécessaire d'utilisation du réseau.

La satisfaction de ces contraintes dépend évidement des spécifités du protocole "jeton temporisé régulier" et donc s'exprime par des relations différentes par rapport au cas du protocole "jeton temporisé".

La condition 1 implique que la période P_i soit supérieur au temps maximum de cycle (T_{TRT}):

$$P_i \geq T_{TRT}, \quad \forall i: i = 1...n \tag{4.9}$$

La condition 2 (qui traduit la contrainte de l'échéance) nécessite de déterminer le nombre minimum v_i de passages du jeton dans une station pendant une période P_i (on doit considérer

l'asynchronisme entre les requêtes et les passages du jeton dans le cas pire à la fois de déphasage et de passage du jeton; c.f. figure 4.5).

On obtient: $v_i = \lfloor P_i/T_{TRT} \rfloor$, ce qui donne $X_i = v_i \cdot H_i = \lfloor P_i/T_{TRT} \rfloor \cdot H_i$, où H_i est l'allocation de bande pour le flux de messages M_i à chaque passage de jeton.

La contrainte de l'échéance est donc:

$$\lfloor P_i/T_{TRT} \rfloor \cdot H_i \geq C_i \qquad \forall_{i=1...n} \tag{4.10}$$

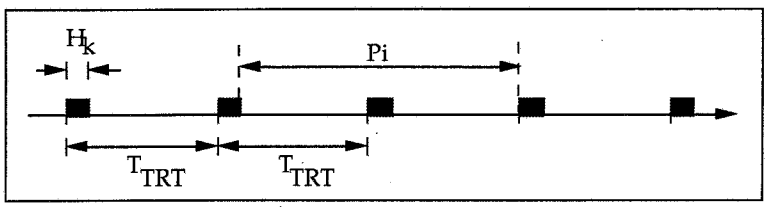

Figure 4.5: Évaluation de v_i dans la station k

3.1.3. Algorithmes d'allocation de bande passante

Nous proposons l'adaptation des algorithmes d'allocation de bande passante "durée complète" et "proportionnel adapté" définis pour le protocole "jeton temporisé" (c.f. III.4.4.1).

Cependant, nous considérons ici que l'on peut avoir plusieurs flux de messages dans une station k (la bande passante nécessaire à la station k (H_k) lors d'un passage de jeton est la somme des bandes passantes nécessaires aux différents flux de la station k lors de ce passage: $H_k = \sum_{nœud_k} H_i$) et nous généralisons donc les analyses faites avec le protocole "jeton temporisé" [ACZD, 92] [ACZ, 93] [ACZD, 94].

Concernant l'algorithme "proportionnel adapté", qui nécessite une fragmentation (contrairement à l'algorithme "durée complète"), nous l'analysons tout d'abord en négligeant le coût de la fragmentation (nécessite des en-têtes pour les fragments) et ensuite nous considérons ce surplus. A notre connaissance, une telle étude sur les aspects de la fragmentation n'a jamais été entreprise (voir les travaux présentés dans [ACZD, 92], [ACZ, 93], [ACZD, 94] et [MZ, 95]).

1. Algorithme "durée complète"

On prend $H_k = \sum_{nœud_k} C_i$, c'est-à-dire, on transmet à l'arrivée d'un jeton tous les messages temps-réel en attente. Notons que la contrainte de l'échéance est automatiquement satisfaite pour chaque flux: en effet, il y a au moins un passage de jeton par période d'un flux ($T_{TRT} \leq \min_i \{P_i\}$) et le message est transmis entièrement sur un passage.

Théorème:

La borne minimale de l'utilisation maximale U_x est asymptotiquement nulle

Démonstration

Considérons, réparti sur l'ensemble des stations, l'ensemble de flux de messages temps-réel $M = \{M_1, M_2 \ldots M_n\}$ suivant:

- on a *1* flux M_1 avec une période $P_1 = T_{TRT}$ et une durée C_1 infinitésimale: $C_1 = \varepsilon \ (\varepsilon \to 0)$;
- on a $n-1$ flux avec des périodes P_i, telles que $P_i = T_{TRT}/\varepsilon \ (\varepsilon \to 0)$, c'est-à-dire de longues périodes, et des durées identiques pour tous les flux $C_i = C$;

Cet ensemble de flux, où le flux M_1 sert a fixer la valeur de T_{TRT} ($T_{TRT} \leq min\, P_i$), représente un cas extrême de mauvaise efficacité pour l'algorithme (les $n-1$ flux de période $P_i = T_{TRT}/\varepsilon$ (période très grande) n'utilisent qu'un seul passage de jeton sur un grand nombre de passages).

Le taux d'utilisation U du réseau pour cet ensemble de flux de messages est:

$$U = \sum_{i=1}^{n} \frac{C_i}{P_i} = \frac{\varepsilon}{T_{TRT}} + \frac{(n-1)\cdot C\cdot \varepsilon}{T_{TRT}} = \frac{\varepsilon}{T_{TRT}}\cdot\left(1+(n-1)\cdot C\right) \qquad \varepsilon \to 0 \qquad (4.11)$$

La contrainte du protocole est:

$$\sum_{k=1}^{m} H_k = \sum_{k=1}^{m}\left\{\sum_{nœud_k} C_i\right\} = \sum_{i=1}^{n} C_i = \varepsilon + (n-1)\cdot C \leq T_{TRT}(1-\alpha)$$

Intégrons la contrainte du protocole dans l'expression du taux d'utilisation. Nous obtenons: $U \leq \frac{\varepsilon}{T_{TRT}}\left(1+\left(T_{TRT}(1-\alpha)-\varepsilon\right)\right)$ et, lorsque $\varepsilon \to 0$, on a $U \to 0$. Ceci montre que quand $\varepsilon \to 0$, la condition d'ordonnançabilité $U \leq U_x$ n'est vraie que pour $U_x = 0$.

Fin démonstration

Commentaire: En conséquence de l'utilisation maximale qui peut devenir asymtoptiquement nulle, on est obligé pour ordonnancer toute configuration de messages de vérifier la validité des contraintes temporelles.

2. Algorithme "proportionnel adapté" sans prise en compte du coût de fragmentation

On prend $H_k = \sum_{nœud_k} \frac{C_i}{v_i} = \sum_{nœud_k} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor}$, c'est-à-dire on répartit la transmission de chaque message sur les v_i arrivées successives du jeton (donc on envoie des fragments de message). Notons que la contrainte de l'échéance est automatiquement satisfaite pour chaque flux car tous les fragments sont transmis au moyen du nombre minimal de passages de jeton par période P_i d'un flux: en effet ce nombre minimal est $\lfloor P_i/T_{TRT} \rfloor$ et on transmet un fragment de durée $\frac{C_i}{\lfloor P_i/T_{TRT} \rfloor}$ par passage de jeton.

Théorème:

La borne minimale de l'utilisation maximale U_x est: $U_x = (1-\alpha)/2$

Afin de démontrer ce théorème, nous devons considérer le lemme suivant:

Lemme

Avec le protocole "jeton temporisé régulier", compte tenu de la contrainte $T_{TRT} \leq \min_i P_i$, on a:

$$\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}} \geq \frac{1}{2} \qquad (4.12)$$

Nous démontrons tout d'abord le lemme et ensuite le théorème.

Démonstration du lemme

Considérons le diagramme de la figure 4.6 qui visualise l'expression d'une période P_i d'un flux en fonction du nombre minimal d'arrivées du jeton pendant la période P_i ($\lfloor P_i/T_{TRT} \rfloor$) et du

T_{TRT}, en se plaçant dans le pire cas de déphasage entre les occurrences des messages de ce flux et les passages de jeton.

On voit que l'on a $P_i = \lfloor P_i/T_{TRT} \rfloor \cdot T_{TRT} + \delta_i$, avec $0 \le \delta_i < T_{TRT}$ et $\lfloor P_i/T_{TRT} \rfloor$ étant un entier x supérieur ou égal à 1, soit $\left\lfloor \frac{P_i}{T_{TRT}} \right\rfloor = \frac{P_i}{T_{TRT}} - \frac{\delta_i}{T_{TRT}}$. On obtient:

$$\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}} = \frac{\frac{P_i}{T_{TRT}} - \frac{\delta_i}{T_{TRT}}}{P_i/T_{TRT}} = 1 - \frac{\delta_i}{P_i} = 1 - \frac{\delta_i}{x \cdot T_{TRT} + \delta_i}$$

Le minimum de $\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}}$ est obtenu pour $x = 1$ et vaut $1/2$.

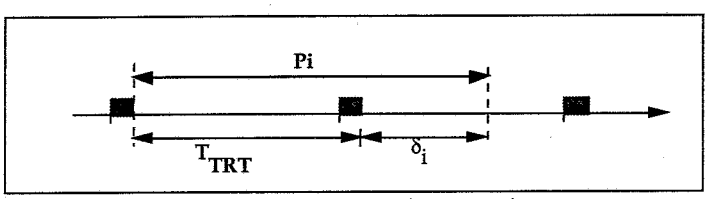

Figure 4.6: Relation entre P_i, $\lfloor P_i/T_{TRT} \rfloor$ et T_{TRT}

Fin démonstration (lemme)

Démonstration (théorème)

Considérons la contrainte du protocole $\sum_{k=1}^{m} \left\{ \sum_{nœud_k} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} \right\} \le T_{TRT} - \tau$, qui peut (en considérant seulement l'ensemble des flux) encore s'écrire: $\sum_{i=1}^{n} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} \le T_{TRT} - \tau$.

En intégrant le lemme $\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}} \ge \frac{1}{2}$ dans la contrainte du protocole, on obtient la relation suivante:

$$\sum_{i=1}^{n} \frac{C_i}{P_i} \cdot 2 \cdot T_{TRT} \le T_{TRT} - \tau$$

Soit comme $U = \sum_{i=1}^{n} \frac{C_i}{P_i}$, la relation précédente devient: $U \le \frac{1-\alpha}{2}$. Donc on a: $U_x = \frac{1-\alpha}{2} \rightarrow 50\%$ (α étant voisin de zéro).

Fin démonstration (théorème)

Commentaire: ce résultat ($U_x \rightarrow 50\%$) traduit le principal intérêt de ce protocole en terme de garantie d'ordonnançabilité de configuration de messages temps-réel, et montre sa supériorité (du point de vue du trafic temps-réel) par rapport au protocole du jeton temporisé ($U_x \rightarrow 33.3\%$). Dans le cas où α n'est pas voisin de zéro, il y a une dégradation de la borne U_x, quelque soit le protocole.

Intuitivement, on peut prévoir cette valeur de 50% pour U_x car le plus grand retard du jeton étant T_{TRT}, si la "périodicité" de tous les flux est de *"presque* $2 \cdot T_{TRT}$*"*, dans le pire cas chaque message n'aura qu'une arrivée du jeton pour être émis; Donc, comme la relation entre la "périodicité" des flux et le temps de cycle de $1/2$, l'utilisation maximale ne peut pas dépasser les 50%.

3. Algorithme "proportionnel adapté" avec prise en compte du coût de fragmentation

Caractérisation de la fragmentation

Appelons φ la durée de l'en-tête d'un fragment (nous considérons que la durée φ ne dépend que de l'implémentation du protocole, c'est-à-dire φ a une valeur constante qui ne dépend pas de la longueur du fragment). Compte tenu qu'un message de durée d'utilisation du réseau C_i est fragmenté en v_i fragments, on peut définir la durée effective d'utilisation C_i^* du réseau pour ce message: on a $C_i^* = C_i + v_i \cdot \varphi$ avec $v_i = \lfloor P_i/T_{TRT} \rfloor$. La composante $v_i \cdot \varphi$ de C_i^* représente le surplus dû à la fragmentation.

Théorème:

La borne minimale de l'utilisation maximale U_x, en considérant le coût de fragmentation est $U_x = \frac{1-\alpha}{2} - \frac{n \cdot \varphi}{2 \cdot T_{TRT}}$, où n représent le nombre de flux de messages

Démonstration

Reprenons la contrainte du protocole exprimée dans le paragraphe précédent: $\sum_{i=1}^{n} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} \le T_{TRT} - \tau$ et, en remplaçant C_i par C_i^*, on obtient la contrainte du protocole avec la prise en compte du coût de fragmentation:

$$\sum_{i=1}^{n} \frac{C_i^*}{\lfloor P_i/T_{TRT} \rfloor} \le T_{TRT} - \tau$$

Le membre de gauche de cette relation peut encore s'écrire en tenant compte, d'une part, de l'expression C_i^* ($C_i + \lfloor P_i/T_{TRT} \rfloor \cdot \varphi$) et, d'autre part, du lemme $\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}} \ge \frac{1}{2}$:

$$\sum_{i=1}^{n} \frac{C_i^*}{\lfloor P_i/T_{TRT} \rfloor} = \sum_{i=1}^{n} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} + \sum_{i=1}^{n} \frac{\lfloor P_i/T_{TRT} \rfloor \cdot \varphi}{\lfloor P_i/T_{TRT} \rfloor} = \sum_{i=1}^{n} \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} + n \cdot \varphi \ge \sum_{i=1}^{n} \frac{C_i}{P_i} \cdot 2 \cdot T_{TRT} + n \cdot \varphi$$

La contrainte du protocole peut donc être exprimée par la relation suivante:

$2 \cdot T_{TRT} \cdot \sum_{i=1}^{n} \frac{C_i}{P_i} + n \cdot \varphi \le T_{TRT}(1-\alpha)$, soit $U \le \frac{1-\alpha}{2} - \frac{n \cdot \varphi}{2 \cdot T_{TRT}}$. Donc on a: $U_x = \frac{1-\alpha}{2} - \frac{n \cdot \varphi}{2 \cdot T_{TRT}}$

Fin démonstration

Commentaire

La borne U_x devient nulle lorsque $n \cdot \varphi = T_{TRT}(1-\alpha)$ (soit $T_{TRT} \approx n \cdot \varphi$, α étant voisin de zéro) ce qui est logique: ça traduit que les longueurs des en-têtes de tous les fragments occupent tout le temps du T_{TRT}.Cette expression est intéressante car elle montre bien, d'une part, l'influence de la fragmentation et, d'autre part, la nécessité de ne pas trop fragmenter si on veut avoir des valeurs de U_x supérieures à zéro.

3.2. Conditions d'ordonnançabilité du trafic temps-réel dans le réseau Profibus [VJ, 93] [VJ, 94]

3.2.1. Réflexions sur l'algorithme d'arbitrage d'accès

L'algorithme d'arbitrage d'accès du protocole Profibus est représenté dans la procédure suivante (nous remplaçons la notation utilisée dans [DIN, 91] par une notation équivalente à celle du jeton temporisé: les variables T_{TR}, T_{TC}, T_{TH} et T_{RR} [DIN, 91] sont remplacées par, respectivement, T_{TRT}, τ/m, THT_k et TRT_k):

```
Algorithme d'arbitrage d'accès de Profibus

Dans chaque station k, (k = 1,2,...m):
    THT_k ← 0;                                   /*procédure d'initialisation */
    TRT_k ← 0;
    Démarrer TRT_k (compteur incrémental)

Dans chaque station k, (k = 1,2,...m), à l'arrivée du jeton, faire:
        THT_k ← T_TRT − TRT_k;
        TRT_k ← 0;
        Démarrer TRT_k (compteur incrémental)
        Si THT_k > 0:
            Démarrer THT_k (compteur decrémental)
        (fin si)
        transfert "1" message "haute-priorité"
        Tant que THT_k > 0 et (messages "haute-priorité" en attente)
            transfert messages "haute-priorité"
        Tant que THT_k > 0 et (messages "basse-priorité" en attente)
            transfert messages "basse-priorité"
        passage du jeton à la station (k+1)
```

Ce protocole utilise des concepts que l'on retrouve dans le protocole du "jeton temporisé", à savoir le concept de T_{TRT} et les concepts de jeton en avance ($THT_k > 0$) et de jeton en retard ($THT_k < 0$).

La différence fondamentale avec le protocole du "jeton temporisé" est qu'il n'y a pas d'allocation de bande passante fixant les possibilités du trafic temps-réel (pas de spécification de H_k), et par conséquent il n'y a pas de contrainte du protocole, c'est-à-dire, une relation entre les H_k et le T_{TRT}. Plus précisément, on peut voir que lorsque le jeton qui arrive dans une station k est en avance ($THT_k > 0$), et si le trafic non-temps-réel dans cette station n'est pas contraint, on peut avoir des valeurs de $THT_x < 0$ pour les stations suivantes ($x = (k+1), \ldots m, 1, \ldots (k-1)$). Cela peut affecter les possibilités de transfert de trafic temps-réel dans ces stations, qui ne vont pouvoir transmettre qu'un seul message temps-réel. On a donc une influence du trafic non-temps-réel d'une station sur les possibilités en trafic temps-réel des stations suivantes. En conséquence, il est important de noter que si le trafic non-temps-réel n'est pas contraint, les possibilités en trafic temps-réel sont toujours réduites à un seul message temps-réel par arrivée du jeton dans une station et ceci quel que soit le choix du T_{TRT}. Donc, du point de vue strictement de la mise en oeuvre du trafic temps-réel, sur la base de 1 seul message par passage de jeton, on n'a pas de contrainte sur la valeur associée au T_{TRT}. Cette modalité de fonctionnement définit ce que nous appelons le profil originel ou profil 1 du réseau Profibus.

On peut cependant considérer un autre profil de fonctionnement, en supposant que toute station contrôle le débit associé à ses flux de messages non-temps-réel de façon à ce qu'elle puisse toujours transmettre tous les messages des flux temps-réel présents lors de l'arrivée d'un

jeton. Cette modalité de fonctionnement, qui est basée sur la satisfaction d'une contrainte de protocole (le temps de cycle doit être tel que tous les besoins de trafic temps-réel, présents lors de l'arrivée d'un jeton, puissent être satisfaits (valeur de THT_k suffisante et donc temps de cycle inférieur à T_{TRT})) définit ce que nous appelons le profil 2 du réseau Profibus. Il est important de remarquer que, par rapport au profil 1, on a maintenant une contrainte sur la valeur de T_{TRT}.

Nous évaluons maintenant ces deux profils par rapport aux possibilités de trafic temps-réel.

3.2.2. Évaluation du profil 1 (trafic non-temps-réel non contraint)

Considérons le cas pire des conséquences de ce profil, c'est-à-dire, une station qui utilise tout le temps défini par le T_{TRT}, ce qui amène ensuite les autres stations à ne pouvoir transmettre qu'un seul message temps-réel à l'arrivée du jeton.

Nous supposons que la durée maximale d'utilisation du réseau nécessaire pour chaque flux temps-réel est égale à la durée permise par le transfert du seul message temps-réel qu'une station peut envoyer quand elle reçoit le jeton (appelons C cette durée).

1. Contrainte de l'échéance

Appelons $T_{CYCLE_{max}}$ la valeur maximale du temps de cycle du jeton. Afin que la contrainte de l'échéance d'un flux de messages temps-réel Mp_i (période P_i) appartenant à une station k soit satisfaite, il faut que le nombre de passages du jeton dans la station k, pendant la période P_i, soit supérieur ou égal à 1, c'est-à-dire on doit avoir la relation suivante: $\lfloor P_i/T_{CYCLE_{max}} \rfloor \geq 1$. Cette relation peut encore s'écrire comme $1/\lfloor P_i/T_{CYCLE_{max}} \rfloor \leq 1$, où $1/\lfloor P_i/T_{CYCLE_{max}} \rfloor$ représente le taux d'utilisation des passages du jeton pendant la période P_i, ce qui traduit encore les besoins du service du jeton pour le flux P_i appartenant à la station k.

En conséquence, les échéances de tous les flux appartenant à une station k seront satisfaites si les besoins de service du jeton sont inférieurs ou égaux à la capacité de service des passages du jeton dans la station k, c'est-à-dire:

$$\sum_{M_i \in nœud_k} \frac{1}{\lfloor P_i/T_{CYCLE_{max}} \rfloor} \leq 1, \quad \forall k,\ 1 \leq k \leq m \tag{4.13}$$

Cette formule traduit une contrainte sur la valeur de $T_{CYCLE_{max}}$, valeur qui est fixée par la station k qui possède le flux avec la plus petite période P_i et qui dépend du nombre total de flux que possède cette station k. Plus précisément, la valeur de $T_{CYCLE_{max}}$ est d'autant plus faible par rapport à la plus petite période P_i qu'il y a un nombre important de flux dans la station k. Par voie de conséquence, on aura dans le cas le plus général, des passages de jetons dans des stations qui fonctionneront "à vide" (pas de message à transmettre).

2. Autre contrainte sur la valeur $T_{CYCLE_{max}}$

Théorème:

La valeur maximale du temps de cycle $T_{CYCLE_{max}}$ est:

$$T_{CYCLE_{max}} = T_{TRT} + \tau + (m-1) \cdot C \tag{4.14}$$

où τ est le temps de propagation du jeton sur l'anneau logique, C est la durée maximale d'un message temps-réel et m est le nombre de stations.

Démonstration:

Supposons que dans le cycle $(l+1)$ le jeton arrive au plus tôt à la station k, c'est-à-dire, aucun trafic n'est transmis par aucune station pendant le cycle de jeton précèdent (figure 4.7); donc, $TRT_k = \tau$ et THT_k est initialisé à $T_{TRT} - \tau$. Dans ce cas, la station peut transférer du trafic pendant $T_{TRT} - \tau$ unités de temps, jusqu'à ce que son THT_k soit écoulé.

Cependant, si la station k laisse écouler son THT_k, la station $k+1$ recevra le jeton à l'instant où $TRT_{k+1} = T_{TRT}$ et donc THT_{k+1} sera initialisé à zéro (car si $t_{k+1}(l) - t_k(l) = \tau/m$ et $t_{k+1}(l+1) - t_k(l) = T_{TRT} + \tau/m$, alors $TRT_k = t_{k+1}(l+1) - t_{k+1}(l) = T_{TRT}$ et donc la valeur de THT_{k+1} est nulle).

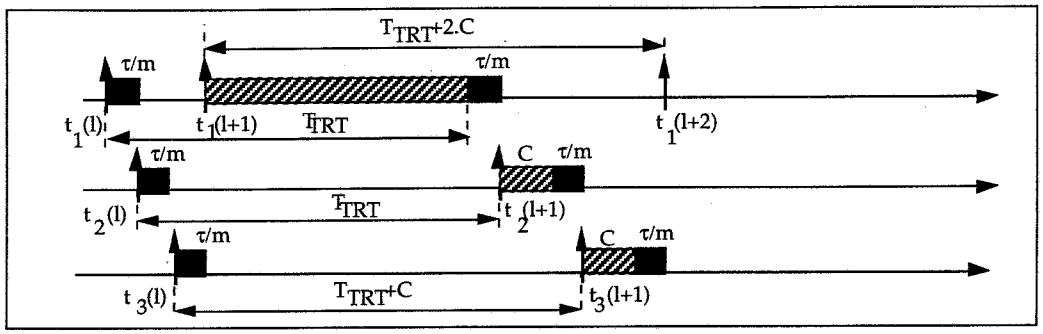

Figure 4.7: Évaluation du temps maximal de cycle du jeton

Si, à l'arrivée du jeton à la station $k+1$, la valeur de THT_{k+1} est nulle, la station n'a le droit de transférer qu'un seul message; par conséquent, à l'arrivée du jeton sur la station suivante: $TRT_{k+2} = T_{TRT} + C$, et ainsi de suite. Dans ce cas, $T_{CYCLE_{max}} = T_{TRT} + (m-1) \cdot C$.

Dans la figure 4.7, nous présentons le cas où $k = 3$ et donc $TRT_{max} = T_{TRT} + 2 \cdot C$.

Cependant, ceci n'est pas le temps de cycle maximal car, dans le cas particulier du démarrage du système, les TRT_k's démarrent simultanément et donc le temps de cycle maximal est $T_{CYCLE_{max}} = T_{TRT} + \tau + (m-1) \cdot C$ (figure 4.8).

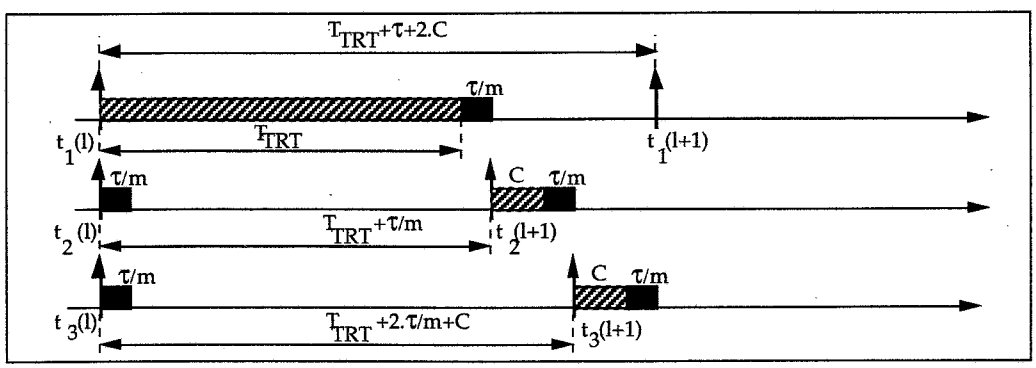

Figure 4.8: Évaluation du temps maximal de cycle du jeton

3.2.3. Évaluation du profil 2 (trafic non-temps-réel contraint)

Considérons m stations, n flux de messages temps-réel, m flux de messages non-temps-réel et T_{CYCLE} la durée d'un cycle. Nous exprimons, tout d'abord, la contrainte de protocole à partir de laquelle nous obtenons la valeur maximale du temps de cycle et le délai d'accès. Ensuite, nous donnons la condition pour que la contrainte de l'échéance soit satisfaite.

1. Contrainte de protocole

Afin que tous les messages présents lors de l'arrivée du jeton dans les stations puissent être transmis, on doit avoir la relation suivante:

$$\sum_{i=1}^{n}\left\{C_i \cdot \left\lceil \frac{T_{CYCLE}}{P_i} \right\rceil\right\} + \sum_{j=1}^{m}\left\{C_j \cdot \left\lceil \frac{T_{CYCLE}}{P_j} \right\rceil\right\} + \tau + \max_{k=1\ldots m}\left\{\sum_{nœud_k} C_i\right\} \leq T_{TRT} \qquad (4.15)$$

où le premier terme et le deuxième terme représentent les bornes maximales, respectivement, du trafic temps-réel et du trafic non-temps-réel sur un cycle; la somme de ces deux termes représente la charge maximale sur un cycle. Le troisième terme représente toujours le temps de propagation du jeton sur l'anneau logique plus la durée du jeton. La somme de ces trois premiers termes représente la durée du temps de cycle. Le quatrième terme représente la borne maximale du trafic temps-réel qu'une station doit transmettre lorsque le jeton lui revient après un cycle (ce terme représente la valeur maximale que doit avoir le *THT* de la station lorsque le jeton lui revient).

2. Contrainte de l'échéance

Comme à l'arrivée du jeton dans une station, on a toujours le temps suffisant pour transférer tous les messages temps-réel en attente , la garantie de l'échéance d'un flux de période P_i est obtenue si le jeton arrive à la station au moins une fois pendant la période P_i.

$$P_i \geq T_{CYCLE_{max}}, \quad \forall i,\ 1 \leq i \leq p \qquad (4.16)$$

C'est donc le flux de plus petite période ($\min_i\{P_i\}$) qui représente la borne supérieure de $T_{CYCLE_{max}}$.

3. Expression de $T_{CYCLE_{max}}$

Compte tenu de la contrainte de l'échéance (qui implique que $\left(\lceil T_{CYCLE}/P_i \rceil = 1,\ \forall_i\right)$) et en considérant également que $\lceil T_{CYCLE}/P_j \rceil = 1,\ \forall_j$, on obtient l'expression de la durée de $T_{CYCLE_{max}}$:

$$T_{CYCLE_{max}} = \sum_{i=1}^{n} C_i + \sum_{j=1}^{m} C_j + \tau \qquad (4.17)$$

Étant donnée la contrainte de l'échéance et l'expression de $T_{CYCLE_{max}}$, on doit encore avoir l'expression suivante:

$$\sum_{i=1}^{n} C_i + \sum_{j=1}^{m} C_j + \tau \leq \min_i\{P_i\} \qquad (4.18)$$

Il est important de remarquer, à partir de cette dernière relation, que si presque tous les besoins de trafic doivent être transmis sur un temps borné par $\min_i\{P_i\}$, on aura ici aussi des passages de jetons dans des stations qui fonctionnent "à vide" (stations où on a des flux de périodes P_i grandes par rapport à la valeur de la plus petite période).

4. Contrainte sur la valeur de T_{TRT}

En considérant, à la fois, la contrainte de protocole et la valeur de $T_{CYCLE_{max}}$, on obtient:

$$T_{CYCLE_{max}} + \max_{k=1\ldots m}\left\{\sum_{nœud_k} C_i\right\} \leq T_{TRT} \qquad (4.19)$$

3.2.4. Exemple d'application

Nous considérons le système suivant: on a m stations ($m = 10$; $\tau = 1$), qui ont chacune 3 flux temps-réel identiques $\{C_i, P_i\} = \{(1, 100), (1, 100), (C^*, 400)\}$, la durée C^* du dernier flux étant variable mais borné inférieurement par 1, c'est-à-dire $C^* > 1$ (notre objectif étant de voir les bornes maximales admissibles pour la durée C^* dans chaque profil); on a aussi, sur deux stations, 2 flux non-temps-réel: $\{C_j, P_j\} = \{(5, 500), (5, 500)\}$.

Profil 1:

Contrainte de l'échéance: $\sum_{M_i \in nœud_k} \frac{1}{\lfloor P_i / T_{CYCLE_{max}} \rfloor} \leq 1$, $\forall k$, $1 \leq k \leq m$; en développant, nous avons que pour chaque station, $\frac{2}{\lfloor 100/T_{CYCLE_{max}} \rfloor} + \frac{1}{\lfloor 400/T_{CYCLE_{max}} \rfloor} \leq 1$, et donc il faut que $\lfloor 100/T_{CYCLE_{max}} \rfloor > 2$, c'est-à-dire $T_{CYCLE_{max}} \leq 33.3$.

Temps de cycle: $T_{CYCLE_{max}} \leq T_{TRT} + \tau + (m-1) \cdot C$, c'est-à-dire: $T_{CYCLE_{max}} \leq T_{TRT} + 1 + 9 \cdot C^*$

Sachant que le temps de cycle est borné par $T_{CYCLE_{max}} \leq T_{TRT} + 1 + 9 \cdot C^*$, pour que la contrainte de l'échéance soit elle aussi satisfaite il faut que $T_{CYCLE_{max}} \leq (T_{TRT} + 1 + 9 \cdot C^*) \leq 33.3$, c'est-à-dire, il faut que $T_{TRT} \leq 32.3 - 9 \cdot C^*$.

Si on diminue T_{TRT} jusqu'à la valeur nulle, la durée C^* peut donc, augmenter jusqu'à 3.6; néanmoins, si on choisi $T_{TRT} = 0$, le trafic non-temps-réel ne pourra plus utiliser la largeur de bande laissée disponible par le trafic temps-réel.

Profil 2:

Contrainte de l'échéance: $P_i \geq T_{CYCLE_{max}}$; donc, $T_{CYCLE_{max}} = 100$;

Contrainte de protocole: $\sum_{i=1}^{n} \left\{ C_i \cdot \left\lceil \frac{T_{CYCLE}}{P_i} \right\rceil \right\} + \sum_{j=1}^{m} \left\{ C_j \cdot \left\lceil \frac{T_{CYCLE}}{P_j} \right\rceil \right\} + \tau + \max_{k=1...m} \left\{ \sum_{nœud_k} C_i \right\} \leq T_{TRT}$;

sachant que $T_{CYCLE} \leq P_i$, la contrainte de protocole impose $T_{TRT} \geq 11 \cdot (2 + C^*) + 11$.

$T_{CYCLE_{max}}$: Il faut que $\sum_{i=1}^{n} C_i + \sum_{j=1}^{m} C_j + \tau \leq \min_i \{P_i\}$, c'est-à-dire: $10 \cdot (2 + C^*) + 10 + 1 \leq 100$, ce qui impose $C^* \leq 6.9$.

La valeur de T_{TRT} découle de la contrainte du protocole: $T_{TRT} \geq 108.9$.

On peut donc, sur cet exemple, dire que pour le profil 1, l'ensemble de flux de messages est ordonnançable tant que $C^* \leq 3.6$ (pour $T_{TRT} = 0$), ce qui correspond à un taux d'utilisation des flux de messages temps-réel qui ne dépasse pas les 29%. Par contre, le profil 2 peut ordonnancer des flux de messages avec $C^* \leq 6.9$ ($T_{TRT} \geq 108.9$), ce qui correspond à un taux d'utilisation pour les flux de messages temps-réel nettement supérieur: $U = 37.25\%$.

En ce qui concerne le délai d'accès au réseau, dans les cas du profil 1 et 2, celui-ci est borné à, respectivement, $T_{CYCLE_{max}} = 33.3$ et $T_{CYCLE_{max}} = 100$, ce qui correspond à un délai d'accès plus intéressant dans le cas du profil 1.

3.2.5. Conclusion

Nous avons proposé deux profils de fonctionnement, qui correspondent à deux types de comportements des sources de trafic non-temps-réel (comportement non contraint (profil 1) et

comportement contraint (profil 2)) et nous avons donné pour chaque profil des conditions d'ordonnançabilité du trafic temps-réel.

Le profil 1 est caractérisé par un temps de cycle faible (ce qui permet des temps d'accès très courts pour les messages les plus prioritaires) et impose des durées courtes pour les messages.

Le profil 2 a un temps de cycle plus long et permet des durées de messages plus élevées.

4. Conclusion

Nous voulons souligner les points suivants:

- concernant les contributions sur les protocoles de classe 1:
  - nous avons conceptualisé un algorithme de changement en ligne de la séquence temporelle de flux périodiques ordonnancés par l'algorithme RM pour le réseau DQDB et donné, d'une part, les conditions pour effectuer des changements corrects (ni perte de message, ni duplication de message) et, d'autre part, le temps maximum pour rendre effectif un changement de mode (aspect très important dans un contexte temps-réel);
  - nous avons exprimé des conditions pour l'ordonnançabilité, dans le pire cas, du service du trafic apériodique urgent dans le réseau FIP en termes, d'une part, d'une contrainte de protocole afin de garantir une non surcharge du macro-cycle et, d'autre part, d'une contrainte de l'échéance basée sur des notions de retard d'avertissement de l'arbitre du bus et de retard d'ordonnancement. Ces expressions montrent que l'efficacité de ce service est faible (ce qui découle de la prépondérance accordée au trafic périodique dans le réseau FIP);
- concernant les contributions sur les protocoles de classe 2:
  - nous avons défini un protocole appelé "jeton temporisé régulier" qui, par rapport au protocole du "jeton temporisé" présente une régularité de cycle (le temps de cycle est borné par T_{TRT}, alors que la borne est de $2 \cdot T_{TRT}$ dans le "jeton temporisé") ce qui permet une meilleure borne minimale de l'utilisation maximale avec l'algorithme d'allocation de bande passante proportionnel adapté (50% par rapport à 33.3%) et donc présente un intérêt supérieur pour le trafic temps-réel; nous avons également évalué le coût de la fragmentation inhérente à l'algorithme proportionnel adapté.
  - nous avons défini deux profils de fonctionnement pour le réseau Profibus, à savoir le profil 1 (on considère que le trafic non-temps-réel n'est pas contraint) et le profil 2 (on considère que le trafic non-temps-réel est contraint), et nous avons défini des conditions d'ordonnançabilité pour ces deux profils; le profil 1 est caractérisé par un temps de cycle faible (ce qui permet des temps d'accès très courts pour les messages les plus prioritaires) et impose des durées courtes pour les messages; le profil 2 a un temps de cycle plus long mais contraint moins les durées des messages.

5. Références

[ACZD, 92] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines in High-Speed Token-Ring Networks with the Timed Token Protocol; in Proc. of 12th IEEE Distributed Computing Systems, Yokohama, pages 468-475, 1992;

[ACZ, 93] G. Agrawal, B. Chen and W. Zhao; Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the Timed Token Protocol; Proc. of INFOCOM'93

[ACZD, 94] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol; in IEEE Transactions on Computers, vol. 43, no 3, pages 327-339, March 1994;

[CVJ, 94] R. Carmo, F. Vasques, G. Juanole; Real-Time Communication Services in a DQDB Network; IEEE RTSS'94, pages 249-258, S. Juan, 1994

[JCV, 94] G. Juanole, R. Carmo, F. Vasques; Connection Oriented Isochronous Services in a DQDB Network: Specification of a Service-Protocol pair and a Bandwidth Allocation Scheme; Proc. of International Conference on Local and Metropolitan Communication Systems, pages 377-396, Kyoto, Japan, 1994

[DIN, 91] DIN 19245 Part 1 & 2: Process Field Bus (English translation of German version); April 1991

[Gro, 82] R. Grow; A Timed Token Protocol for Local Area Networks; IEEE Electro'82, Token Access Protocols, 17/3, 1982

[IEEE, 90] Distributed Queue Dual Bus (DQDB) Subnetwork of Metropolitan Area Network, IEEE Std 802.6 - 1990

[LL, 73] C. Liu, J. Layland; Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment; Journal of the ACM 20 (1), pages 46-61, 1973;

[MZ, 95] N. Malcolm, W. Zhao; Hard Real-Time Communication in Multiple-Access Networks; The Journal of Real-Time Systems, 8(1), pages 35-77, 1995;

[NFa, 90] FIP Bus for Exchange of Information between Transmitters, Actuators and Programmable Controllers, Data Link Layer; NF C 46-603, June 1990

[VJ, 93] F. Vasques, G. Juanole; Fieldbus MAC Mechanisms for Hard Real-Time Data Communication Support; proc. of OBS'93: Open Bus Systems, pages 225-232; Munich, Germany, November 1993.

[VJ, 94] F. Vasques, G. Juanole; Pre-Run-Time Schedulability Analysis in Fieldbus Networks; proc. of IECON'94, pages 1200-1204; Bologna, Italy, September 1994.

Chapitre V

Proposition, modélisation et analyse de mécanismes pour l'ordonnancement de trafic temps-réel dans une sous-couche MAC

1. Introduction

Un critère fondamental des protocoles de la sous-couche MAC est la justesse temporelle de la mise en circulation, sur la ressource de transmission, des messages des différents flux temps-réel. Cette justesse temporelle dépend évidemment de la différence entre l'instant d'occurrence d'un message dans un flux et l'instant de la mise en circulation de ce message.

Nous avons, au chapitre III, classé les principaux protocoles MAC suivant deux classes de base (classe 1 et 2) qui peuvent considérer des trafics périodiques et apériodiques temps-réel et implanter des techniques d'ordonnancement hors-ligne ou en-ligne.

En ce qui concerne le trafic périodique, un ordonnancement hors-ligne est très efficace [B_all, 92]lorsqu'on a un trafic bien défini une fois par toutes; par contre, si on a des fréquents changements de mode (un changement de mode nécessite, tout d'abord, de connaître les caractéristiques du nouveau trafic, puis de calculer une nouvelle séquence d'ordonnancement, ce qui introduit donc un retard dans la mise en oeuvre de la nouvelle séquence) on doit se poser la question du choix d'un algorithme en-ligne ou hors-ligne.

En ce qui concerne le trafic apériodique temps-réel, comme il n'est pas connu a priori, la notion d'ordonnancement hors-ligne n'est pas adaptée (nous disons même qu'elle n'a pas de sens car, par définition, on ne peut pas construire, au préalable une séquence temporelle) et on doit donc mettre en oeuvre un ordonnancement en-ligne.

Dans le contexte le plus général possible des réseaux temps-réel, il faut considérer que l'on a, à la fois, les deux types de trafic temps-réel (périodique et apériodique) et également du trafic apériodique non-temps-réel. En conséquence, il faut définir des mécanismes qui permettent de réaliser un ordonnancement conjoint et efficace de tous les types de trafic.

C'est cette idée clef qui a introduit la proposition que nous allons présenter, proposition qui est basée sur la norme IEC-65C [IEC, 94] qui considère un accès centralisé des stations à la

ressource de transmission (le contrôleur d'accès est la station appelée "Link Active Scheduler": LAS) et qui peut mettre en oeuvre en même temps des protocoles de classe 1 et de classe 2 (c.f. III.5), donc permettre la réalisation d'un ordonnancement conjoint des différents types de trafic.

Notons de plus que, dans le domaine de complexité d'architectures de communication avec des contraintes temporelles, il est absolument indispensable de vérifier et d'évaluer toute proposition.

En conséquence, ce chapitre comprend trois parties:

- la première partie est concernée par la présentation de la proposition;
- la deuxième partie présente la modélisation des différents mécanismes au moyen du modèle "Réseaux de Petri Temporisés Stochastiques";
- la troisième partie présente l'analyse effectuée sur le trafic temps-réel.

2. Proposition

2.1. Réflexion sur l'ordonnancement conjoint de tous les types de trafic

Considérons donc les trois types de trafic (périodique, apériodique temps-réel et non-temps-réel) qui doivent être supportés au niveau de la sous-couche MAC des réseaux temps-réel:

- Le trafic *périodique*, dans un mode de fonctionnement, a des caractéristiques bien définies (période, durée maximale d'utilisation du réseau, échéance). Une garantie du respect des échéances doit être assurée.
- Le trafic *apériodique temps-réel* a, à la fois, des caractéristiques connues (intervalle minimum entre deux arrivées consécutives, durée maximale d'utilisation du réseau, échéance) et inconnues (instants d'arrivée des messages). Ici aussi une garantie du respect des échéances doit être assurée.
- Le trafic *apériodique non-temps-réel* n'a pas de caractéristiques connues. Un effort de transfert au plus tôt doit être effectué afin de minimiser le retard de transfert.

Dans un contexte distribué, c'est l'ordonnancement en-ligne du trafic apériodique temps-réel qui pose le plus grand problème, compte tenu de la difficulté de la connaissance au plus tôt des instants d'arrivées des messages (c'est la grande différence par rapport à un système localisé en un point géographique, où cette connaissance est instantanée). On peut considérer deux grandes techniques pour le trafic apériodique temps-réel:

- soit on a un ordonnancement à un instant t basé sur une connaissance globale des arrivées des messages (notion d'ordonnancement global): les prises de décision de l'ordonnancement sont basées sur une connaissance globale des messages dans les différentes stations (cette vue est cependant plus au moins en retard avec la vraie situation à l'instant t). Ce type d'ordonnancement peut être implanté de manière distribuée (CAN; IEEE 802.5).
- soit on n'a pas un ordonnancement basé sur une connaissance globale, mais on a un ordonnancement à deux niveaux: un niveau d'ordonnancement local des messages dans les stations et un niveau d'ordonnancement global des stations pour l'utilisation de le ressource de transmission.

L'ordonnancement global des stations peut être implanté de manière distribuée (intervalle de temps accordé aux stations, comme dans le cas des protocoles basés sur le jeton temporisé, ce qui réalise un anneau logique) ou de manière centralisé (un contrôleur d'accès interroge successivement les stations ou accorde un intervalle de temps successivement à chaque station, ce qui revient encore à réaliser un anneau logique). Dans l'implantation centralisée, la technique d'interrogation est lourde et peu efficace (il faut interroger les stations pour savoir si elles ont des messages à envoyer). Par contre, la technique de l'anneau logique (qui peut être implantée de manière distribuée (la manière traditionnelle) ou centralisée (une station maître possède le contrôle de l'anneau) est intéressante. En effet, les protocoles basés sur le jeton temporisé (c.f. III.4.4.1) offrent (dans la mesure où ils sont conçus en intégrant bien les caractéristiques connues du trafic apériodique temps-réel) des possibilités de satisfaire en-ligne les exigences des ordonnancements locaux (avec un retard borné supérieurement). En outre, les protocoles sur la base de la technique d'allocation de bande proportionnelle adaptée, offrent des bornes minimales U_x de l'utilisation maximale (33% pour le jeton temporisé et 50% pour le jeton temporisé régulier, dans le cas où il n'y a pas de fragmentation de messages), ce qui, d'une part, garantit automatiquement pour des ensembles de messages M, tels que $U(M) < U_x$, le respect des contraintes temporelles et, d'autre part, laisse de la bande passante pour ordonnancer le trafic périodique.

Dans le contexte de la norme IEC-65C, qui est basée sur un contrôle d'accès centralisé dans la station LAS, nous adoptons, pour ordonnancer le trafic apériodique temps-réel, les mécanismes de mise en oeuvre du concept "jeton temporisé" qu'elle offre.

En ce qui concerne le trafic périodique, la norme IEC-65C permet, en introduisant dans le LAS la connaissance des caractéristiques des différents flux périodiques, de mettre en oeuvre un ordonnancement centralisé (le LAS donnant successivement la parole aux flux dans les stations). La prise de décision de l'ordonnancement, c'est-à-dire la génération de la séquence de droits de parole accordés aux flux, peut être réalisée soit hors-ligne (technique utilisée dans FIP avec la notion de table de scrutation), soit en-ligne. Par la suite, nous utilisons les termes d'ordonnancement hors-ligne et en-ligne pour traduire, respectivement, le premier et deuxième types de décision. Les algorithmes en-ligne sont évidemment mieux adaptés à la mise en oeuvre du changement de mode. Dans le cadre de ce travail, nous considérons des algorithmes en-ligne qui nécessitent, en particulier, des activités de vérification que la séquence temporelle générée dynamiquement permet bien la satisfaction des échéances.

En ce qui concerne l'ordonnancement des flux de trafic apériodique non-temps-réel, le LAS permet de le réaliser en donnant la parole à ces flux dans la mesure du temps laissé disponible par les ordonnancements des deux autres types de trafic.

En conclusion, dans le contexte de la norme IEC-65C et des possibilités d'ordonnancement centralisés dans le LAS qu'elle offre, nous proposons les modalités d'ordonnancement conjoint suivantes:

- ordonnancement en-ligne pour le trafic périodique (algorithmes RM ou ED);
- ordonnancement basé sur le concept du jeton temporisé pour le trafic apériodique temps-réel;
- ordonnancement suivant la politique du "meilleur effort" pour le trafic apériodique.

Nous présentons maintenant cette proposition.

2.2. Cadre de la proposition

1. Hypothèses

Le réseau considéré comporte m stations, où se trouvent les différents flux de messages à ordonnancer, et la station LAS qui réalise l'ordonnancement.

On considère que sont répartis sur les m stations:

- p flux de messages périodiques $\left(Mp_1, Mp_2, \ldots Mp_i, \ldots Mp_p\right)$ à échéance sur requête; on appelle, respectivement, P_i et C_i la période et la durée d'utilisation du réseau par un message du flux Mp_i;
- s flux de messages apériodiques temps-réel $\left(Ma_1, Ma_2, \ldots Ma_j, \ldots Ma_s\right)$ à échéance sur requête; on appelle, respectivement, P_j et C_j l'intervalle minimal entre deux arrivées successives et la durée d'utilisation du réseau par un message du flux Ma_j;

On suppose de plus que l'on a un flux de **M**essages **n**on-temps-réel (Mn) par station; on appelle C_n la durée d'utilisation maximale du réseau par un message du flux Mn.

2. Définition de trois types de jetons

L'ordonnancement est basée sur l'utilisation de trois types de jetons: EP ("**E**xecute **P**ériodique"), EA ("**E**xecute **A**périodique") et EN ("**E**xecute **N**on-temps-réel"). Ils permettent de déclencher le transfert de trafics en provenance, respectivement, de flux périodiques (Mp), de flux apériodiques (temps-réel (Ma) et/ou non-temps-réel (Mn)) et de flux non-temps-réel (Mn).

Plus précisément, le LAS génère une séquence temporelle de jetons EP, EA et EN de manière à ce que:

- d'une part, les contraintes temporelles des flux temps-réel (périodiques et apériodiques) soient respectées;
- d'autre part, le "meilleur effort" pour les flux non-temps-réel soit équitablement réparti.

Dans ce document, nous n'analysons que la génération d'une séquence temporelle de jetons EP et EA pour servir le trafic temps-réel (périodique et apériodique), et dans quelle mesure le respect des contraintes temporelles associées à ce type de trafic est garanti. En ce qui concerne l'inclusion de la séquence des jetons EN, nous présentons brièvement une stratégie possible d'implantation.

3. Objectif de l'ordonnancement dans le LAS

L'objectif est de générer une séquence temporelle de jetons telle que:

- cas des flux de messages périodiques: chaque flux Mp_i reçoit, durant sa période P_i, un nombre v_i de jetons EP ($v_i = 1$ si pas de fragmentation; $v_i > 1$ si fragmentation) de manière à ce que tous les fragments d'un message de ce flux puissent être transférés avant qu'un nouveau message de ce flux ne soit produit;
- cas des flux de messages apériodiques temps-réel: chaque flux Ma_j reçoit, durant chaque période P_j, un nombre v_j de jetons EA ($v_j = 1$ si pas de fragmentation; $v_j > 1$ si fragmentation) de manière à ce que tous les fragments d'un message de ce flux puissent être transférés avant l'arrivée la plus rapide possible d'un nouveau message de ce flux;

- cas des flux de messages apériodiques non-temps-réel: chaque station reçoit des jetons EN pour servir le flux *Mn* de son trafic non-temps-réel (compte tenu de la définition de ce trafic, nous ne définissons aucune contrainte en termes de nombre de passages de jeton, contrairement aux autres trafics).

2.3. L'ordonnancement du trafic apériodique temps-réel

1. Principe

Nous proposons d'implanter un mécanisme d'anneau logique entre les stations, c'est-à-dire, on a une séquence de fenêtres temporelles de durée T et, à l'intérieur de chaque fenêtre temporelle, chaque station k pourra disposer d'un temps H_k pour utiliser la ressource de transmission.

2. Le mécanisme

Le mécanisme est implanté au moyen de passages successifs du jeton EA entre le LAS et les différentes stations. Le LAS envoie à chaque station k, durant chaque fenêtre temporelle T, un jeton EA porteur d'une durée H_k pendant laquelle la station k doit pouvoir transmettre le trafic apériodique temps-réel prévu. Si les besoins du trafic apériodique temps-réel n'épuisent pas la durée H_k, la durée restante pourra être utilisée pour transférer du trafic apériodique non-temps-réel qui est en attente. La station k renvoie le jeton EA au LAS soit à la fin de la durée H_k soit avant, dés qu'il n'y a plus de trafic apériodique non-temps-réel en attente (retour anticipé du jeton EA).

Ce mécanisme est visualisé sur la figure 5.1 (on appelle τ_k la somme de la durée du jeton EA et du temps de propagation entre le LAS et la station k).

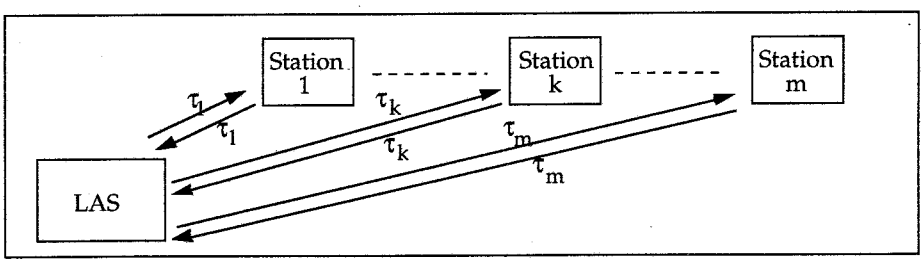

Figure 5.1: Le mécanisme du jeton EA

Le temps sur une fenêtre temporelle T, où le jeton n'est ni en transit entre le LAS et les stations, ni utilisé par les stations, est un temps dont dispose le LAS pour ordonnancer les autres trafics (périodique et apériodique non-temps-réel). Appelons H_{LAS} ce temps et posons $\sum_{k=1}^{m} 2 \cdot \tau_k = \tau$; on obtient:

$$T - \left(\sum_{k=1}^{m} H_k + \tau \right) \leq H_{LAS} \leq T - \tau \qquad (5.1)$$

H_{LAS} est égal à $T - \left(\sum H_k + \tau \right)$ si toutes les durées H_k sont utilisées par les stations (fonctionnement à pleine charge). H_{LAS} est égal à $T - \tau$ si les durées H_k ne sont pas utilisées par les stations.

Il est évidemment impératif que l'on ait la relation $T - \left(\sum_{k=1}^{m} H_k + \tau \right) > 0$ afin de pouvoir ordonnancer les autres types de trafic.

3. Caractérisation de l'anneau logique

Le temps séparant deux arrivées consécutives du jeton EA dans une station k peut varier. En effet, en considérant, d'une part, que l'on a un ordre total sur les passages des jetons EA dans les stations (ordre: $1, 2, \ldots k, \ldots m$) et, d'autre part, que les stations utilisent tout le temps dont elles disposent pour transmettre du trafic apériodique, on peut distinguer deux cas extrêmes de fonctionnement: le cas de la régularité des passages du jeton EA appelé cas 1 (figure 5.2); le cas de l'irrégularité des passages du jeton EA appelé cas 2 (figure 5.3):

- cas 1: Le LAS envoie sans interruption les jetons EA au début de chaque période T. En conséquence, la valeur de l'intervalle de temps séparant deux arrivées successives du jeton à une station k est toujours T (notion de régularité). Le LAS utilise ensuite le temps restant sur la période T pour servir les trafics périodique et apériodique non-temps-réel.

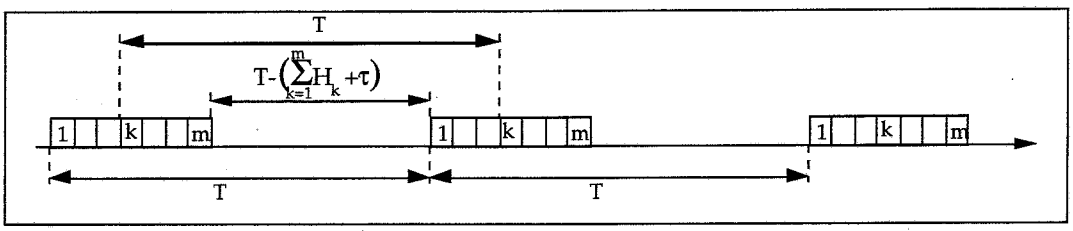

Figure 5.2: Cas 1 de la séquence de génération de jetons EA

Remarque: dans le cas où il y a un retour anticipé du jeton vers le LAS (une station n'utilise pas tout le temps attribué) et si on veut maintenir la régularité, il faut que le LAS utilise le temps résiduel pour générer du trafic périodique et/ou apériodique non-temps-réel, avant de passer le jeton EA à la station suivante.

- cas 2: Le LAS mixe l'envoi de jetons EA avec la sollicitation des autre types de trafic, ce qui induit des passages non réguliers du jeton EA à chaque station (notion d'irrégularité).

  On voit sur la figure 5.3, que l'intervalle de temps séparant deux arrivées successives du jeton à une station k est borné supérieurement par la valeur $\left\{2 \cdot T - \left(\sum_{k=1}^{m} H_k + \tau\right)\right\}$ et donc la valeur de la borne supérieure n'excède pas $2 \cdot T$.

Le fonctionnement du cas 2 peut être vu comme un fonctionnement "type protocole jeton temporisé" (c.f. III.4.4.1), avec $T = T_{TRT}$. Le fonctionnement du cas 1 peut être vu comme un fonctionnement "type protocole jeton temporisé régulier" (c.f. IV.3.1) avec $T = T_{TRT}$. Le fait que l'on puisse avoir un fonctionnement cas 1 ou cas 2 dépend de la modalité de l'ordonnancement conjoint (ceci va être présenté après au paragraphe 1.3.4).

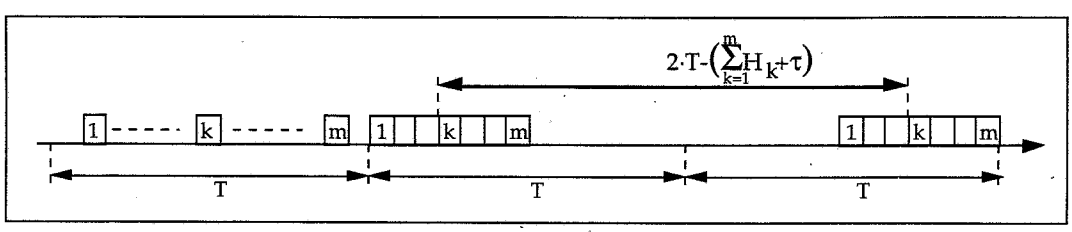

Figure 5.3: Cas 2 de la séquence de génération de jetons EA

Cependant, quel que soit le cas de fonctionnement, nous considérons un algorithme d'allocation de bande passante proportionnel adapté (c.f. III.4.4.1.3 et IV.3.1.3) compte tenu du taux d'utilisation U_x permis (valeur la plus élevée).

La spécification des protocoles concernés sera faite de la manière suivante:

- Considération relative aux flux apériodiques:

  $\forall j: j = 1 \ldots s$, il faut $P_j \geq T$ dans le cas 1 et $P_j \geq 2 \cdot T$ dans le cas 2 (c'est le flux qui a la plus petite valeur pour le temps minimal entre deux arrivées qui fixe T); ceci découle de la contrainte de l'échéance.

- Considération relative aux stations:

  Il faut, tout d'abord, calculer le nombre minimum v_j de passages de jeton dans la station k où se trouve le flux Ma_j (ceci découle de l'algorithme d'allocation de bande passante proportionnel adapté):

  - cas 1: $v_j = \lfloor P_j / T \rfloor$
  - cas 2: $v_j = \lfloor (P_j / T) - 1 \rfloor$

  A partir de la valeur de v_j, on obtient la longueur du fragment du flux Ma_j envoyé à chaque passage du jeton: C_j / v_j. On obtient finalement la bande passante nécessaire à la station k, à chaque passage du jeton EA, pour la transmission de ses flux apériodiques temps-réel: $H_k = \sum_{nœud\ k} C_j / v_j$.

4. Remarque

Le fonctionnement du LAS, pour l'ordonnancement du trafic apériodique temps-réel, peut être vu en termes du concept de serveur [SSL, 89], c'est-à-dire un mécanisme qui offre périodiquement une capacité de service:

- le serveur doit, au début de chaque intervalle de temps T, mettre dans une file d'attente un ensemble de m jetons EA (chaque jeton étant porteur d'une durée H_k) qui devront être envoyés aux m stations avant la fin de l'intervalle de temps T;
- l'intervalle de temps T représente la période du serveur et la somme des durées portées par les jetons $\left(\sum_{k=1}^{m} H_k\right)$ représente la capacité de service.

2.4. L'ordonnancement du trafic périodique

Nous proposons un ordonnancement en-ligne par assignation de priorités (protocole de classe 1; c.f. chapitre III) aux flux de messages Mp_i, $(i = 1, 2, \ldots p)$ qui peut être effectué soit avec l'algorithme RM non-préemptif, soit avec l'algorithme ED non-préemptif (algorithmes présentés au chapitre II et pour lesquels des tests d'ordonnançabilité ont été donnés).

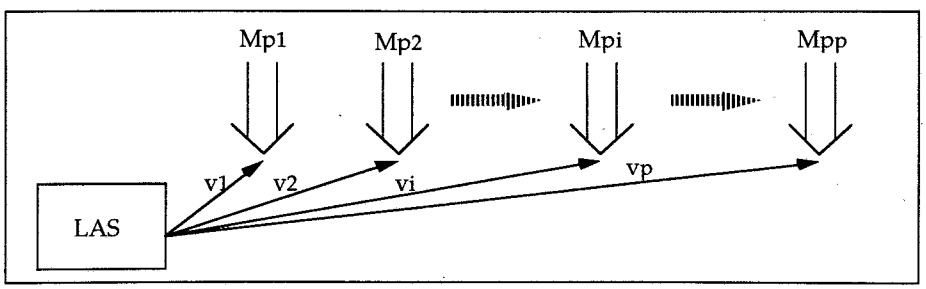

Figure 5.4: L'ordonnancement des flux de messages Mp_i

Le LAS implante cet ordonnancement en envoyant, à chaque flux Mp_i et durant sa période P_i, v_i jetons EP (figure 5.4); à chaque réception du jeton EP, le flux Mp_i envoie un fragment de

message de ce flux (notons qu'un jeton EP n'est pas porteur d'une durée: il entraîne une réponse immédiate).

En ce qui concerne la prise de décision de l'ordonnancement relativement aux flux à solliciter, elles sont de moindre complexité avec l'algorithme RM qu'avec l'algorithme ED, du fait que dans le premier les priorités sont statiques, alors qu'elles sont dynamiques dans le deuxième.

2.5. L'ordonnancement conjoint

1. Problématique

On doit définir un ordonnancement non-préemptif (c.f. I.4.2) et donc il est essentiel de limiter la durée maximale d'inversion de priorité (nécessité de spécifier une durée d'utilisation maximale du réseau pour un transfert). Nous appelons D cette durée d'utilisation maximale.

C'est dans le cadre, d'une part, des caractéristiques des flux temps-réel (périodiques et apériodiques) et, d'autre part, de cette contrainte sur l'utilisation maximale, que nous devons définir des échanges, relatifs à ces flux temps-réel, qui satisfont un test d'ordonnançabilité et qui permettent de faire le "meilleur effort" pour le trafic apériodique non-temps-réel.

En outre, comme l'ordonnancement du trafic apériodique temps-réel est basé sur le concept d'un serveur périodique, l'ordonnancement de ce serveur doit être situé par rapport aux flux périodiques et donc la spécification de l'ordonnancement des flux de trafic apériodique temps-réel est faite par rapport au choix initial que l'on fait pour ordonnancer le trafic périodique à savoir l'algorithme RM ou l'algorithme ED.

Nous présentons maintenant la méthode de spécification de l'ordonnancement, ainsi que le test d'ordonnançabilité.

2. Méthode de spécification

Échanges relatifs au trafic périodique

Il faut comparer la durée C_i relative à chaque flux Mp_i avec la durée D. Selon que l'on ait $C_i \leq D$ ou $C_i > D$, il n'y a pas ou il y a la nécessité de fragmenter les messages du flux Mp_i. Dans le cas où on fragmente (soit v_i fragments), ceci revient à remplacer un flux Mp_i par v_i flux Mp_i^*, chaque flux Mp_i^* ayant une période égale à $P_i^* = (P_i/v_i)$ et une durée $C_i^* = (C_i/v_i)$ (méthode de la transformation de la période [AB, 90].

Échanges relatifs au trafic apériodique temps-réel

La spécification de base est la spécification de la période T du serveur qui est conditionnée par le choix de l'algorithme RM ou de l'algorithme ED pour le trafic périodique.

En effet, avec l'algorithme RM si on décide d'accorder la plus haute priorité au serveur (choix de T tel que: $T < P_i$, $\forall_i$ si pas de fragmentation des messages des flux périodiques; $T < P_i^*$, $\forall_i$ si fragmentation des messages des flux périodiques), on aura des passages réguliers des jetons EA au début de chaque période T (fonctionnement du type "jeton temporisé régulier" (c.f. IV.3.1).

Par contre avec l'algorithme ED, même si on choisit pour le serveur une période plus faible que toutes les autres, comme la priorité dépend de l'échéance, on n'aura pas de passages

réguliers des jetons EA sur une période T (fonctionnement du type "jeton temporisé" (c.f. III.4.4.1)).

Donc, suivant que l'on choisit pour le trafic périodique l'algorithme RM ou l'algorithme ED, on a des passages réguliers ou irréguliers du jeton, et on doit donc considérer des formules différentes pour le nombre minimal de passages de jeton pendant la période d'un flux Ma_j.

En conséquence, nous résumons les contraintes que doit satisfaire le choix de la période T du serveur:

- si on fait le choix de l'algorithme RM pour le trafic périodique:
  1- $T \leq P_i$ ou $T \leq P_i^*$, $\forall i: i = 1...n$; cette contrainte exprime que l'on assigne la plus grande priorité au serveur;
  2- $T \leq P_j$, $\forall j: j = 1...s$; cette contrainte exprime la contrainte de l'échéance d'un message d'un flux apériodique temps-réel Ma_j;
  3- $H_k = \sum_{nœud_k} \frac{C_j}{\lfloor P_j/T \rfloor} \leq D$, $\forall k: k = 1...m$; cette contrainte exprime la contrainte de la durée d'utilisation maximale afin de limiter la durée de l'inversion de priorité (notons que si elle n'est pas satisfaite il faut que la station fragmente les messages des flux Ma_j sur la base de la technique de la transformation de la période [AB, 90]; notons que cette fragmentation n'est pas vue par l'ordonnanceur);
  4- $\sum_{k=1}^{m} H_k < T - \tau$, $\forall k: k = 1...m$; cette contrainte exprime, à la fois, la contrainte d'un protocole du type "jeton temporisé" et la contrainte pour pouvoir faire du transfert de trafic des flux périodiques Mp_i (si on n'avait pas de trafic périodique, la contrainte serait: ...≤...);
- si on fait le choix de l'algorithme ED pour le trafic périodique:
  1- Nous n'avons plus maintenant la première contrainte exprimée pour l'utilisation de l'algorithme RM;
  2- $T \leq P_j/2$, $\forall j: j = 1...s$; c'est la contrainte de l'échéance d'un message d'un flux Ma_j;
  3- $H_k = \sum_{nœud_k} \frac{C_j}{\lfloor (P_j/T) - 1 \rfloor} \leq D$, $\forall k: k = 1...m$; c'est la contrainte de la durée d'utilisation maximale pour limiter la durée de l'inversion de priorité;
  4- $\sum_{k=1}^{m} H_k < T - \tau$, $\forall k: k = 1...m$; c'est la contrainte de protocole du type "jeton temporisé" et la contrainte du transfert de trafic périodique;

Échanges relatifs au trafic apériodique non-temps-réel

Il faut borner la durée d'utilisation maximale du réseau par les messages des flux non-temps-réel afin de limiter la durée maximale d'inversion de priorité sur le bus. Il faut donc que, quel que soit le flux Mn, C_n soit inférieur à D.

Test d'ordonnançabilité conjoint des flux de messages temps-réel

- si on fait le choix de l'algorithme RM pour le trafic périodique:
  - Nous indiquons ici le deuxième test d'ordonnançabilité présenté au chapitre II (paragraphe II.4.3.1) et nous considérons le cas le plus général où on a de la fragmentation (soit n^* fragments pour l'ensemble de tous les flux périodiques):

$$\sum_{i^*=1}^{n^*} \frac{C_{i^*}}{P_{i^*}} + \frac{\sum_{k=1}^{m} H_k}{T} + \frac{D}{T} \leq (n^* + 1)\left(\sqrt[(n^*+1)]{2} - 1\right)$$

où le premier terme, le deuxième terme et le troisième terme du membre de gauche représentent, respectivement, les flux périodiques, les flux apériodiques temps-réel et la composante "inversion de priorité".

- si on fait le choix de l'algorithme ED pour le trafic périodique (nous indiquons ici le test présenté au chapitre II (paragraphe II.3.3 et II.4.1)):
  - $\sum_{i^*=1}^{n^*} \frac{C_{i^*}}{P_{i^*}} + \frac{\sum_{k=1}^{m} H_k}{T} \leq 1$; cette contrainte représente une garantie de non-surcharge du réseau par, simultanément, les messages des flux périodiques Mp_i (premier terme) et le serveur des flux de messages apériodiques temps-réel (deuxième terme);
  - $W = \max_{1<i\leq n}\left\{ \max_{k\cdot P_j^+ \in S_i}\left(\frac{C_i + \sum_{l=1}^{i-1}\left\lfloor \left(k\cdot P_j^+\right)/P_l \right\rfloor \cdot C_l}{k\cdot P_j^+} \right)\right\} < 1$, pour $S_i = \bigcup_{j=1}^{i-1}\left\{ \bigcup_{k=1}^{\lceil P_i/P_j - 1\rceil}\left(k\cdot P_j^+\right)\right\}$; cette contrainte assure que, quel que soit le flux de messages périodiques Mp_i ou le serveur, la demande normalisée maximale (c.f. II.4.1.3) demandée au réseau reste toujours inférieure à l'unité, c'est-à-dire, que quel que soit l'occurrence d'un message dans le pire cas de déphasage, celui-ci n'induit pas de dépassement d'échéances sur les messages des autres flux.

3. L'ordonnancement des flux de messages non-temps-réel

En ce qui concerne ces flux de messages, le LAS envoie un jeton EN successivement à chaque station sur le bus, tant qu'il n'y a pas de jetons EP ou EA en attente d'ordonnancement. Un jeton EN est un jeton auquel est associé une durée (comme le jeton EA). Ce mode de fonctionnement implante encore un anneau logique entre les stations sur le bus, de telle sorte que chacune puisse avoir un droit d'accès équitable pour transférer son trafic non-temps-réel (nous rappelons que ce trafic est aussi véhiculé par les jetons EA, quand, dans une station, il n'y a plus de trafic apériodique temps-réel à transférer).

La durée maximale associé aux jetons EN dépend elle aussi évidemment de la durée maximale admissible pour les inversions de priorité sur le bus.

2.6. La syntaxe des jetons proposés

Nous proposons la syntaxe suivante pour les trois jetons:

- $EP\langle adr\rangle$, où *adr* représente l'adresse d'un flux périodique Mp_i. Nous ne spécifions pas une durée pour ce jeton, car il ne permet que l'envoi d'un seul fragment de message par transaction;
- $EA\langle adr; durée\rangle$, où *adr* et *durée* représentent, respectivement, l'adresse d'un ensemble de flux de messages apériodiques d'une station k et l'allocation de bande passante H_k;
- $EN\langle adr; durée\rangle$, où *adr* et *durée* représentent, respectivement, l'adresse d'une station k et la durée de séjour du jeton;

Ces jetons peuvent être situés par rapport aux types de jetons que nous avons présentés au chapitre III sur le protocole de classe 1&2 IEC-65C (paragraphe III.5.2) de la manière suivante:

- Le jeton EP peut être facilement supporté par la technique du jeton délégué à réponse immédiate (type 1 ou 2), car cette technique permet l'adressage d'un flux de messages défini comme étant périodique:
- Les jetons EA et EN peuvent être supportés par la technique du jeton délégué temporisé (type 3), car cette technique permet l'adressage d'un ensemble de flux apériodiques dans une station dans le cas de EA et l'adressage d'une station dans le cas EN.

3. Modélisation

3.1. Contexte

L'architecture du système à modéliser au moyen des Réseaux de Petri Temporisés Stochastiques (RdPTS) consiste en une station LAS et m stations utilisatrices du bus dans lesquelles on trouve des besoins en trafic temps-réel répartis sous la forme de p flux périodiques et de s flux apériodiques. On a également un flux apériodique non-temps-réel par station

En conséquence, il est nécessaire d'établir 4 modèles:

- le modèle du LAS (mise en oeuvre de l'ordonnancement, c'est-à-dire, la génération et l'envoi des jetons EP, EA et EN);
- le modèle du service d'un flux de messages périodiques Mp_i (génération des messages périodiques du flux et transfert de ces messages au moyen du jeton EP);
- le modèle du service des flux de messages apériodiques temps-réel d'une station k (génération des messages des différents flux et transfert de ces messages au moyen du jeton EA);
- le modèle du service du flux de messages non-temps-réel d'une station k (génération des messages du flux et transfert de ces messages au moyen du jeton EN). Pratiquement, comme notre objectif principal est l'analyse de la gestion du trafic temps-réel, nous représenterons le trafic non-temps-réel de manière globale au sein du réseau, c'est-à-dire sans distinguer dans la génération de ce trafic les différentes stations.

Nous considérons l'algorithme RM pour l'ordonnancement du trafic périodique, ce qui implique donc un fonctionnement régulier pour le passage du jeton EA sur le bus (c.f. V.2.3.3). Les priorités de l'algorithme RM, qui sont des priorités statiques, se représentent aisément avec les RdPTS au moyen de schémas de restrictions de tir d'un ensemble de transitions, restrictions qui sont basées sur des conditions binaires de places (marquées / non marquées). Par contre, le schéma de priorités dynamiques de l'algorithme ED ne peut pas être représenté car nous n'avons pas les moyens de restreindre les tirs d'un ensemble de transitions sur la base de la durée du marquage de places (pour ce faire il faudrait pouvoir associer l'attribut *temps* au jeton lui-même).

On distingue deux types de transitions: des transitions temporisées (le temps de tir d'une transition a une durée non nulle) et des transitions immédiates (le temps de tir a une durée nulle). Dans les modèles, les transitions immédiates sont représentées par des traits et les transitions temporisées sont représentées par des rectangles.

3.2. Le modèle du LAS

Le modèle du LAS, qui est représenté sur la figure 5.5, comprend deux sous-modèles: le sous-modèle "l'ordonnancement" qui représente, d'une part, la vue dans le LAS des différents flux de jetons à générer (parties grisées du haut) et, d'autre part, l'utilisation du médium pour l'envoi des jetons (partie du bas); le sous-modèle "les restrictions RM" qui représente les restrictions sur des tirs de certaines transitions afin de réaliser le schéma de priorités de l'algorithme RM.

1. Sous-modèle "L'ordonnancement"

Génération des jetons EA, EP et EN

Dans les parties grisées, nous avons les modules EA, EP et EN qui représentent respectivement la génération de jetons EA (par le serveur) vers les m stations pour le trafic apériodique temps-réel, la génération de jetons EP vers les p flux de trafic périodique et la génération de jetons EN vers la globalité de l'ensemble des sources de trafic non-temps-réel du réseau.

- Module EA: la séquence $P_{1.serv}$, $t_{1.serv}$, $P_{2.serv}$, $t_{2.serv}$ et $(P_{3.EA.1}...P_{3.EA.m})$ visualise la génération de jetons EA pour les stations $1,...m$ (jetons $EA_1...EA_m$): la place $P_{1.serv}$ représente l'état initial du serveur; la transition $t_{1.serv}$ représente le déphasage initial du serveur par rapport à une référence temporelle globale; la place $P_{2.serv}$ représente le serveur dans la condition "prêt à générer les jetons EA qui devront être envoyés aux m stations pendant la période T" et la transition $t_{2.serv}$ représente à la fois la génération de ces jetons (places $(P_{3.EA.1}...P_{3.EA.m})$) et la périodicité du serveur (retour à la place $P_{2.serv}$). Les places $(P_{3.EA.1}...P_{3.EA.m})$ visualisent des buffers d'attente, c'est-à-dire elles représentent les buffers où les jetons EA, après avoir été produits, attendent qu'un droit d'accès au bus leur soit accordé (rôle du sous-modèle "les restrictions RM").

  Les distributions des transitions temporisées sont: $t_{1.serv}\left(\delta(x-\textit{déphasage du serveur})\right)$, $t_{2.serv}\left(\delta(x-T)\right)$ et $t_{4.EA.1}\left(\delta(x-H_1)\right).....t_{4.EA.m}\left(\delta(x-H_m)\right)$; la notation $\delta(x-a)$ représente une distribution déterministe de durée a.

- Module EP: les séquences ($P_{1.EP.1}$, $t_{1.EP.1}$, $P_{2.EP.1}$, $t_{2.EP.1}$, $P_{3.EP.1}$) ... ($P_{1.EP.p}$, $t_{1.EP.p}$, $P_{2.EP.p}$, $t_{2.EP.p}$, $P_{3.EP.p}$) visualisent la génération de jetons EP relatifs aux flux $Mp_1...Mp_p$ (jetons $EP_1...EP_p$): les places $P_{1.EP.1}...P_{1.EP.p}$ représentent les états initiaux du LAS relativement à chaque flux; les transitions $t_{1.EP.1}...t_{1.EP.p}$ représentent les déphasages initiaux; les places $P_{2.EP.1}...P_{2.EP.p}$ représentent l'état du LAS "prêt à générer par période le nombre de jetons nécessaires pour chaque flux; les transitions $t_{2.EP.1}...t_{2.EP.p}$ représentent à la fois la génération de jetons EP (places $P_{3.EP.1}...P_{3.EP.p}$) et la périodicité de cette génération (retour aux places $P_{2.EP.1}...P_{2.EP.p}$); le poids sur les arcs de sortie ($\#_{EP.1}...\#_{EP.p}$) des transitions $t_{2.EP.1}...t_{2.EP.p}$ représente le nombre de jetons générés par période pour chaque flux. Les places $P_{3.EP.1}...P_{3.EP.p}$ visualisent des buffers d'attente des jetons EP.

  Les distributions des transitions temporisées sont: $t_{1.EP.i}\left(\delta(x-\textit{déphasage du jeton } EP_i)\right)$, $t_{2.EP.1}\left(\delta(x-P_1)\right).....t_{2.EP.p}\left(\delta(x-P_p)\right)$ et $t_{4.EP.1}\left(\delta(x-C_i/v_i)\right).....t_{4.EP.p}\left(\delta(x-C_p/v_p)\right)$.

- Module EN: il comporte seulement la place $P_{3.EN}$ qui visualise le buffer d'attente du jeton EN. On suppose qu'un jeton EN est toujours prêt.

  La transition temporisée $t_{4.EN}$ a une distribution $\delta(x-\textit{durée moyenne d'un message})$.

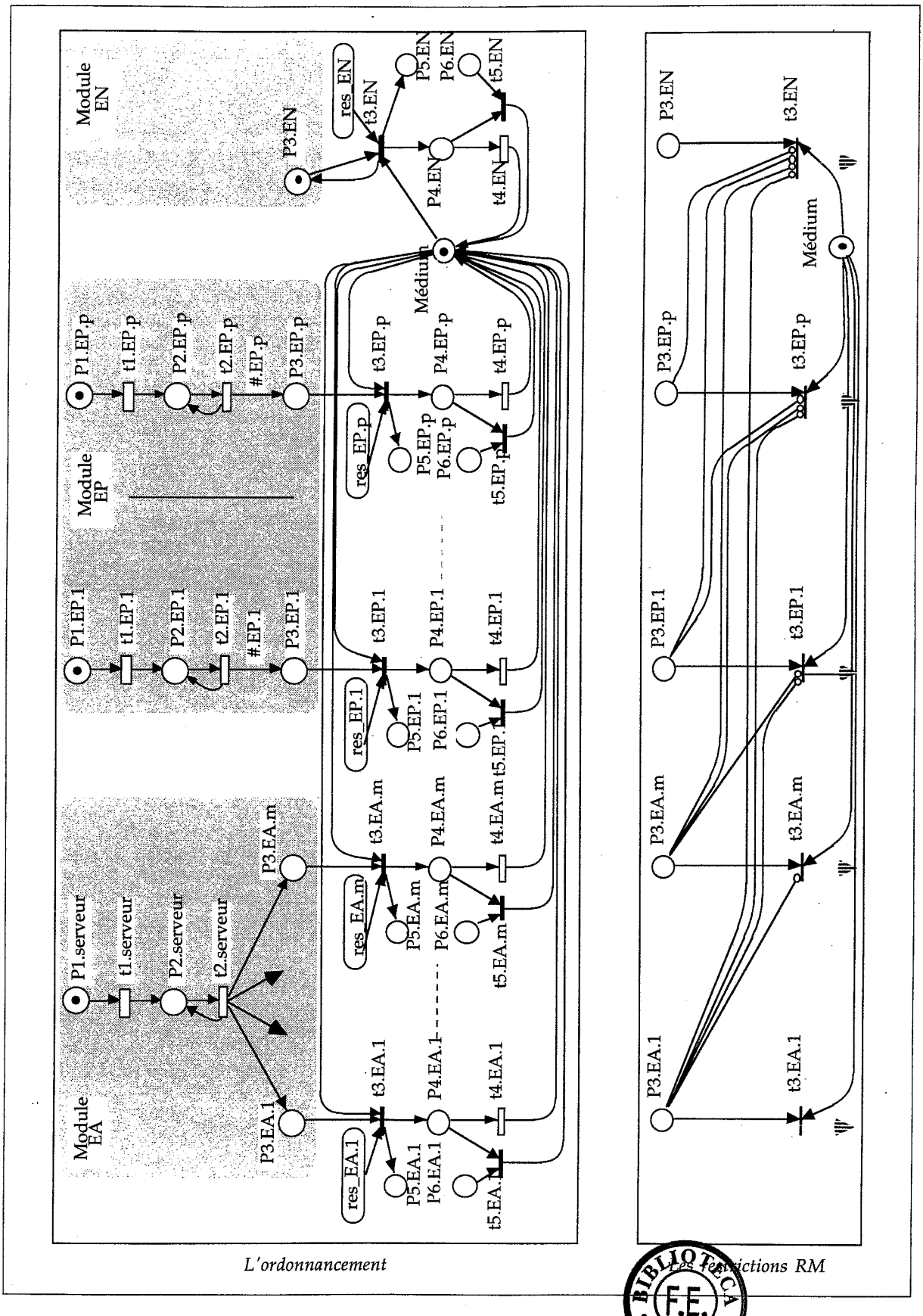

Figure 5.5: Le LAS

BIBLIOTECA F.E. PORTO

Utilisation du médium

On a le même motif qui se reproduit relativement à chaque jeton EA et EP et au jeton EN (place Médium, transitions t_3, places P_4 et P_5, transitions t_4, transitions t_5 et places P_6): la place Médium représente le médium (le médium est libre quand elle est marquée); la transition t_3 visualise l'envoi du jeton (un jeton ne peut être envoyé que si le médium est libre et si les conditions de restriction de l'algorithme RM (ovale avec la notation *res*) autorisent l'ordonnancement du jeton); la place P_5 représente l'envoi du jeton; la place P_4 représente l'attente de la fin du temps dont dispose le jeton (ce temps est représenté par la transition t_4); la transition t_5 représente le retour prématuré du jeton (ce retour est représenté par P_6).

Notons que la condition "place Médium non marquée" (qui empêche l'ordonnancement des flux) visualise l'aspect non-préemptif de l'algorithme.

2. Sous-modèle "Les restrictions RM"

Les restrictions sur les tirs des transitions t_3 sont représentées par des ensembles d'arcs inhibiteurs en entrée de ces transitions t_3. Un ensemble d'arcs inhibiteurs associé à une transition t_3 est tel que cette transition ne peut être tirée que s'il n'y a aucun jeton en attente dans les places buffers d'attente des flux de priorité supérieure. Dans le schéma de la figure 5.5, les priorités des flux sont décroissantes en allant de gauche vers la droite (les flux du serveur ont la priorité la plus élevée; les flux périodiques ont des priorités qui décroissent du flux Mp_1 au flux Mp_p; le flux de trafic non-temps-réel a la priorité la plus faible).

3.3. Le modèle du service d'un flux de messages périodiques Mp_i

Ce modèle est représenté sur la figure 5.6.

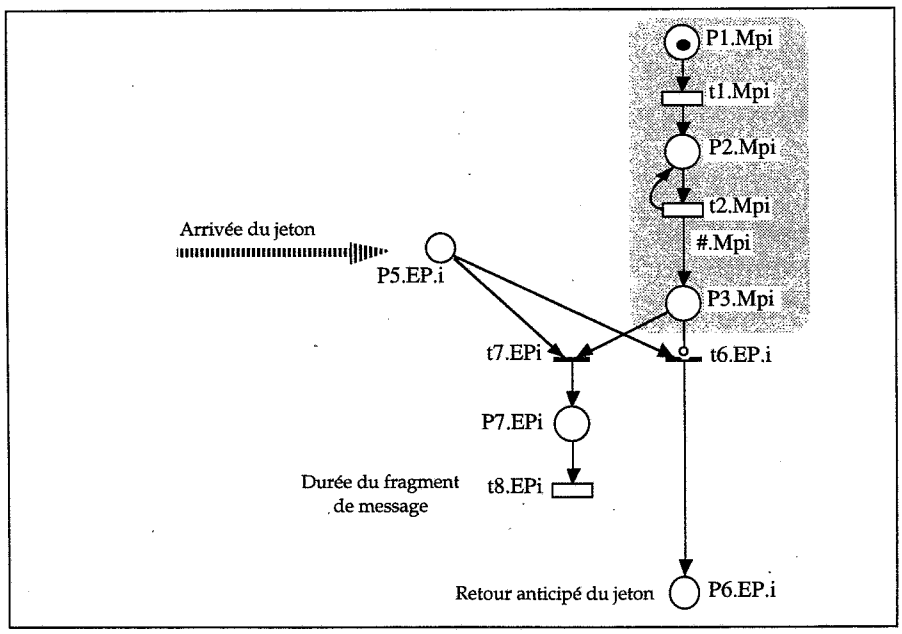

Figure 5.6: Service de flux de messages périodique Mp_i

La séquence $P_{1.Mp_i}$, $t_{1.Mp_i}$, $P_{2.Mp_i}$, $t_{2.Mp_i}$ et $P_{3.Mp_i}$ visualise la production de messages par le flux Mp_i de manière identique à la génération de jetons dans le LAS (la transition $t_{1.Mp_i}$ visualise le déphasage initial, la place $P_{2.Mp_i}$ visualise la condition "prêt à produire" et la transition $t_{2.Mp_i}$ visualise, à la fois, la production d'un nombre v_i de fragments qui sont mis dans

la place buffer $P_{3.Mp_i}$ (v_i est représenté par le poids $\#Mp_i$ de l'arc de sortie de la transition $t_{2.Mp_i}$) et la périodicité (retour à la place $P_{2.Mp_i}$)).

L'ensemble des places $P_{5.EP.i}$ et $P_{7.EP.i}$ et des transitions $t_{7.EP.i}$ et $t_{8.EP.i}$ représentent l'arrivée du jeton EP_i (place $P_{5.EP.i}$) et la prise d'un fragment de la place buffer $P_{3.Mp_i}$ (si elle est marquée): cette prise est représentée par la transition $t_{7.EP.i}$. La place $P_{7.EP.i}$ représente la condition "en cours de transmission" pour le fragment et la transition $t_{8.EP.i}$ représente la durée de la transmission du fragment. Notons que si, lors de l'arrivée du jeton EP_i, il n'y a pas de fragment dans le buffer, on a le renvoi anticipé du jeton EP_i (transition $t_{6.EP.i}$ et place $P_{6.EP.i}$).

Les distributions des transitions temporisées sont: $t_{1.Mp_i}\left(\delta(x - déphasage\ du\ flux\ Mp_i)\right)$ $t_{2.Mp_i}\left(\delta(x-P_i)\right)$ et $t_{8.EP_i}\left(\delta(x-C_i/v_i)\right)$.

3.4. Le modèle du service des flux de messages apériodiques temps-réel d'une station *k*

Ce service est mis en oeuvre au moyen du jeton EA porteur d'une durée. Nous savons que cette durée si n'est pas totalement utilisée par le trafic apériodique temps-réel, le réseau peut être utilisé pour transférer du trafic apériodique non-temps-réel. En conséquence, la modélisation du service des flux de messages apériodiques temps-réel d'une station k doit également représenter la possibilité de transférer du trafic apériodique non-temps-réel (trafic représenté globalement).

Nous considérons que l'on a x flux de messages apériodiques temps-réel dans la station k. Le modèle est représenté sur la figure 5.7.

L'ensemble (place $P_{5.EA.k}$, transition $t_{6.EA.k}$, place $P_{7.EA.k}$ et transition $t_{7.EA.k}$) représente l'arrivée du jeton et le temps de séjour du jeton EA_k jusqu'à l'expiration de la durée H_k. La partie grisée de gauche, représentant le service des flux apériodiques temps-réel, est composée, d'une part, d'un autre ensemble de motifs représentant la production des flux (places $P_{1.k.Ma_j}$, $P_{2.k.Ma_j}$, $P_{3.k.Ma_j}$ et transitions $t_{1.k.Ma_j}$ et $t_{2.k.Ma_j}$) et qui sont identiques à ceux représentés sur les figures précédentes et, d'autre part, des motifs (place $P_{4.k.Ma_j}$, transition $t_{4.k.Ma_j}$, place $P_{5.k.Ma_j}$, transition $t_{5.k.Ma_j}$, place $P_{6.k.Ma_j}$ et transition $t_{6.k.Ma_j}$) qui veulent visualiser la prise d'un fragment d'un flux dans la place $P_{3.k.Ma_j}$, la durée d'émission de ce fragment (transition $t_{5.k.Ma_j}$) et le passage au flux suivant (transition $t_{6.k.Ma_j}$).

Après le tir de la transition $t_{6.k.Ma_{(j+x)}}$, c'est-à-dire quand la place $P_{9.EA.k}$ est marquée, on peut envoyer du trafic non-temps-réel présent dans la place buffer $P_{3.Mn}$ (qui visualise la présence de besoins de trafic non-temps-réel dans la station k; ces besoins découlent de la source globale qui modélise le trafic non-temps-réel dans le réseau (paragraphe suivant).

Les mécanismes relatifs au transfert du trafic non-temps-réel sont représentés de la manière suivante:

- la prise d'un message de trafic non-temps-réel de la place buffer $P_{3.Mn}$ est visualisée par le tir de la transition $t_{8.EA.k}$, ce qui induit la condition de "transmission en cours" (place $P_{8.EA.k}$), la durée de la transmission étant représentée par la transition $t_{9.EA.k}$;
- la fin du trafic non-temps-réel se produit de deux manières: soit (cas 1) lorsque la durée H_k du jeton EA_k expire (tir de la transition $t_{7.EA.k}$, ce qui annule le marquage de la place $P_{7.EA.k}$ et donc supprime l'autorisation de trafic non-temps-réel (tir de la transition $t_{11.EA.k}$)), soit (cas 2) lorsqu'il n'y a plus de messages non-temps-réel en attente (place buffer $P_{3.Mn}$ avec un marquage nul):

- cas 1: la transmission en cours est interrompue et le message est remis dans le buffer (tir de la transition $t_{12.EA.k}$);
- cas 2: on a un renvoi anticipé du jeton EA_k au LAS (tir de la transition $t_{10.EA.k}$).

Les distributions des transitions temporisées sont: $t_{2.k.Ma_j}\left(\delta(x-P_j)\right)$, $t_{5.k.Ma_j}\left(\delta(x-C_j/v_j)\right)$, $t_{1.k.Ma_j}\left(\delta(x-déphasage\ du\ flux\ Ma_j)\right)$, $t_{7.EA.k}\left(\delta(x-H_k)\right)$ et $t_{9.EN}\left(\delta(x-durée\ moy.\ d'un\ message)\right)$.

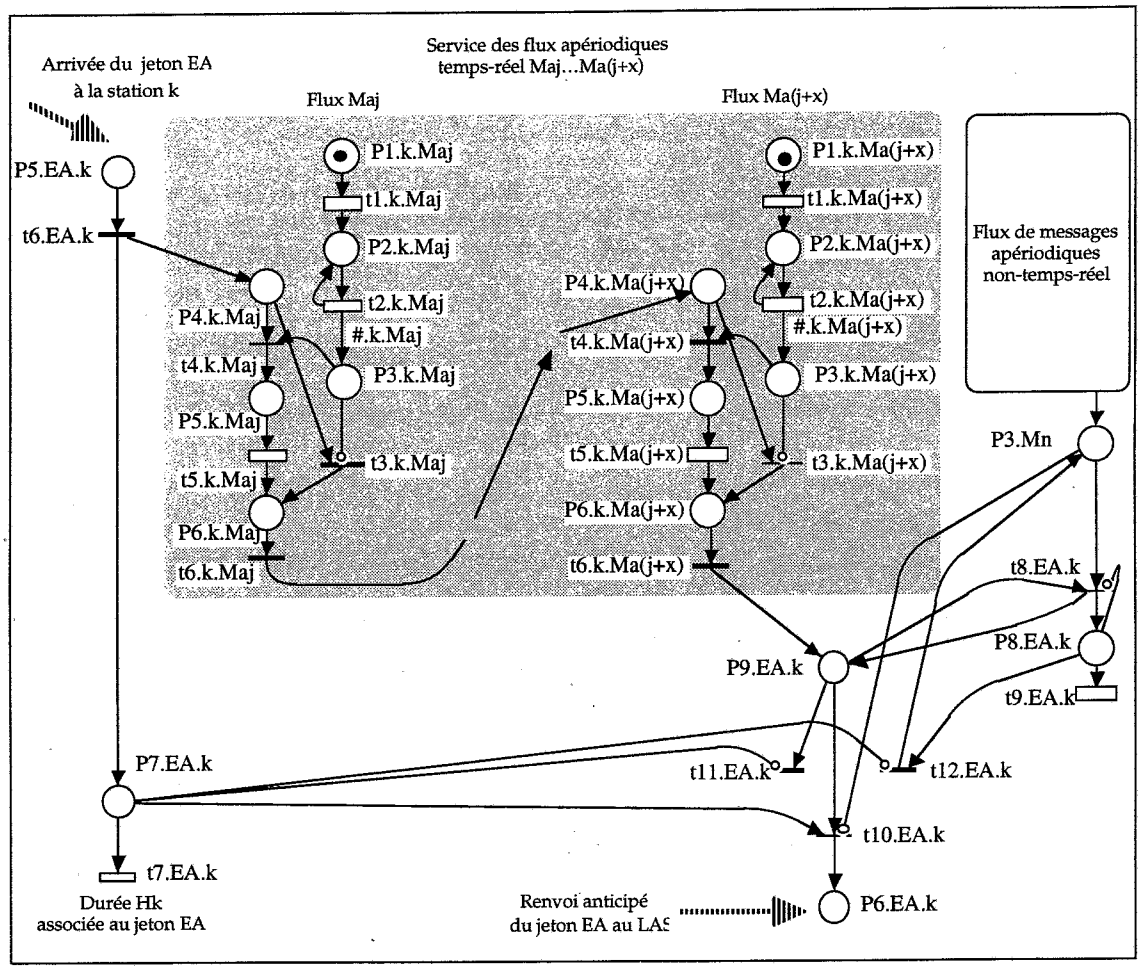

Figure 5.7: Service des flux de messages apériodiques d'une station k

3.5. Le modèle du service des flux de messages apériodiques non-temps-réel

Le modèle est représenté sur la figure 5.8: la séquence $P_{1.Mn}$, $t_{1.Mn}$, $P_{2.Mn}$, $t_{2.Mn}$ et $P_{3.Mn}$ visualise la production des messages du flux global de trafic non-temps-réel (cette séquence est identique à celle présentée pour la production des autres types de trafic dans les figures 5.6 et 5.7); l'ensemble des autres places et transitions ($P_{5.EN}$, $P_{6.EN}$, $P_{7.EN}$, $P_{8.EN}$ et $P_{9.EN}$, et $t_{6.EN}$, $t_{7.EN}$, $t_{8.EN}$, $t_{9.EN}$, $t_{10.EN}$, $t_{11.EN}$ et $t_{12.EN}$) représente des mécanismes strictement identiques à ceux représentés par l'ensemble des places et des transitions ($P_{5.EA.k}$, $P_{6.EA.k}$, $P_{7.EA.k}$, $P_{8.EA.k}$ et $P_{9.EA.k}$ et $t_{6.EA.k}$, $t_{7.EA.k}$, $t_{8.EA.k}$, $t_{9.EA.k}$, $t_{10.EA.k}$, $t_{11.EA.k}$ et $t_{12.EA.k}$) de la figure précédente (figure 5.7).

Les distributions des transitions temporisées sont: $t_{1.Mn}\left(\delta(x-déphasage\ du\ flux\ Mn)\right)$, $t_{2.Mn}$ ($f(x)=\lambda\cdot e^{-\lambda\cdot x}$ avec λ = taux d'arrivées des messages; dans le cas d'un trafic non-temps-

réel, des arrivées du type Poisson sont la meilleure représentation) $t_{7.EN}\left(\delta(x - durée\ associée\ au\ jeton)\right)$, et $t_{9.EN}\left(\delta(x - durée\ moy.\ d'un\ message)\right)$.

3.6. Le modèle global

Le modèle global est obtenu en interconnectant les différents modèles par fusion des places qui portent le même nom sur les différents modèles. Les places fusionnées sont les suivantes: $P_{5.EA.1}$, ... $P_{5.EA.m}$, $P_{6.EA.1}$, ... $P_{6.EA.m}$, $P_{5.EP.1}$, ... $P_{5.EP.m}$, $P_{6.EP.1}$, ... $P_{6.EP.m}$, $P_{5.EN}$, $P_{6.EN}$.

Nous adoptons cette technique d'interconnexion car notre but n'est pas ici d'analyser les problèmes de transmission, mais seulement les problèmes d'ordonnancement.

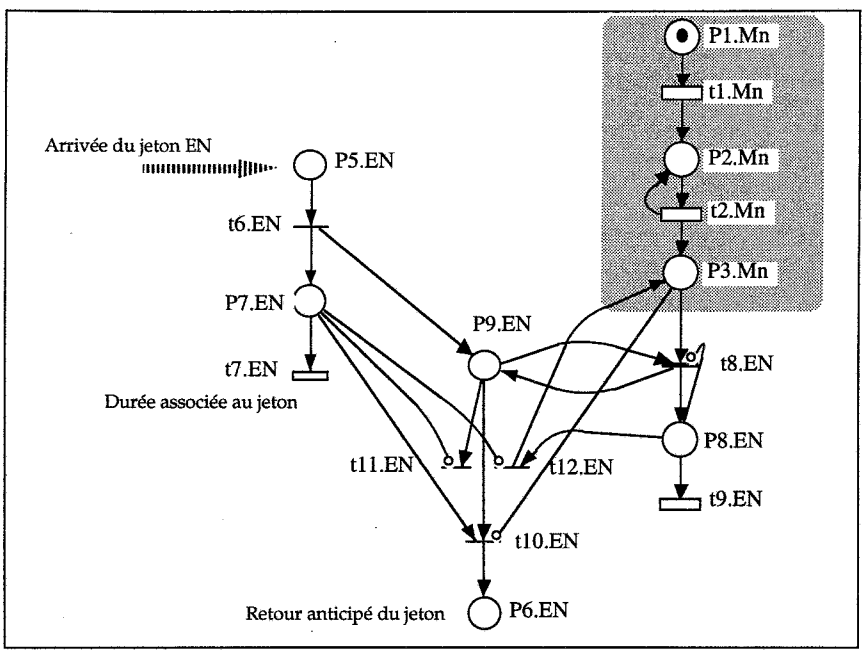

Figure 5.8: Service du flux de messages non-temps-réel

4. Analyse

4.1. Cadre de l'analyse

1. Hypothèses

Nous considérons uniquement des flux temps-réel:

- flux périodiques: 27 flux dont les caractéristiques initiales sont données sur le tableau de la figure 5.9;

Mp_i	C_i	P_i	U(réseau)= $\sum_{i=1}^{27} C_i/P_i$
i = 1 ... 6	2.5	150	36.667%
i = 7 ... 11	2.5	250	
i = 12 ... 17	2.5	300	
i = 18 ... 22	10	500	
i = 23 ... 27	10	750	

Figure 5.9: Flux périodiques

- flux apériodiques: on a *10* flux qui sont répartis sur *3* stations et qui ont les caractéristiques données sur le tableau de la figure 5.10.

Nous fixons la durée maximale d'utilisation du réseau pour un message à une valeur *10* (ce qui définit donc la durée maximale de l'inversion de priorité). En outre, nous considérons des déphasages très faibles (incréments de *0.01*) qui sont associés aux différents flux de manière inversement proportionnelle à leur priorité (ceci permet de faire apparaître les instants critiques non-préemptifs).

	Ma_j	C_j	P_j	U(réseau)= $\sum_{j=1}^{10} C_j/P_j$
station 1	j = 1	2.5	100	23.833 %
	j = 2, 3	5	250	
station 2	j = 4	5	150	
	j = 5, 6	7.5	300	
station 3	j = 7, 8	5	250	
	j = 9, 10	7.5	300	

Figure 5.10: Flux apériodiques

2. Définition du serveur du trafic apériodique

Ses caractéristiques sont définies en appliquant la méthode développé au paragraphe V.1.4:

- la valeur de sa période T est prise égale à *100* (c'est-à-dire, on a bien $T \leq \min_j\{P_j\}$); de plus, compte tenu de cette valeur, le serveur a la plus grande priorité: $T < \min_i\{P_i\} = 150$;
- les durées H_1, H_2 et H_3 allouées par les jetons EA_1, EA_2 et EA_3 (application de la formule $H_k = \sum_{nœud_k}\left(C_j/\left(\lfloor P_j/T \rfloor\right)\right)$) sont donc: $H_1 = 7.5$, $H_2 = 10$ et $H_3 = 10$; notons que ces valeurs assurent le respect de la durée maximale d'inversion de priorité permise et donc il n'est pas nécessaire d'effectuer de la fragmentation.

Compte tenu des valeurs de T et de H_k, $k = 1...m$, la relation $T > \sum_{k=1}^{m} H_k + \tau$ est vérifiée (τ étant négligé dans notre étude) et donc la contrainte du protocole du jeton temporisé et la contrainte du transfert de trafic périodique sont satisfaites.

Notons que l'utilisation maximale du réseau par le serveur est: $\frac{\sum_{k=1}^{m} H_k}{T} = 27.5\%$. L'utilisation du réseau par les flux de messages apériodiques étant de *23.833%* (tableau de la figure 5.10), on a donc une efficacité de *86.67%* ce qui est supérieur au taux minimal d'utilisation maximale de *50%* donné pour le jeton temporisé régulier (c.f. IV.3.1). Ceci ne doit pas étonner car la borne *50%* représente une condition suffisante.

3. Analyses effectuées

Nous considérons deux cas de configurations initiales du réseau en termes de trafic:

- cas 1: on a uniquement le trafic périodique dont les caractéristiques sont données sur le tableau de la figure 5.9; ce trafic satisfaisant le test d'ordonnançabilité (c.f. V.2.5);
- cas 2: on a à la fois le trafic périodique (tableau de la figure 5.9) et du trafic apériodique (tableau de la figure 5.10) géré au moyen du serveur que nous avons défini.

L'objectif de l'analyse est, à partir des configurations initiales du cas 1 et du cas 2, d'introduire des modifications dans les caractéristiques des flux périodiques (augmentation progressive du taux d'utilisation du réseau par ces flux) et de déterminer:

- d'une part, le retard maximal des messages (temps écoulé entre la requête et la fin de l'envoi); ce retard est mesuré au moyen du modèle RdPTS par le temps écoulé entre l'instant où le message d'un flux M_{**} arrive dans une place buffer (places $P_{3.M_{**}}$ dans les modèles des services périodique et apériodique temps-réel) et l'instant où ce message finit d'utiliser le médium (tir des transitions t_8 dans le modèle du service périodique et des transitions t_5 dans le modèle du service apériodique temps-réel);
- d'autre part, la limite des ordonnancements permis, c'est-à-dire lorsque le retard maximal des messages dépasse l'échéance temporelle; cette condition est visualisée par le tir des transitions t_2 des modèles service périodique et apériodique temps-réel alors que les places de sortie de ces transitions (places buffer $P_{3.M_{**}}$) sont encore marquées.

4.2. Résultats

1. Cas 1

Les principaux résultats sont résumés sur le tableau de la figure 5.11. La première ligne rappelle les caractéristiques initiales des flux périodiques et donc les résultats conséquents. Les lignes suivantes donnent les modifications introduites dans les flux périodiques (valeurs des C_i) et les résultats conséquents qui sont pertinents.

C_i					U(réseau)	Test d'ordonnançabilité	Retard max. de message du flux Mp_{27}	Respect des échéances
i = 1, ...6	i = 7, ...11	i = 12, ...17	i = 18, ...22	i = 23, ...27				
2.5	2.5	2.5	10	10	36.67%	positif	132.5	oui
2.5	2.5	7.5	10	10	46.67%	positif	177.5	oui
2.5	5	10	10	10	56.67%	positif	205	oui
2.5	10	10	10	10	66.67%	positif	230	oui
5	10	10	10	10	76.67%	négatif	400	oui
6	10	10	10	10	80.67%	négatif	418	oui
7	10	10	10	10	84.67%	négatif	436	oui
7.5	10	10	10	10	86.67%	négatif	445	oui
8	10	10	10	10	88.67%	négatif	710	oui
8.5	10	10	10	10	90.67%	négatif	725	oui
9	10	10	10	10	92.67%	négatif	740	oui
9.25	10	10	10	10	93.67%	négatif	749	oui
9.5	10	10	10	10	94.67%	négatif	862	non
10	10	10	10	10	96.67%	négatif	880	non

Figure 5.11: Résultat du cas 1

Ces résultats montrent:

- le taux d'utilisation à partir duquel l'ensemble de flux de messages n'est plus ordonnançable (entre *93.6%* et *94.6%*) est supérieur à celui qui découle du test d'ordonnançabilité [LL, 73] (vers les *70%*), ce qui confirme bien le caractère pessimiste de ce test.

- le premier flux a avoir, tout d'abord, des retards d'ordonnancement, et ensuite des échéances non respectées, est le flux qui a la plus faible priorité Mp_{27} (on voit que tant que le retard est inférieur à *750* (période du flux Mp_{27}), le flux reste ordonnançable.

2. Cas 2

Les principaux résultats sont résumés sur le tableau de la figure 5.12 (la première ligne est relative à la spécification initiale des flux périodiques et les autre lignes sont relatives aux flux périodiques modifiés).

C_i					U(réseau)	Test d'ordon nançabilité	Retard max. de message du flux Mp_{27}	Respect des échéances
i = 1, ...6	i = 7, ...11	i = 12, ...17	i = 18, ...22	i = 23, ...27				
2.5	2.5	2.5	10	10	60.5%	positif	230	oui
2.5	2.5	5	10	10	65.5%	positif	245	oui
2.5	2.5	7.5	10	10	70.5%	négatif	272.5	oui
2.5	2.5	10	10	10	75.5%	négatif	287.5	oui
2.5	5	10	10	10	80.5%	négatif	442.5	oui
2.5	6	10	10	10	82.5%	négatif	467.5	oui
2.5	6.5	10	10	10	83.5%	négatif	472.5	oui
2.5	7	10	10	10	84.5%	négatif	477.5	oui
2.5	7.5	10	10	10	85.5%	négatif	482.5	oui
2.5	8	10	10	10	86.5%	négatif	487.5	oui
2.5	8.5	10	10	10	87.5%	négatif	492.5	oui
2.5	9	10	10	10	88.5%	négatif	497.5	oui
2.5	9.5	10	10	10	89.5%	négatif	847.5	non
2.5	10	10	10	10	90.5%	négatif	857.5	non

Par rapport à l'analyse du cas précédent, il faut remarquer que l'on a une réduction de la limite de l'ordonnançabilité (conditions pour lesquelles les échéances ne sont plus respectées) du système (fourchette ($88.5\% - 89.5\%$) par opposition à la fourchette ($93.67\% - 94.67\%$)). Cette réduction s'explique par le fonctionnement du serveur (taux d'utilisation du réseau de 27.5% pour fournir un taux d'utilisation des flux de messages apériodiques de 23.83%, soit une perte d'efficacité de l'ordre de 4% que l'on retrouve dans les limites de l'ordonnançabilité.

5. Conclusion

Les points suivants doivent être soulignés:

- en ce qui concerne la proposition d'architecture pour une sous-couche MAC, nous avons défini une modalité d'ordonnancement conjoint des trois différents types de trafic (périodique, apériodique temps-réel et apériodique non-temps-réel) qui consiste en un ordonnancement en-ligne pour le trafic périodique (algorithme RM non-préemptif ou ED non-préemptif), un ordonnancement basé sur le concept du jeton temporisé pour le trafic apériodique temps-réel et un ordonnancement suivant la politique du meilleur effort pour le trafic apériodique non-temps-réel. Cet ordonnancement conjoint est mis en oeuvre au moyen de trois types de jetons générés par une station de contrôle appelée LAS ("Link Active Scheduler");
- en ce qui concerne la modélisation de l'architecture au moyen des "Réseaux de Petri Temporisés Stochastiques" en considérant l'algorithme RM, nous avons défini un modèle

basé sur quatre modèles de base, le modèle du LAS (qui intègre au moyen de restrictions de tir sur des transitions le système de priorités de l'algorithme RM), le modèle du service des flux de messages périodiques, le modèle du service des flux de messages apériodiques temps-réel et le modèle du service des flux de messages apériodiques non-temps-réel;

- en ce qui concerne l'analyse qui a été faite sur le trafic temps-réel, nous avons évalué l'ordonnancement permis par l'architecture proposée en termes de taux d'utilisation permis et de limites de l'ordonnançabilité (échéances non satisfaites);
- nous voudrions enfin insister sur l'intérêt du modèle "Réseau de Petri Temporisé Stochastique", qui permet d'évaluer l'ordonnançabilité d'une configuration arbitraire de flux de messages dans le cas le plus dur de déphasage, et donc, par voie de conséquence, de vérifier si des configurations qui ne satisfont pas le test analytique d'ordonnançabilité (pessimiste) sont toujours ordonnançables. Cette possibilité constitue, à notre avis, un apport important d'un outil de modélisation et validation au problème de l'étude de l'ordonnançabilité de messages temps-réel.

6. Références

[AB, 90] N. Audsley, A. Burns; Real-Time Systems Scheduling; YCS 134, Departement of Computer Science, University of York, 1990

[B_all, 92] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, F. Wang; On the Comptitiveness of On-Line Real-Time Task Scheduling; Journal of Real-Time Systems, 4, pages 125-144, 1992

[IEC, 94] Digital Data Communications for Measurements and Control - Field Bus for use in Industrial Control Systems; Parts 3 (105 v1.1) & 4 (106 v1.1), IEC SC65C/WG6

[LL, 73] C. Liu and J. Layland, Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, Journal of ACM, vol. 29, n° 1, 1973, pp 46-61

[SSL, 89] B. Sprunt, J. Lehoczky, L. Sha; Aperiodic Task Scheduling for Hard Real-Time Systems; Journal of Real-Time Systems, 1, pages 27-60, 1989

Conclusion

L'objectif de ce travail de thèse est la spécification d'une architecture de communication MAC temps-réel, c'est-à-dire, la définition de mécanismes pour assurer des garanties pour le trafic périodique et le trafic apériodique temps-réel. Il s'est organisé autour de trois axes:

- classification des protocoles MAC temps-réel en mettant en exergue l'aspect ordonnancement qui est un aspect essentiel pour spécifier et concevoir rationnellement des protocoles MAC,
- développement de contributions sur des mécanismes d'ordonnancement et des mécanismes protocolaires, à la fois, en termes conceptuels et en termes de réflexions sur les normes existantes;
- proposition , représentation formelle et évaluation d'une architecture de communication temps-réel.

Nous présentons maintenant les points principaux de ces trois axes et nous donnons ensuite quelques perspectives.

Points principaux des trois axes développés:

En ce qui concerne la classification des protocoles MAC temps-réel:

- nous avons proposé une classification des protocoles MAC en deux grandes classes: la classe 1, où l'ordonnancement est basé sur une assignation de priorités aux flux de messages, et la classe 2, où l'ordonnancement est basé sur une garantie de temps d'accès borné aux stations;
- nous avons développé une analyse des protocoles de MAC temps-réel existants dans les deux classes en termes de deux processus: l'arbitrage d'accès et le contrôle de la durée de transmission; les protocoles CAN, FIP, DQDB et IEEE 802.5 se situent dans la classe 1; les protocoles IEEE 802.3DCR, TTP, "jeton temporisé" et ses dérivés FDDI, IEEE 802.4 et Profibus se situent dans la classe 2;
- nous avons fait une présentation complète du protocole "jeton temporisé" en développant, en particulier, les schémas d'allocation de bande passante qu'il offre, ainsi que les taux d'utilisation permis;
- nous avons montré l'intérêt de la norme ISA SP-50 / IEC-65C qui permet, à la fois, la mise en oeuvre de mécanismes de la classe 1 et de la classe 2 et donc offre de grandes possibilités d'ordonnancement;

- nous avons effectué une réflexion comparative entre les protocoles de classe 1 et les protocoles de classe 2, par rapport aux attributs robustesse, borne minimale de l'utilisation maximale, potentialité de trafic non-temps-réel et temps d'arbitrage d'accès.

En ce qui concerne les contributions sur des mécanismes d'ordonnancement et des mécanismes protocolaires:

Ordonnancement:

- nous avons réalisé des travaux sur les algorithmes non-préemptifs, l'attribut non-préemptif étant une différence fondamentale entre l'ordonnancement des flux de messages et l'ordonnancement des tâches.
  - algorithme ED non-préemptif (deuxième condition d'ordonnançabilité [JSM, 91], qui est la condition spécifique de l'attribut "non-préemptif" et qui a été exprimée dans une échelle temporelle discrète): nous avons, d'une part, développé une extension de cette deuxième condition à une échelle temporelle continue (ce qui permet de considérer des périodes et des durées quelconques pour les tâches, ainsi que des déphasages initiaux infiniment petits), et, d'autre part, proposé une méthode de calcul relativement simple et facilement implantable en-ligne;
  - algorithme RM non-préemptif: ne connaissant pas de travaux dans ce domaine, nous avons proposé une condition d'ordonnançabilité qui exploite le résultat sur les techniques à priorité héritée pour le partage de ressources.
- nous avons conceptualisé un algorithme de changement en ligne de la séquence temporelle de flux périodiques ordonnancés par l'algorithme RM et donné, d'une part, les conditions pour effectuer des changements corrects (ni perte de messages, ni duplication de messages) et, d'autre part, le temps maximum pour rendre effectif un changement de mode (aspect très important dans un contexte temps-réel);

Protocoles

- nous avons défini un protocole appelé "jeton temporisé régulier" qui, par rapport au protocole du "jeton temporisé" présente une régularité de cycle (le temps de cycle est borné par T_{TRT}, alors que la borne est de $2 \cdot T_{TRT}$ dans le "jeton temporisé") ce qui permet une meilleure borne minimale de l'utilisation maximale avec l'algorithme d'allocation de bande passante proportionnel adapté (50% par rapport à 33.3%) et donc présente un intérêt supérieur pour le trafic temps-réel; nous avons également évalué le coût de la fragmentation inhérent à l'algorithme proportionnel adapté.
- nous avons exprimé des conditions pour l'ordonnançabilité, dans le pire cas, du service du trafic apériodique urgent dans le réseau FIP en termes, d'une part, d'une contrainte de protocole afin de garantir une non surcharge du macro-cycle et, d'autre part, d'une contrainte de l'échéance basée sur des notions de retard d'avertissement de l'arbitre du bus et de retard d'ordonnancement. Ces expressions montrent cependant que l'efficacité de ce service est faible (ce qui découle de la prépondérance accordée au trafic périodique dans le réseau FIP);
- nous avons défini deux profils de fonctionnement pour le réseau Profibus, à savoir le profil 1 (on considère que le trafic non-temps-réel n'est pas contraint) et le profil 2 (on considère que le trafic non-temps-réel est contraint), et nous avons défini des conditions d'ordonnançabilité pour ces deux profils; le profil 1 est caractérisé par un temps de cycle faible (ce qui permet des temps d'accès très courts pour les messages les plus prioritaires)

et impose des durées courtes pour les messages; le profil 2 a un temps de cycle plus long mais contraint moins les durées des messages.

En ce qui concerne l'architecture de communication MAC temps-réel proposée, modélisée et évaluée:

- la proposition d'architecture définit une modalité d'ordonnancement conjoint des trois différents types de trafic (périodique, apériodique temps-réel et apériodique non-temps-réel) qui consiste en un ordonnancement en-ligne pour le trafic périodique (algorithme RM non-préemptif ou ED non-préemptif), un ordonnancement basé sur le concept du jeton temporisé pour le trafic apériodique temps-réel et un ordonnancement suivant la politique du meilleur effort pour le trafic apériodique non-temps-réel. Cet ordonnancement conjoint est mis en oeuvre au moyen de trois types de jetons générés par une station de contrôle appelée LAS ("Link Active Scheduler"); il est important de voir que cette architecture correspond bien au souci d'intégration des mécanismes d'ordonnancement et de protocole qui a guidé notre travail de thèse;
- la modélisation de l'architecture, au moyen des "Réseaux de Petri Temporisés Stochastiques" en considérant l'algorithme RM, est effectuée à partir de quatre modèles de base, le modèle du LAS (qui intègre au moyen de restrictions de tir sur des transitions le système de priorités de l'algorithme RM), le modèle du service des flux de messages périodiques, le modèle du service des flux de messages apériodiques temps-réel et le modèle du service des flux de messages apériodiques non-temps-réel; Notons que l'algorithme RM, du fait que les priorités sont statiques, peut être facilement représenté avec le modèle "Réseau de Petri Temporisé Stochastique", ce qui n'est pas le cas pour l'algorithme ED qui utilise des priorités dynamiques (il faudrait pouvoir restreindre le tir de transitions sur la base de la durée du marquage de places).
- l'analyse qui a été faite sur le trafic temps-réel, a permis d'évaluer l'ordonnancement en termes de taux d'utilisation permis et de limites de l'ordonnançabilité (échéances non satisfaites); nous voudrions insister sur l'intérêt du modèle "Réseau de Petri Temporisé Stochastique" qui permet d'évaluer automatiquement l'ordonnançabilité d'une configuration de flux de messages et donc, par voie de conséquence, de vérifier que des configurations qui ne satisfont pas le test d'ordonnançabilité (pessimiste) sont cependant ordonnançables.

Perspectives:

Deux types de travaux pourraient être entrepris:

- l'architecture de communication présentée est une architecture à contrôle centralisé (rôles du LAS et des jetons) et donc il faudrait prendre en compte des hypothèses de pannes au niveau du LAS et des passages de jeton, définir et évaluer des mécanismes de reprise. De plus, le bon fonctionnement de l'ordonnancement dépend de la synchronisation entre le LAS et les différentes stations et donc il faudrait, d'une part, étudier des mécanismes de synchronisation de horloges et, d'autre part, voir comment on peut inclure dans l'ordonnancement des flux effectué par le LAS, un ordonnancement de PDUs spéciaux afin de synchroniser les horloges des stations;
- l'architecture présentée se situe strictement au niveau MAC. C'est une première vue, mais il est évident qu'il faut considérer le fonctionnement des couches supérieures et, en particulier, de la sous-couche LLC. Dans ce contexte il serait important:
  - d'étudier les conséquences de l'asynchronisme entre la sous-couche LLC et la sous-couche MAC (déphasage entre les productions de messages et les envois de jetons);

- de considérer la présence d'attributs temporels spécifiques à la sous-couche LLC, tels que la validité temporelle par exemple, et de voir, d'une part, comment ces attributs doivent être gérés par rapport à l'ordonnancement et, d'autre part, comment on peut assurer des propriétés de cohérence lorsqu'on a plusieurs consommateurs d'une donnée (notion de cohérence spatiale [NFa, 90];
- d'étudier comment on peut implanter des services de groupe.

Annexe

BIBLIOTECA F.E. PORTO

1. Liste de publications effectuées dans le cadre de cette thèse

Conférences Internationales

[CVJ, 94] R. Carmo, F. Vasques and G. Juanole; Real-Time Communication in a DQDB Network; in Proc. of Real-Time Systems Symposium (RTSS'94), pages 249-258, San Juan (Puerto Rico), December 1994; (Rapport LAAS N94164).

[JCV, 94] G. Juanole, R. Carmo and F. Vasques; Connection Oriented Isochronous Services in a DQDB Network: Specification of a Service-Protocol pair and a Bandwidth Allocation Scheme; in Proc. of International Conference on Local and Metropolitan Communication Systems; pages 377-396, Kyoto (Japan), December 1994; (Rapport LAAS N94206).

[VJ, 94] F. Vasques and G. Juanole; Pre-Run-Time Schedulability Analysis in Fieldbus Networks; in Proc. of International Conference on Industrial Electronics, Control and Instrumentation (IECON'94), pages 1200-1204, Bologna (Italy), September 1994; (Rapport LAAS N94233).

[RVV, 94a] P. Rosa, F. Vasques and R. Valette; The Conception of Distributed Applications and the Fieldbus Configuration: A Concurrent Engineering Vision; in Proc. of International Conference on Industrial Electronics, Control and Instrumentation (IECON'94), pages 1135-1140, Bologna (Italy), September 1994; (Rapport LAAS N94232).

[RVV, 94b] P. Rosa, F. Vasques and R. Valette; The Network Transparency Concept in Fieldbus Based Distributed Systems; in Proc. of International Symposium on Industrial Electronics (ISIE'94), pages 247-251, Santiago (Chile), May 1994; (Rapport LAAS N94049).

[VJ, 93] F. Vasques and G. Juanole; Fieldbus MAC Mechanisms for Hard Real Time Data Communication Support; in Proc. of International Conference on Open Bus Systems (OBS'93), pages 225-232, Munich (Germany), November 1993; (Rapport LAAS N93396).

Conférences Nationales

[RVV, 94c] P. Rosa, F. Vasques and R. Valette; A Concepção de Aplicações e o Dimensionamento de Rede Barramento de Campo: Uma Visão Orientada à

Engenharia Concorrente; em Anais do 12° Simpósio Brasileiro de Redes de Computadores (SBRC'94), páginas 34-49, Curitiba (Brasil), Maio 1994.

[ACJV, 93] Y. Atamna, R. Carmo, G. Juanole et F. Vasques; Le Modèle "Réseau de Petri Temporisé Stochastique" (RdPTS) pour l'Analyse Qualitative et/ou Quantitative des Systèmes Temps-Réel; dans Actes des Conférences "Le Salon des Solutions Informatiques Temps-Réel" (RTS'93), pages IV-5 a IV-20, Paris (France), Janvier 1993; (Rapport LAAS N92438).

Rapports de contrat

[JV, 93] G. Juanole et F. Vasques; Objectif de Modélisation et d'Analyse de la Couche de Liaison de Données du Réseau de Terrain Défini par la Norme CEI-65C 92/93, Contrat EDF-GMAT, N M63/7B7619, Janvier 1993; (Rapport LAAS N93023).

[JV, 94] G. Juanole et F. Vasques; Modélisation et analyse de scénarios de trafic périodique; Contrat EDF/DER, Mars 1994 (Rapport LAAS N94107).

[JV, 94] G. Juanole, L. Gallon et F. Vasques; Modélisation et analyse basées sur les réseaux de Petri temporises stochastiques: méthodologie et application a des mécanismes de la couche liaison de données du réseau de terrain défini par la norme CEI 65C 92/39; Contrat EDF/DER, Contrat EDF-GMAT NM63/7B7619, Décembre 1994 (Rapport LAAS N94476).

Thèse de Francisco VASQUES de CARVALHO

Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC de réseaux locaux temps-réel

RESUME :

Cette thèse se situe dans le contexte des réseaux de communication temps-réel et son objectif est de proposer une architecture de communication pour la sous-couche MAC, qui définit des mécanismes pour assurer les contraintes temporelles du trafic temps-réel.

Tout d'abord, une classification des protocoles MAC temps-réel existants, en mettant en exergue l'aspect ordonnancement de flux de messages ou ordonnancement de stations, est effectuée. En particulier, le protocole "jeton temporisé" et des mécanismes de la norme "ISA SP-50 / IEC-65C" sont détaillés.

Ensuite, des contributions sur les mécanismes d'ordonnancement et les mécanismes protocolaires sont développées, à la fois en termes conceptuels et en termes de réflexions sur les normes existantes: proposition d'un algorithme non-préemptif ED avec, en particulier, une extension des conditions classiques d'ordonnançabilité; définition d'un algorithme de changement de mode de fonctionnement pour un système ordonnancé par l'algorithme RM; définition d'un protocole appelé "jeton temporisé régulier" qui améliore les performances temps-réel du protocole "jeton temporisé"; définition des conditions d'ordonnançabilité du trafic apériodique urgent dans le réseau FIP et proposition de deux profils de fonctionnement temps-réel pour le réseau Profibus.

Enfin, nous proposons, modélisons avec le modèle "Réseaux de Petri Temporisés Stochastiques" et évaluons une architecture de communication pour la sous-couche MAC de réseaux locaux temps-réel. Cette architecture met en œuvre, de manière centralisée, un ordonnancement conjoint des trafics périodique et apériodique temps-réel, sur la base d'un algorithme non-préemptif pour le trafic périodique et d'une technique de jeton temporisé pour le trafic apériodique temps-réel. L'analyse permet, d'une part, d'évaluer l'ordonnancement en termes de taux d'utilisation permis et des limites de l'ordonnançabilité et, d'autre part, de montrer tout l'intérêt des modèles "Réseaux de Petri Temporisés Stochastiques" pour représenter et évaluer automatiquement l'ordonnançabilité d'un ensemble de configurations de flux de messages.

MOTS-CLES :

Ordonnancement temps-réel; Réseaux de communication temps-réel; Ordonnancement non-préemptif; Changement de mode de fonctionnement; Jeton temporisé; Réseaux de terrain; Modélisation et analyse; Réseaux de Petri temporisés stochastiques

On the integration of scheduling and communication mechanisms in the MAC sub-layer of real-time local area networks

ABSTRACT

In this thesis, it is proposed an architecture for the MAC sub-layer of a real-time communication network, which integrates mechanisms to guarantee traffic timing constraints.

First, a taxonomy of real-time MAC protocols based on scheduling attributes is presented. In particular, the timed token protocol and some mechanisms of the "ISA SP-50 / IEC-65C" standard are detailed.

After, contributions about scheduling and protocol mechanisms are presented, such as: extension of the non-preemptive ED scheduling algorithm schedulability test; definition of new mode change algorithm for RM scheduled networks; definition of a new timed token protocol, which enhances the traditional timed token protocol real-time performance; definition of schedulability tests for the FIP network sporadic traffic; proposal of two real-time profiles for the Profibus network.

Finally, an architecture for the MAC sub-layer of a real-time communication network is proposed, modelled and evaluated using Stochastic Timed Petri Nets. In this architecture, we propose a centralised scheduling of periodic and sporadic traffic, using a non-preemptive scheduling algorithm to schedule periodic traffic and a timed token based mechanism to schedule sporadic traffic. In this way the traffic timing constraints are ensured. We briefly analyse some traffic scenarios in terms of real-time traffic schedulability bounds. The effectuated analysis allows us to show the interest of the Stochastic Timed Petri Nets model to automatically evaluate the schedulability of generic message stream sets.

KEYWORDS :

Real-time scheduling; Real-time communication networks; Non-preemptive scheduling; Mode change; Timed token; Fieldbus; Modelling and analysis; Stochastic Timed Petri Nets.

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000050899