

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	Examineur
Z.	MAMMERI	Rapporteur
M.	SAMAAN	Examineur

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

*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. SAMAAAN, 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	1
Contexte de l'étude	3
1. Introduction	3
2. Sur les systèmes temps-réel	3
2.1. Présentation générale	3
2.2. Sur les exécutifs temps-réel	4
3. L'ordonnancement temps-réel de tâches	5
3.1. Généralités	5
3.2. Les caractéristiques temporelles des tâches	5
3.3. Les principaux types d'algorithmes	6
4. Sur les réseaux de communication temps-réel	7
4.1. Architecture	7
4.2. Ordonnancement temps-réel de messages	7
4.3. Modélisation de messages au niveau du service MAC	8
5. Sur les Réseaux de Petri Temporisés Stochastiques	9
3.1. Définition formelle	9
3.2. Analyse	10
3.3. L'outil RdPTS	11
Conclusion	11
Références	12
Ordonnancement monoprocesseur de tâches temps-réel: état de l'art et contributions	15
1. Introduction	15
2. Algorithmes préemptifs	16
2.1. Tâches périodiques	16
3. Algorithmes non-préemptifs: l'algorithme ED non-préemptif	21
4. Contributions	23

4.1.	Algorithme ED non-préemptif	23
4.2.	Algorithme RM non-préemptif	27
5.	Conclusion	28
6.	Références	29
Réseaux de communication temps-réel: les principaux protocoles MAC		31
1.	Introduction	31
2.	Classification des protocoles MAC temps-réel	32
2.1.	La problématique de l'ordonnancement de messages temps-réel	32
2.2.	L'arbitrage d'accès et le contrôle de la durée de transmission	33
2.3.	Les techniques d'accès	34
3.	Protocoles de la "classe 1"	35
3.1.	Principales normes	35
3.2.	Le protocole CAN	36
3.3.	Le protocole FIP	37
3.4.	Le réseau DQDB	42
3.5.	Le protocole IEEE 802.5	44
4.	Protocoles de la "classe 2"	46
4.1.	Principales normes	46
4.2.	Le protocole IEEE 802.3 DCR	47
4.3.	Le protocole TTP	48
4.4.	Les protocoles basés sur le jeton temporisé: IEEE 802.4, FDDI et Profibus	49
5.	Un protocole mixte: classe 1&2	55
6.	Évaluation comparative des classes 1 et 2	57
6.1.	Critères d'évaluation	57
7.	Conclusion	58
8.	Références	59
Sur les réseaux de communication temps-réel: contributions		61
1.	Introduction	61
2.	Sur les protocoles de classe 1	62
2.1.	Algorithme de changement de mode de fonctionnement	62
2.2.	Conditions d'ordonnançabilité dans le réseau FIP	70
3.	Sur les protocoles de classe 2	73
3.1.	Le protocole "jeton temporisé régulier"	73
3.2.	Conditions d'ordonnançabilité dans le réseau Profibus	78
5.	Conclusion	84
5.	Références	84
Proposition, modélisation et analyse de mécanismes pour l'ordonnancement de trafic temps-réel dans une sous-couche MAC		87
1.	Introduction	87

2.	Proposition	88
2.1.	Réflexion sur l'ordonnancement conjoint de tous les types de trafic	88
2.2.	Cadre de la proposition	90
2.3.	L'ordonnancement du trafic apériodique temps-réel	91
2.4.	L'ordonnancement du trafic périodique	93
2.5.	L'ordonnancement conjoint	94
2.6.	La syntaxe des jetons proposés	96
3.	Modélisation	97
3.1.	Contexte	97
3.2.	Le modèle du LAS	97
3.3.	Le modèle du service d'un flux de messages périodiques Mp_i	100
3.4.	Le modèle du service des flux de messages apériodiques temps-réel	101
3.5.	Le modèle du service des flux de messages apériodiques	102
3.6.	Le modèle global	102
4.	Analyse	103
4.1.	Cadre de l'analyse	103
4.2.	Résultats	105
5.	Conclusion	106
6.	Références	107
	Conclusion	109
	Annexe	115

Introduction

L'évolution technologique de ces dernières 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 aperiodique) 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 assujéti 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 (monoprocasseur ou multiprocasseur), 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:

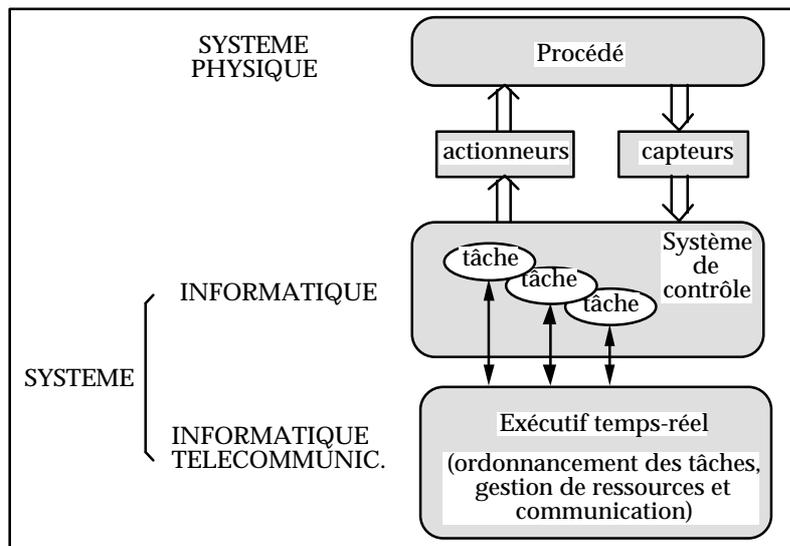


Figure 1.1: Système temps-réel

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 aperiodique). 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, $\{t_1, t_2, \dots, t_i, \dots, t_n\}$, chaque tâche t_i étant caractérisée par un tuple $\{C_i, d_i, P_i, I_i\}$, $1 \leq i \leq n$, où:

- C_i représente la durée maximale d'exécution d'un travail de t_i ;
- d_i est l'échéance de t_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 t_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 t_i (pour des tâches aperiodiques);
- I_i représente la différence de phase entre la première requête d'exécution de la tâche t_i et une référence temporelle commune à l'ensemble de tâches;

Considérons le cas des tâches périodiques: chaque tâche t_i génère une séquence infinie de travaux. La requête du $j^{\text{ème}}$ 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.

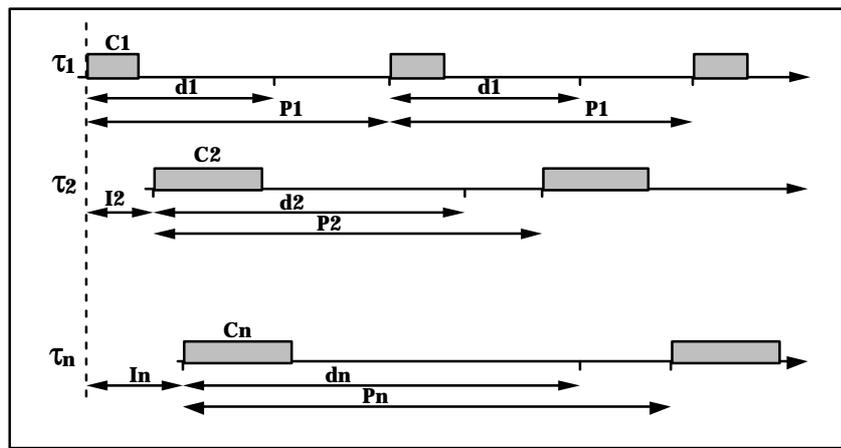


Figure 1.2: Modèle des tâches

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 aperiodique 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'ordonnabilité* 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 diffèrent 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

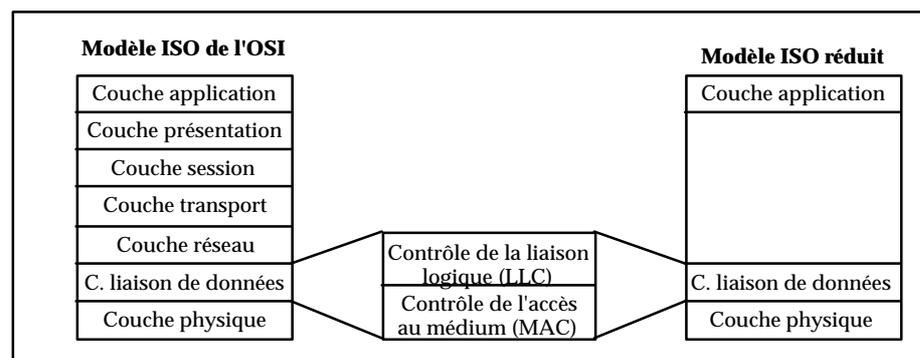


Figure 1.3: Le modèle de référence ISO de l'OSI

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'ordonnement 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'ordonnement temps-réel des messages.

4.2. Ordonnement 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'ordonnement des tâches. Nous ne considérons pas non plus les aspects de précedence 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 $Mp = \{Mp_1, \dots, Mp_i, \dots, Mp_p\}$, comportant p flux de messages périodiques et un sous-ensemble $Ma = \{Ma_1, \dots, Ma_j, \dots, Ma_s\}$, comportant s flux de messages apériodiques 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: $M = \{M_1, \dots, M_i, \dots, M_n\}$; 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: $N = \{N_1, \dots, N_k, \dots, N_m\}$.

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

Il nous paraît 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 $[q_{mi}, q_{Mi}]$. Nous avons: $0 \leq q_{mi} \leq q_{Mi} \leq \infty$; q_{mi} étant la date de tir au plus tôt et q_{Mi} étant la date de tir au plus tard. q_{mi} et q_{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: $\int_{q_{mi}}^{q_{M_i}} f_i(x) dx = 1$.

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 $\langle t_i, p_i, q_i \rangle$ 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 q_i le temps moyen de tir ou temps conditionnel de séjour dans l'état de départ de la transition. Les valeurs p_i et q_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 q_i avec une probabilité p_i sont :

- t_i sensibilisée dans le réseau de Petri sous-jacent ;
- q_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 q_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 q_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 Q (matrice des temps de transitions):

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

$$g \cdot P = g, \text{ avec } \sum g_i = 1 \quad (1.1)$$

- les temps inconditionnels de séjour (h_i) sont obtenus en combinant les matrices P et Q ;
- les probabilités en régime permanent p_i des états sont calculées à partir du vecteur des probabilités à l'équilibre g et des temps inconditionnels de séjour h_i :

$$p_i = \frac{h_i g_i}{\sum_i h_i g_i}$$

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 nécessité 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 Stochastic Timed Petri Nets", in Proc. of Workshop on Petri Nets and Performance Models, Tools Exhibition (PNPM'93), Toulouse, France, 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 réflexion 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 Polytechnique 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

