Le modèle réseaux de Petri temporisés stochastiques: extensions et applications

Laurent Gallon

N° ordre : 2868 - Année 1997

Thèse

Présentée devant **L'Université Paul Sabatier (Sciences)**
en vue de l'obtention du TITRE de
DOCTEUR de l'Université Paul Sabatier de TOULOUSE.
spécialité : **Informatique Industrielle**

par
Laurent GALLON

LE MODÈLE RÉSEAUX DE PETRI TEMPORISÉS STOCHASTIQUES : EXTENSIONS ET APPLICATIONS

Soutenue le 16 Décembre 1997 devant la commission d'examen :

R. CASTANET	*Professeur à l'École Nationale Supérieure d'Électronique et de Radio-électricité de Bordeaux*	*- Rapporteur*
J.P. THOMESSE	*Professeur à l'Institut National de Polytechnique de Lorraine (Nancy)*	*- Rapporteur*
F. COTTET	*Professeur à l'École Nationale Supérieure de Mécanique et d'Aérotechnique (Poitiers)*	*- Examinateur*
M. DIAZ	*Directeur de recherche au CNRS. LAAS-CNRS (Toulouse)*	*- Examinateur*
F. DURAND	*Ingénieur, Aérospatiale (Toulouse)*	*- Examinateur*
M. SAMAAN	*Ingénieur, DER EDF-GDF (Chatou)*	*- Examinateur*
G. JUANOLE	*Professeur à l'Université Paul Sabatier de Toulouse*	*- Directeur de thèse*

Rapport LAAS N° 97556
Thèse préparée au Laboratoire d'Analyse et d'Architecture
des Systèmes du CNRS.

7, avenue du Colonel Roche
31077 Toulouse Cedex 4.

Avant-propos

Les travaux présentés dans ce manuscrit ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique, au sein du groupe Outils Logiciels pour la Communication. Je tiens à exprimer ma gratitude à Messieurs Alain Costes et Jean Claude Laprie, Directeurs successifs du LAAS pendant mon séjour au laboratoire, et à Monsieur Michel Diaz, responsable du groupe OLC, pour leur accueil.

Je tiens à remercier tout particulièrement Monsieur Guy Juanole, qui a accepté d'être mon directeur de thèse. Ses encouragements, son soutien et ses conseils m'ont permis de mener à bien ce travail. Qu'il soit assuré ici de ma profonde gratitude.

Je suis très reconnaissant à Monsieur Michel Diaz, qui a accepté de présider le jury de soutenance. Je remercie Messieurs Richard Castanet et Jean-Pierre Thomesse d'avoir accepté de rapporter sur ce travail, ainsi que Messieurs Francis Cottet, Frédéric Durand et Mazen Samaan pour avoir accepté de participer au jury de soutenance.

Merci à tous ceux qui ont rendus mon séjour au LAAS des plus agréables. Outre les footballeurs, pétanqueurs, et autres Tortues Véloces, mes amis Isabelle, Nathalie, Nabil, Adel, Philippe et Philippe, Slim, Thierry et Francisco me laissent des souvenirs inoubliables. J'ai une pensée toute particulière pour trois personnes qui me sont très chères, Éric Zamaï, Jean Marc Castel et Jean Luc Albacete. Je leur dédie ce mémoire.

Je n'oublie pas non plus les personnes fantastiques qui constituent le personnel technique du laboratoire. Ils sont le cœur du LAAS, et rien ne pourrait aboutir sans eux. Merci à Jean Christophe, Christian M., Christian B., Joëlle, et tous les autres.

Ce travail est aussi un aboutissement pour mes parents, Lucie et Jérôme, pour ma sœur Nathalie et mon frère Nicolas, qui m'ont soutenus dans tous les moments difficiles. A force d'écoute et d'amour, ils m'ont donné les moyens de réaliser mon rêve d'étudiant. Ce mémoire, et tout ce qu'il représente, leur appartient autant qu'à moi.

Enfin, Nathalie est ma force et ma conscience depuis plusieurs années. Sans toi, je ne serais pas arrivé au bout de cette aventure. Notre mariage n'est qu'une preuve infime de ce que tu représente pour moi. Ce manuscrit est une page commune de notre histoire qui se tourne. Bien d'autres suivront ...

Table des matières

Introduction

L'évolution technologique de ces dernières années, en particulier dans le domaine des Réseaux de Communication et de l'Informatique, a induit et multiplié de nouvelles classes de systèmes très sophistiqués (Systèmes de productique, systèmes de contrôle-commande, Systèmes multimédias, ...), qui sont des systèmes distribués (distribution géographique et/ou distribution du contrôle) et temps-réel (les contraintes temporelles sont un élément fondamental du fonctionnement). Ces systèmes repôsent sur des mécanismes nombreux et complexes, dont il est nécessaire, lors de la phase conceptuelle, de garantir le comportement (à la fois en termes qualitatifs et quantitatifs, et ce au niveau des aspects fonctionnels, et de sûreté de fonctionnement). L'obtention de cette garantie passe par l'utilisation de modèles formels. Parmi les modèles formels très utilisés, citons les Réseaux de Petri [Bra83, DeHA89], les files d'attente [Kle75] et les processus markoviens à espace d'états discret [KS60].

Les Réseaux de Petri sont un modèle de haut niveau, qui permet l'expression des principaux mécanismes des systèmes distribués (parallélisme, synchronisation, choix non déterministe, ...). Par contre, ils n'offrent que des possibilités d'analyses qualitatives. Les files d'attente et les processus markoviens à espace d'états discret sont utilisés pour l'évaluation de performances. Les réseaux de files d'attente, qui ont été et sont le modèle le plus utilisé pour l'évaluation de performances, présentent cependant une lacune en terme de pouvoir d'expression (ils ne formalisent pas les aspects de synchronisation). Les processus markoviens à espace d'états discret, qui sont très utilisés dans le domaine de la sûreté de fonctionnement, présentent l'inconvénient de nécessiter l'énumération des états et des transitions entre états (ce qui peut induire des erreurs).
Le plus souvent les études qualitatives et les études quantitatives ont été faites de manière disjointe, et on est en droit de se poser la question de savoir, dans ces cas là, si ce que l'on a évalué est ce que l'on a vérifié.

C'est dans ce contexte que l'idée d'utiliser un modèle de haut niveau comme les Réseaux de Petri, puis d'augmenter leur pouvoir d'expression par l'adjonction du paramètre temps, a germé et a donné lieu à la définition de modèles Réseaux de Petri étendus temporellement, de manière à pouvoir analyser qualitativement et quantitativement les comportements de systèmes avec le même modèle.

L'extension temporelle des Réseaux de Petri a été envisagée sur les transitions [Ram74,

Mol81, MBCC84, DTGN84, Mol85, RJ87, CL93, Ata94, RH80, Zub80, RP84, HV87] et sur les places [Sif77]. Une manière très naturelle d'introduire le temps repose sur l'interprétation du Réseau de Petri dans lequel, étant donné un marquage (ensemble de conditions), un temps doit s'écouler depuis l'instant de sensibilisation de cette transition (marquage qui conditionne la continuité du comportement) avant qu'un événement (tir d'une transition), qui est le résultat de l'activité effectuée dans ce marquage, ne se produise. Cette interprétation amène à associer le temps aux transitions, ce que nous considérons ici (c'est l'extension temporelle qui a donné lieu au plus grand nombre de travaux).

Une critique que l'on peut faire aux différentes extensions temporelles associées aux transitions est qu'elles considèrent soit un intervalle de temps (mais sans distribution temporelle associée à cet intervalle, ce qui ne permet pas d'évaluer des performances de type probabiliste), soit des distributions qui sont le plus souvent exponentielles, ce qui ne permet pas de représenter des contraintes temporelles. Or, en considérant des systèmes distribués temps-réel (systèmes fondamentaux dans le contexte technologique actuel), il faut être à même de pouvoir exprimer des contraintes temporelles et d'évaluer des performances. C'est précisément la raison pour laquelle a été développé voici trois ans, dans le groupe *Outils Logiciels pour la Communication* (OLC) du LAAS-CNRS, le modèle Réseaux de Petri Temporisés Stochastiques [Ata94]. Ce modèle associe aux transitions un intervalle de temps et une densité de probabilité définie sur cet intervalle. Cependant, ce modèle, tel qu'il a été défini, présente des insuffisances en termes de pouvoir d'expression et de pouvoir d'analyse. L'objectif du travail présenté ici est précisément d'introduire les améliorations nécessaires et de montrer leur intérêt pour les systèmes distribués temps-réel et les réseaux locaux temps-réel plus précisément.

Ce mémoire est structuré suivant cinq chapitres :

- le premier chapitre fait une présentation des principaux modèles concernant les extensions temporelles des transitions des Réseaux de Petri, avec en particulier les caractéristiques du modèle Réseaux de Petri Temporisés Stochastiques lors du début de notre travail de thèse;
- le deuxième chapitre concerne les améliorations que nous avons apportées au modèle Réseaux de Petri Temporisés Stochastiques au niveau de son pouvoir d'expression;
- le troisième chapitre est une réflexion sur le pouvoir d'expression du modèle Réseaux de Petri Temporisés Stochastiques (suite à certaines améliorations) à travers une comparaison avec la référence comportementale qui est fournie par l'analyse des Réseaux de Petri étendus avec des intervalles de temps;
- le quatrième chapitre concerne les améliorations que nous avons apportées au modèle Réseaux de Petri Temporisés Stochastiques au niveau de son pouvoir d'analyse;
- le cinquième et dernier chapitre est consacré à la modélisation, l'analyse et la validation du protocole temps critique ARINC 629 CP, au moyen du modèle Réseaux de Petri Temporisés Stochastiques. Cette application a été effectuée dans le cadre d'un contrat entre le groupe OLC du LAAS-CNRS et Aérospatiale-Toulouse.

Chapitre I

Sur des extensions temporelles des Réseaux de Petri

I.1 Introduction

L'objectif de ce chapitre est de présenter des modèles très caractéristiques des extensions temporelles des Réseaux de Petri: les Réseaux de Petri Temporels (intervalle de temps associé aux transitions); les Réseaux de Petri Stochastiques (variable aléatoire temporelle associée aux transitions); les Réseaux de Petri Temporisés Stochastiques, modèle développé dans le groupe OLC du LAAS-CNRS, qui regroupe les extensions temporelles des deux modèles précédents (intervalle de temps et variable aléatoire temporelle sur cet intervalle).

Ce chapitre comprend cinq parties:

- la première partie fait un rappel sur le modèle de base: les Réseaux de Petri Place-Transition;
- la deuxième partie présente le modèle Réseaux de Petri Temporels qui a été défini par Merlin et Farber [MF76];
- la troisième partie présente les éléments fondamentaux des modèles Réseaux de Petri Stochastiques, en insistant sur les notions essentielles que sont les politiques d'exécution;
- la quatrième partie présente trois exemples importants de Réseaux de Petri Stochastiques;
- la cinquième partie présente le modèle Réseaux de Petri Temporisés Stochastiques, tel qu'il était défini au début de cette thèse.

I.2 Rappel sur le modèle Réseaux de Petri Place-Transition (PN)

Un réseau de Petri (RdP) est un 5-uplet $RP =< P, T, Pre, Post, M_0 >$ [DeHA89, Bra83, Mur89] dans lequel :

- $P = (p_1, p_2, \ldots, p_m)$ est un ensemble fini de places;
- $T = (t_1, t_2, \ldots, t_n)$ est un ensemble fini de transitions;
- $Pre : P \times T \rightarrow N$ est l'application incidence avant, qui donne pour chaque transition du réseau le nombre de jetons (marques) nécessaire dans chaque place pour pouvoir tirer cette transition;
- $Post : P \times T \rightarrow N$ est l'application incidence arrière, qui donne pour chaque transition le nombre de jetons créés dans chaque place par le tir de cette transition;

  $M_0 : P \rightarrow N$ est l'application marquage initial, qui donne pour chaque place le nombre initial de jetons.

D'un point de vue graphique, un RdP est un graphe biparti orienté, dont les nœuds sont des places et des transitions, représentées respectivement par des cercles et des traits. Un arc valué relie la place p_i à la transition t_j si $Pre(p_i, t_j) > 0$. Un arc valué relie une transition t_j à une place p_i si $Post(p_i, t_j) > 0$.

Un RdP peut être représenté sous forme matricielle par la matrice $C = [c_{ij}]$, définie par :

$$\begin{aligned} C : P \times T &\rightarrow Z \\ (p_i, t_j) &\rightarrow c_{ij} = Post(p_i, t_j) - Pre(p_i, t_j) \end{aligned}$$

Les règles d'évolution d'un RdP sont les suivantes :

- une transition t est dite « sensibilisée » par un marquage M si et seulement si :

$$\forall p \in P, M(p) \geq Pre(p, t)$$

(on note aussi $M \geq Pre(\bullet, t)$)

- si une transition t est sensibilisée, alors elle peut être « tirée »; son tir provoque un changement de marquage tel que :

$$\forall p \in P, M'(p) = M(p) + C(p, t)$$

(on note aussi $M' = M + C(\bullet, t)$)

Un marquage M' sera dit « immédiatement accessible » à partir du marquage M s'il existe une transition t sensibilisée par M et dont le tir amène à M'. Une séquence de franchissement (ou trajectoire) $\sum$ est une séquence de marquages obtenus en tirant une séquence de transitions : $\sum = \{M_0; t_1 M_1; \cdots; t_i M_i; \cdots; t_n M_n\}$, t_i étant la i-ème transition tirée, et M_i le i-ème marquage atteint. La relation entre le marquage initial M_0 et un

marquage M_i, atteint depuis M_0 par la séquence de franchissement s_i, s'exprime au moyen de « l'équation fondamentale » des réseaux de Petri :

$$M_i = M_0 + C\ \bar{s_i}$$

.

On peut définir une relation d'accessibilité sur l'ensemble des marquages d'un RdP. Le comportement dynamique du RdP est alors défini par l'ensemble de ses marquages accessibles depuis le marquage initial M_0 (noté $A(M_0)$). On appelle « graphe des marquages » (ou « graphe d'accessibilité ») le graphe orienté et étiqueté dont les sommets sont les marquages accessibles du Réseau de Petri, et les arcs représentent les accessibilités immédiates entre ces marquages. Ce graphe est noté $GA(M_0)$. Ses chemins donnent l'ensemble des séquences de franchissement (cet ensemble est noté $S(M_0)$).

Remarque : on peut augmenter le pouvoir d'expression des réseaux de Petri classiques Place-Transition en ajoutant la possibilité d'utilisation d'arcs inhibiteurs, c'est à dire d'arcs ne sensibilisant une transition que si le nombre de jetons contenus dans la place d'entrée considéré est inférieur à une certaine valeur (poids de l'arc inhibiteur). L'ensemble des arcs inhibiteurs d'un réseau de Petri sera noté H. Lors du tir de la transition, aucun jeton n'est enlevé des places d'entrée « testées » par un arc inhibiteur.

I.3 Le modèle Réseaux de Petri Temporels (RdPT)

I.3.1 Définition

Un Réseau de Petri Temporel $RPT =< RP, M_0, I_0 >$ [Men82, Rou85, DB91, BB93] est un triplet où :

- $RP =< P, T, Pre, Post >$ est un réseau de Petri,
- $M_0 : P \rightarrow N$ est le marquage initial,
- $$IO : \quad T \rightarrow (Q^+ \cup 0) \times (Q^+ \cup \infty)$$
  $$t_i \rightarrow IO(t_i) = [\alpha_i^s, \beta_i^s] \text{ avec } 0 \leq \alpha_i^s \leq \beta_i^s$$

IO est la fonction « intervalle de tir statique »; elle associe à chaque transition un intervalle de tir qui correspond à l'intervalle des dates de tir possibles de la transition lors de sa sensibilisation.

D'un point de vue graphique, un RdPT est représenté de manière identique à un RdP, si ce n'est que l'intervalle de tir donné par la fonction IO pour chaque transition est précisé sur cette représentation. La forme matricielle est similaire.

I.3.2 Notion d'état et de classe d'états

[BM82, BM83] défini un état E d'un RdPT par une paire $E = (M, I)$ où :

- M est le marquage courant du RdPT;

- I est la fonction qui assigne à chaque transition sensibilisée un intervalle de tir $[\alpha_i, \beta_i]$ représentant l'intervalle de temps dans lequel la transition doit être tirée. Notons que cet intervalle peut être différent de celui donné par la fonction IO, car la référence temporelle n'est pas l'instant de sensibilisation de la transition, mais l'instant d'arrivée dans l'état E.

A partir d'un état $E = (M, I)$, une transition t_i sensibilisée par M peut être tirée à la date θ_i si les deux conditions suivantes sont vérifiées:

α_i, borne inférieure de l'intervalle de tir assigné à t_i par I, est plus petite que la plus petite des bornes supérieures des intervalles de tir des transitions sensibilisées par M : $\alpha_i \leq \min_E(\beta_k)$
Nous distinguerons dans la suite de ce manuscrit par *Dernier Instant de Tir* la valeur $DIT_E = \min_E(\beta_k)$, ce qui permettra d'exprimer la condition précédente sous la forme $\alpha_i \leq DIT_E$.

- θ_i appartient à l'intervalle $[\alpha_i, DIT_E]$

Si ces deux conditions peuvent être réalisées, on dira que la transition t_i est tirable.

Le tir d'une transition t_i depuis l'état $E = (M, I)$ à la date θ_i, conduit à l'état $E' = (M', I')$ défini par:

- M' tel que $M' = M + C(\bullet, t_i)$;
- I' est telle que:
  - pour toute transition t_j nouvellement sensibilisée (sensibilisée par M', mais pas par $M - Pre(\bullet, t_i)$), $I'(t_j) = IO(t_j)$;
    Le fait de tester si une transition t_j est sensibilisée par les marquages M' et $M - Pre(\bullet, t_i)$ permet de considérer comme nouvellement sensibilisée toute transition qui est désensibilisée et resensibilisée immédiatement par le tir de la transition t_i.
  - pour toute transition t_j qui reste sensibilisée lors du tir de t_i (sensibilisée par M' et par $M - Pre(\bullet, t_i)$), $I'(t_j)$ est définie par:

$$I'(t_j) = [\alpha'_j, \beta'_j] = [max(0, \alpha_j - \theta_i, \beta_j - \theta_i] \text{ avec } I(t_j) = [\alpha_j, \beta_j]$$

    (politique de mémoire temporelle de la dernière sensibilisation).

Cette règle de tir permet de définir une relation d'accessibilité sur les états d'un RdPT [BM83]. Le comportement du RdPT est caractérisé par l'ensemble des états accessibles depuis le marquage initial. On peut donc construire un graphe des états accessibles qui représente le comportement du Réseau de Petri.

La deuxième partie de la règle de calcul de I' permet de calculer ce que l'on appelle le vieillissement des transitions qui restent sensibilisées lors du tir de la transition t_i. Cette règle dépend de la date de tir θ_i. Comme il y a une infinité de valeurs possibles pour θ_i ($\theta_i \in [\alpha_i, DIT_E]$), il y a une infinité d'états E' possibles. Le graphe des états accessibles est donc infini, et ne peut permettre une quelconque analyse du comportement du système.

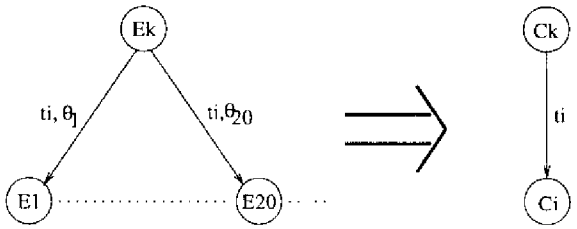


FIG. I.1 – *Passage des états à une classe d'états*

Notons que l'ensemble des états E_i issus du tir d'une transition t_i à une date $\theta_i \in [\alpha_i, DIT_E]$ (un état par date θ_i choisie) autorisent les mêmes comportements futurs du système en termes de marquage. En effet, tous ces états correspondent au même marquage du Réseau de Petri, et ne diffèrent que par l'aspect temporel des transitions qui restent sensibilisées lors du tir de t_i (d'un état à un autre, ces transitions vieillissent d'une durée égale à la date de tir θ_i, et donc puisque θ_i est différente pour chaque état, le vieillissement n'est pas le même pour chaque état). [BM82] propose de représenter cet ensemble d'états par une classe d'états (figure I.1). L'avantage de cette représentation est de transformer des infinités d'états en classes d'états, et donc de rendre possible la construction d'un graphe donnant les comportements du système. Ce graphe s'appelle le graphe des classes d'états.

Nous détaillons dans la suite de ce paragraphe la construction du graphe des classes détats (nous nous baserons sur cette construction dans le chapitre III consacré à l'étude du pouvoir d'expression offert par le modèle Réseaux de Petri Temporisés Stochastiques).

I.3.3 Construction du graphe des classes d'états

Le graphe des classes d'états est basé sur la notion de classe d'états. Une classe d'états C est définie par $C = (M, D)$ avec :

- M est le marquage du Réseau de Petri, commun à tous les états constituants la classe,
- D est le domaine de tir de la classe, défini comme l'union de tous les domaines de tir des états constituant la classe.

Il a été montré dans [BM82] que le domaine D peut être considéré comme l'ensemble des solutions du système d'inéquations :

$$D = \{t | A.t \geq b\}$$

où A est une matrice, b et t sont des vecteurs constitués d'autant de composantes qu'il y a de transitions sensibilisées dans la classe. Chaque composante du vecteur t est une variable $t(i)$ associée à la $i^{ème}$ transition sensibilisée par le marquage M. Dans un but de simplification des notations, nous noterons t_i cette transition.

Une transition t_i est dite tirable dans la classe C si et seulement si :

1. t_i est sensibilisée par le marquage M de la classe C

$$\forall p \in P, M(p) \geq Pre(p, t_i)$$

2. l'intervalle de tir $I(t_i)$ associé à la transition t(i) vérifie le système suivant :

$$\begin{aligned} &A.t \geq b \\ &t(i) \leq t(j), \forall j, j \neq i \end{aligned}$$

La première condition indique que la transition t_i est sensibilisée structurellement par le marquage M. La seconde condition indique que la transition t_i est sensibilisée temporellement (que ses attributs temporels, en comparaison avec ceux des autres transitions sensibilisées, lui permettent d'être tirée) et qu'elle est tirée avant toute autre transition sensibilisée.

Le tir de la transition t_f depuis la classe $C = (M, D)$, dans l'intervalle de tir $[\alpha_f; DIT_C]$ (où DIT_C est la plus petite des bornes supérieures des intervalles de tir des transitions tirables dans C), permet d'atteindre la classe $C' = (M', D')$ définie par :

1. $M' = M + C(\bullet, t_f)$;

2. Le domaine D' est calculé à partir du domaine D en trois phases :

   (a) on ajoute au système $A.t \geq b$ la condition de tir de la transition t_f :

$$\begin{aligned} &A.t \geq b \\ &t(f) \leq t(j), \forall j, j \neq f \end{aligned}$$

   On effectue le changement de variable $t(j) = t''(j) + t(f)$ et on élimine du système la variable $t(f)$, de manière à obtenir un nouveau système de la forme :

$$\begin{aligned} &A''.t'' \geq b'' \\ &0 \leq t'' \end{aligned}$$

   Les nouvelles variables t'' correspondent aux dates de tir des transitions sensibilisées par le marquage M, en prenant comme origine des temps l'instant de tir $t(f)$ de la transition t_f.

   (b) on élimine du système toutes les variables $t(k)$ correspondantes à des transitions qui sont désensibilisées structurellement par le tir de t_f, c'est à dire toutes les transitions qui sont sensibilisées par M et pas par $M - Pre(\bullet, t_f)$;

   (c) on ajoute au système les variables $t(l)$ correspondantes aux transitions nouvellement sensibilisées par le tir de t_f, c'est à dire les transitions qui ne sont pas structurellement sensibilisées par $M - Pre(\bullet, t_f)$ et qui le sont par M'. Ce nouveau système, que l'on peut écrire sous la même forme que celle de D (forme générale) [BM82], défini le domaine D'.

I.3.4 Équivalence des classes d'états

Chaque classe d'états peut être représentée par plusieurs systèmes d'inéquations, chacun de ces systèmes caractérisant le même domaine de tir. Pour mettre en œuvre une vérification et rendre finie la construction du graphe des classes d'états, il est nécessaire de chercher une procédure de détection d'équivalence entre deux classes.

Deux classes seront dites équivalentes si et seulement si :

- elles correspondent au même marquage,
- leurs domaines de tir sont équivalents.

Le fait que chaque domaine puisse être représenté par plusieurs systèmes d'inéquations différents rend l'équivalence de deux domaines de tir D et D' difficile à démontrer en règle générale. Dans [Men82], un algorithme itératif en deux phases, basé sur la recherche de la forme canonique d'un domaine de tir, est proposé pour détecter l'équivalence de deux classes. Son principal inconvénient est qu'il est très coûteux en temps.

I.3.5 Analyses permises

On peut citer deux types d'analyses possibles sur le graphe des classes d'états:

- la vérification que les contraintes temporelles du système sont satisfaites, basée sur l'analyse des séquences d'événements données par le graphe des classes d'états,

  l'évaluation des durées de chemins : les durées minimales et maximales de chemins dans le graphe des classes d'états peuvent être calculés [BB93, Tou97].

I.4 Généralités sur les Réseaux de Petri Stochastiques

I.4.1 Définition d'un Réseau de Petri Stochastique (SPN)

Les Réseaux de Petri Stochastiques sont des réseaux de Petri dans lesquels une variable aléatoire est associée à chaque transition. Cette variable spécifie la durée qui sépare l'instant de sensibilisation de la transition et l'instant de tir de cette transition. Avant de donner la définition formelle d'un Réseau de Petri Stochastique, nous donnons quelques définitions, issues des travaux présentés dans [MBAB+89, FN85, Mol81, MC86, Nat80].

I.4.1.1 Généralités

En s'appuyant sur les définitions données dans le paragraphe I.2, on peut définir la notion de trajectoire temporisée. Une trajectoire temporisée $\sum_{temp}$ est une séquence de franchissement $\sum$ de $S(M_0)$ augmentée d'un ensemble de valeurs réelles croissantes τ_i représentant les dates de tir des différentes transitions :

$$\sum_{temp} = \{(\tau_0, M_0), (t_1, \tau_1, M_1), \cdots, (t_i, \tau_i, M_i), \cdots\}$$

L'intervalle de temps $[\tau_i, \tau_{i+1}]$ représente la durée pendant laquelle le réseau reste dans le marquage M_i.

Une séquence temporisée $\sum_{temp}$ arrêtée à la k-ième date de tir τ_k est appelée « histoire » du réseau de Petri jusqu'à la k-ième date de tir, et est noté $Z(k)$.

Par la suite, nous supposerons que $\tau_0 = 0$.

En posant $Z = Z(i)$, $M = M_i$, on peut définir la « distribution de tir » $D_k(x|M, Z)$ de la transition t_k sensibilisée par M_i, par :

$$D_k(x|M, Z) = Prob\{t_k \text{ tirée}, d_f \leq x|M, Z\}$$

où la variable aléatoire d_f représente le « délai de tir », c'est à dire le temps $\tau_{i+1} - \tau_i$ entre l'instant d'arrivée dans le marquage M et l'instant de tir de la transition t_k.

La probabilité que t_k, dans le marquage M, soit la prochaine transition tirée est :

$$p_k(M, Z) = \int_0^\infty d_x D_k(x|M, Z) = Prob\{t_k \text{ tirée}|M, Z\}$$

et la distribution de la variable aléatoire représentant le temps passé dans le marquage M avant le tir d'une transition t_k :

$$F(x|M, Z) = \sum_{t_k \text{ sens. par } M} D_k(x|M, Z) = Prob\{d_f \leq x|M, Z\}$$

A partir de ces définitions, nous pouvons donner une définition formelle d'un Réseau de Petri Stochastique.

I.4.1.2 Définition

Un Réseau de Petri Stochastique (SPN) est un réseau de Petri PN dans lequel :

1. une variable aléatoire θ_k est associée à chaque transition $t_k \in T$ de PN; cette variable représente le temps nécessaire à l'activité représentée par t_k pour s'achever;
2. chaque variable aléatoire θ_k est caractérisée par une fonction de répartition $G_k(x|M)$ du temps de tir de la transition, quand celle-ci est considérée comme étant la seule transition sensibilisée par le marquage M;
3. une politique d'exécution est définie qui induit les mesures probabilistes $D_k(x|M, Z)$ sur l'ensemble de toutes les séquences temporisées $\sum_{temp}$. Elle précise notamment la prise en compte de l'histoire Z dans ces mesures. Cette politique consiste en deux points : la façon dont, dans chaque marquage, la transition à tirer est choisie, et la manière dont on garde la trace de l'histoire passée du système;
4. la densité de probabilité initiale de l'état initial est définie. On supposera que le système est dans l'état initial correspondant au marquage M_0 à la date $\tau_0 = 0$, avec une probabilité de 1.

Avec cette définition, l'ensemble des séquences possibles du SPN, avec la mesure de probabilité induite par la politique d'exécution, définit un processus stochastique à espace discret et à temps continu. Si l'on suppose que $p_k(x|M, Z) > 0, \forall t_k$ sensibilisée par M,

alors l'espace d'état de ce processus est isomorphe au graphe d'accessibilité $GA(M_0)$ du réseau de Petri sous-jacent.

Comme nous l'avons vu dans le 3. de la définition précédente, la mesure probabiliste associée au graphe d'accessibilité dépend de la politique d'exécution choisie. Cette mesure est primordiale car c'est elle qui nous permet de faire des analyses quantitatives du comportement du système. Il convient donc d'étudier de plus près les principales politiques d'exécution que l'on rencontre dans les Réseaux de Petri Stochastiques, et l'influence de ces politiques sur la mesure probabiliste.

1.4.2 Les politiques d'exécution dans les Réseaux de Petri Stochastiques

La politique d'exécution d'un Réseau de Petri Stochastique est composée de deux éléments :

- la politique de *sélection* de la transition à tirer quand plusieurs transitions sont sensibilisées par un marquage du réseau de Petri,
- la *mémoire temporelle* qui autorise la prise en compte, dès l'instant d'arrivée dans un marquage, du passé du système.

1.4.2.1 Politiques de sélection

On distingue deux types différents de politiques de sélection [MBB+85, MBAB+89, Don97] :

- la *présélection* (« preselection policy ») qui permet de définir, de manière statique, des priorités de tir entre les transitions. On peut définir ces priorités soit au niveau du graphe des marquages (table de switch [MBC86]), soit au niveau du réseau de Petri (poids associés aux transitions);
- la *compétition* (« race policy ») qui autorise le tir de la transition dont la variable aléatoire associée a statistiquement la valeur la plus petite;

La politique de présélection a plusieurs inconvénients : elle est statique (pas de variation avec l'évolution du comportement), lourde à gérer dans le cas des tables de switch, et peut souffrir d'erreurs de spécification.

La politique de compétition, quant à elle, n'est pas déterministe dans le choix de la transition à tirer si deux transitions sensibilisées par le même marquage ont des variables aléatoires qui ont statistiquement la même valeur.

1.4.2.2 Politiques de mémoire temporelle

On distingue trois types de politique de mémoire temporelle [MBB+85, MBAB+89, Don97] :

- la *réinitialisation* (« resampling ») qui ne prend pas en compte le passé du système; chaque variable aléatoire est réinitialisée après le tir d'une transition;

- la *mémoire de la dernière sensibilisation* (« enabling memory ») où seule la dernière période de sensibilisation (qui débute au dernier instant de sensibilisation de la transition) est prise en compte;
- la *mémoire de toutes les sensibilisations* (« age memory ») où toutes les périodes de sensibilisation de la transition sont prises en compte;

La réinitialisation permet d'obtenir des processus stochastiques indépendants du passé, mais qui peuvent dépendre, par exemple, du marquage courant du réseau de Petri.

Les mémoires de sensibilisation font apparaître la notion de temps résiduels. Ces temps correspondent aux temps nécessaires aux processus associés aux transitions pour s'achever. Dans le cas de la mémoire de la dernière sensibilisation, le processus débute à l'instant de sensibilisation de la transition, et est réinitialisé dès que la transition est désensibilisée. Dans le cas de la mémoire de toutes les sensibilisations, ce processus débute lors de la première sensibilisation de la transition, et n'est pas réinitialisé jusqu'au tir de la transition.

La mémoire de la dernière sensibilisation modélise naturellement des processus comme les temporisations (« time out ») qui sont réinitialisées dès qu'elles ne sont plus actives. La mémoire de toutes les sensibilisations modélise naturellement les mécanismes de préemption, qui permettent de reprendre un travail dans l'état où on l'avait laissé.

I.4.2.3 Politiques d'exécution et processus stochastique associé au Réseau de Petri Stochastique

Les différentes politiques d'exécution que l'on peut associer aux transitions d'un Réseau de Petri Stochastique permettent de préciser la mesure probabiliste donnant les temps de séjour dans les différents états du système. A partir de ces mesures, le processus stochastique associé au réseau de Petri peut être défini.

Nous nous proposons dans ce paragraphe de préciser ce processus stochastique pour chaque type de politique d'exécution, dans le cas où toutes les transitions suivent la même politique d'exécution. On supposera que pour chaque transition, la distribution $G_k(x|M)$ peut être déterminée, quel que soit le marquage M de l'ensemble des marquages accessibles. Cette distribution représente la distribution du temps nécessaire à l'achèvement de l'activité associée à t_k, si on suppose que t_k est la seule transition sensibilisée par M.

Politique de Présélection

Dans ce cas, on peut écrire:

$$D_k(x|M,Z) = Prob\{t_k \text{ tirée}, d_f \leq x|M,Z\} = p_k(M,Z).D'_k(x|M,Z)$$

où $D'_k(x|M,Z)$ représente la distribution du délai de tir d_f sachant que t_k est la transition tirée. Les probabilités $p_k(M,Z)$ sont telles que

$$\sum_{t_j \text{ sens. par } M} p_j(M,Z) = 1$$

Notons que dans ce type de politique, le choix de la transition prochainement tirée est fait dès l'entrée dans le marquage M. Le temps de séjour dans ce marquage, issu du calcul d'un

délai aléatoire à partir de la distribution $D'_k(x|M,Z)$, ne dépend donc pas de la probabilité de tir de la transition.

Dans le cadre de la présélection, la réinitialisation est la seule politique de mémoire temporelle qui ait un sens. En effet, dès l'arrivée dans un marquage M, la transition prochainement tirée est choisie. Puis un temps correspondant au temps nécessaire à l'achèvement de l'activité associée à cette transition s'écoule, avant le changement de marquage. Dans le cas où plusieurs transitions sont en concurrence, il est clair qu'une politique de mémoire de sensibilisation peut donner lieu à des incohérences. Par exemple, supposons que deux transitions t_1 et t_2 soient en concurrence dans le marquage M, et que le temps nécessaire à l'achèvement de l'activité associée à t_1 (par exemple 10 secondes) soit supérieur au temps nécessaire à l'achèvement de l'activité associée à la transition t_2 (par exemple 5 secondes). Si t_1 est la transition choisie pour être tirée, une incohérence apparaît : l'activité associée à t_2 s'achève forcément avant celle associée à t_1 (puisqu'elle ne dure que 5 secondes, alors que l'activité associée à t_1 dure 10 secondes), mais t_2 ne pourra être tirée qu'après t_1 (c'est à dire 5 secondes après l'achèvement de l'activité qui lui est associée). On voit donc que dans ce cas, le réseau de Petri ne peut modéliser de manière satisfaisante la concurrence entre ces deux transitions.

Aussi, seule la réinitialisation peut être associée à une politique de présélection (P-R).

Le sens que l'on peut donner à la politique P-R est le suivant : dès l'arrivée du système dans un état, une activité est choisie parmi les activités possibles. Le travail correspondant à cette dernière est alors accompli en totalité, et permet de passer à l'état suivant. Les autres activités ne peuvent débuter que dans ce nouvel état.

On a alors :

$$D'_k(x|M,Z) = D'_k(x|M) = G_k(x|M)$$

On défini par α_{ij} la probabilité de passage immédiat du marquage M_i au marquage M_j :

$$\alpha_{ij} = \begin{cases} p_k(M_i) \text{ si } M_i \xrightarrow{t_k} M_j \\ 0 \text{ sinon} \end{cases}$$

La matrice $Q(x)$ est définie par les valeurs $q_{ij}(x) = \alpha_{ij}.G_k(x|M_i)$, représentant la probabilité que l'événement $M_i \xrightarrow{t_k} M_j$ arrive avant la date x.

Le couple $(Q(x), M_0)$ défini alors un processus semi-markovien sous-jacent à l'ensemble des marquages accessibles du réseau de Petri. Les solutions de ce processus peuvent être obtenus par les techniques classiques.

Politique de compétition

Quand le réseau de Petri arrive dans le marquage M, une valeur θ_k est déterminée pour chaque transition t_k sensibilisée par ce marquage, à partir de la distribution

$$\Phi(x_1, x_2, \cdots | M, Z) = Prob\{\theta_1 \leq x_1, \theta_2 \leq x_2, \cdots | M, Z\}$$

La plus petite de ces valeurs détermine à la fois la transition qui sera tirée et le temps de séjour dans le marquage M.

Si on suppose que les θ_k sont indépendants, alors leur calcul se fait par l'intermédiaire des distributions marginales

$$\Phi_k(x|M,Z) = Prob\{\theta_k \leq x|M,Z\}$$

On peut alors écrire :

$$D_k(x|M,Z) = \int_0^x \Big(\prod_{\substack{j \neq k \\ t_j \text{ sens. par } M}} [1 - \Phi_j(u|M,Z)] \Big) d_u\Phi_k(u|M,Z)$$

$$p_k(M,Z) = \int_0^\infty \Big(\prod_{\substack{j \neq k \\ t_j \text{ sens. par } M}} [1 - \Phi_j(x|M,Z)] \Big) d_x\Phi_k(x|M,Z)$$

$$F(x|M,Z) = 1 - \prod_{t_j \text{ sens. par } M} [1 - \Phi_j(x|M,Z)]$$

La spécification des distributions marginales $\Phi_k(x|M,Z)$ associées à chaque transition permet de définir parfaitement le modèle Réseau de Petri Stochastique. Cette spécification dépend de la mémoire temporelle utilisée dans la politique d'exécution.

Politique de compétition avec réinitialisation (R-R)

Dans ce cas, les distributions marginales $\Phi_k(x|M,Z)$ s'écrivent :

$$\Phi_k(x|M,Z) = G_k(x|M)$$

On peut définir la matrice des probabilités de transition immédiate $Q(x) = [q_{ij}(x)]$ telle que :

$$q_{ij}(x) = \begin{cases} D_k(x|M) \text{ si } M_i \xrightarrow{t_k} M_j \\ 0 \text{ sinon} \end{cases}$$

Le couple $(Q(x), M_0)$ défini alors un processus semi-markovien sous-jacent à un sous-ensemble des marquages accessibles du réseau de Petri. Les solutions de ce processus peuvent être obtenus par les techniques classiques.

Politique de compétition avec mémoire de sensibilisation (R-E, R-A)

Dans le cas des mémoires de sensibilisation, les distributions marginales $\Phi_k(x|M,Z)$ sont calculées à partir des distributions $G_k(x|M)$.

Dans ce but, on défini pour chaque transition t_k sensibilisée par le marquage M une variable a_k qui donne la quantité de travail déjà accompli par l'activité associée à la transition t_k à l'instant d'arrivée dans le marquage M. Cette variable permet donc de mémoriser le passé de la transition.

La différence entre la mémoire de la dernière sensibilisation (R-E) et la mémoire de toutes les sensibilisations (R-A) se situe dans la mise à jour de la variable a_k lors d'un

changement de marquage $M \rightarrow M'$ du réseau. La politique R-E remet à zéro la variable a_k dès que la transition t_k est désensibilisée. La politique R-A mémorise la valeur de a_k quand la transition t_k est désensibilisée, et repart de cette valeur quand t_k est resensibilisée. Dans les deux cas, lorsque t_k est tirée, la variable a_k est remise à zéro.

En pratique, la variable a_k mesure le temps de sensibilisation de la transition t_k.

Le calcul des distributions $\Phi_k(x|M,Z)$ est effectué à partir de la définition de ces variables a_k. Si on suppose que les distributions $G_k(x|M,Z)$ ne dépendent pas du marquage ($G_k(x|M,Z) = G_k(x|Z)$), alors la distribution marginale de la transition t_k est calculée comme suit :

$$\Phi_k(x|M,Z) = \frac{G_k(x+a_k) - G_k(x)}{1 - G_k(a_k)}$$

L'utilisation des variables a_k permet de caractériser le processus stochastique sous-jacent au réseau comme étant markovien, avec une partie de l'espace d'état discrète (les états du graphe d'état associé au réseau de Petri) et une partie continue (le produit cartésien des domaines définis par les variables a_k).

Le calcul des solutions de ce processus ne peut être effectué si les distributions $G_k(x|Z)$ sont de forme générale. Aussi, plusieurs études ont été faites pour des formes particulières de ces distributions, comme les formes exponentielle, discrète, uniforme, ... Ces différentes études ont donné lieu à la définition de différents modèles Réseaux de Petri Stochastiques. Nous nous proposons de préciser dans la suite de ce chapitre trois de ces modèles: le modèle GSPN, le modèle DSPN et le modèle RdPTS.

I.5 Quelques modèles Réseaux de Petri Stochastiques importants

I.5.1 Le modèle Réseaux de Petri Stochastiques (SPN)

I.5.1.1 Définition

Un Réseau de Petri Stochastique [Nat80, Mol81, Flo85] est un 6-uplet

$$SPN = < P, T, Pre, Post, H, M_0, \lambda$$

avec :

- $< P, T, Pre, Post, H, M_0 >$ est un Réseau de Petri avec arcs inhibiteurs, tel que nous l'avons décrit dans le paragraphe I.2,
- λ est la fonction qui associe à chaque transition un taux de transition :

$$\begin{array}{rl} \lambda : & P \rightarrow Z^+ \\ & t_k \rightarrow \lambda_k \end{array}$$

La politique de choix de la transition à tirer est la politique de compétition. Les fonctions de répartition des variables aléatoires associées aux transitions étant toutes exponentielles, on a :

- le processus de marquage est un processus markovien homogène. Les méthodes classiques de calcul markovien sont applicables;

- du fait de la propriété d'absence de mémoire temporelle de la loi exponentielle, il y a perte, à chaque instant, de la mémoire des travaux cumulés. Cela implique que lors du tir d'une transition, les attributs temporels des transitions en parallèle ne sont pas modifiés.

I.5.1.2 Analyses permises

Outre des analyses qualitatives effectuées sur le graphe des marquages, on peut évaluer les mêmes types de performances qu'avec les chaînes de Markov (le graphe des marquages étant isomorphe à une chaîne de Markov).

Si on a un graphe cyclique, on peut évaluer, en régime permanent :

- les probabilités des états,
- le temps de séjour dans un marquage M_i $T_i = \frac{1}{\sum_{t_i \in T_i} \lambda_i}$, où T_i est l'ensemble des transitions tirables dans M_i;
- la probabilité de tirer une transition t_k dans le marquage M_i $p_k = \frac{\lambda_k}{\sum_{t_i \in T_i} \lambda_i}$.

Si on a un graphe acyclique, on peut évaluer les temps moyens d'absorption.

I.5.2 Le modèle Réseaux de Petri Stochastiques Généralisés (GSPN)

Dans sa dernière définition [MBC86], un GSPN est un 8-uplet

$$GSPN = < P, T, \Pi, Pre, Post, H, W, M_0 >$$

avec :

- $< P, T, Pre, Post, H, M_0 >$ est un Réseau de Petri avec arcs inhibiteurs et sans confusion, tel que nous l'avons décrit dans le paragraphe I.2;
- Π est la fonction priorité qui assigne à chaque transition un niveau de priorité (nombre naturel positif) :

$$\begin{aligned} \Pi : \quad & T \rightarrow \mathrm{IN}^+ \\ & t_k \rightarrow \Pi(t_k) \end{aligned}$$

- W est la fonction permettant de définir le processus stochastique associé au modèle. W associe à chaque transition t_k un nombre réel positif w_k. Si la transition est une transition temporisée, ce nombre est le taux de la distribution exponentielle du délai de tir de t_k. Si la transition est immédiate, w_k correspond au poids associé à t_k.

$$\begin{aligned} W : \quad & T \rightarrow Z^+ \\ & t_k \rightarrow w_k \end{aligned}$$

Dans un marquage où plusieurs transitions sont sensibilisées, les transitions immédiates sont obligatoirement tirées avant les transitions temporisées. On définit ainsi deux types de marquage : les marquages « tangibles », dans lesquels seules des transitions temporisées

sont sensibilisées, et les marquages « évanescents » dans lesquels au moins une transition immédiate est sensibilisée.

Dans les marquages tangibles, la politique de sélection est la politique de compétition. La politique de mémoire temporelle n'est pas précisée puisque la distribution exponentielle n'a pas de mémoire temporelle.

La probabilité de tir d'une transition t_k à partir d'un marquage M est :

$$p_k(M) = \frac{w_k}{\sum_{t_l} w_l}$$

où t_l appartient à l'ensemble des transition sensibilisées dans le marquage M.

La manière dont on choisit la transition à tirer à partir d'un marquage évanescent M est complexe, et n'est pas rappelée ici. Elle est basée sur une politique de présélection. Toutefois, il a été démontré que quelle que soit la séquence de transitions immédiates tirées à partir d'un marquage M où plusieurs transitions sont sensibilisées, on atteint le même marquage tangible. Aussi, seul l'ensemble des marquages tangibles est nécessaire pour caractériser le processus stochastique sous-jacent (ces états constituent les points de régénération d'une chaîne immergée). L'obtention de la chaîne immergée associée au GSPN passe donc par une première phase d'élimination des marquages évanescents du graphe des marquages accessibles. Plusieurs algorithmes ont été proposés pour éliminer ces marquages ([Bla89, BCFMR87, Chi85]).

A partir de la chaîne immergée associée, les indices de performance classiques peuvent être obtenus (cf paragraphe I.5.1).

I.5.3 Le modèle Réseaux de Petri Stochastiques et Déterministes (DSPN)

Le modèle Réseaux de Petri Stochastiques et Déterministes [MC86, CL93, Cia93] peut être vu comme une extension du modèle GSPN. Dans un DSPN, on trouve trois types de transitions :

- les transitions immédiates, qui sont prioritaires sur les autres types de transitions,
- les transitions à distribution exponentielle,
- les transitions discrètes, de durée τ, dont la distribution est une impulsion de Dirac $\delta(x - \tau)$.

La politique de sélection de la transition à tirer dans un marquage est la politique de compétition pour les transitions non immédiates, et de présélection pour les transitions immédiates (comme pour le modèle GSPN).

Les transitions immédiates et à distribution exponentielle n'ont pas de mémoire temporelle. Il n'est donc pas nécessaire de préciser la politique de mémoire temporelle pour ces types de transitions. Par contre, les transitions discrètes ont cette propriété.

Dans le cas d'une transition discrète t_d dans le marquage M_i, lors du tir d'une autre transition t_i qui fait passer au marquage M_j, la distribution de t_d va être modifiée. Cette modification dépend de la politique de mémoire temporelle choisie :

- réinitialisation : la distribution après le tir est $\delta(x - \tau_d)$;
- mémoire de toutes les sensibilisations : la nouvelle distribution est $\delta(x - (\tau_d - \theta_i))$, où θ_i est la date de tir de la transition t_i. Notons que si τ_d dépend du marquage courant M_j, la différence $\tau_d - \theta_i$ est multipliée par le facteur $\frac{\tau_d(M_j)}{\tau_d(M_i)}$;
- mémoire de la dernière sensibilisation : si la transition reste sensibilisée, la modification de la distribution est identique à celle effectuée dans le cas de la mémoire de toutes les sensibilisations, sinon on utilise la modification utilisée pour la réinitialisation.

Il a été démontré dans [CKT93] et [CGL93] que le processus de marquage sous-jacent au DSPN constitue un processus stochastique markovien régénératif (encore appelé processus semi-régénératif).

I.6 Le modèle Réseaux de Petri Temporisés Stochastiques (RdPTS)

Le modèle Réseaux de Petri Temporisés Stochastiques est le modèle que nous avons utilisé. Nous allons donc, dans ce paragraphe, détailler sa définition telle qu'elle était au début de notre thèse.

I.6.1 Définition

Un RdPTS [Ata94] est un 8-uplet $< P, T, Pre, Post, H, IO, FO, M_0 >$ où :

- $< P, T, Pre, Post, H, M_0 >$ est un Réseau de Petri avec arcs inhibiteurs, tel que nous l'avons décrit dans le paragraphe I.2; notons qu'il n'y a pas de mécanisme de priorité explicite dans ce modèle;

  IO est la fonction intervalle de tir initial. A chaque transition $t_i \in T$ est associé un intervalle de tir $IO(t_i) = [\theta_{m_i}, \theta_{M_i}]$. Nous avons $0 \leq \theta_{m_i} \leq \theta_{M_i}$ avec $\theta_{M_i} \leq \infty$; θ_{m_i} est la date de tir au plus tôt (*Earliest Firing Time*: EFT) et θ_{M_i} la date de tir au plus tard (*Lastest Firing Time*: LFT). θ_{m_i} et θ_{M_i} se réfèrent à l'instant de sensibilisation de la transition t_i;

  FO est la fonction densité de probabilité de tir initiale. A chaque transitions $t_i \in T$ est associée une densité de probabilité initiale $FO(t_i) = f_i(x)$ définie sur son intervalle de tir initial $IO(t_i)$. Cette densité de probabilité est telle que $\int_{\theta_{m_i}}^{\theta_{M_i}} f_i(x)dx = 1$

Les densités de probabilité associées aux transitions peuvent être de trois types :

- *densité continue*: on utilise la densité exponentielle ($f_i(x) = \lambda_i e^{-\lambda_i x}$) et la densité uniforme ($f_i(x) = \frac{1}{\theta_{M_i} - \theta_{m_i}}$); la densité exponentielle est la densité la plus utilisée en évaluation de performance; la densité uniforme permet de modéliser des phénomènes comme les phénomènes de gigue dans les réseaux de communication;

- *densité discrète*: on utilise ici des densités de type impulsion de dirac ($f_i(x) = \delta(x-\theta_i)$); le cas particulier des transitions immédiates correspond à une distribution discrète avec $\theta_i = 0$; une densité discrète permet de définir la date d'occurrence d'un événement, comme par exemple la fin de l'écoulement d'une temporisation;
- *densité mixte*: on a une partie de la densité qui est uniforme, et l'autre partie discrète ($f_i(x) = f_{i_c}(x) + f_{i_d}(x) = \frac{1-\sum_1^n K_i}{\theta_{M_i}-\theta_{m_i}} + \sum_1^n K_i\delta(x-\theta_i)$); ce type de densité permet de privilégier certaines dates d'un intervalle temporel pour l'occurrence de l'événement associé à la transition;

Par convention, les transitions immédiates seront représentées du point de vue graphique par un trait fin. Les autres types de transitions, que nous qualifierons de transitions temporisées, seront représentées par un rectangle.

I.6.2 Évaluation d'un Réseau de Petri Temporisé Stochastique

Il y a évidemment priorité des transitions immédiates sur les autres transitions (à cause des attributs temporels, et pas d'une priorité explicite comme c'est le cas, par exemple, dans le modèle GSPN). La politique d'exécution est la politique de compétition avec mémoire de la dernière sensibilisation, à l'exception du cas où on a des transitions avec des distributions déterministes identiques ($= 0$ ou $\neq 0$) : on a alors une présélection équiprobable.

La règle d'évolution d'un RdPTS est basée sur le tir des transitions. Une transition t_k peut être tirée à partir d'un marquage M_i si :

- elle est structurellement sensibilisée, c'est à dire s'il y a assez de jetons dans ses places d'entrée pour qu'elle soit tirée,
- elle est temporellement sensibilisée, c'est à dire si la probabilité qu'elle soit tirée depuis le marquage M_i n'est pas nulle.

En ce qui concerne les multi-sensibilisations, une politique de mono-service au niveau des transitions est mise en œuvre.

La probabilité de tir d'une transition t_k à partir d'un marquage M_i est donné par

$$P_k = \int_{\theta_{m_k}}^{DIT_i} f_{ke}(x)dx \tag{I.1}$$

où DIT_i est la plus petite des bornes supérieures des intervalles de tir des transitions sensibilisées par M_i, et $f_{ke}(x)$ est la densité de probabilité extrinsèque de t_k. Par exemple, si toutes les transitions sensibilisées dans le marquage M_i ont une densité de probabilité continue, alors $f_{ke}(x) = f_k(x)(\prod_{i\neq k}\int_x^\infty f_i(y)dy)$. Le calcul dans le cas général est donné dans [Ata94].

La date de tir de la transition t_k est une *date de tir moyenne*, donnée par

$$\theta_k = \int_{\theta_{m_k}}^{DIT_i} x.f_{ke}(x)dx \tag{I.2}$$

Lors du tir d'une transition t_k à la date θ_k, la mémoire temporelle des transitions t_i qui restent sensibilisées intervient sur deux points :

- l'intervalle de tir des transitions t_i devient :

$$I'_i = [max(0, \theta_{m_i} - \theta_k), \theta_{M_i} - \theta_k] = [\theta'_{m_i}, \theta'_{M_i}]$$

- la densité de probabilité $f_i(x) = f_{i_c}(x) + f_{i_d}(x)$ des transitions t_i devient :
  - en ce qui concerne la composante continue de la densité

$$f'_{i_c}(x) = \frac{f_i(x + \theta_k)}{1 - F_i(\theta_k^-)}$$

où $F_i(\theta_k^-)$ est la probabilité cumulée de la densité $f_i(x)$ sur l'intervalle $[0; \theta_k[$;
  - en ce qui concerne la composante discrète de la densité

$$f'_{i_d}(x) = \sum_n \frac{K_{in}\delta(x + \theta_k - \theta_n)}{1 - F_i(\theta_k^-)} = \sum_m K'_{im}\delta(x - \theta_m)$$

I.6.3 Le graphe d'états probabilisé

Le comportement dynamique d'un RdPTS est donné par le graphe d'états probabilisé. Ce graphe est défini par :

- un ensemble d'états, chaque état E_i étant un triplet $< M_i, D_i, F_i >$ où :
  - M_i est le marquage du réseau de Petri sous-jacent,
  - D_i est le domaine de tir, c'est à dire l'ensemble des intervalles de tir des transitions sensibilisées par M_i; on note DIT_i la plus petite des bornes supérieures de ces intervalles ($DIT_i = min_j(LFT_j)$),
  - F_i est le processus stochastique de tir, défini par l'ensemble des densités de probabilité des transitions sensibilisées par M_i.

- un ensemble d'arcs orientés entre les états, que l'on appelle transitions entre états, qui sont des triplets $< t_i, p_i, \theta_i >$ où :
  - t_i est la transition du réseau de Petri sous-jacent dont le tir provoque le changement de marquage $M_i \rightarrow M_j$,
  - p_i est la probabilité de branchement, c'est à dire la probabilité de tirer t_i entre les dates EFT_i et DIT_i (calculée grâce à l'équation I.1),
  - θ_i est la durée de séjour inconditionnelle dans l'état M_i (calculé grâce à l'équation I.2),

Les matrices P des probabilités de transitions et Θ des temps inconditionnels de séjour permettent de caractériser la chaîne immergée associée, et de faire les mesures de performance classiques.

Le processus stochastique associé aux états est markovien si toutes les transitions sont exponentielles, et semi-markovien si les transitions exclusives ou compétitives ont n'importe quelle distribution et les transitions parallèles ont des distributions exponentielles. Dans le cas général (distributions quelconques), il est difficile de caractériser le processus stochastique (si on a un graphe cyclique, on a un processus semi-régénératif).

I.6.4 Analyses permises par le graphe d'états probabilisé

Deux types d'analyses sont possibles à partir du graphe d'états probabilisé :

- des analyses qualitatives, dans lesquelles on ne prend pas en compte les étiquettes p_i et θ_i du graphe d'états probabilisé. On distingue deux types d'analyses qualitatives :
  - les analyses qualitatives sur le graphe global, basées sur des séquences de tous les événements du graphe d'états probabilisé;
  - les vues abstraites qualitatives, basées sur la notion d'équivalence observationnelle [Ver89], qui permettent d'obtenir des graphes réduits relatifs à des sous-ensembles d'événements du graphe des classes d'états.
- les analyses quantitatives, dans lesquelles on prend en compte les étiquettes p_i et θ_i du graphe d'états probabilisé. On distingue deux types d'analyses quantitatives :
  - sur le graphe global, on peut calculer les probabilités et temps de séjour dans les états du graphe si le graphe est cyclique, et les probabilités et temps d'absorption si le graphe est acyclique;

    les vues abstraites quantitatives [AJ95] permettent d'obtenir un graphe réduit du graphe d'états probabilisé en se focalisant sur un sous-ensemble d'états de ce graphe.

I.7 Conclusion

Nous avons présenté dans ce chapitre trois types de modèles résultant d'extensions temporelles des Réseaux de Petri :

- les Réseaux de Petri Temporels, dont l'objectif est de pouvoir représenter des contraintes temporelles et de pouvoir vérifier si ces contraintes sont satisfaites. Le graphe des classes d'états est l'objet qui fournit la référence comportementale exhaustive de tous les comportements temporels;
- les Réseaux de Petri Stochastiques, dont l'objectif est d'ajouter une analyse quantitative à une analyse qualitative qui a pu être effectuée sur le réseau de Petri sous-jacent : les modèles SPN et GSPN ne permettent cependant pas d'exprimer des contraintes temporelles; le modèle DSPN (qui admet une transition déterministe au plus par marquage) permet d'aborder de manière limitée le problème des contraintes temporelles. Cependant, il faut noter que pour tous ces modèles, le processus stochastique

associé est bien identifié (markovien pour les modèles SPN et GSPN, semi-régénératif pour le modèle DSPN);

- les Réseaux de Petri Temporisés Stochastiques, qui permettent à la fois d'exprimer, sans restriction, des contraintes temporelles, et de faire, au moyen du graphe d'états probabilisé qui représente le comportement dynamique du système (la structure de ce graphe dépend des contraintes temporelles), à la fois des analyses qualitatives et quantitatives.

  La critique fondamentale est que, d'une part et dans le cas général, le processus associé est difficile à identifier, et d'autre part, comme les transitions sont tirées en un temps moyen, on n'a pas de couverture exhaustive (qualitative et quantitative) du comportement. Il serait intéressant d'augmenter cette couverture. De plus les analyses permises sont soit qualitatives, soit quantitatives, mais ne sont pas reliées. Il serait souhaitable d'établir des relations entre ces analyses.

Le but du travail présenté dans les prochains chapitres est précisément d'améliorer les potentialités du modèle Réseaux de Petri Temporisés Stochastiques dans le sens indiqué.

Chapitre II

Extension du pouvoir d'expression du modèle Réseaux de Petri Temporisés Stochastiques

II.1 Introduction

Le modèle Réseaux de Petri Temporisés Stochastiques, tel qu'il a été défini dans [Ata94], nécessite quelques améliorations afin d'être bien adapté au contexte de la modélisation et de l'analyse des systèmes temps-réel. La vue moyenne offerte par le graphe d'états probabilisé n'est pas suffisante pour avoir une bonne couverture qualitative et quantitative du comportement d'un système. De plus, des mécanismes importants comme la préemption ne peuvent pas être modélisés avec la politique d'exécution de compétition basée sur la mémoire de la dernière sensibilisation.

Nous présentons dans ce chapitre les diverses améliorations que nous avons apportées au modèle Réseaux de Petri Temporisés Stochastiques afin de résoudre ces problèmes :

- le première partie présente les différentes politiques d'exécution que l'on peut utiliser dans le modèle RdPTS, leurs implications sur la construction des graphes d'états probabilisés, et le pouvoir de modélisation supplémentaire qu'elles apportent. Ce pouvoir d'expression est montré à travers l'étude d'un processus panne-réparation;
- la deuxième partie présente les différentes règles de tir possibles, et les différents graphes d'états probabilisés que l'on peut construire à partir de ces règles; ces règles nous permettent d'obtenir d'autres vues que la vue moyenne, et d'étudier des scénarii particuliers du comportement dynamique de systèmes. L'intérêt de ces différentes règles de tir est montré à travers l'étude d'un protocole de communication temps-critique (analyse des cas pires);

II.2 Multiples politiques d'exécution

II.2.1 Justification

Une politique d'exécution est composée de la politique de sélection de la transition qui sera tirée à partir d'un état, et de la politique de la mémoire temporelle pour les transitions qui restent sensibilisées lors du tir d'une autre transition, comme nous l'avons vu dans le premier chapitre de ce mémoire.

Le modèle RdPTS a été défini dans [Ata94] avec une politique de sélection dite de compétition, qui permet de modéliser les aspects de compétition entre les différentes activités qui sont actives en parallèle dans les systèmes temps-réels. Dans le cas où plusieurs transitions sensibilisées ont la même date de tir possible, on applique une politique de présélection, en considérant que ces transitions sont équiprobables (même probabilité de tir).

La politique de mémoire temporelle choisie est la politique de la dernière sensibilisation, car le modèle RdPTS a trouvé ses principales applications, dans un premier temps, dans le domaine des protocoles de communication, où les temporisations sont un mécanisme couramment rencontré (cette politique modélise de manière naturelle les « Time-Out », temporisations qui mesurent la durée de validité d'une donnée ou d'un message).

Par la suite, le champs d'application du modèle RdPTS s'est étendu, et il a été utilisé dans la modélisation et l'analyse d'autres systèmes à contraintes de temps que les protocoles de communication, comme les algorithmes d'ordonnancement [Vas96, Liè97]. Certains mécanismes, comme par exemple la préemption, n'étant pas modélisables avec la politique de mémoire temporelle de la dernière sensibilisation, nous avons augmenté le pouvoir d'expression du modèle en autorisant le choix de la politique de mémoire de toutes les sensibilisations. Nous avons également ajouté la politique de réinitialisation. Ainsi, trois politiques d'exécution sont aujourd'hui définies dans le modèle RdPTS :

- la politique de compétition avec réinitialisation, qui permet la modélisation de systèmes dans lesquels la fin d'une activité provoque la réinitialisation des autres activités,
- la politique de compétition avec mémorisation de la dernière sensibilisation, qui permet de modéliser, par exemple, des temporisations qui s'écoulent en parallèle,
- la politique de compétition avec mémorisation de toutes les sensibilisations, qui permet de modéliser, par exemple, des algorithmes d'ordonnancement préemptifs.

On peut affecter sans contrainte à chacune des transitions du réseau de Petri une de ces trois politiques. On peut ainsi modéliser la plupart des mécanismes rencontrés dans le domaine des systèmes temps-réel.

Ces trois politiques d'exécution possédant toutes la technique de compétition pour sélectionner la transition à tirer, nous les caractériserons par la suite uniquement par le type de mémoire temporelle.

II.2.2 Notion d'état avec les différents types de mémoire temporelle

II.2.2.1 Justification de la modification de la notion d'état

L'introduction de la politique de compétition avec mémorisation de toutes les sensibilisations nécessite une nouvelle définition de la notion d'état utilisée dans la construction du graphe d'états probabilisé. Jusqu'ici ([Ata94]), un état était composé de deux éléments: le marquage du réseau de Petri sous-jacent, qui définit la sensibilisation structurelle des transitions, et le domaine de tir des transitions structurellement sensibilisées, qui définit la sensibilisation temporelle des transitions. Toute transition qui est à la fois sensibilisée structurellement et temporellement est tirable à partir de l'état considéré.

Il est nécessaire de faire apparaître dans les états les attributs temporels des transitions qui suivent la politique de mémoire de toutes les sensibilisations. En effet, ces transitions peuvent, même si elles ne sont pas sensibilisées structurellement, avoir des attributs temporels différents de leurs attributs de départ, ce qui indique que les activités associées à ces transitions ont déjà effectué une partie de leur travail.

Aussi, pour pouvoir prendre en compte cette politique, nous devons ajouter cet élément dans la définition de l'état pour pouvoir différencier les états qui sont équivalents du point de vue sensibilisation structurelle et temporelle, mais qui sont différents en ce qui concerne les attributs temporels des transitions du type mémoire de toutes les sensibilisations. Nous proposons donc une nouvelle définition d'un état.

II.2.2.2 Nouvelle définition de la notion d'état

Un état E est un triplet $< M, D, F >$ où :

- M est le marquage du réseau de Petri sous-jacent,
- D est le domaine de tir, c'est à dire l'ensemble des intervalles de tir des transitions sensibilisées par le marquage de cet état et des transitions qui ne sont pas sensibilisées par ce marquage et qui ont pour mémoire temporelle la mémoire de toutes les sensibilisations,
- F est le processus stochastique de tir, c'est à dire l'ensemble des densités de probabilité associées à l'ensemble des intervalles de tir du domaine de tir.

Notons que si l'ensemble des transitions qui ont pour mémoire temporelle la mémoire de toutes les sensibilisations est vide, alors on retrouve la définition de l'état proposée dans [Ata94].

II.2.2.3 Exemple

Prenons l'exemple du Réseau de Petri Temporisé Stochastique représenté sur la figure II.1, dans laquelle nous considérons (figure II.2) les cas 1 et 2 de mémoire temporelle (notons que la notion de mémoire temporelle est inopérante pour les transitions t_1 et t_3 qui sont immédiates). Dans le cas 1, on obtient le graphe et les états représentés respectivement sur les figures II.3 et II.4. Dans le cas 2, on obtient le graphe et les états représentés respectivement sur les figures II.5 et II.6.

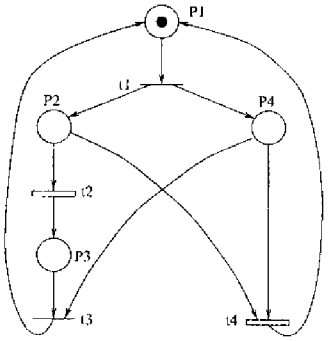


FIG. II.1 – *Exemple de réseau de Petri Temporisé Stochastique*

		mémoire temporelle	
transitions	densités	cas 1	cas 2
t_1, t_3	$\delta(x)$	-	-
t_4	$\delta(x-10)$	dernière sens.	toutes sens.
t_2	$\delta(x-3)$	dernière sens.	dernière sens.

FIG. II.2 – *Densités de probabilité et mémoire temporelle associées aux transitions*

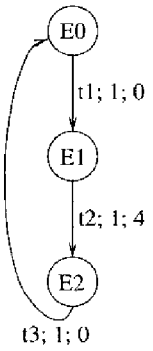


FIG. II.3 – *Graphe d'états probabilisé (mémoire de la dernière sensibilisation)*

État	Description de l'état
E_0	$M_0 = p_1$ $F_0 : f_1(x) = \delta(x)$
E_1	$M_1 = p_2 p_4$ $F_1 : \begin{cases} f_2(x) = \delta(x-4) \\ f_4(x) = \delta(x-10) \end{cases}$
E_2	$M_2 = p_3 p_4$ $F_2 : f_3(x) = \delta(x)$

FIG. II.4 – *Description des états du graphe d'états probabilisé*

Dans le cas 1, on voit que la transition t_4 ne peut jamais être tirée : t_2 sera toujours tirée avant elle.

Dans le cas 2, la transition t_4 pourra être tirée au bout de trois tirs successifs de t_2, puisqu'elle mémorise le temps passé de sensibilisation.

II.2.2.4 Remarque

Avec cette nouvelle définition de l'état, on peut donc construire des graphes d'états à partir de réseaux de Petri dans lesquels on trouve les trois types de politiques d'exécution sur différentes transitions. Plusieurs exemples d'application (algorithmes Rate Monotonic Préemptif, d'ordonnancement à priorité héritée) peuvent être trouvés dans [Liè97].

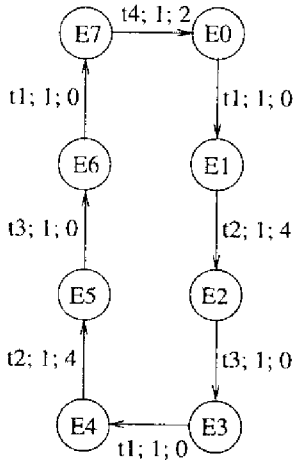


FIG. II.5 – *Graphe d'états probabilisé*

État	Description de l'état
E_0	$M_0 = p_1$ $F_0 : \begin{cases} f_1(x) = \delta(x) \\ f_4(x) = \delta(x-10) \end{cases}$
E_1	$M_1 = p_2p_4$ $F_1 : \begin{cases} f_2(x) = \delta(x-4) \\ f_4(x) = \delta(x-10) \end{cases}$
E_2	$M_2 = p_3p_4$ $F_2 : \begin{cases} f_3(x) = \delta(x) \\ f_4(x) = \delta(x-6) \end{cases}$
E_3	$M_3 = p_1$ $F_3 : \begin{cases} f_1(x) = \delta(x) \\ f_4(x) = \delta(x-6) \end{cases}$
E_4	$M_4 = p_2p_4$ $F_4 : \begin{cases} f_2(x) = \delta(x-4) \\ f_4(x) = \delta(x-6) \end{cases}$
E_5	$M_5 = p_3p_4$ $F_5 : \begin{cases} f_3(x) = \delta(x) \\ f_4(x) = \delta(x-2) \end{cases}$
E_6	$M_6 = p_1$ $F_6 : \begin{cases} f_1(x) = \delta(x) \\ f_4(x) = \delta(x-2) \end{cases}$
E_7	$M_7 = p_2p_4$ $F_7 : \begin{cases} f_2(x) = \delta(x-4) \\ f_4(x) = \delta(x-2) \end{cases}$

FIG. II.6 – *Description des états du graphe d'états probabilisé*

II.2.2.5 Exemple : processus de panne-réparation

Prenons l'exemple d'un processus de panne-réparation, dont le modèle Réseau de Petri Temporisé Stochastique est représenté sur la figure II.7. La place P_1 représente la machine en panne, la place P_3 la machine réparée. La place P_2 représente le réparateur libre (prêt à réparer la machine si celle-ci est en panne), et la place P_4 le réparateur préempté (occupé par une autre tâche plus prioritaire que la réparation de la machine).

La transition t_1 représente la fin de la réparation de la machine. La transition t_2 représente l'occurrence d'une panne. La transition t_3 représente la préemption du réparateur par un processus plus prioritaire, la transition t_4 la fin de cette préemption.

Les densités de probabilités associées à chaque transition sont données dans la table II.8. Le taux de panne de la machine est $\lambda = \frac{1}{100}$. La préemption du réparateur est

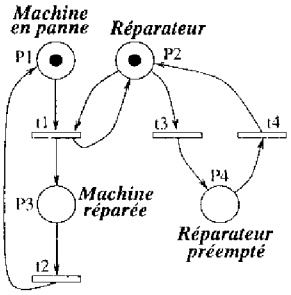


FIG. II.7 – *Modèle du processus de panne-réparation*

transitions	densités
t_1	$\delta(x-10)$
t_2	$\frac{1}{100}e^{-\frac{1}{100}x}$
t_3	$\delta(x-4)$
t_4	$\delta(x-5)$

FIG. II.8 – *Densités de probabilité des transitions du modèle de la figure II.7*

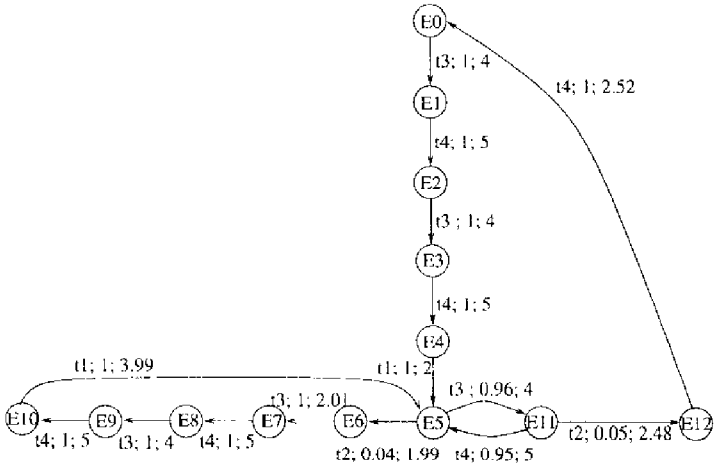


FIG. II.9 – *Graphe d'états probabilisé moyen*

périodique, toutes les 4 unités de temps, et dure 5 unités de temps. La réparation de la machine dure 10 unités de temps. Notons que toutes les transitions ont pour mémoire temporelle la mémoire de la dernière sensibilisation, sauf la transition t_1 qui utilise la mémoire de toutes les sensibilisations. Cela lui permet de mémoriser le travail de réparation effectué par le réparateur, même si ce dernier est interrompu (t_3) avant la fin de ce travail.

L'état initial est tel que le réparateur débute la réparation de la machine (places P_1 et P_2 marquées).

Le graphe d'états probabilisé moyen relatif à cette configuration temporelle est donné sur la figure II.9. On vérifie bien que le réparateur est interrompu deux fois au cours de la réparation de la machine (séquence $t_3t_4t_3t_4t_1$ entre E_0 et E_5, et E_6 et E_5). Si l'on mesure le temps de réparation *dans les états où le réparateur répare la machine* (états E_0 (4 u.t.), E_2 (4 u.t.), E_4 (2 u.t.) d'une part, E_6 (2,01 u.t.), E_8 (4 u.t.), E_{10} (3,99 u.t.) d'autre part),

on trouve bien que cette réparation dure 10 unités de temps.

Des exemples plus significatifs d'utilisation dans le domaine du temps-réel pourront être trouvés dans [Liè97]

II.3 Multiples règles de tir

II.3.1 Justification de la définition de plusieurs règles de tir

La construction du graphe d'états probabilisé effectuée sur la base du tir des transitions à la valeur moyenne masque des comportements possibles (excepté si toutes les distributions sont déterministes et/ou exponentielles).

Ce problème doit être étudié particulièrement dans le cas du parallélisme, où plusieurs transitions sont simultanément sensibilisées et tirables. Si ces transitions ont des densités de probabilité qui ont la propriété de mémoire temporelle (densités uniformes ou mixtes), alors le tir à la valeur moyenne peut nous faire « manquer » des comportements (des chemins), et donc ne donne pas tous les états du processus stochastique.

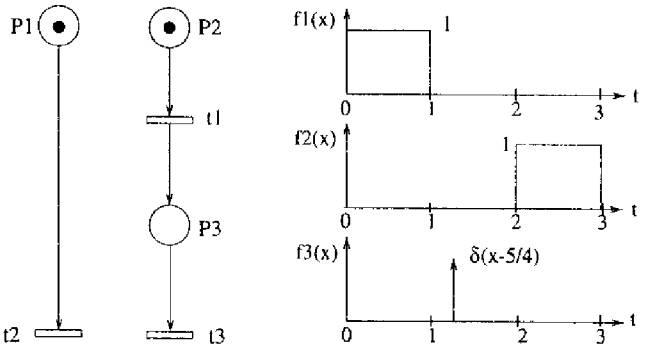


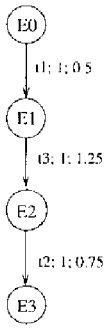
FIG. II.10 – *Exemple de réseau de Petri Temporisé Stochastique*

L'exemple de la figure II.10 illustre ce point. La spécification temporelle choisie est donnée sur la figure II.10.

La figure II.11 donne le graphe d'états probabilisé (sur la base de la valeur moyenne) construit à partir de cet exemple. La table de cette figure donne les attributs temporels des transitions sensibilisées dans chaque état du graphe. On peut voir sur ce graphe que même si t_2 est sensibilisée en même temps que t_1 et que t_3 dans les états E_0 et E_1, elle ne peut être tirée qu'après ces transitions.

Or, on imagine facilement que si t_1 était tirée à une date très voisine de la borne supérieure de son intervalle de tir (par exemple, 0.9), t_2 pourrait être tirée avant t_3 dans l'état E_1. Cette séquence de tirs des transitions n'est pas visible sur le graphe d'états probabilisé de la figure II.11.

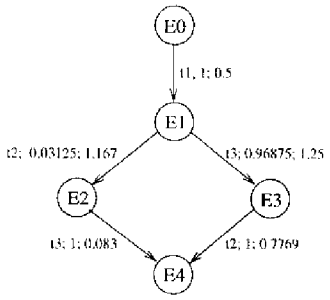
La solution a ce problème serait de prendre en compte toutes les dates de tir possibles



Etat	Marquage	Attributs Temporels
E0	P_1=1, P_2=1	t1 : $f_1(x)$= 1; $0 \leqslant x \leqslant 1$; 0 ailleurs t2 : $f_2(x)$= 1; $2 \leqslant x \leqslant 3$; 0 ailleurs
E1	P_1=1, P_3=1	t2 : $f_2(x)$= 1; $3/2 \leqslant x \leqslant 5/2$; 0 ailleurs t3 : $f_3(x)=\delta(x-5/4)$
E2	P_3=1	t2 : $f_2(x)$= 1; $1/4 \leqslant x \leqslant 5/4$; 0 ailleurs
E3		*Etat puits*

FIG. II.11 – *Graphe d'états probabilisé (tir à la valeur moyenne)*

pour chaque transition. La difficulté réside dans le calcul de la nouvelle densité de probabilité des transitions qui restent sensibilisées (transitions en parallèle) lors du tir d'une autre transition (calcul de l'âge pris par ces transitions). Ce calcul ne peut être fait que par produit de convolution entre la densité de probabilité de la transition qui est tirée, et la densité de probabilité de la transition qui reste sensibilisée. Le résultat est la densité de probabilité de la transition qui reste sensibilisée, dans le nouvel état.



Etat	Marquage	Attributs Temporels
E0	P_1=1, P_2=1	t1 : $f_1(x)$= 1; $0 \leqslant x \leqslant 1$; 0 ailleurs t2 : $f_2(x)$= 1; $2 \leqslant x \leqslant 3$; 0 ailleurs
E1	P_1=1, P_3=1	t2 : $f_2^*(x)$= $x-1$; $1 \leqslant x \leqslant 2$; $3-x$; $2 \leqslant x \leqslant 3$; 0 ailleurs t3 : $f_3(x)=\delta(x-5/4)$
E2	P_3=1	t3 : $f_3^*(x)$= $8-32x$; $0 \leqslant x \leqslant 1/4$; 0 ailleurs
E3	P_1=1	t2 : $f_2^*(x)$= $(32/31)x + 8/31$; $0 \leqslant x \leqslant 3/4$; $56/31 - (32/31)x$; $3/4 \leqslant x \leqslant 7/4$; 0 ailleurs
E4		*Etat puits*

FIG. II.12 – *Graphe d'états probabilisé (comportement complet)*

La figure II.12 donne le graphe d'états probabilisé issu de l'exemple de la figure II.10, en utilisant le produit de convolution dans le calcul de l'âge des transitions [Ata94]. Les attributs temporels des transitions sensibilisées dans chaque état sont donnés dans la table de la même figure. Les fonctions notées f^* sont issues d'un produit de convolution. Sur ce graphe, nous voyons apparaître le comportement complet du système (il y a un chemin de plus que dans le graphe de la figure II.11).

Malheureusement, le produit de convolution modifie la forme d'une densité de pro-

babilité, et augmente la complexité de cette dernière. Aussi, quand une transition reste sensibilisée pendant plusieurs tirs de transitions, ce calcul devient trop complexe et ne peut plus être effectué. Dans l'exemple de la figure II.12, on peut remarquer cette augmentation de complexité sur la densité de probabilité de la transition t_2 sur le chemin $E_0 \rightarrow E_1 \rightarrow E_3$.

Le tir de la transition à une date moyenne ne nous permet donc d'obtenir qu'un point de vue du système. L'utilisation du produit de convolution permet d'obtenir tous les comportements du système (tous les points de vue), mais ne peut pas être utilisé à cause de la complexité de calcul qu'il introduit. Nous nous proposons donc de définir une solution intermédiaire : définir plusieurs dates de tir possibles pour chaque transition, ce qui permet de conserver la simplicité de calcul du tir à la date moyenne, et d'obtenir plusieurs points de vue du comportement du système. Chacune de cette date de tir définie une règle de tir de transition.

II.3.2 Définition des différentes règles de tir

Nous définissons cinq règles de tir différentes pour une transition t_i :

1. **la règle de tir *MIN***; la transition t_i est tirée à la borne inférieure de son intervalle de tir : $\theta_i = \theta_{m_i}$. Cette règle permet d'obtenir une évolution au plus tôt de la transition t_i;

2. **la règle de tir *min***; la transition t_i est tirée à la valeur moyenne moins l'écart type, si cette valeur n'est pas à l'extérieur de l'intervalle de tir, sinon à la borne inférieure de l'intervalle de tir : $\theta_i = \max(\theta_{m_i}, \theta_{moyen_i} - \sigma_i)$. Cette règle permet d'obtenir une date d'évolution rapide de la transition t_i dans l'ensemble des dates les plus probables;

3. **la règle de tir *moyenne***; la transition t_i est tirée à la valeur moyenne : $\theta_i = \theta_{moyen_i}$. Cette règle de tir permet d'obtenir la date d'évolution de la transition la plus probable en moyenne;

4. **la règle de tir *max***; la transition t_i est tirée à la valeur moyenne plus l'écart type, si cette valeur n'est pas à l'extérieur de l'intervalle de tir, sinon au $DIT = \min_j(\theta_{Mj}), j \in \{\text{transitions sensibilisées}\}$ (plus petite borne supérieure des intervalles de tir des transitions qui sont sensibilisées) : $\theta_i = \min(DIT, \theta_{moyen_i} + \sigma_i)$. Cette règle permet d'obtenir la date d'évolution la plus lente dans l'ensemble des dates les plus probables;

5. **la règle de tir *MAX***; la transition t_i est tirée au DIT, plus petite borne supérieure des intervalles de tir des transitions sensibilisées : $\theta_i = DIT = \min_j(\theta_{Mj}), j \in \{\text{transitions sensibilisées}\}$. Cette règle permet d'obtenir une évolution au plus tard de la transition t_i.

Les valeurs de θ_{moyen_i} et σ_i sont calculées comme suit :

- dans [Ata94], la notion de densité de probabilité extrinsèque $f_{ie}(x)$ a été définie à partir des densités de probabilité intrinsèques des transitions sensibilisées. A partir

de cette densité de probabilité, la date de tir moyenne se calcule comme suit :

$$\theta_{moyen_i} = \frac{1}{P_i} \int_{\theta_{mi}}^{DIT} x.f_{ie}(x).dx \quad \text{(II.1)}$$

avec P_i, probabilité de branchement :

$$P_i = \int_{\theta_{mi}}^{DIT} f_{ie}(x).dx \quad \text{(II.2)}$$

- la valeur de l'écart type σ_i est calculée comme suit :

$$\sigma_i = \sqrt{\mathcal{M}_{2_i} - \theta^2_{moyen_i}} \quad \text{(II.3)}$$

avec $\mathcal{M}_{2_i}$, moment d'ordre 2 :

$$\mathcal{M}_{2_i} = \frac{1}{P_i} \int_{\theta_{mi}}^{DIT} x^2.f_{ie}(x).dx \quad \text{(II.4)}$$

II.3.3 Définition des différents types de graphes d'états probabilisés

L'affectation de ces cinq règles de tir aux transitions du modèle RdPTS étudié nous permet de construire différents graphes d'états probabilisés, chacun d'eux donnant une vue d'une partie du comportement du système. Nous définissons deux types de graphes d'états probabilisés : les graphes homogènes et les graphes hétérogènes.

II.3.3.1 Les graphes d'états probabilisés homogènes

Lors de la construction de ce type de graphe, on affecte à toutes les transitions la même règle de tir. Il y a donc 5 graphes homogènes différents :

- le graphe *MIN* issu du tir des transitions suivant la règle *MIN*; ce graphe permet d'obtenir le comportement dynamique au plus tôt du système,
- le graphe *min* issu du tir des transitions suivant la règle *min*; ce graphe permet d'obtenir un comportement dynamique rapide du système,
- le graphe *moyen* issu du tir des transitions suivant la règle *moyen*; ce graphe permet d'obtenir le comportement dynamique statistiquement le plus probable du système,
- le graphe *max* issu du tir des transitions suivant la règle *max*; ce graphe permet d'obtenir un comportement dynamique lent,
- le graphe *MAX* issu du tir des transitions suivant la règle *MAX*; ce graphe permet d'obtenir le comportement dynamique au plus tard du système.

Ces graphes permettent d'étudier les différents comportements dynamiques du systèmes. Dans de nombreux cas, leur construction et leur comparaison permettent d'en retirer le comportement complet du système (les différents points de vue obtenus donnent l'ensemble des comportements possibles du système). Malheureusement, cette propriété n'est pas vérifiée dans le cas général. On retrouve donc les mêmes incertitudes qu'avec la règle de tir moyen seule, mais le fait de construire ces cinq graphes nous permet, d'une part, d'obtenir dans beaucoup plus de cas le comportement global du système, et d'autre part, d'avoir différentes informations quantitatives pour chaque point de vue.

II.3.3.2 Les graphes d'états probabilisés hétérogènes

Lors de la construction de ce type de graphe, on affecte à chaque transition une règle de tir. On peut donc ainsi définir cinq sous-ensembles de transitions : les transitions qui suivent la règle *MIN*, les transitions qui suivent la règle *min*, les transitions qui suivent la règle *moyen*, les transitions qui suivent la règle *max*, et les transitions qui suivent la règle *MAX*.

Notation : on notera, par exemple, MIN($\{t_{MIN}\}$)MAX($\{t_{MAX}\}$) un graphe hétérogène qui est construit à partir des règles *MIN*, *moyen* et *MAX*. $\{t_{MIN}\}$ est l'ensemble des transitions qui suivent la règle *MIN* et $\{t_{MAX}\}$ l'ensemble des transitions qui suivent la règle *MAX*. Les transitions du réseau de Petri sous-jacent qui n'appartiennent pas à ces deux ensembles suivent la règle *moyen*.

On peut construire un grand nombre de graphes hétérogènes : si n est le nombre de transitions dans le modèle RdPTS, on pourra construire $5^n - 5$ graphes hétérogènes différents.

Les graphes hétérogènes ont une utilité indéniable dans l'étude des cas pires du système : ils permettent d'affecter un comportement bien particulier à chaque transition du modèle, et donc d'étudier les cas extrêmes du comportement dynamique du système.

II.3.4 Application à un protocole de communication temps critique

Le but de ce paragraphe est de montrer, à travers un exemple, l'intérêt des différentes règles de tir, et notamment les règles MIN, moyen et MAX.

Considérons un bus de terrain [NOR94] qui utilise comme ressource de transmission un bus série, dans lequel une station envoie périodiquement une donnée à une station consommatrice (la station consommatrice attend périodiquement la donnée qui doit être reçue dans un certain intervalle temporel). L'architecture (éléments de service et de protocole) considérée est montrée sur la figure II.13.

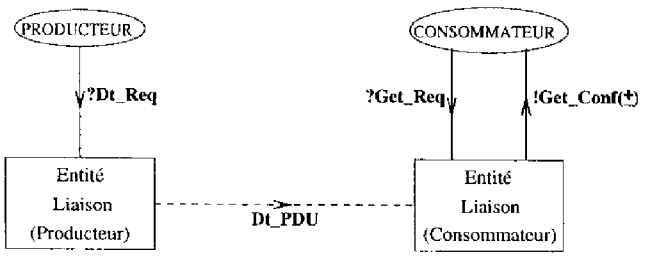


FIG. II.13 – *Éléments de service et de protocole*

L'entité liaison (producteur) reçoit périodiquement un *Dt_Req* du Producteur et envoie un *Dt_PDU*.

L'entité liaison (consommateur) reçoit périodiquement le *Get_Req* du Consommateur. En fonction de l'arrivée ou non du *Dt_PDU* dans l'intervalle temporel, et à partir de

l'instant de réception du *Get_Req*, l'entité liaison (consommateur) envoie $Get\_Conf(+)$ ou $Get\_Conf(-)$. Le *Dt_PDU* peut être reçu en dehors de la fenêtre temporelle parce qu'il y a de la gigue dans le phénomène de production.

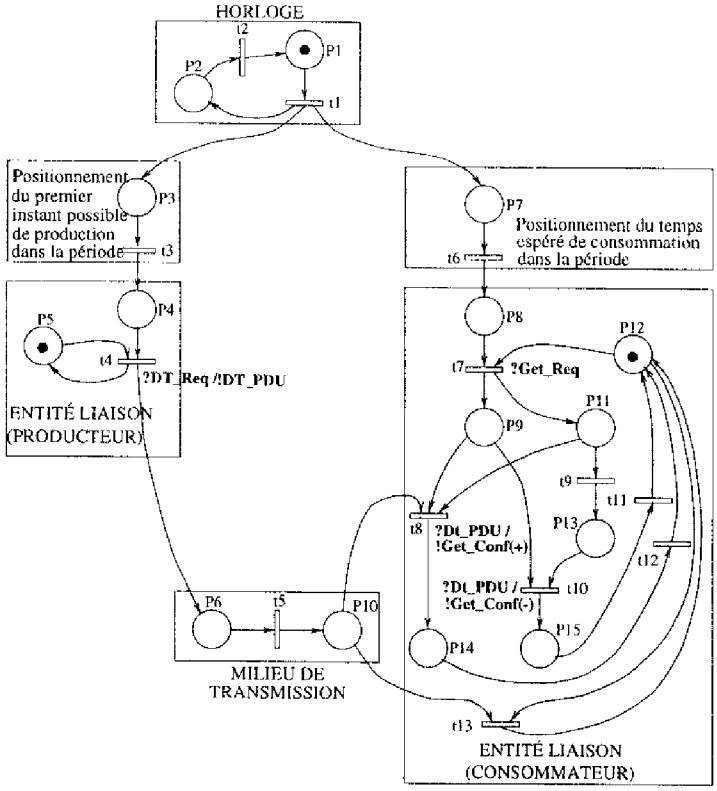


FIG. II.14 – *Modèle du réseau de Petri sous-jacent*

II.3.4.1 Le modèle RdPTS

Le réseau de Petri sous-jacent

Le modèle du réseau de Petri sous-jacent (figure II.14) consiste en une entité liaison de production (la place $P5$ modélise l'entité de production; la place $P4$ modélise l'autorisation de production (respect de la périodicité); la transition t_4 modélise la gigue dans la production), en un milieu de transmission supposé parfait, c'est à dire sans perte (places $P6$ et $P10$; transition t_5 qui donne le temps de propagation), en une entité liaison de consommation (places $P8$, $P9$, $P11$, $P12$, $P14$ et $P15$; transitions t_7, t_8, t_9, t_{10}, t_{11}, t_{12} et t_{13}; la transition t_9 modélise la fenêtre temporelle, c'est à dire la durée au delà de laquelle la réception d'une donnée n'est plus acceptable), et en un modèle représentant la périodicité (horloge avec les places $P1$ et $P2$, la transition t_1 qui modélise le top d'horloge sur les deux entités, et la transition t_2 qui donne la périodicité de l'horloge) et le positionnement, dans la période, du premier instant possible de production (place $P3$ et transition t_3) et

du temps espéré de consommation (place $P7$ et transition t_6).

Spécifications temporelles

Nous considérons trois spécifications temporelles qui diffèrent uniquement par la spécification des transitions t_3 et t_4 (qui modélisent la gigue dans la production).

Les autres transitions ont toujours la spécification suivante :

$t_1, t_7, t_8, t_{10}, t_{11}, t_{12}, t_{13} : \delta(x)$;
$t_2 : \delta(x - 200)$
$t_5 : \delta(x - 5)$
$t_6 : \delta(x - 100)$
$t_9 : \delta(x - 10)$

Le trois spécifications des transitions t_3 et t_4 sont :

- spécification 1 : $t_3 : \delta(x - 94)$; $t_4 : f_4(x) = \frac{1}{12}$ pour $0 \leq x \leq 12$, $f_4(x) = 0$ ailleurs;
- spécification 2 : $t_3 : \delta(x - 96)$; $t_4 : f_4(x) = \frac{1}{8}$ pour $0 \leq x \leq 8$, $f_4(x) = 0$ ailleurs;
- spécification 3 : $t_3 : \delta(x - 98)$; $t_4 : \delta(x - 4)$;

La figure II.15 illustre ces trois spécifications.

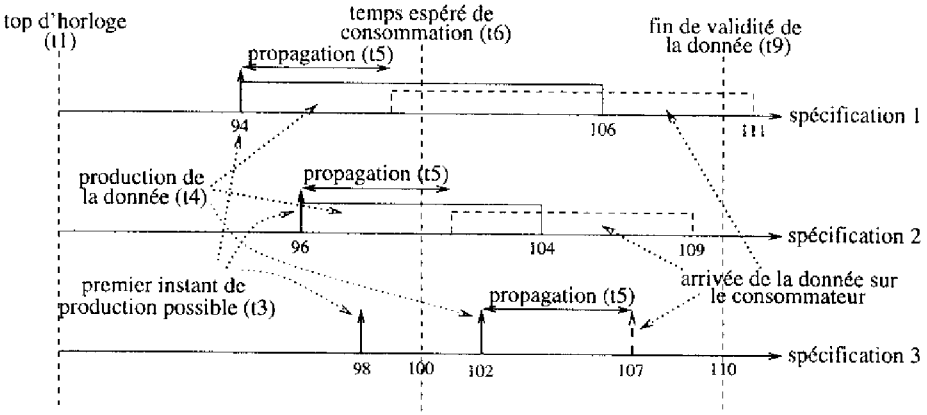


FIG. II.15 – *Spécifications temporelles étudiées*

II.3.4.2 Quelles analyses?

Comme on a un processus de production avec gigue (notion d'intervalle avec bornes inférieure et supérieure), on a un système qui n'a pas un comportement markovien, et donc une analyse classique ne peut être effectuée.

Il faut identifier les *deux cas pires* qui représentent les bornes de tous les comportements possibles : si au moins un de ces cas pires donne un comportement incorrect, la spécification temporelle ne peut être acceptée; si les deux cas pires donnent un comportement correct, la spécification temporelle peut être acceptée (on est sûr que l'on aura toujours un comportement correct).

Les deux cas pires sont :

le producteur produit au plus tôt; on doit considérer le graphe MIN(t_4) que nous appellerons MIN par la suite;

- le producteur produit au plus tard; on doit considérer le graphe MAX(t_4) que nous appellerons MAX par la suite.

Les analyses suivantes doivent plus particulièrement être effectuées :

- analyse qualitative : l'inévitabilité ou la potentiabilité de consommations correctes; le service fourni du côté consommateur (vue abstraite qualitative (cf chapitre IV.2.1) relative aux événements $?Get\_Req$, $!Get\_Conf(+)$ et $!Get\_Conf(-)$).
- analyse quantitative : le temps moyen de production t_p; le temps moyen de consommation correcte t_c; le temps moyen entre une production et une consommation correcte t_{cc}.

II.3.4.3 Analyse de la spécification 1

Comme toutes les transitions ont une distribution déterministe, exceptée la transition t_4 qui est uniformément distribuée, cinq graphes peuvent être obtenus pour visualiser l'influence de la gigue. Ces cinq graphes sont :

les graphes min, moyen, max qui montrent un comportement correct (Fig II.16);

- le graphe MIN qui montre à la fois un comportement correct et un comportement incorrect (arrivée de la donnée en avance chez le consommateur : tir de la transition t_{13} et ensuite tirs des transitions t_9 et t_{10}). Ce graphe est représenté sur la figure II.17;

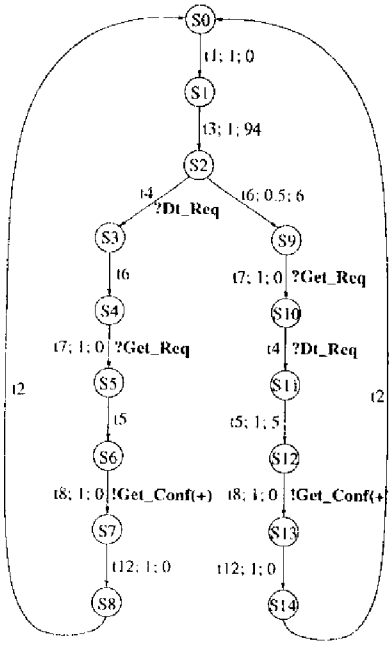
le graphe MAX qui montre à la fois un comportement correct et un comportement incorrect (arrivée tardive de la donnée chez le consommateur : tirs des transitions t_9 et t_{10}, et ensuite tir de la transition t_{13}), comme on peut le voir sur la figure II.18.

Analyse qualitative

La propriété de potentialité de consommation correcte apparaît sur les graphes MIN et MAX. Par contre, avec les graphes min, moyen et max, on a l'inévitabilité d'une consommation correcte.

Ces propriétés peuvent être visualisées à travers une vue abstraite qualitative relative aux événements locaux du service consommateur (figure II.19) :

- graphes min, moyen, max : ils présentent un comportement dynamique correct ($?Get\_Req$ toujours suivi de $!Get\_Conf(+)$);
- graphe MIN : ce graphe montre deux comportements dynamiques possibles : d'une part $?Get\_Req$ est suivi de $!Get\_Conf(-)$ (si la transition invisible t_4 ($?Dt\_Req$) est tirée avant t_7 ($?Get\_Req$) : comportement incorrect), d'autre part $?Get\_Req$ est suivi de $!Get\_Conf(+)$ (si t_4 est tirée après t_7 : comportement normal);



	graphe min	graphe moyen	graphe max
S2⇨S3	0.5; 1.27	0.5; 3	0.5; 4.73
S3⇨S4	1; 4.73	1; 3	1; 1.27
S5⇨S6	1; 0.27	1; 2	1; 3.73
S8⇨S0	1; 99.73	1; 98	1; 96.27
S10⇨S11	1; 1.27	1; 3	1; 4.73
S14⇨S0	1; 93.73	1; 92	1; 90.27

FIG. II.16 – *Graphes min, moyen, max (spécification 1)*

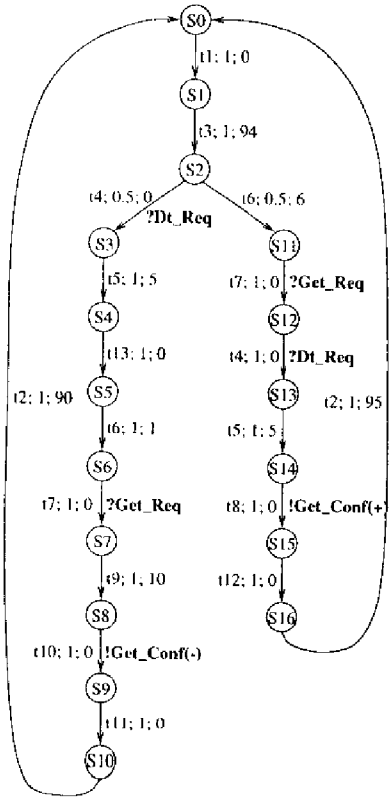


FIG. II.17 – *Graphe MIN (spécification 1)*

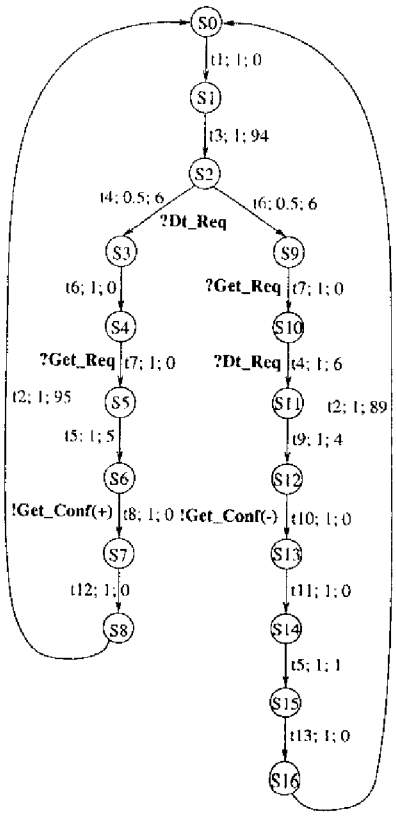


FIG. II.18 – *Graphe MAX (spécification 1)*

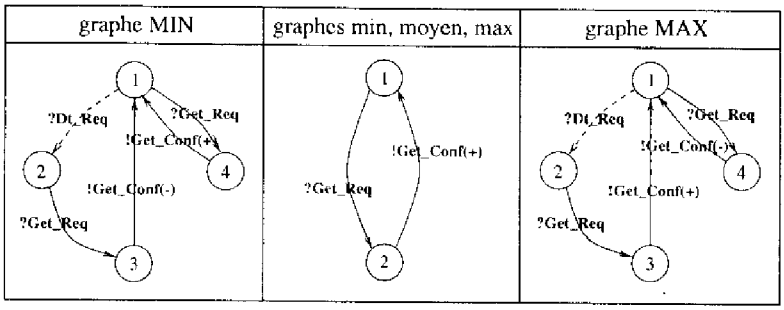


FIG. II.19 – *Vues abstraites qualitatives: service local*

- graphe MAX : ici encore, deux comportements dynamiques peuvent être observés : d'une part $?Get\_Req$ est suivi de $!Get\_Conf(+)$ (si la transition invisible t_4 ($?Dt\_Req$) est tirée avant t_7 ($?Get\_Req$) : comportement normal), d'autre part $?Get\_Req$ est suivi de $!Get\_Conf(-)$ (si t_4 est tirée après t_7 : comportement anormal).

	t_p	t_c	t_{cc}
graphe min	200	200	3,27
graphe moyen	200	200	5
graphe max	200	200	6,73
graphe MIN	200	400	208
graphe MAX	200	400	202

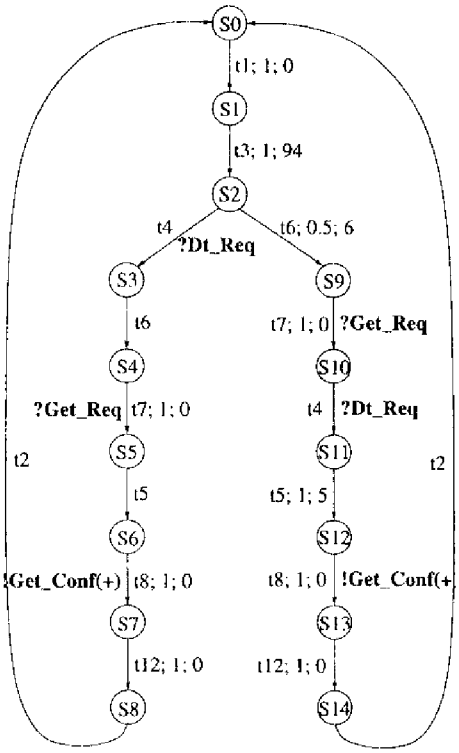
FIG. II.20 – *Vues abstraites quantitatives*

Analyse quantitative

Nous résumons sur le tableau de la figure II.20 les principaux résultats obtenus qui traduisent de manière quantitative l'analyse qualitative précédente (on voit avec les graphes MIN et MAX le phénomène de « sous-consommation » puisque $t_c > t_p$). Les calculs de t_c, t_p et t_{cc} sont effectués par application des règles de réduction de Beizer (cf chapitre IV.3.1) sur les différents graphes d'états probabilisés.

Commentaire

La spécification 1 ne peut être acceptée puisque les cas pires donnent des situations erronées.



	graphe MIN	graphe moyen	graphe MAX
S2⇨S3	0.5; 4	0.5; 2	0.5; 4
S3⇨S4	1; 4	1; 2	1; 0
S5⇨S6	1; 1	1; 3	1; 5
S8⇨S0	1; 99	1; 97	1; 95
S10⇨S11	1; 0	1; 2	1; 4
S14⇨S0	1; 95	1; 93	1; 91

FIG. II.21 – *Graphes MIN, moyen, MAX (spécification 2)*

II.3.4.4 Analyse de la spécification 2

Tous les graphes donnent maintenant un comportement correct. Nous représentons sur la figure II.21 uniquement les graphes MIN, moyen et MAX.

Analyse

On peut voir l'inévitabilité de la consommation correcte. Le service local au niveau du consommateur , quel que soit le graphe considéré, est comme sur le graphe du milieu de la figure II.19. Les performances obtenues sont recensées sur le tableau de la figure II.22. Elles montrent quantitativement que le processus se déroule correctement ($t_c = t_p$) mais fait apparaître de la dispersion sur le temps t_{cc} (ce qui est logique).

	t_p	t_c	t_{cc}
graphe MIN	200	200	3
graphe moyen	200	200	5
graphe MAX	200	200	7

FIG. II.22 – *Analyse quantitative (spécification 2)*

Commentaire

Si la dispersion sur t_{cc} est acceptable, la spécification peut être acceptée. Dans le cas contraire, elle doit être rejetée.

II.3.4.5 Analyse de la spécification 3

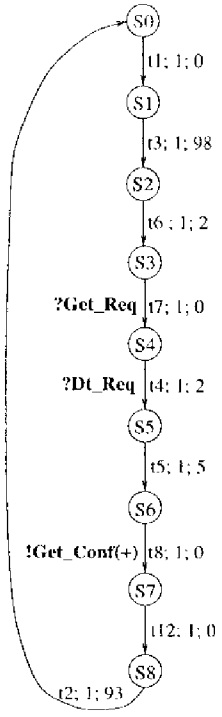


FIG. II.23 *Graphe d'états probabilisé (spécification 3)*

Maintenant, on a un seul graphe (puisqu'on a des distributions déterministes) qui montre (figure II.23) que l'on a un fonctionnement correct (le service local sur le consommateur est identique à celui du graphe du milieu de la figure II.19). Les performances sont : $t_p = 200$; $t_c = 200$; $t_{cc} = 7$. Si la valeur de t_{cc} est acceptable, la spécification temporelle peut être considérée.

II.4 Conclusion

L'introduction dans le modèle RdPTS de plusieurs types de mémoire temporelle associés à la politique de sélection de compétition permet d'augmenter le pouvoir d'expression du modèle. En particulier, la mémoire de toutes les sensibilisations permet la modélisation des mécanismes de préemption. Elle nécessite, pour pouvoir être prise en compte dans

la structure même des graphes d'états probabilisés, une nouvelle définition de la notion d'états. Un exemple de processus de panne-réparation illustre ce dernier point.

La définition des différentes règles de tir apportent un plus indéniable au modèle Réseaux de Petri Temporisés Stochastiques pour l'analyse qualitative et quantitative des modèles :

- la construction des différents graphes, et notamment des graphes hétérogènes comme les cas pires, permet d'étudier des comportements particuliers de systèmes (aspect important dans un contexte temps-réel);
- les différents graphes d'états particuliers apportent, pour chaque comportement étudié, des informations quantitatives différentes. De ces informations peuvent être tirées différentes propriétés quantitatives

L'analyse d'un protocole temps critique producteur-consommateur (production périodique avec gigue; fenêtre de validité temporelle de consommation) a montré comment l'utilisation des différents graphes d'états probabilisés permet de déterminer (à travers l'étude des cas pires) la validité de spécifications temporelles.

Chapitre III

Quelques réflexions sur les modèles RdPTS et RdPT

III.1 Introduction

Il est important de situer les différents graphes d'états probabilisés, qui sont issus des différentes règles de tir, et qui permettent donc de représenter des points de vue temporels et probabilistes, par rapport à un graphe qui représente le cadre reconnu de l'exhaustivité comportementale, c'est-à-dire qui donne toutes les séquences de tirs de transitions possibles en tenant compte des attributs temporels associés aux transitions. Ce graphe est le graphe des classes d'états, qui décrit le comportement dynamique du modèle RdPT, modèle à partir duquel on définit le modèle RdPTS en associant des densités de probabilités aux intervalles des transitions. Notons que comme le modèle RdPT est basé sur la notion de mémoire de la dernière sensibilisation, nous considérons que les graphes d'états probabilisés étudiés utilisent aussi cette politique de mémoire temporelle.

Le but de ce chapitre est précisément, à partir d'exemples de modélisation basé sur le modèle RdPT, de faire une analyse critique du graphe des classes d'états, et ensuite de se servir de cette analyse pour, d'une part, positionner les différents graphes d'états probabilisés basés sur le modèle RdPTS (ici, nous considérons uniquement les graphes MIN, moyen et MAX), et, d'autre part, proposer des modifications relatives au graphe des classes d'états.

Ce chapitre est composé de trois parties :

- dans la première partie, nous donnons deux exemples de modélisation au moyen du modèle RdPT qui nous permettent de faire des commentaires sur le graphe des classes d'états;
- la deuxième partie est consacrée au positionnement des graphes d'états probabilisés MIN, moyen et MAX dans le cadre d'un graphe des classes d'états;

- la troisième partie présente le concept de graphe des sous-classes d'états qui définit, dans des cas particuliers de Réseaux de Petri, un objet représentatif du comportement temporel plus rigoureux que l'objet graphe des classes d'états.

III.2 Commentaires sur le graphe des classes d'états

Nous allons, à travers deux exemples de Réseaux de Petri Temporels, mettre en exergue deux points particuliers du graphe des classes d'états, points très importants pour la suite de ce chapitre.

III.2.1 Exemple 1

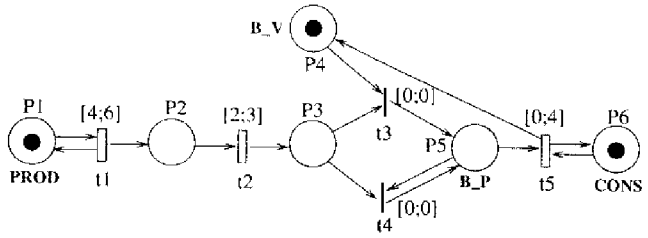


FIG. III.1 *Exemple de Réseau de Petri Temporel*

III.2.1.1 Description du réseau de Petri

L'exemple de la figure III.1 modélise un système producteur-consommateur, avec transmission des données par un réseau. Un buffer d'une capacité d'un message permet de stocker les données sur le consommateur avant qu'elles ne soient consommées.

La signification des places est la suivante : la place P_1 (PROD) modélise le producteur, la place P_6 (CONS) le consommateur. La place P_2 représente la donnée produite par le producteur, la place P_3 représente l'arrivée de la donnée sur le consommateur. La place P_4 modélise le buffer vide (B_V), la place P_5 modélise le buffer plein (B_P).

La signification des transitions est la suivante : la transition t_1 représente la production, la transition t_5 la consommation. La transition t_2 modélise la propagation de la donnée sur le médium de communication. La transition t_3 modélise une bufferisation correcte de la donnée (place P_4 marquée), la transition t_4 un écrasement du buffer (place P_5 marquée). Notons que cette transition t_4, si elle visualise l'écrasement, induit lorsqu'elle est tirée une réinitialisation de la transition t_5 (ce qui n'est pas d'un point de vue temporel le comportement correct du système). La représentation correcte serait obtenue en remplacant les deux arcs reliant t_4 et P_5 par un arc inhibiteur entre P_4 et t_4. Mais ce qui nous importe ici est l'étude d'un Réseau de Petri Temporel et non pas la représentation précise d'un mécanisme.

La production peut s'effectuer dans un temps entre 4 et 6 unités de temps (u.t.). La consommation s'effectue dans un temps entre 0 et 4 u.t., la propagation dure entre 2 et 3 u.t.

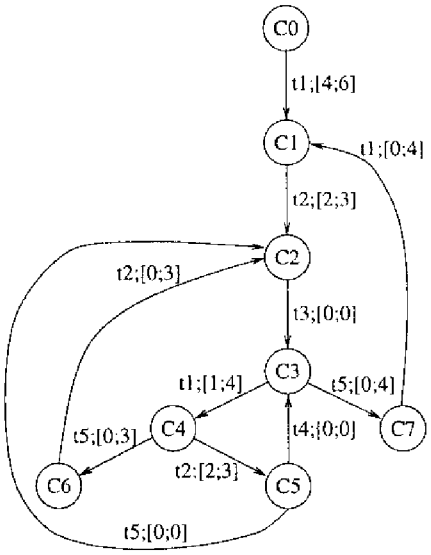


FIG. III.2 – *Graphe des classes d'états*

$$C_0 : \begin{cases} M_0 = p_1 p_4 p_6 \\ D_0 = \{\ 4 \le t(1) \le 6 \end{cases} \qquad C_1 : \begin{cases} M_1 = p_1 p_2 p_4 p_6 \\ D_1 = \begin{cases} 4 \le t(1) \le 6 \\ 2 \le t(2) \le 3 \end{cases} \end{cases}$$

$$C_2 : \begin{cases} M_2 = p_1 p_3 p_4 p_6 \\ D_2 = \begin{cases} 1 \le t(1) \le 4 \\ 0 \le t(3) \le 0 \end{cases} \end{cases} \qquad C_3 : \begin{cases} M_3 = p_1 p_5 p_6 \\ D_3 = \begin{cases} 1 \le t(1) \le 4 \\ 0 \le t(5) \le 4 \end{cases} \end{cases}$$

$$C_4 : \begin{cases} M_4 = p_1 p_2 p_5 p_6 \\ D_4 = \begin{cases} 4 \le t(1) \le 6 \\ 2 \le t(2) \le 3 \\ 0 \le t(5) \le 3 \end{cases} \end{cases} \qquad C_5 : \begin{cases} M_5 = p_1 p_3 p_5 p_6 \\ D_5 = \begin{cases} 1 \le t(1) \le 4 \\ 0 \le t(4) \le 0 \\ 0 \le t(5) \le 1 \\ t(1) - t(5) \le 6 \\ t(5) - t(1) \le -1 \end{cases} \end{cases}$$

$$C_6 : \begin{cases} M_6 = p_1 p_2 p_4 p_6 \\ D_6 = \begin{cases} 1 \le t(1) \le 6 \\ 0 \le t(2) \le 3 \\ t(1) - t(2) \le 4 \\ t(2) - t(1) \le -1 \end{cases} \end{cases} \qquad C_7 : \begin{cases} M_7 = p_1 p_4 p_6 \\ D_7 = \{\ 0 \le t(1) \le 4 \end{cases}$$

TAB. III.1 – *Description des classes d'états du graphe de la figure III.2*

III.2.1.2 Graphe des classes d'états et critique

Le graphe des classes d'états correspondant à cette configuration temporelle est décrit sur la figure III.2 et la table III.1.

Étudions plus précisément les classes C_1 et C_6 de la figure III.2, qui correspondent au

même marquage $p_1p_2p_4p_6$ du réseau de Petri sous-jacent. Dans ces deux classes, seules t_1 et t_2 sont tirables (table III.1). Dans C_1, il n'y a pas de mémoire croisée entre ces deux transitions. Dans C_6, il y a une mémoire croisée qui apparaît.

Si on calcule la mémoire croisée entre les transitions t_1 et t_2 de la classe C_1 (elle n'apparaît pas dans C_1, car elle est redondante avec les inéquations qui définissent les intervalles de tir de t_1 et t_2, mais on peut quand même la calculer), on obtient $t(1)-t(2) \leq 4$ et $t(2) - t(1) \leq -1$. On a donc la même mémoire croisée entre t_1 et t_2 dans les classes C_1 et C_6.

Si on compare les intervalles de tir de t_1 et t_2 dans les deux classes C_1 et C_6, on s'aperçoit que les intervalles de C_1 sont inclus dans ceux de C_6.

Finalement, on peut considérer à partir de cette étude comparative que *la classe C_1 est incluse dans la classe C_6*. Par inclusion, nous voulons dire que tous les états qui constituent la classe C_1 sont des états de la classe C_6 (donc un état du système peut appartenir à plusieurs classes).

Si on regarde sur le graphe des classes d'états, les séquences qui partent de la classe C_3 et qui vont aux classes C_1 et C_6, c'est-à-dire respectivement t_1t_5 et t_5t_1, on pourrait en conclure que ces deux séquences ne sont jamais équivalentes puisqu'elles amènent à deux classes différentes. Or comme C_1 est incluse dans C_6, ces séquences peuvent être équivalentes, mais ce n'est pas visible sur le graphe des classes d'états. Cela peut entrainer une mauvaise interprétation du comportement dynamique du système. [BM82, Men82, BM83] ont proposé un algorithme pour détecter les classes équivalentes, mais aucun travail n'a été mené pour détecter les inclusions de classes, comme également les intersections.

Le problème de l'inclusion pourrait être résolu, dans notre exemple, si on pouvait découper la classe C_6 en deux sous-classes telles que $C_6 = C_1 \cup C_6''$.

III.2.2 Exemple 2

III.2.2.1 Description du réseau de Petri

Le réseau de Petri de la figure III.3 représente deux activités qui s'exécutent en parallèle. Le début de leur exécution est simultané.

La première activité est composée de deux tâches séquentielles : la première tâche, modélisée par la transition t_1, a une durée comprise entre 0 et 1 unités de temps (u.t.). La seconde tâche, modélisée par la transition t_2, a une durée comprise entre 3 et 4 u.t..

La seconde activité est constituée d'une seule tâche, modélisée par la transition t_3, et a une durée comprise entre 5 et 6 u.t..

III.2.2.2 Graphe des classes d'états et critique

Le graphe des classes d'états correspondant à ce Réseau de Petri Temporel est donné sur la figure III.4. Le détail des classes d'états est donné dans la table III.2.

D'après le graphe de la figure III.4, après le tir de t_1, on peut tirer indifféremment t_2 ou t_3. **Une conclusion attive serait de déduire de ce graphe l'inévitabilité de tir**

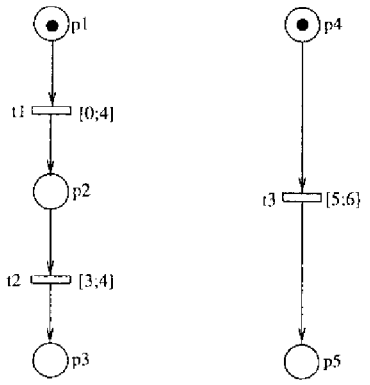


FIG. III.3 – *Modèle Réseau de Petri Temporel*

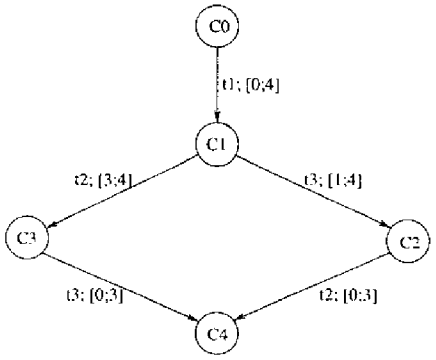


FIG. III.4 – *Graphe des classes d'états*

$$
\begin{array}{|ll|ll|}
\hline
C_0 : & M_0 = p_1 p_4 \quad D_0 = \left\{ \begin{array}{l} 0 \leq t(1) \leq 4 \\ 5 \leq t(3) \leq 6 \end{array} \right. & C_1 : & M_1 = p_2 p_4 \quad D_1 = \left\{ \begin{array}{l} 3 \leq t(2) \leq 4 \\ 1 \leq t(3) \leq 6 \end{array} \right. \\
C_2 : & M_2 = p_2 p_5 \quad D_2 = \left\{ 0 \leq t(2) \leq 3 \right. & C_3 : & M_3 = p_3 p_4 \quad D_3 = \left\{ 0 \leq t(3) \leq 3 \right. \\
C_4 : & M_4 = p_3 p_5 \quad D_4 = \emptyset & & \\
\hline
\end{array}
$$

TAB. III.2 – *Description des classes d'états du graphe de la figure III.4*

de t_2 ou de t_3 après le tir de t_1. [BM82, Men82, BM83, DB91] ont précisé qu'il n'était pas possible, en règle générale, de faire des analyses qualitatives de ce type sur le graphe des classes d'états. Nous allons montrer pourquoi en étudiant plus en détail les états de la classe C_1.

Si on considère le cas particulier du tir de t_1 à la date $\theta_1 = 0$, on obtient un état de C_1 dans lequel la transition t_2 est tirable dans l'intervalle $[3;4]$ et la transition t_3 est tirable dans l'intervalle $[5;6]$. Pour cet état, il n'y a pas potentialité de tir de t_2 ou de t_3, mais inévitabilité de tir de t_2 après le tir de t_1!

De la même manière, si on envisage le tir de t_1 à la date $\theta_1 = 4$, on obtient un état de C_1 dans lequel t_2 est tirable dans l'intervalle $[2;3]$ et t_3 est tirable dans l'intervalle $[1;2]$. Ici encore, il n'y a pas inévitabilité de tir de t_2 ou t_3, mais inévitabilité de tir de t_3!

On peut conclure de cette analyse que la classe C_1 est composée d'états n'autorisant que le tir de t_2, d'états n'autorisant que le tir de t_3, et d'états autorisant le tir de t_2 ou de t_3. Il y a donc trois sous-ensembles d'états, qui proposent des comportements futurs (des tirs de transitions) différents.

Quand, dans une classe, apparaissent plusieurs sous-ensembles de transitions proposant plusieurs comportements futurs différents, nous dirons que la classe fait apparaître une

hétérogénéité de comportements futurs.

Le graphe des classes d'états, par construction, permet de prendre en compte toutes les dates de tir possibles pour les transitions (de la date au plus tôt à la date au plus tard). Il montre toutes les séquences de tirs de transitions possibles (réalisables compte-tenu des attributs temporels des transitions), mais ne permet pas de les analyser. Dans notre exemple et dans le cas du sous-ensemble d'états de C_1 qui ne proposent que le tir de t_2 (nouvellement sensibilisée) comme comportement futur (inévitabilité de tir de t_2), le graphe des classes d'états fait apparaître le tir possible de t_3 (précédemment sensibilisée) au niveau de C_1. Pour ces états de C_1, le tir de cette transition est temporellement impossible. Nous appellerons ce type de comportement du graphe des classes d'états (le tir impossible de t_3 dans ces états) *comportement irréalisable* dans la suite de ce chapitre.

L'hétérogénéité de comportements futurs et l'apparition de comportements irréalisables est une déficience du graphe des classes d'états, puisqu'elle ne nous permet pas de faire des analyses qualitatives, et de tirer des propriétés de type inévitabilité de tir de transition. Une solution à ce problème serait de découper la classe de manière à séparer les sous-ensembles d'états offrant les mêmes comportements futurs. C'est ce que nous étudierons dans la troisième partie de ce chapitre.

III.3 Positionnement des différents graphes d'états probabilisés par rapport au graphe des classes d'états

Dans ce paragraphe, nous nous proposons de montrer, à travers un exemple, comment on peut situer les graphes d'états probabilisés MIN, moyen et MAX dans le graphe des classes d'états, c'est-à-dire intégrer chaque état et chaque arc de ces graphes d'états probabilisés dans le graphe des classes d'états.

III.3.1 Graphe MIN

L'algorithme de positionnement d'un graphe d'états probabilisé dans le graphe des classes d'états est le suivant :

1. placer l'état initial du graphe d'états probabilisé dans la classe initiale du graphe des classes;

2. positionner les autres états dans les différentes classes, en s'appuyant sur les séquences de tir proposées par le graphe d'états probabilisé.

Nous illustrons cet algorithme par le positionnement du graphe d'états probabilisé MIN (figure III.5, qui correspond à la configuration temporelle donnée dans la table III.3) de l'exemple 1 traité en début de ce chapitre (figure III.1), dans le graphe des classes d'états de la figure III.2.

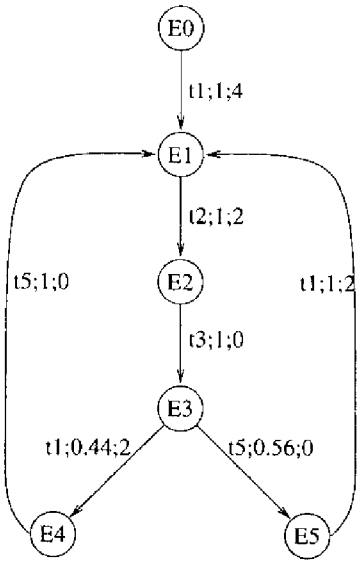


FIG. III.5 – *Graphe d'états probabilisé MIN de l'exemple de la figure III.2*

Transition	Densité de probabilité associée
t_1	$f_1(x) = \frac{3}{4}\delta(x-4) + \frac{1}{8}$ pour $4 \leq x \leq 6$
t_2	$f_2(x) = 1$ pour $2 \leq x \leq 3$
t_3	$f_3(x) = \delta(x)$
t_4	$f_4(x) = \delta(x)$
t_5	$f_5(x) = \frac{1}{4}$ pour $0 \leq x \leq 4$

TAB. III.3 – *Attributs temporels des transitions de l'exemple de la figure III.1*

Remarque importante : dans les états des graphes d'états probabilisés, il n'y a pas de mémoire croisée, puisque le temps s'écoule de la même manière pour toutes les transitions.

Le point de départ du graphe d'états probabilisé et du graphe des classes d'états est le même : l'état initial (la classe d'états initiale n'est composée que de cet état), qui s'écrit :

$$E_0 : \left[\begin{array}{l} M_0 = p_1 p_4 p_6 \\ I_0 = \left\{ \; t_1 : 4 \leq x \leq 6 \right. \\ F_0 = \left\{ \; f_1(x) = \frac{3}{4}\delta(x-4) + \frac{1}{8} \right. \end{array} \right.$$

Dans cet état, seule la transition t_1 est tirable. C'est aussi la seule possibilité de tir qui apparaît depuis la classe C_0 sur le graphe des classes d'états, vers la classe C_1. L'état E_1 résultant du tir de t_1 à la date au plus tôt $\theta_1 = 4$ s'écrit :

$$E_1 : \left[\begin{array}{l} M_1 = p_1 p_2 p_4 p_6 \\ I_1 = \left\{ \begin{array}{l} t_1 : 4 \leq x \leq 6 \\ t_2 : 2 \leq x \leq 3 \end{array} \right. \\ F_1 = \left\{ \begin{array}{l} f_1(x) = \frac{3}{4}\delta(x-4) + \frac{1}{8} \\ f_2(x) = 1 \end{array} \right. \end{array} \right.$$

L'état E_1 fait partie de la classe C_1, puisque $I_1 \subset D_1$. De cet état, on ne peut tirer que la transition t_2. La date de tir au plus tôt est $\theta_2 = 2$, ce qui nous amène à l'état E_2 :

$$E_2 : \left[\begin{array}{l} M_2 = p_1 p_3 p_4 p_6 \\ I_2 = \left\{ \begin{array}{l} t_1 : 2 \leq x \leq 4 \\ t_3 : 0 \leq x \leq 0 \end{array} \right. \\ F_2 = \left\{ \begin{array}{l} f_1(x) = \frac{3}{4}\delta(x-2) + \frac{1}{8} \\ f_3(x) = \delta(x) \end{array} \right. \end{array} \right.$$

Cet état fait partie de la classe C_2 ($I_2 \subset D_2$), atteinte depuis C_1 par le tir de t_2.

Dans l'état E_2, seule la transition t_3 est tirable. La date de tir au plus tôt est $\theta_3 = 0$, ce qui amène à l'état E_3 :

$$E_3 : \left[\begin{array}{l} M_3 = p_1 p_5 p_6 \\ I_3 = \left\{ \begin{array}{l} t_1 : 2 \leq x \leq 4 \\ t_5 : 0 \leq x \leq 4 \end{array} \right. \\ F_3 = \left\{ \begin{array}{l} f_1(x) = \frac{3}{4}\delta(x-2) + \frac{1}{8} \\ f_5(x) = \frac{1}{4} \end{array} \right. \end{array} \right.$$

Cet état fait partie de la classe C_3 ($I_3 \subset D_3$), atteinte depuis C_2 par le tir de t_3.

Dans cet état, deux transitions sont tirables : t_1 à la date (au plus tôt) $\theta_1 = 2$, et t_5 à la date (au plus tôt) $\theta_5 = 0$. Le tir de t_1 amène à l'état E_4, le tir de t_5 à l'état E_5 :

$$E_4 : \left[\begin{array}{l} M_4 = p_1 p_2 p_5 p_6 \\ I_4 = \left\{ \begin{array}{l} t_1 : 4 \leq x \leq 6 \\ t_2 : 2 \leq x \leq 3 \\ t_5 : 0 \leq x \leq 2 \end{array} \right. \\ F_4 = \left\{ \begin{array}{l} f_1(x) = \frac{3}{4}\delta(x-4) + \frac{1}{8} \\ f_2(x) = 1 \\ f_5(x) = \frac{1}{2} \end{array} \right. \end{array} \right. \qquad E_5 : \left[\begin{array}{l} M_5 = p_1 p_4 p_6 \\ I_5 = \left\{ \begin{array}{l} t_1 : 2 \leq x \leq 4 \end{array} \right. \\ F_5 = \left\{ \begin{array}{l} f_1(x) = \frac{3}{4}\delta(x-2) + \frac{1}{8} \end{array} \right. \end{array} \right.$$

Ces états appartiennent respectivement aux classes C_4 ($I_4 \subset D_4$) et C_7 ($I_5 \subset D_7$) du graphe des classes d'états.

Dans l'état E_5, seule t_1 est tirable. Son tir à la date de tir au plus tôt $\theta_1 = 2$ amène à l'état E_2 que nous avons déjà étudié (classe C_2).

Plusieurs remarques sont à faire sur l'état E_4 :

- le tir de t_5 à la date au plus tôt $\theta_5 = 0$ nous amène, dans le graphe d'états probabilisé,

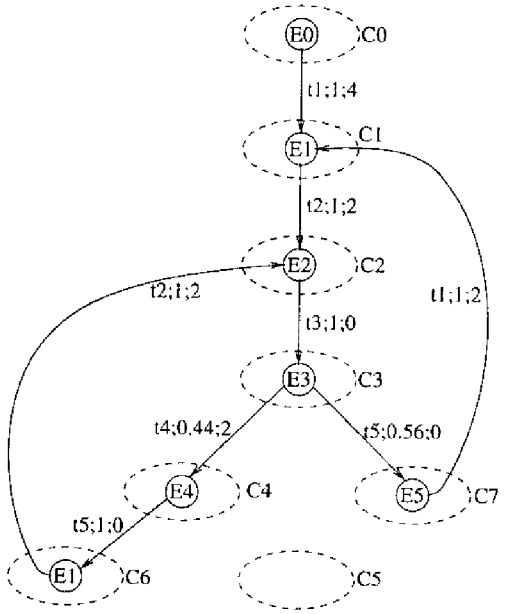


FIG. III.6 – *Graphe MIN positionné dans le graphe des classes d'états*

de l'état E_5 à l'état E_1, que nous avons déjà étudié, et qui appartient à la classe C_1. Or, si l'on regarde le graphe des classes d'états, le tir de la transition t_5 mène de la classe C_4 (à laquelle l'état E_4 appartient) à la classe C_6, et non pas à la classe C_1 (à laquelle l'état E_1 appartient). Cela veut donc dire que l'état E_1 appartient à la fois à la classe C_1 et à la classe C_6 (nous avons déjà vu que la classe C_1 est incluse dans la classe C_6);

- Par définition du graphe des classes d'états, la double appartenance de E_1 à C_1 et C_6 montre qu'il n'est pas évident de positionner directement (sans suivre les séquences de tirs de transitions) les états du graphe d'états probabilisé dans les différentes classes d'états: chaque état peut appartenir à plusieurs classes, et on ne sait pas a priori si on doit le positionner dans une seule, plusieurs ou toutes ces classes. La seule façon de procéder est de suivre les séquences de tirs de transitions, comme nous l'avons fait;

- Pour représenter le fait que l'état E_1 appartient à la fois à la classe C_1 et à la classe C_6, on « duplique » cet état dans les deux classes, de manière à pouvoir positionner les arcs $E_0 \rightarrow E_1$ et $E_4 \rightarrow E_1$ dans le graphe des classes d'états, sans violer la structure de ce dernier. Depuis E_1 (de la classe C_6), un arc (correspondant au tir de t_2 à la date $\theta_2 = 2$) part vers l'état E_2 de la classe C_2. Cet arc est lui aussi dupliqué, puisqu'il existe déjà entre l'état E_1 de la classe C_1 et l'état E_2. Pour pouvoir positionner un graphe d'états probabilisé dans le graphe des classes d'états, on peut donc être amené à dupliquer toute une partie du graphe d'états probabilisé.

La figure III.6 représente le graphe MIN positionné dans la graphe des classes d'états. Aucun état du graphe MIN n'a été positionné dans la classe C_5.

III.3.2 Graphe moyen

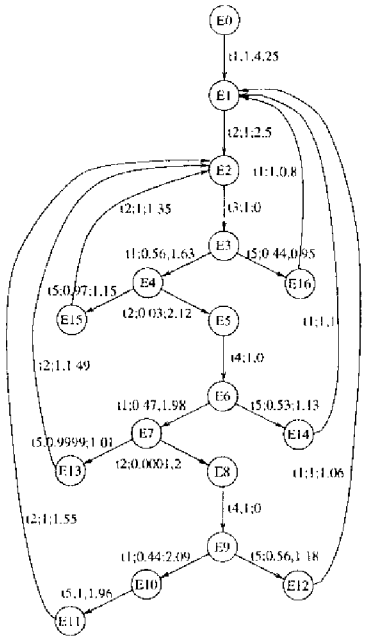


FIG. III.7 – *Graphe d'états probabilisé moyen de l'exemple de la figure III.2*

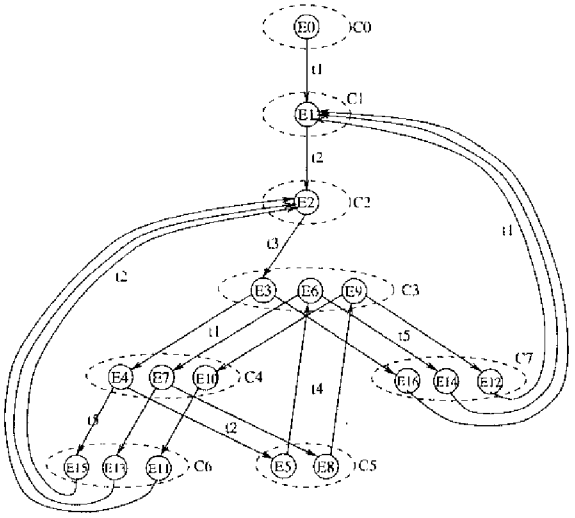


FIG. III.8 – *Graphe moyen positionné dans le graphe des classes d'états*

Le positionnement du graphe moyen (figure III.7) de l'exemple 1 dans le graphe des classes d'états (figure III.2) se fait de la même manière que pour le graphe MIN.

Jusqu'à l'état E_4 (classe C_4), la construction est similaire à celle que nous avons décrit avec le graphe MIN. Dans l'état E_4, les transitions t_2 et t_5 peuvent être tirées, contrairement à l'état E_4 du graphe MIN où seule t_5 était tirable. Le tir de t_5 à la date moyenne $\theta_5 = 1.15$ nous amène dans l'état E_{15} de la classe C_6. Notons qu'ici, l'état E_{15} est différent de l'état E_1, et donc qu'il n'y a pas de duplication de l'état E_1 comme avec le graphe MIN.

Le tir de t_2 à la date moyenne $\theta_2 = 2.12$ nous amène de l'état E_4 à l'état E_5 de la classe C_5 ($I_5 \subset D_5$). Dans ce nouvel état, seule t_4 est tirable. Son tir amène à l'état E_6. Cet état, si on suit la structure du graphe des classes d'états, se situe dans la classe C_3 ($I_6 \subset D_3$), mais est différent de l'état E_3 que nous avons déjà étudié dans cette classe ($I_6 \neq I_3$). Donc nous avons deux états différents du même graphe d'états probabilisé dans une même classe d'états.

Le reste des états du graphe d'états probabilisé moyen peut être intégré aisément dans les classes d'états. Nous avons représenté le positionnement complet du graphe moyen dans le graphe des classes d'états sur la figure III.8 (dans un soucis de clarté, nous n'avons pas

fait apparaître les étiquettes quantitatives sur les arcs entre états). On voit apparaître sur cette figure la propriété essentielle du graphe des classes d'états (c'est dans ce but qil a été construit) qui permet de représenter par une seule séquence de tir (par exemple $t_1t_2t_4$ entre les classes C_3, C_4 et C_5) un comportement répétitif du système ($t_1t_2t_4$ se répète deux fois dans le graphe moyen de l'état E_3 à l'état E_9). Cette propriété lui donne un pouvoir d'abstraction dans la représentation du comportement d'un système. Notons qu'ici, il y a au moins un état du graphe d'états probabilisé dans chaque classe d'états.

III.3.3 Graphe MAX

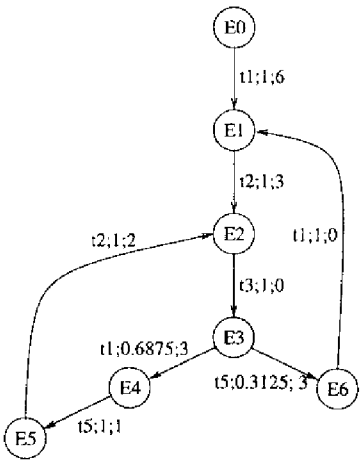


FIG. III.9 – *Graphe d'états probabilisé MAX de l'exemple de la figure III.2*

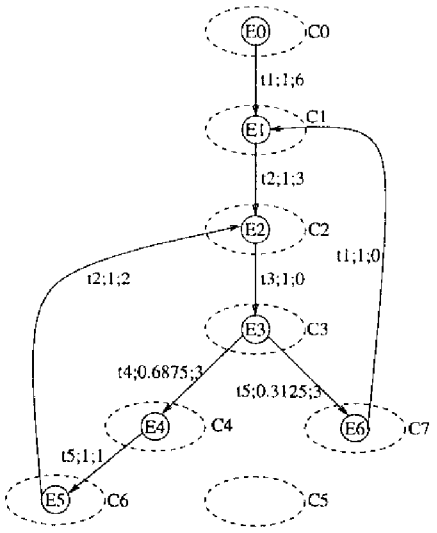


FIG. III.10 – *Graphe MAX positionné dans le graphe des classes d'états*

Le positionnement du graphe MAX (figure III.9) dans le graphe des classes d'états est présenté sur la figure III.10. Nous ne développons pas cette construction qui est strictement similaire aux deux positionnements déjà étudiés. Il n'y a aucun état de ce graphe dans la classe C_5.

III.3.4 Conclusion

En résumé, le positionnement d'un graphe d'états probabilisé dans le graphe des classes d'états peut nous amener à :

- dupliquer des états dans plusieurs classes pour respecter les séquences de tirs de transitions du graphe des classes d'états,
- positionner plusieurs états dans des classes, ce qui peut indiquer que le graphe d'états probabilisé possède des séquences répétitives de tirs de transitions.

De plus, on constate que des graphes d'états probabilisés ne passent pas par toutes les classes. Ceci découle logiquement du fait qu'un graphe d'états probabilisé décrit un

scénario temporel particulier et donc certaines séquences de tir, visualisées au moyen du graphe des classes d'états, ne sont pas possibles dans le scénario temporel considéré. Notons cependant que le positionnement des trois graphes d'états probabilisés MIN, moyen et MAX au sein du graphe des classes d'états donne des informations qualitatives et quantitatives intéressantes.

III.4 Concept de graphe des sous-classes d'états

III.4.1 Idée principale et hypothèse de travail

Le graphe des sous-classes d'états est un raffinement du graphe des classes d'états. Il est obtenu en découpant les classes faisant apparaître une hétérogénéité de comportements futurs en autant de sous-classes qu'il y a de sous-ensembles d'états (qui ont des comportements futurs différents) dans ces classes. Par exemple, la classe C_1 de l'exemple 2 du début de ce chapitre sera découpée en trois sous-classes d'états, puisqu'elle est composée de trois sous-ensembles d'états (voir paragraphe III.2.2.2).

Ce découpage ayant été fait, les comportements irréalisables du graphe des classes d'états sont éliminés, puisque les sous-ensembles d'états sont caractérisés par l'homogénéité des comportements futurs qu'ils proposent.

Nous avons vu que les comportements irréalisables d'une classe C_i sont issus du masquage d'impossibilités de tirs de transitions précédemment sensibilisées et nouvellement sensibilisées. Dans la suite de ce paragraphe, nous allons supposer que les transitions précédemment sensibilisées ont été seulement sensibilisées dans la classe d'états C_{i-1} qui précède la classe C_i, c'est-à-dire de la classe depuis laquelle, par le tir d'une transition t_i, on arrive dans la classe C_i (le chemin entre C_{i-1} et C_i ne comprend qu'un tir de transition). Cette hypothèse de travail, que nous appellerons *hypothèse de parallélisme de profondeur (d'ordre) 1*, sera considérée et examinée après avoir étudié l'algorithme de construction du graphe des sous-classes d'états, pour pouvoir étendre cet algorithme au cas général.

III.4.2 Principe de construction du graphe des sous-classes d'états

Nous allons dans un premier temps construire le graphe des sous-classes d'états de l'exemple 2 de ce chapitre (figure III.3). Puis nous donnerons l'algorithme (basé sur l'algorithme de construction du graphe des classes d'états donné dans [BM82]) de cette construction.

Le graphe des sous-classes d'états est construit de manière similaire au graphe des classes d'états. Seule la procédure de calcul de la classe d'états résultant du tir d'une transition est modifiée. Reprenons le calcul des classes du graphe des classes d'états de l'exemple de la figure III.3, et étudions l'obtention de la classe C_1, issue du tir de la transition t_1 depuis la classe C_0 :

$$C_0 : \left[\begin{array}{l} M_0 = p_1 p_4 \\ D_0 = \left\{ \begin{array}{l} 0 \leq t(1) \leq 4 \\ 5 \leq t(3) \leq 6 \end{array} \right. \end{array} \right.$$

Soit le changement de variable $t(3) = t''(3) + t(1)$. On a :

$$\left| \begin{array}{c} 0 \leq t(1) \leq 4 \\ 5 - t(1) \leq t''(3) \leq 6 - t(1) \end{array} \right.$$

Le tir de t_1 sensibilise nouvellement la transition t_2. Dans son algorithme, [BM82] introduit l'intervalle de tir de t_2 de manière indépendante des transitions précédemment sensibilisées : il fait d'abord « vieillir » les transitions qui restent sensibilisées (ici t_3), puis rajoute le domaine de tir des transitions nouvellement sensibilisées (ici t_2). Nous nous proposons d'introduire ces domaines de tir (ici, celui de t_2) dès à présent dans le calcul :

$$\left| \begin{array}{c} 0 \leq t(1) \leq 4 \\ 5 - t(1) \leq t(3) \leq 6 - t(1) \\ 3 \leq t(2) \leq 4 \end{array} \right.$$

Dans l'algorithme de [BM82], on donne à $t(1)$ les valeurs extrêmes de son domaine de tir, et on en déduit le nouvel intervalle de tir de t_3 (on a ainsi pris en compte toutes les dates de tir possibles). Ici, nous allons appliquer dans un premier temps une recherche des différents comportements futurs possibles à travers l'étude de la compétition, après le tir de t_1, entre t_2 et t_3. Nous généraliserons en même temps cette étude à la compétition entre une transitions t_{ns} nouvellement sensibilisée, tirable dans l'intervalle $[\alpha_{ns}; \beta_{ns}]$, et une transition t_{as} anciennement sensibilisée (en accord avec notre hypothèse de travail), tirable dans l'intervalle de tir $[\alpha_{as} - t(i); \beta_{as} - t(i)]$, où $t(i)$ est la date de tir de la transition que l'on étudie ($t(1)$ dans notre exemple).

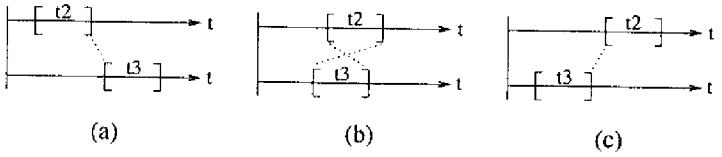


FIG. III.11 – *Différents types de compétition entre t_2 et t_3*

La figure III.11 nous montre les trois types de compétition différents que l'on peut obtenir entre les transitions t_2 et t_3 ([ALL83] a recensé 13 cas différents) :

- t_2 sera tirée obligatoirement avant t_3 (figure III.11 (a)), si la borne supérieure de son intervalle de tir est strictement plus petite que la borne inférieure de l'intervalle de tir de t_3 :

  $$4 < 5 - t(1) \Rightarrow t(1) < 1$$

  Pour les transitions t_{as} et t_{ns}, t_{ns} sera tirée obligatoirement avant t_{as} si :

  $$\beta_{ns} < \alpha_{as} - t(i) \Rightarrow t(i) < \alpha_{as} - \beta_{ns}$$

- t_3 sera tirée obligatoirement avant t_2 (figure III.11 (c)) si la borne supérieure de son intervalle de tir est strictement plus petite que la borne inférieure de l'intervalle de tir de t_2 :

  $$6 - t(1) < 3 \Rightarrow t(1) > 3$$

Pour les transitions t_{as} et t_{ns}, t_{as} sera tirée obligatoirement avant t_{ns} si :

$$\beta_{as} - t(i) < \alpha_{ns} \Rightarrow t(i) > \beta_{as} - \alpha_{ns}$$

- t_2 et t_3 seront tirables en même temps (figure III.11 (b)) si la borne inférieure de l'intervalle de tir de t_3 est plus petite ou égale à la borne supérieure de l'intervalle de tir de t_2, et si la borne inférieure de l'intervalle de tir de t_2 est plus petite ou égale à la borne supérieure de l'intervalle de tir de t_3 :

$$\left| \begin{array}{l} 5 - t(1) \leq 4 \\ 3 \leq 6 - t(1) \end{array} \right\} \Rightarrow 1 \leq t(1) \leq 3$$

Pour les transitions t_{as} et t_{ns}, cette condition s'écrit :

$$\left| \begin{array}{l} \alpha_{as} - t(i) \leq \beta_{ns} \\ \alpha_{ns} \leq \beta_{as} - t(i) \end{array} \right\} \Rightarrow \alpha_{as} - \beta_{ns} \leq t(i) \leq \beta_{as} - \alpha_{ns}$$

L'étude de ces trois compétitions différentes nous a permis d'obtenir des contraintes supplémentaires sur l'intervalle de tir de t_1. Si on ajoute à ces contraintes l'intervalle de tir de t_1, on obtient les sous-intervalles de cet intervalle permettant d'obtenir les trois comportements futurs du système :

- $t(1) \in [0;1[$: t_2 tirable obligatoirement avant t_3,
- $t(1) \in [1;3]$: t_2 et t_3 tirables en même temps,
- $t(1) \in]3;4]$: t_3 tirable obligatoirement avant t_2.

Si on applique alors le calcul de classe de [BM82] en prenant pour intervalle de tir de t_1 les trois intervalles que l'on vient de calculer, on va obtenir les trois sous-classes d'états de la classe C_1.

Avant de donner la description de ces sous-classes, nous devons résoudre une difficulté qui est l'apparition d'intervalles ouverts comme intervalles de tir ($t(1) \in [0;1[$ et $t(1) \in]3;4]$). La définition du calcul des classes d'états a été donnée dans [BM82, DB91] uniquement pour des intervalles fermés, et pour des bornes rationnelles (condition suffisante pour obtenir un graphe fini). Donc, on ne peut pas utiliser directement les intervalles que l'on a trouvés dans un calcul de classe d'états.

Le côté ouvert d'un intervalle indique une limite (à gauche ou à droite) qui tend vers une borne.

Notations : Une limite à droite d'une valeur A sera notée A^+. Une limite à gauche d'une valeur A sera notée A^-. On notera un intervalle ouvert du type $[A;B[$ ou $]A;B]$ respectivement sous la forme $[A;B^-]$ ou $[A^+;B]$. Nous appellerons A^+ et B^- dates limites de l'intervalle.

Cette nouvelle notation nous permet de travailler avec des intervalles fermés. Mais elle ne donne pas des bornes d'intervalle rationnelles, ce qui ne permet pas de construire le

graphe des classes d'états données par [BM82, DB91] (qui serait alors infini). Pour pallier à ce problème, on peut dire que lorsqu'on calcule, dans notre exemple, la sous-classe d'états issue du tir de t_1 dans l'intervalle $[0;1^-]$, tout se passe comme si on calculait la sous-classe issue du tir de t_1 dans l'intervalle $[0;1]$ (intervalle fermé avec bornes rationnelles, donc en accord avec les hypothèses de calcul des classes d'états), *et que l'on enlevait à cette sous-classe l'état qui correspond au tir de* t_1 *à la date* $t(1) = 1$. Au niveau de la description du domaine de définition des sous-classes, l'exclusion de cet état sera effectuée en utilisant les notations A^- et A^+.

Avec ces nouvelles notations, les intervalles de tir de t_1 s'écrivent donc :

- $t(1) \in [0;1^-]$: t_2 tirable obligatoirement avant t_3,
- $t(1) \in [1;3]$: t_2 et t_3 tirables en même temps,
- $t(1) \in [3^+;4]$: t_3 tirable obligatoirement avant t_2.

En généralisant aux transitions t_{as} et t_{ns}, on a :

- $t(i) \in [\alpha_i;(\alpha_{as} - \beta_{ns})^-]$: t_{ns} tirable obligatoirement avant t_{as},
- $t(i) \in [\alpha_{as} - \beta_{ns}, \beta_{as} - \alpha_{ns}]$: t_{as} et t_{ns} tirables en même temps,
- $t(i) \in [(\beta_{as} - \alpha_{ns})^+;DIT]$: t_{as} tirable obligatoirement avant t_{ns}.

où α_i est la date de tir au plus tôt de t_i, et DIT sa date de tir au plus tard.

La définition d'une sous-classe d'états est la même que celle d'une classe d'états. Une sous-classe est caractérisée par le marquage du réseau de Petri sous-jacent, et par le domaine temporel des transitions sensibilisées. La notion de mémoire temporelle est strictement identique à celle donnée dans [BM82, DB91], puisqu'elle n'intervient qu'entre transitions précédemment sensibilisées (elle ne dépend pas des transitions nouvellement sensibilisées, donc n'est pas modifiée ici).

Depuis C_0 le calcul de la sous-classe issue du tir de t_1 dans l'intervalle $[0;1^-]$ s'effectue comme suit :

$$\left| \begin{array}{c} 0 \leq t(1) \leq 1^- \\ 5 - t(1) \leq t(3) \leq 6 - t(1) \\ 3 \leq t(2) \leq 4 \end{array} \right.$$

Si on donne à $t(1)$ sa valeur minimale 0, on obtient pour $t(3)$: $5 \leq t(3) \leq 6$; Si on donne à $t(1)$ sa valeur maximale 1^-, on obtient pour $t(3)$:

$$5 - 1^- \leq t(3) \leq 6 - 1^- \Leftrightarrow 4^+ \leq t(3) \leq 5^+$$

En prenant la plus petite des bornes inférieures et la plus grande des bornes supérieures de ces deux intervalles, on peut donner la description de la sous-classe d'états :

$$C_{1.1} : \begin{bmatrix} M_1 = p_2 p_4 \\ D_{1.1} = \begin{cases} 3 \leq t(2) \leq 4 \\ 4^+ \leq t(3) \leq 6 \end{cases} \end{bmatrix}$$

La borne inférieure 4^+ de l'intervalle de tir de $t(3)$ notifie l'exclusion de l'état issu du tir de $t(1)$ à la date 1.

Remarque: on peut utiliser les notations A^+ et A^- dans les opérations d'addition et de soustraction. Le résultat de ces opérations doit être en accord avec le fondement de ces notations, à savoir la suppression d'une valeur d'un intervalle. Ainsi, par exemple, l'opération $5 - 1^-$ est égale à une des deux limites de la valeur $5 - 1 = 4$. Comme on a soustrait une limite à gauche, il est naturel d'obtenir une limite à droite: 4^+.

Nous donnons ci-dessous l'algorithme général du calcul du domaine de tir d'une sous-classes d'état. Le reste de l'algorithme de construction du graphe des classes d'états est identique à celui proposé dans [DB91].

III.4.3 Algorithme du calcul d'une sous-classe d'états

Cet algorithme est un algorithme généré dans l'hypothèse uniquement d'un parallélisme de profondeur 1 (on a dans une classe un nombre quelconque de transitions nouvellement et précédemment sensibilisées).

Soit le domaine de tir D et le marquage M d'une sous-classe d'états SC; D peut s'écrire sous la forme:

$$\begin{vmatrix} \alpha_i \leq t(i) \leq \beta_i \\ t(j) - t(k) \leq \gamma_{jk} \end{vmatrix}$$

On suppose que l'on tire la transition t_f dans l'intervalle $[\alpha_f; DIT_{SC}]$.

Soit NS, l'ensemble des transitions nouvellement sensibilisées par le tir de t_f depuis SC (non sensibilisées par $M - Pre(\bullet, t_f)$ et sensibilisées par $M - Pre(\bullet, t_f) + Post(\bullet, t_f)$).

Soit AS, l'ensemble des transitions qui restent sensibilisées lors du tir de t_f depuis SC (sensibilisées par $M - Pre(\bullet, t_f)$ et sensibilisées par $M - Pre(\bullet, t_f) + Post(\bullet, t_f)$).

Soit DL, l'ensemble des dates de tir limites de t_f; Initialement, $DL = \{\alpha_f, DIT_{SC}\}$.

- *Étape 1 : calcul des intervalles de tir de t_f*

$$\forall (t_{as}, t_{ns}) \in AS \times NS, DL \longleftarrow DL \cup \{(\alpha_{as} - \beta^s_{ns})^-, \alpha_{as} - \beta^s_{ns}, \beta_{as} - \alpha^s_{ns}, (\beta_{as} - \alpha^s_{ns})^+\}$$

La transition nouvellement sensibilisée par le tir de t_f a pour attributs temporels son intervalle statique $[\alpha^s_{ns}; \beta^s_{ns}]$; la transition précédemment sensibilisée a pour attributs temporels dans la classe SC l'intervalle $[\alpha_{as}; \beta_{as}]$;

On suppose que l'ensemble DL est ordonné par dates croissantes. On cherche la date α_f dans cet ensemble. A partir de cette date, tout couple $(\theta_{inf}, \theta_{sup})$ de dates successives de DL définit un intervalle de tir possible de t_f, et ce jusqu'à ce qu'on ait $\theta_{sup} = DIT_{SC}$. Soit CI l'ensemble de ces intervalles.

• *Étape 2 : calcul de la sous-classe résultante du tir de t_f dans un intervalle $I \in CI$* (issu de [DB91])

Pour tout intervalle $I = [\alpha^i_f, \beta^i_f] \in CI$, le domaine de tir D de SC devient :
(a) vieillissement des transitions :

(OP1)
- α_i devient $\max(0, -\gamma_{fi}, \alpha_i - \beta^i_f)$
- β_i devient $\min(\gamma_{if}, \beta_i - \alpha^i_f)$
- γ_{jk} devient $\min(\gamma_{jk}, \beta_j - \alpha_k)$

(b) élimination des transitions désensibilisées :

(OP2)
- α_i devient $\max(0, -\gamma_{fi}, \alpha_i - \beta^i_f)$
- β_i devient $\min(\gamma_{if}, \beta_i - \alpha^i_f)$
- γ_{jk} devient $\min(\gamma_{jk}, \beta_j - \alpha_k)$

(c) ajout des transitions nouvellement sensibilisées :

(OP3) de nouvelles variables sont ajoutées au domaine de tir, telles que
$\alpha^s_k \leq t(k) \leq \beta^s_k$
et que
$\forall i \neq k,\ t(i) - t(k) \leq \gamma_{nk}$ avec $\gamma_{nk} = \beta_i - \alpha^s_k$
$\forall j \neq k,\ t(k) - t(j) \leq \gamma_{kj}$ avec $\gamma_{kj} = \beta^s_k - \alpha_j$

III.4.4 Exemple de graphe des sous-classes d'états

La table III.4 donne la description des différentes sous-classes d'états obtenues en appliquant cet algorithme à l'exemple de la figure III.3. La figure III.12 montre le graphe des sous-classes d'états correspondant (notons qu'ici nous n'avons pas de mémoire croisée, car il n'y a jamais, dans un marquage, au moins deux transitions anciennement sensibilisées). La table III.4 donne la description des sous-classes.

III.4.5 Levée de l'hypothèse de travail

Notre hypothèse de travail était de considérer des transitions précédemment sensibilisées et nouvellement sensibilisées telles que le parallélisme entre elles soit d'une profondeur maximale de 1. Cette hypothèse restrictive est la limite principale de notre algorithme. Nous allons étudier dans ce paragraphe la levée de cette hypothèse, pour étendre notre

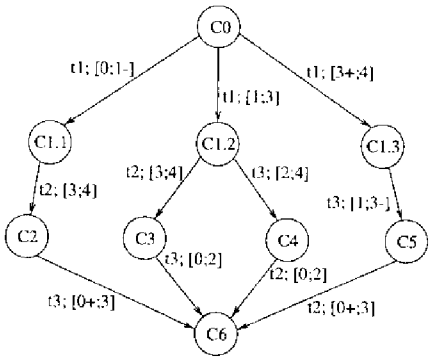


FIG. III.12 – *Graphe des sous-classes d'états*

$$
\begin{array}{|ll|ll|}
\hline
C_0 : & \left[\begin{array}{l} M_0 = p_1 p_4 \\ D_0 = \left\{\begin{array}{l} 0 \le t(1) \le 4 \\ 5 \le t(3) \le 6 \end{array}\right. \end{array}\right. & C_{1.1} : & \left[\begin{array}{l} M_{1.1} = p_2 p_4 \\ D_{1.1} = \left\{\begin{array}{l} 3 \le t(2) \le 4 \\ 4^+ \le t(3) \le 6 \end{array}\right. \end{array}\right. \\
C_{1.2} : & \left[\begin{array}{l} M_{1.2} = p_2 p_4 \\ D_{1.2} = \left\{\begin{array}{l} 3 \le t(2) \le 4 \\ 2 \le t(3) \le 5 \end{array}\right. \end{array}\right. & C_{1.3} : & \left[\begin{array}{l} M_{1.3} = p_2 p_4 \\ D_{1.3} = \left\{\begin{array}{l} 3 \le t(2) \le 4 \\ 1 \le t(3) \le 3^- \end{array}\right. \end{array}\right. \\
C_2 : & \left[\begin{array}{l} M_2 = p_3 p_4 \\ D_2 = \left\{\begin{array}{l} 0^+ \le t(3) \le 3 \end{array}\right. \end{array}\right. & C_3 : & \left[\begin{array}{l} M_3 = p_3 p_4 \\ D_3 = \left\{\begin{array}{l} 0 \le t(3) \le 2 \end{array}\right. \end{array}\right. \\
C_4 : & \left[\begin{array}{l} M_4 = p_2 p_5 \\ D_4 = \left\{\begin{array}{l} 0 \le t(2) \le 2 \end{array}\right. \end{array}\right. & C_5 : & \left[\begin{array}{l} M_5 = p_2 p_5 \\ D_5 = \left\{\begin{array}{l} 0^+ \le t(2) \le 3 \end{array}\right. \end{array}\right. \\
C_6 : & \left[\begin{array}{l} M_6 = p_4 p_5 \\ D_6 = \emptyset \end{array}\right. & & \\
\hline
\end{array}
$$

TAB. III.4 – *Description des sous-classes d'états du graphe de la figure III.12*

algorithme à des cas de parallélisme de profondeur supérieure à 1.

Dans l'hypothèse d'une profondeur de parallélisme de 1, l'algorithme que nous avons donné examine la dernière transition tirée pour savoir si son tir peut faire apparaître une hétérogénéité de comportements futurs. Si c'est le cas, il découpe l'intervalle de tir de cette transition pour faire disparaître cette hétérogénéité.

Pour un parallélisme de profondeur supérieure à 1, il faut que l'algorithme soit capable :

- de détecter une hétérogénéité de comportements futurs dans une classe C_j,
- de remonter dans le passé du système pour trouver le tir de la ou les transitions qui provoquent cette hétérogénéité.

III.4.6 Détection d'une hétérogénéité de comportements futurs

La détection d'une hétérogénéité de comportement dans la classe C_j passe par l'étude de la compétition entre chaque couple de transitions (t_{as}, t_{ns}) de C_j. On suppose que

la transition anciennement sensibilisée t_{as} est sensibilisée depuis la classe C_i, et que la séquence de tir $t_1 t_2 \ldots t_n$ permet de passer de la classe C_i à la classe C_j sans désensibiliser t_{as}.

On pose : $[\alpha_{as}, \beta_{as}]$ est l'intervalle de tir de la transition t_{as} dans C_j, $[\alpha^s_{as}, \beta^s_{as}]$ est l'intervalle de tir statique de la transition t_{as} (l'intervalle de tir qui lui est affectée lors de sa sensibilisation dans C_i), et $[\alpha^s_{ns}, \beta^s_{ns}]$ est l'intervalle de tir statique de la transition t_{ns} dans C_j. Ces intervalles sont donnés par la description des classes d'états du graphe des classes d'états. Il est donc nécessaire, avant de chercher les hétérogénéités de comportements futurs dans la classe C_j, de construire cette classe (avec l'algorithme de [BM82]), pour avoir ces valeurs.

Dans un premier temps, nous devons calculer l'intervalle des valeurs possibles pour la durée pendant laquelle t_{as} reste sensibilisée entre C_i et C_j, durée égale à $\theta_1 + \theta_2 + \cdots + \theta_n = \sum_1^n \theta_i$, où les θ_i sont les dates de tir des transitions t_i de la séquence de tir $t_1 t_2 \ldots t_n$. La valeur minimale de cette durée est $d_{min} = \min(\sum_1^n \theta_i)$. La valeur maximale de cette durée est $d_{max} = max(\sum_1^n \theta_i)$.

Dans C_j, l'intervalle de tir de t_{as} est l'union des intervalles de tir de tous les états issus du tir de $t_1 t_2 \ldots t_n$ depuis C_i ($\sum_1^n \theta_i \in [d_{min}; d_{max}]$). α_{as} est donc la plus petite valeur de $\alpha^s_{as} - \sum_1^n \theta_i$, et β_{as} la plus grande valeur de $\beta^s_{as} - \sum_1^n \theta_i$. On a donc :

$$\alpha_{as} = \alpha^s_{as} - max(\sum_1^n \theta_i) \Rightarrow d_{max} = max(\sum_1^n \theta_i) = \alpha^s_{as} - \alpha_{as}$$

$$\beta_{as} = \beta^s_{as} - min(\sum_1^n \theta_i) \Rightarrow d_{min} = min(\sum_1^n \theta_i) = \beta^s_{as} - \beta_{as}$$

On peut donc écrire que $\sum_1^n \theta_i \in [\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$. Cet intervalle donne l'ensemble des valeurs possibles pour la durée du chemin entre C_i et C_j.

Sur la base de l'étude de la compétition entre les transitions t_2 et t_3 du paragraphe précédent, on peut dire que :

- si, quelle que soit la durée du chemin entre C_i et C_j, la borne supérieure de l'intervalle de tir de t_{as} est inférieure à la borne inférieure de l'intervalle de tir de t_{ns} ($\beta^s_{as} - \sum_1^n \theta_i < \alpha^s_{ns} \Leftrightarrow \sum_1^n \theta_i > \beta^s_{as} - \alpha^s_{ns}$), alors t_{as} sera toujours tirée avant t_{ns}. Si $\beta^s_{as} - \alpha^s_{ns} \in [\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$, c'est-à-dire si une partie des états de C_j n'autorise que le tir de t_{as} (états issus d'une durée du chemin entre C_i et C_j comprise dans l'intervalle $]\beta^s_{as} - \alpha^s_{ns}; \alpha^s_{as} - \alpha_{as}]$, et une autre partie des états autorise le tir des deux transitions (états issus d'une durée du chemin entre C_i et C_j comprise dans l'intervalle $[\beta^s_{as} - \beta_{as}; \beta^s_{as} - \alpha^s_{ns}]$, il y a hétérogénéité dans la classe C_j;
- si, quelle que soit la durée du chemin entre C_i et C_j, la borne inférieure de l'intervalle de tir de t_{as} est supérieure à la borne supérieure de l'intervalle de tir de t_{ns} ($\alpha^s_{as} - \sum_1^n \theta_i > \beta^s_{ns} \Leftrightarrow \sum_1^n \theta_i < \alpha^s_{as} - \beta^s_{ns}$), alors t_{ns} sera toujours tirée avant t_{as}. De la même manière que pour le point précédent, si $\alpha^s_{as} - \beta^s_{ns} \in [\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$, il y a hétérogénéité dans la classe C_j.

On tire de cette étude la condition qui permet de détecter une hétérogénéité comportementale dans C_j :

si
$(\beta^s_{as} - \alpha^s_{ns}) \in [\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$
ou si
$(\alpha^s_{as} - \beta^s_{ns}) \in [\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$
alors

il y a hétérogénéité de comportements futurs dans la classe C_j due à la compétition entre les deux transitions t_{as} et t_{ns}.

III.4.7 Découpage en sous-classes d'états

Supposons que la classe C_j fasse apparaître une hétérogénéité de comportements futurs due à la compétition entre les transitions t_{as} et t_{ns}. Pour découper cette classe en sous-classes d'états, il faut trouver les valeurs limites de la durée $d = \theta_1 + \theta_2 + \cdots + \theta_n = \sum_1^n \theta_i$ du chemin $t_1 t_2 \ldots t_n$ qui provoquent les changements de comportements futurs. En reprenant les équations du paragraphe précédent, on peut écrire que ces valeurs limites sont telles que $d_1 = \beta^s_{as} - \alpha^s_{ns}$ et $d_2 = \alpha^s_{as} - \beta^s_{ns}$. La durée d étant comprise dans l'intervalle $[\beta^s_{as} - \beta_{as}; \alpha^s_{as} - \alpha_{as}]$, on peut écrire, par exemple, que t_{ns} sera toujours tirée avant t_{as} si

$$\beta^s_{as} - \beta_{as} \leq \theta_1 + \theta_2 + \cdots + \theta_n < \alpha^s_{as} - \beta^s_{ns}$$

ce qui nous donne deux relations dynamiques entre les dates de tir θ_i. En rajoutant les différentes contraintes sur ces dates de tir θ_i, contraintes issues des classes d'états entre C_i et C_j sur le chemin étudié (2n relations statiques), on voit que l'on obtient un système à $2(n+1)$ inéquations. Il est impossible de tirer de ce système des intervalles statiques (bornes fixes) à affecter à chaque date de tir θ_i, qui permette de caractériser l'ensemble des états de C_j qui n'autorise que le tir de t_{ns}. En effet, il faudrait pour cela que nous ayons, en plus des 2n relations statiques, 2n relations dynamiques pour pouvoir caractériser une solution du système d'inéquations obtenu.

En conclusion, pour un parallélisme de profondeur supérieure à 1, il est impossible dans le cas général de construire le graphe des sous-classes d'états avec notre méthode. On ne peut que détecter les classes d'états qui font apparaître une hétérogénéité de comportements futurs.

III.5 Conclusion

Trois points intéressants, à notre avis, peuvent être retirés de la réflexion effectuée dans ce chapitre :

- le graphe des classes d'états, par construction, peut faire apparaître des hétérogénéité de comportements futurs dans les classes d'états. Ceci ne permet pas, comme cela a été souligné dans [BM82, Men82, BM83, DB91], de déduire des propriétés qualitatives;

- les graphes d'états probabilisés (MIN, moyen, MAX) peuvent être positionnés au sein du graphe des classes d'états, ce qui permet d'offrir des points de vue quantitatifs de certains scénarios;

- le concept de graphe des sous-classes d'états définit, dans l'hypothèse de Réseaux de Petri avec ce que nous appelons un parallélisme d'ordre 1 uniquement, un objet qui ne présente plus d'hétérogénéité comportementale, et qui assure donc de l'inévitabilité temporelle du tir de toutes les transitions sortantes des sous-classes.

Une poursuite de ces réflexions devrait être faite, en particulier au niveau du graphe des sous-classes d'états, afin de l'exploiter pour obtenir simplement des informations quantitatives des chemins minimum, moyen et maximum.

Chapitre IV

Extension du pouvoir d'analyse du modèle Réseaux de Petri Temporisés Stochastiques

IV.1 Introduction

Le graphe d'états probabilisé est un graphe qui possède sur ses transitions, à la fois, des étiquettes qualitatives (transitions du modèle Réseaux de Petri Temporisés Stochastiques, et parfois les actions associées, comme l'envoi d'un message (*!message*) et la réception d'un message (*?message*)) et des étiquettes quantitatives (les probabilités de tir des transitions et les durées écoulées entre les sensibilisations et les tirs).

Nous avons indiqué dans le premier chapitre que ce graphe était utilisé, en particulier :

1. pour obtenir des vues abstraites qualitatives (projection sur un sous-ensemble d'étiquettes qualitatives (événements) et obtention d'un automate quotient sur la base de la relation d'équivalence observationnelle),
2. pour obtenir des vues abstraites quantitatives (projection sur un sous-ensemble d'états et obtention d'un graphe réduit à ces seuls états, les étiquettes quantitatives associées à ce graphe réduit dépendant des passages dans les états non observés).

Les vues abstraites sont des objets très intéressants pour l'analyse des points de vue, très important pour l'utilisateur.

Cependant, ces deux techniques de vues abstraites offrent des points de vue disjoints, et donc il est apparu intéressant d'utiliser ces deux techniques pour obtenir un objet donnant des informations qualitatives et quantitatives. Plus précisément, l'automate quotient obtenu par une projection sur un sous-ensemble d'événements (vue abstraite qualitative)

est représentatif de la sémantique du système modélisé, et donc il est important de compléter cette sémantique par des informations quantitatives (obtention d'une vue abstraite qualitative quantifiée).

Le but de ce chapitre est de développer une méthodologie d'obtention des vues abstraites qualitatives quantifiées et de montrer, à travers des exemples d'application, l'intérêt de ce concept.

Ce chapitre est décomposé en quatre parties :

- la première partie rappelle la notion de vue abstraite qualitative basée sur la relation d'équivalence observationnelle;
- la deuxième partie rappelle la notion de vue abstraite quantitative basée sur les règles de réduction de Beizer;
- la troisième partie présente le concept de vue abstraite qualitative quantifiée (qui combine les deux premiers types de vues abstraites);
- la dernière partie montre l'intérêt du concept de vue abstraite qualitative quantifiée pour l'étude de mécanismes protocolaires et d'architectures de communication. Nous étudierons trois exemples d'application : le phénomène de divergence, c'est-à-dire l'apparition de comportements répétitifs du système, que l'on rencontre fréquemment dans les protocoles de communication et dans les mécanismes d'exclusion (application 1); la modélisation du service offert par un protocole de transfert de données (application 2); la modélisation d'un architecture multicouches (application 3).

IV.2 Vues abstraites qualitatives

Les vues abstraites qualitatives sont basées sur une approche abstraite de la vérification formelle. Cette approche a été définie dans [Ver89] par : « *Il est plus intéressant d'étudier un système par rapport à son comportement externe (d'étudier la façon dont il réagit à des sollicitations de son environnement) que de l'étudier par rapport à sa structure interne* ».

De nombreuses propositions d'équivalences basées sur la réaction des systèmes à des événements externes ont été proposées. Elles sont basées sur la notion d'observation, aussi appelée abstraction.

Nous ne nous intéressons ici qu'à la notion d'équivalence observationnelle définie dans [Mil80], qui est basée sur les notions d'observation et d'expérimentation. Nous ne donnons ici que la définition générale de cette équivalence ([Ver89]).

IV.2.1 Obtention du graphe quotient par équivalence observationnelle

Un graphe d'états fini labellé avec des étiquettes qualitatives est un triplet $< S, E, \Delta >$ où :

- $S = (E_0, E_1, \ldots)$ est l'ensemble des sommets du graphe. Ces sommets représentent les états du système étudié. E_0 est l'état initial;

- $E = (e_0, e_1, \ldots)$ est l'ensemble des labels (événements);
- Δ est l'ensemble des arcs labellés du graphe : $\Delta \subset S \times E \times S$. Un élément $(m, \mu, p) \in \Delta$ est noté $m \xrightarrow{\mu} p$. La notation $\xrightarrow{\mu}$ est la relation de transition.

Soit E' l'ensemble des événements que l'on veut observer, c'est-à-dire l'ensemble des événements que l'on veut voir apparaître dans l'automate quotient. Tout événement de E n'appartenant pas à E' sera dit non observable et sera noté τ.

La relation de transition $\stackrel{\mu}{\Rightarrow}$, $\mu \in E' \cup \{\epsilon\}$ est définie par :

- $m \stackrel{\epsilon}{\Rightarrow} p$: $m = p$ où $m \xrightarrow{\tau} x_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} x_p \xrightarrow{\tau} p$
- $m \stackrel{\mu}{\Rightarrow} p$: $m \stackrel{\epsilon}{\Rightarrow} x_1 \xrightarrow{\mu} x_2 \stackrel{\epsilon}{\Rightarrow} p$

Ces relations sont appelées ϵ-expérimentation ou μ-expérimentation.

L'équivalence observationnelle, notée $\equiv$, peut être définie comme la limite d'une suite décroissante de relations d'équivalence. Cette suite est définie par récurrence en posant :

- $\forall m, p \in S, m \equiv_0 p$
- $m \equiv_{k+1} p$ ssi $m \equiv_k p$ et $\forall t \in E' \cup \{\epsilon\}$
  $[m \stackrel{t}{\Rightarrow} n$ implique $\exists q \in S : p \stackrel{t}{\Rightarrow} q$ et $n \equiv_k q]$
  et
  $[p \stackrel{t}{\Rightarrow} q$ implique $\exists n \in S : m \stackrel{t}{\Rightarrow} n$ et $q \equiv_k n]$
- $\equiv = \bigcap_{k \geq 0} \equiv_k$

On peut résumer cette définition en disant que deux sommets d'un graphe sont équivalents observationnellement si les comportements observables du système à partir de ces deux sommets sont identiques.

L'exemple des figures IV.1, IV.2 illustre l'obtention d'un automate quotient équivalent au graphe initial par rapport aux événements observables (e_1, e_5).

Les différentes relations d'équivalence obtenues lors du calcul de l'automate quotient sont :

$$\equiv_0 = \{\{E_0\}, \{E_1\}, \{E_2\}, \{E_3\}, \{E_4\}, \{E_5\}, \{E_6\}\}$$
$$\equiv_1 = \{\{E_0, E_6\}, \{E_1, E_2, E_3, E_4, E_5\}\}$$
$$\equiv_2 = \{\{E_0, E_6\}, \{E_1, E_2, E_3, E_5\}, \{E_4\}\}$$
$$\equiv_3 = \{\{E_0, E_6\}, \{E_1, E_2, E_3, E_5\}, \{E_4\}\}$$

L'égalité $\equiv_2 = \equiv_3$ indique que la suite a fini de converger, et que donc $\equiv_2$ (ou $\equiv_3$) donne la structure de l'automate quotient (figure IV.2). Les états E_0, E_6 sont considérés comme équivalents et sont regroupés dans une même classe d'états, C_1. Même chose pour les états E_1, E_2, E_3, E_5 qui sont regroupés dans la classe C_2, et l'état E_4 qui est le seul état de la classe C_3.

L'automate quotient de la figure IV.2 ne fait apparaître entre les différentes classes d'états que des arcs labellés par le nom d'un événement observable (e_1, e_5). Néanmoins, l'événement ϵ (aussi noté τ) peut apparaître dans une transition entre deux classes si son occurrence influence le comportement observé du système (modifie le comportement observé qui lui succède).

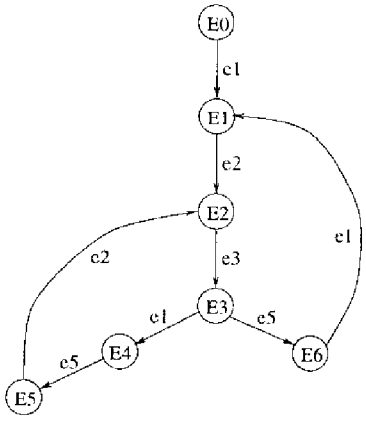


FIG. IV.1 – *Graphe d'états avec des étiquettes qualitatives*

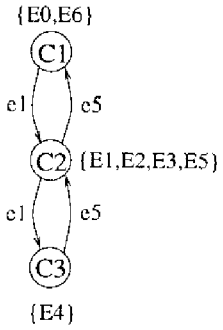


FIG. IV.2 – *Automate quotient*

IV.2.2 Obtention de l'automate quotient détaillé

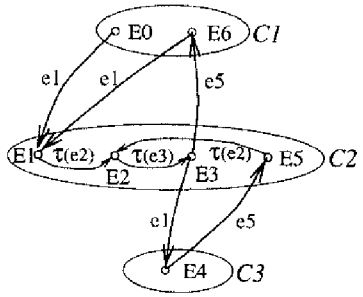


FIG. IV.3 – *Automate quotient détaillé*

Dans l'automate quotient, les liens entre états d'une même classe ne sont pas intéressants du point de vue de l'équivalence observationnelle (ils n'apportent aucune information supplémentaire sur le comportement du système dans le cadre des événements observables choisis). C'est pourquoi ils n'apparaissent pas sur la représentation de cet automate (figure IV.2).

Néanmoins, ces informations nous seront utiles pour construire les vues abstraites qualitatives quantifiées (paragraphe IV.4). Nous nous proposons donc de définir par automate quotient détaillé l'automate quotient (obtenu par équivalence observationnelle) dans lequel nous faisons apparaître tous les états dans chaque classe et les arcs entre ces derniers. Les transitions entre deux états d'une même classe sont issues de l'occurrence d'un événement non observable; elles seront donc toutes labellées par l'étiquette τ. Les transitions entre deux états de classes différentes correspondent aux transitions entre classes de l'automate quotient. Elles seront donc labellées soit par le nom d'un événement observable, soit par

l'étiquette τ. Notons que plusieurs transitions entre des états de deux classes différentes peuvent correspondre à une seule transition de l'automate quotient; elles seront alors labellées par la même étiquette que celle de la transition de l'automate quotient.

La figure IV.3 montre l'automate quotient détaillé de l'exemple de la figure IV.2 (événements observables : e_1 et e_5).

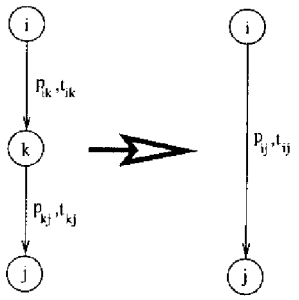
IV.3 Vues abstraites quantitatives

La méthodologie d'obtention de vues abstraites quantitatives est basée sur deux éléments : les règles de réduction de Beizer et une représentation matricielle du graphe étudié. Nous considérons dans ce sous-chapitre que le graphe G étudié est défini par :

- l'ensemble E des états du système,
- l'ensemble U des arcs u_{ij} reliant l'état E_i à l'état E_j, caractérisés par un doublet (p_{ij}, t_{ij}) (p_{ij} est la probabilité de transition de l'état E_i à E_j, t_{ij} est le temps de transition de l'état E_i à l'état E_j).

IV.3.1 Règles de Beizer

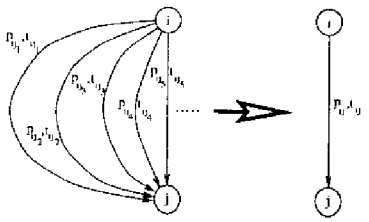
Les règles de réduction de Beizer [Bei71] sont au nombre de trois : la réduction série (figure IV.4), la réduction parallèle (figure IV.5), la réduction de boucle (figure IV.6).



$$p_{ij} = p_{ik} \cdot p_{kj}$$

$$t_{ij} = t_{ik} + t_{kj}$$

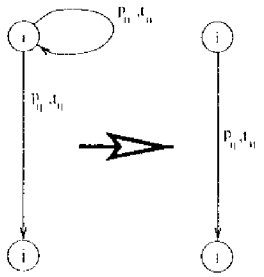
FIG. IV.4 – *Réduction série*



$$p_{ij} = \sum_{k=1}^{k=N} p_{ij_k}$$

$$t_{ij} = \frac{\sum_{k=1}^{k=N} p_{ij_k} . t_{ij_k}}{\sum_{k=1}^{k=N} p_{ij_k}}$$

FIG. IV.5 – *Réduction parallèle*



$$p_{ij} = \frac{p_{ij}}{1 - p_{ii}}$$

$$t_{ij} = t_{ij} + \frac{t_{ii} \cdot p_{ii}}{1 - p_{ii}}$$

FIG. IV.6 *Réduction de boucle*

IV.3.2 Matrices associées au graphe

Deux matrices carrées, d'ordre le nombre d'états dans le graphe, sont associées au graphe [Ata94] :

la matrice des probabilités $P = [p_{ij}]$,

la matrice des temps de transition $T = [t_{ij}]$.

Dans ces matrices, chaque ligne i représente les probabilités ou les temps de transition d'un état E_i vers un état successeur E_j. Chaque colonne j représente les probabilités ou les temps de transition d'un état prédécesseur E_i vers l'état E_j.

Remarque : Ces matrices ne permettent de représenter que des graphes dans lesquels il n'y a pas plusieurs arcs entre deux mêmes états (*1-graphe*). Dans le cas où il existe dans le graphe G plusieurs arcs entre deux mêmes états, une transformation préalable est nécessaire pour transformer ces arcs en un seul arc.

IV.3.3 Définition d'une vue abstraite quantitative

Une vue abstraite quantitative est une projection sur un sous-ensemble S d'états du graphe. Dans une vue abstraite quantitative, toute transition entre deux états E_i et E_j de S est caractérisée par une probabilité P_{ij} et un temps de transition T_{ij}, respectivement probabilité et temps de premier passage [Ata94]. Il existe deux types de vues abstraites quantitatives [Ata94] : les vues abstraites quantitatives sans boucle et les vues abstraites quantitatives avec boucles. Dans la suite de ce chapitre, nous entendrons par vue abstraite quantitative une vue abstraite quantitative sans boucle.

IV.3.3.1 Méthodologie de calcul d'une vue abstraite quantitative

Pour chaque couple d'états $(E_i, E_j) \in S, E_i \neq E_j$, les valeurs de P_{ij} et T_{ij} sont données par :

calcul de la probabilité F_{ij} :

$$[F_{ij}] = [I - [p^0_{ij}]]^{-1} \qquad \text{(IV.1)}$$

$[p^0_{ij}]$ étant la matrice P dans laquelle la ligne j est annulée (l'état E_j n'a plus de successeur), et les colonnes k telles que $E_k \in S - \{E_i, E_j\}$ sont anulées (on ne prend

pas en compte les chemins entre E_i et E_j qui passent par les autres états de S. Le fait d'exclure E_i de S permet d'intégrer les boucles sur E_i dans l'arc $E_i \rightarrow E_j$).

La colonne j de la matrice $[F_{ij}]$ obtenue donne toutes les probabilités de transition d'un état $E_i \neq E_j$ vers l'état E_j;

- calcul du temps de transition T_{ij} :

$$[F_{ij}T_{ij}]_j = [I - [p^0_{ij}]]^{-1}.\{[p_{ij}t_{ij}]_j + [p^0_{ij}t^0_{ij}][F_{ij}]_j\} \qquad \text{(IV.2)}$$

$[p_{ij}t_{ij}]_j$ et $[F_{ij}]_j$ étant respectivement les colonnes j des matrices $[p_{ij}t_{ij}]$ et $[F_{ij}]$; $[p^0_{ij}t^0_{ij}]$ est la matrice $[p_{ij}t_{ij}]$ sur laquelle on applique les mêmes restrictions que pour $[p^0_{ij}]$.

Le résultat de ce calcul est la colonne j de la matrice $[F_{ij}T_{ij}]$; en divisant par F_{ij} (calculé précédemment) l'élément de la ligne i de ce vecteur, on obtient la valeur de T_{ij}.

IV.3.3.2 Exemple

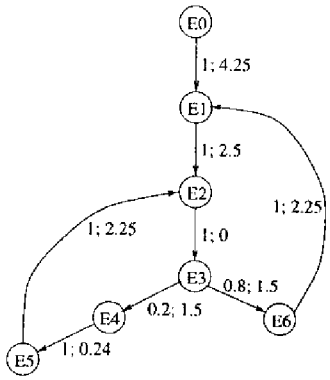


FIG. IV.7 – *Exemple de graphe avec étiquettes quantitatives*

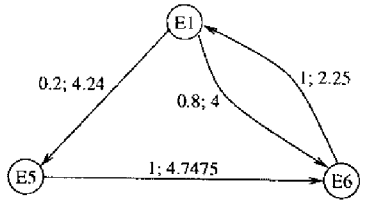


FIG. IV.8 – *Vue abstraite quantitative*

Soit le graphe de la figure IV.7. Nous nous proposons de calculer la vue abstraite quantitative relative aux états E_1, E_5 et E_6 (figure IV.8).

Les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ s'écrivent :

$$[p_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad [p_{ij}t_{ij}] = \begin{bmatrix} 0 & 4.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 1.5 \\ 0 & 0 & 0 & 0 & 0 & 0.24 & 0 \\ 0 & 0 & 2.25 & 0 & 0 & 0 & 0 \\ 0 & 2.25 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- **Calcul de la transition $E_1 \rightarrow E_5$:**
  on annule dans les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ la ligne 6 (E_5 correspond à la ligne 6) et la colonne 7 (E_6 correspond à la colonne 7). Les formules IV.1 et IV.2 nous donnent alors : $F_{15} = 0.2$ et $T_{15} = 4.24$.

- **Calcul de la transition $E_1 \rightarrow E_6$:**
  on annule dans les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ la ligne 7 (E_6) et la colonne 6 (E_5). Les formules IV.1 et IV.2 nous donnent alors : $F_{16} = 0.8$ et $T_{16} = 4$.

- **Calcul de la transition $E_6 \rightarrow E_1$:**
  on annule dans les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ la ligne 2 (E_1) et la colonne 6 (E_5). Les formules IV.1 et IV.2 nous donnent alors : $F_{16} = 1$ et $T_{16} = 2.25$.

- **Calcul de la transition $E_5 \rightarrow E_6$:**
  on annule dans les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ la ligne 7 (E_6) et la colonne 2 (E_1). Les formules IV.1 et IV.2 nous donnent alors : $F_{56} = 1$ et $T_{56} = 4.7475$.

IV.4 Vues abstraites qualitatives quantifiées

La notion de vue abstraite qualitative quantifiée est basée sur les notions de vue abstraite qualitative et vue abstraite quantitative. Son but est de donner une représentation réduite du graphe initial, équivalente à ce dernier relativement à un ensemble d'événements, tout en conservant les informations quantitatives (informations que l'on perd dans les vues abstraites qualitatives).

L'obtention d'une vue abstraite qualitative quantifiée s'effectue en trois phases :

- obtention de l'automate quotient relatif à l'ensemble des événements observables E',
- obtention de l'automate quotient détaillé quantifié,
- obtention de l'automate quotient quantifié.

Nous allons maintenant étudier successivement ces trois phases.

IV.4.1 Automate quotient

L'obtention de l'automate quotient relatif à un ensemble d'événements E' est obtenu par équivalence observationnelle, telle que nous l'avons décrite dans le paragraphe IV.2.1. A la fin de cette phase, nous avons un automate constitué de classes d'états et d'arcs entre classes labellés par une étiquette qualitative.

IV.4.2 Automate quotient détaillé quantifié

Le début de cette phase consiste en l'obtention de l'automate quotient détaillé, telle que nous l'avons décrite dans le paragraphe IV.2.2. Cet automate nous permet de faire

apparaître chaque état du graphe initial dans les classes d'états, et de faire apparaître entre ces états les arcs du graphe d'états initial avec les étiquettes qualitatives.

La seconde partie de cette phase consiste en l'ajout, sur chaque arc de l'automate précédent, des étiquettes quantitatives du graphe d'états initial (dans le cadre d'un graphe d'états probabilisé, ces étiquettes sont constituées d'une probabilité et d'un temps de transition). L'automate ainsi obtenu est l'automate quotient détaillé quantifié.

Notons qu'à ce niveau du calcul, nous n'avons pas diminué la complexité du graphe d'états initial. Nous n'avons fait que regrouper les états en classes d'états, conformément à l'automate quotient.

IV.4.3 Automate quotient quantifié

Le but de cette phase est de réduire la complexité de l'automate quotient détaillé quantifié. Pour cela, nous allons simplifier la représentation du comportement interne des classes puisque seules les informations quantitatives nous intéressent dans ces dernières. Il nous faut donc donner une vue abstraite de ces classes en conservant les informations quantitatives.

Nous allons dans un premier temps proposer des définitions qui nous permettront ensuite de donner la méthodologie de calcul de l'automate quotient quantifié. Nous illustrerons ensuite cette méthodologie sur un exemple simple.

IV.4.3.1 Définitions

Définissons dans un premier temps les différents types d'états constituant une classe d'états, puis la notion de vue abstraite quantifiée d'une classe d'états.

Définition 1 :

Dans une classe d'états C_i, nous appelons état frontière de C_i tout état de la classe qui est relié par un arc (au moins) à un état n'appartenant pas à la classe C_i.

Définition 2 :

Dans une classe d'états C_i, nous appelons état interne de la classe tout état qui n'est relié à aucun état n'appartenant pas à la classe C_i.

Définition 3 :

Dans une classe d'états C_i, un état d'entrée est un état frontière qui est le début d'arcs n'aboutissant qu'à des états de la même classe C_i (frontières ou internes).

Définition 4 :

Dans une classe C_i, un état de sortie est un état frontière qui est la fin d'arcs ne provenant que d'états de la même classe C_i (frontières ou internes).

Définition 5 :

Dans une classe C_i, un état d'entrée-sortie est un état frontière qui est à la fois état d'entrée et état de sortie.

Définition d'une vue abstraite quantifiée d'une classe d'états :

Une vue abstraite quantifiée d'une classe d'états est l'automate résultant de la vue abstraite quantitative sur l'ensemble des états frontières de cette classe d'états.

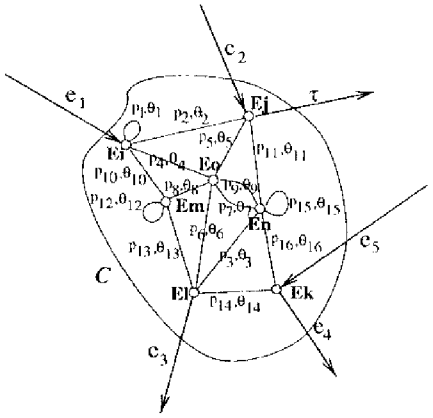


FIG. IV.9 - *Vue abstraite détaillée quantifiée*

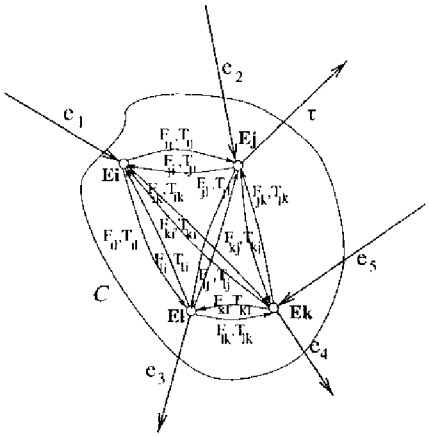


FIG. IV.10 - *Vue abstraite quantifiée*

La figure IV.10 nous montre la vue abstraite quantifiée obtenue à partir de la vue abstraite détaillée quantifiée de la classe C (figure IV.9). Dans cette classe, il y a quatre états frontières : un état d'entrée E_i, un état de sortie E_l, et deux états d'entrée-sortie E_j et E_k. Ce sont sur ces quatre états que la vue abstraite quantitative est faite.

IV.4.3.2 Méthodologie d'obtention de l'automate quotient quantifié

L'automate quotient quantifié est obtenu à partir de l'automate quotient détaillé quantifié en remplaçant chaque classe d'états détaillée par sa vue abstraite quantifiée (classe quantifiée).

En ce qui concerne le calcul d'une classe quantifiée, il faut noter plusieurs points :

- dans le calcul de la vue abstraite quantitative, tous les arcs partant d'un état de sortie ou d'un état d'entrée-sortie et aboutissant à un état extérieur à la classe sont éliminés (mise à 0 des éléments correspondants dans les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$), ceci afin d'éviter la prise en compte de cycles comprenant des états extérieurs à la classe. Ces contraintes s'ajoutent aux contraintes imposées sur les matrices par le calcul de la vue abstraite quantifiée (paragraphe IV.3.3).

Les parties du graphe extérieures à la classe n'étant pas prises en compte, tout se passe comme si le graphe n'était composé que des états et des arcs de la classe C_i. On peut donc, dans le calcul de la vue abstraite quantifiée, réduire les matrices $[p_{ij}]$ et $[p_{ij}t_{ij}]$ à des matrices de dimension le nombre d'états dans la classe C_i (on ne prend en compte que les arcs internes à la classe). Cette remarque permet de diminuer de manière non négligeable le coût du calcul en termes de mémoire et de temps de calcul;

- s'il n'y a pas d'états internes dans la classe, et s'il n'existe pas de boucle sur un état frontière, ni de cycle entre les états frontières, alors la vue abstraite quantifiée de la classe nous est déjà donnée par la vue abstraite détaillée quantifiée. Il est donc inutile dans ce cas d'appliquer l'algorithme précédent.
  En pratique, on applique l'algorithme précédent dans tous les cas (dans le cas où on a déjà la vue abstraite quantifiée, l'algorithme n'aura aucun effet et nous donnera la même vue que la vue abstraite détaillée quantifiée);

IV.4.3.3 Exemple

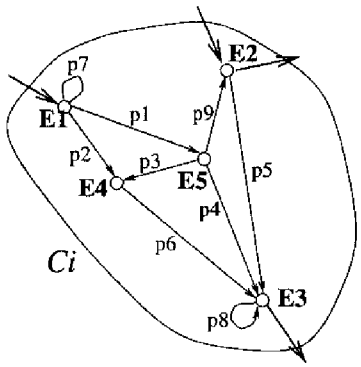


FIG. IV.11 – *Vue abstraite détaillée quantifiée de la classe* C_i

Soit le graphe de la figure IV.11, qui représente une classe d'états C_i d'un automate plus général. Nous nous proposons de calculer la vue abstraite quantitative relative aux états E_1, E_2 et E_3 : E_1 est état d'entrée de C_i, E_2 est état d'entrée-sortie, E_3 est état de sortie.

La matrice $[p_{ij}]$, ramenée aux états de la classe C_i, s'écrit :

$$[p_{ij}] = \begin{bmatrix} p_7 & 0 & 0 & p_2 & p_1 \\ 0 & 0 & p_5 & 0 & 0 \\ 0 & 0 & p_8 & 0 & 0 \\ 0 & 0 & p_6 & 0 & 0 \\ 0 & p_9 & p_4 & p_3 & 0 \end{bmatrix}$$

où la colonne 1 et la ligne 1 correspondent à l'état E_1, ...

Le calcul de la probabilité de chemin entre l'état E_1 et l'état E_3 est calculée en utilisant la formule IV.1, où la matrice $[p^0_{ij}]$ s'écrit :

$$[p^0_{ij}] = \begin{bmatrix} p_7 & \mathbf{0} & 0 & p_2 & p_1 \\ 0 & \mathbf{0} & p_5 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & p_6 & 0 & 0 \\ 0 & \mathbf{0} & p_4 & p_3 & 0 \end{bmatrix}$$

La colonne 2 a été mise à zéro pour éviter de prendre en compte des chemins entre E_1 et E_3 passant par E_2 (*nous considérons les chemins ne passant pas par les autres états frontières de la classe*), et la ligne 3 est mise à zéro pour éviter de prendre en compte des chemins dans lesquels on passe plusieurs fois par E_3 (notion de premier passage).

Dans le cas général de n états frontières, 1 seule ligne est mise à zéro, et $n-2$ colonnes sont mises à zéro.

IV.5 Application 1 : évaluation du phénomène de divergence

Le phénomène de divergence correspond à une répétition à l'infini d'un comportement : un système diverge s'il peut accomplir indéfiniment une séquence d'actions. Il est très important de pouvoir détecter un tel phénomène et le quantifier, surtout dans le cadre de systèmes temps-réels. Dans un graphe d'états, ce phénomène se traduit sous deux formes :

- des circuits, c'est-à-dire une succession de transitions qui amènent d'un état à ce même état,
- des boucles, c'est-à-dire des transitions qui partent et arrivent sur le même état.

Les circuits et/ou boucles (qui visualisent des répétitions d'actions) représentent ainsi l'empêchement des autres actions (notion de famine).

La vue abstraite qualitative, basée sur la relation d'équivalence observationnelle de Milner, masque les phénomènes de divergence quand ils ne sont constitués que d'événements non observés [Ver89].

L'exemple de la figure IV.12 illustre ce phénomène de masquage de la divergence. Le modèle présenté modélise le partage d'une ressource en exclusion mutuelle. La place P_7 représente la ressource. Les places P_1, P_2 et P_3 représentent respectivement, chez le premier processus, les conditions « Attente d'une requête », « Demande de la ressource » et « Libération de la ressource ». Les places P_4, P_5 et P_6 ont les mêmes significations, pour le second processus. Les transitions sont labellées avec l'envoi de la notification de requête (!Req), la réception de l'autorisation de prise de la ressource (?Ack) et l'envoi la notification de libération de la requête (!Rel).Toutes les transitions ont une densité de probabilité exponentielle (taux λ).

Le graphe d'états probabilisé moyen est représenté sur la figure IV.13 (tous les taux λ ont pour valeur 1). L'automate quotient relatif aux événements !Req1, ?Ack1, !Rel1 (com-

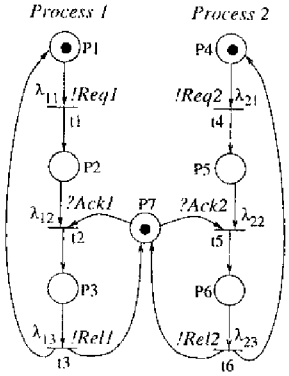


FIG. IV.12 – *Modèle RdPTS de la mutuelle exclusion*

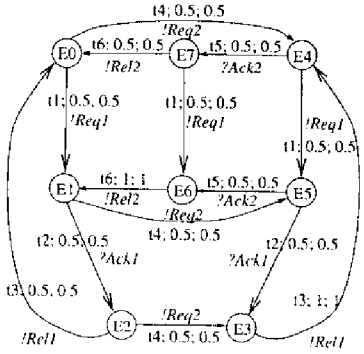


FIG. IV.13 – *Graphe d'états probabilisé moyen du modèle de la figure IV.12*

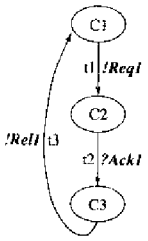


FIG. IV.14 – *Automate quotient relatif au premier processus*

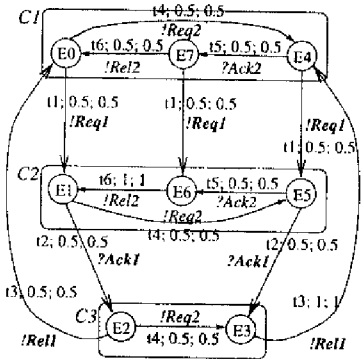


FIG. IV.15 – *Automate quotient quantifié relatif au premier processus*

portement du premier processus) est donné sur la figure IV.14. D'après cet automate, la séquence !Req1 $\rightarrow$?Ack1 $\rightarrow$!Rel1 est inévitable.

L'automate quotient quantifié relatif aux événements !Req1, ?Ack1, !Rel1 (comportement du premier processus) est donné sur la figure IV.15. Notons ici que comme tous les états du graphe d'états probabilisé sont des états frontières des classes de l'automate quotient, ils apparaissent dans l'automate quotient quantifié; l'automate quotient quantifié est donc identique à l'automate quotient détaillé quantifié. Si certains états du graphe détats probabilisé moyen avaient été des états internes aux classes d'états, ils n'apparaitraient pas dans la figure IV.15.

On peut se rendre compte sur cet automate que les classes C_1 et C_2 masquent des divergences : elles comportent chacune un cycle (circuit), qui ne disparait pas lors de la quantification de l'automate quotient, puisqu'il passe par des états frontières de ces classes. Ces circuits sont très importants du point de vue qualitatif, car ils indiquent que le second processus peut utiliser la ressource continuellement, sans que le premier processus puisse

y accéder (famine pour le processeur 1 due à une monopolisation de la ressource par le processeur 2). Cette propriété est restée invisible sur l'automate quotient car les transitions participant à cette divergence ne sont pas observées.

Dans notre exemple, l'automate quotient quantifié fait apparaître ces phénomènes de divergence. Notons que cette propriété n'est pas toujours vérifiée : si la divergence n'implique que des états internes à la classe, la vue abstraite quantifiée de la classe ne fait pas apparaître cette divergence (puisque les états internes aux classes ne sont pas visibles dans cet automate), mais la quantifiera en tenant compte de sa probabilité et de sa durée dans les temps et probabilités de chemin entre états frontières de la classe.

Ici, on peut quantifier ce phénomène de divergence en mesurant la probabilité P_d et le temps moyen T_d d'ocurrence d'un circuit dans la classe C_1 par exemple : P_d est le produit des probabilités des arcs du circuit, T_d est la somme des temps des arcs du circuit. La table de la figure IV.16 donne les valeurs de P_d et de T_d en fonction de λ_{21}, taux d'ocurrence de requête sur le second processus (les autres taux sont tous supposés égaux à 1). On s'aperçoit que plus le taux λ_{21} augmente, plus la probabilité P_d de monopolisation de la ressource augmente, alors que le temps entre deux requêtes !Req2 (temps de circuit) diminue. On a ainsi transformé une vision qualitative de la famine du premier processus en une vue quantitative, en fonction du taux de requêtes du second processus.

λ_{21}	P_d	T_d
1	0.125	1.5
5	0.21	1.17
10	0.23	1.09

FIG. IV.16 *Quantification de la divergence*

λ_{21}	P_c
1	0.57
5	0.21
10	0.12

FIG. IV.17 *Probabilité conditionnelle d'occurrence de* !Req1

Après avoir évalué le phénomène de divergence, on peut calculer la probabilité conditionnelle de l'événement *!Req1*, sachant que l'on est dans un des états E_0, E_4, E_7.

La probabilité conditionnelle d'occurrence de *!Req1* sachant que l'on est dans l'état E_0 se calcule comme suit : on considère que le circuit de la classe C_1 forme une boucle sur l'état E_0 (cela revient à dire que l'événement *!Req1* ne peut pas avoir lieu dans les états E_4 et E_7). La probabilité de cette boucle est le produit des probabilités des arcs qui la compose (P_d). On applique ensuite les règles de Beizer pour intégrer cette boucle dans l'arc $E_0 \rightarrow E_1$. La probabilité du nouvel arc est la probabilité d'occurrence de *!Req1* sachant que l'on est dans l'état E_0.

La table de la figure IV.17 nous donne la valeur de cette probabilité en fonction du taux λ_{21}. Un calcul identique nous permettrait de calculer les probabilités conditionnelles d'occurrence de *!Req1* sachant que l'on dans l'état E_4 et l'état E_7.

IV.6 Application 2 : évaluation du service offert par un protocole de transfert de données

Nous nous proposons de faire une étude du service offert par le protocole de transfert de données Go Back N, et de montrer, à travers cet exemple, l'utilité du concept d'automate quotient qualitatif quantifié.

IV.6.1 Hypothèses sur le fonctionnement du protocole Go Back N

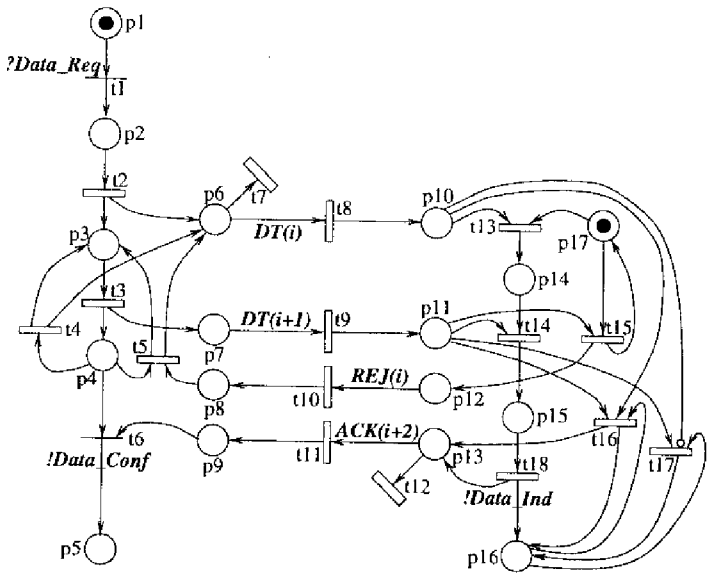


FIG. IV.18 – *Modèle du protocole Go Back N avec une fenêtre de 2 messages*

densité de probabilité	transitions
$\delta(x)$	t_1, t_6
$\delta(x-3)$	t_8, t_9, t_{10}, t_{11}
$\delta(x-5)$	$t_2, t_3, t_5, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}$
$\delta(x-100)$	t_4
$2p_p\delta(x-3)+1-2p_p$ $(3 \leq x \leq 4)$	t_7, t_{12}

FIG. IV.19 – *Densités de probabilités associées aux transitions*

Nous supposerons que ce protocole de transfert de donnée fragmente les données (SDU) en deux fragments. Ces deux PDU doivent être reçus tous les deux par l'entité réceptrice avant qu'elle n'acquitte la donnée (envoi du PDU ACK à l'entité émettrice : acquittement collectif). Cet accusé de réception pourra lui aussi être perdu, Dans ce cas, la donnée est ré-émise au bout d'un certain temps (« time-out ») par l'entité émettrice.

Un mécanisme de détection et de correction d'erreur est mis en place. Ce mécanisme implique, en cas de perte sur le médium de transmission, la ré-émission de tous les fragments de la donnée. Nous supposerons ici que seul le premier fragment de la donnée peut être perdu par le médium. Ainsi, si l'entité réceptrice reçoit le second fragment sans avoir reçu le premier, elle envoi un PDU de rejet (REJ) pour signaler à l'émetteur la perte, et lui demander la ré-émission de la donnée (des deux fragments).

IV.6.2 Description du modèle Réseaux de Petri

La figure IV.18 donne le modèle RdPTS du protocole Go Back N. Les transitions t_1, t_{18} et t_6 correspondent au service transport fourni par la couche transport au niveau supérieur : ?Data_Req, !Data_Ind, !Data_Conf. Les transitions t_2 et t_3 correspondent à l'envoi des deux fragments de la donnée (entité émettrice). Les transitions t_{13} et t_{14} correspondent à la réception de ces fragments (entité réceptrice). Les transitions t_8, t_9, t_{10} et t_{11} modélisent la propagation des divers PDUs échangées entre les deux entités, et les transitions t_7 et t_{12} modélisent les pertes possibles des PDUs DT(i) (premier fragment) et ACK(i+2) (acquittement des deux fragments). Nous considérons que le médium de transmission est bidirectionnel à l'alternat, et que seuls les deux PDUs précédents peuvent être perdus.

Au niveau de l'entité émettrice, la transition t_4 modélise la temporisation qui permet de ré-émettre les deux fragments de donnée si l'accusé de réception ACK(i+2) n'est pas arrivé au bout d'un certain temps. La transition t_5 permet de ré-émettre les deux fragments de donnée en cas de réception d'un rejet REJ(i) de ces fragments.

Au niveau de l'entité réceptrice, la transition t_{18} permet, en plus de la délivrance de la donnée au niveau supérieur (!Data_Ind), d'envoyer le PDU d'acquittement ACK(i+2) des fragments de donnée. La transition t_{15} permet de rejeter les deux fragments (envoi du PDU REJ(i)) en cas de réception du segment DT(i+1) avant le fragment DT(i) (ceci peut se produire s'il y a perte du fragment DT(i) par le médium). La transition t_{16} permet de renvoyer le PDU ACK(i+2) en cas de réception des deux fragments DT(i) et DT(i+1) hors séquence (ceci peut se produire si, après délivrance de la donnée, le PDU ACK(i+2) se perd sur le réseau; la temporisation t_4 s'achève alors sur l'entité émettrice et les deux fragments sont renvoyés). La transition t_{17} permet de renvoyer le PDU ACK(i+2) en cas de réception du seul PDU DT(i+1) hors séquence (ce cas est le même que pour la transition t_{16}, si ce n'est que le fragment DT(i) renvoyé par l'entité émettrice est perdu par le médium).

La place p_{16} de l'entité réceptrice est marquée dès que la donnée est délivrée au niveau supérieur (elle indique l'attente de passage à la fenêtre suivante). Notons que les transitions t_{16} et t_{17} dépendent de cette place (ce qui permet de détecter les fragments hors séquence).

Les transitions $t_2, t_3, t_5, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}$ sont temporisées, et ont la même densité de probabilité discrète. Cette durée représente le temps de traitement des fragments de donnée.

La figure IV.19 donnent les densités de probabilités associées aux transitions du modèle RdPTS. Le temps de propagation (3 unités de temps (u.t.)) a été choisi inférieur au temps de traitement des PDUs (5 u.t.). La temporisation a une durée de 100 u.t..

IV.6.3 Modélisation des pertes sur le médium

En ce qui concerne les transitions de pertes, nous choisissons comme densité de probabilité une densité mixte qui nous permettra de paramétrer la valeur de la probabilité de perte. Prenons l'exemple de la figure IV.20. La transition t_i a une densité de probabilité discrète (Dirac à la date a). La transition t_j a une densité de probabilité mixte (Dirac de poids $2k$ à la date a, uniforme entre a et b de hauteur $\frac{1-2k}{b-a}$, qui est le complément à 1 du poids de l'impulsion de Dirac). Supposons que ces deux transitions soient sensibilisées en même temps. Elles seront alors tirables toutes les deux à la date a. La probabilité de tir de t_i est égale à [Ata94] : $P(t_i) = \frac{1}{2}.(1).2k + 1.\int_a^b \frac{1-2k}{b-a}dx = k + 1 - 2k = 1 - k$; la probabilité de tir de t_j est égale à : $P(t_j) = \frac{1}{2}.(1).2k = k$.

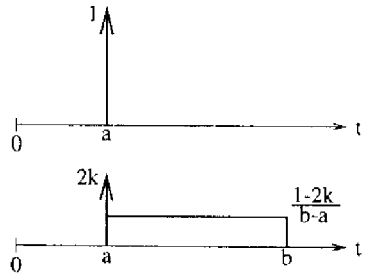


FIG. IV.20 – *Ajustement de la probabilité de perte par la densité de probabilité*

Si, dans l'exemple du protocole Go Back N, on associe la densité de probabilité de la transition t_i à la transition t_8 du modèle de la figure IV.18 (propagation normale du PDU DT(i)), et celle de t_j à la transition t_7 (perte du PDU DT(i)), on voit que la probabilité de perte sur le médium est égale à k. On peut ainsi ajuster la probabilité de perte voulue en choisissant les valeurs appropriées de k, a et b dans la densité de probabilité de t_7 et t_8. Dans nos exemples, les valeurs de a et b sont 3 et 4, 3 étant donc la valeur du temps de propagation; k varie de 10^{-1} à 10^{-5}.

IV.6.4 Extension du modèle

L'extension du modèle du protocole Go Back N à une fenêtre de trois messages est présentée sur la figure IV.21. Ce nouveau modèle peut être facilement étendu à une fenêtre de quatre messages ou plus. Les valeurs temporelles sont les mêmes que celles choisies pour le protocole avec fenêtre de deux messages (voir table de la figure IV.19).

IV.6.5 Analyse du protocole Go Back N

IV.6.5.1 Étude du service (Data_Req, Data_Ind)

Nous nous proposons d'étudier le service (Data_Req, Data_Ind), aussi appelé service distribué, offert par le protocole Go Back N :

- d'un point de vue qualitatif, c'est-à-dire savoir si un Data_Req est toujours suivi d'un Data_Ind,

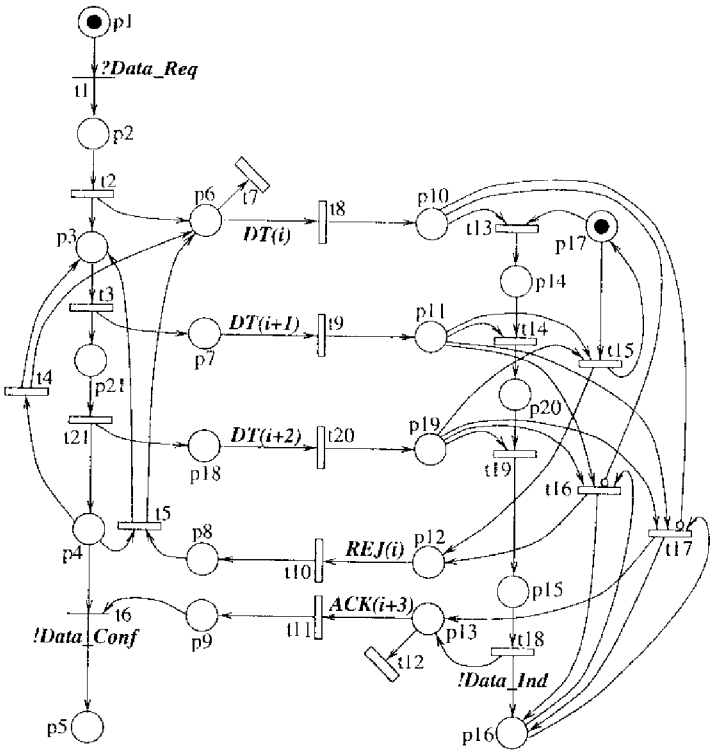


FIG. IV.21 – *Modèle du protocole Go Back N avec une fenêtre de 3 messages*

– d'un point de vue quantitatif, c'est-à-dire calculer et comparer le temps écoulé entre les occurrences des deux primitives, en fonction de la probabilité de perte p_p et de la largeur de la fenêtre.

Pour cela, nous construisons pour chaque valeur choisie de la probabilité de perte, et pour chaque largeur de fenêtre, un graphe d'états probabilisé donnant le comportement du protocole. La structure de ces graphes d'états probabilisés varie avec la largeur de la fenêtre (puisqu'il y a plus ou moins de messages à envoyer), mais ne varie pas avec la probabilité de perte (seules les valeurs quantitatives varient avec p_p). Le nombre d'états pour chacun de ces graphes est : 28 pour une fenêtre de 2 messages, 38 pour une fenêtre de 3 messages, 48 pour une fenêtre de 4 messages. Nous donnons sur la figure IV.22 le graphe d'états probabilisé moyen obtenu pour une fenêtre de 2 messages.

Nous calculons ensuite, à partir de ces graphes, l'automate quotient quantifié relatif aux deux primitives Data_Req et Data_Ind. L'automate obtenu est représenté sur la figure IV.23. Sa structure est la même pour toutes les valeurs de la probabilité de perte et pour toute largeur de fenêtre (pour tous les graphes). Seule la valeur du temps de séjour θ_1 dans la classe C_1 varie. Notons que cette variation dépend à la fois de la valeur de p_p et du nombre de fragments de donnée à envoyer.

Cet automate montre bien que la propriété qualitative « Data_Req toujours suivi d'un Data_Ind » est vérifiée. La table de la figure IV.24 donne la valeur du temps de séjour θ_1

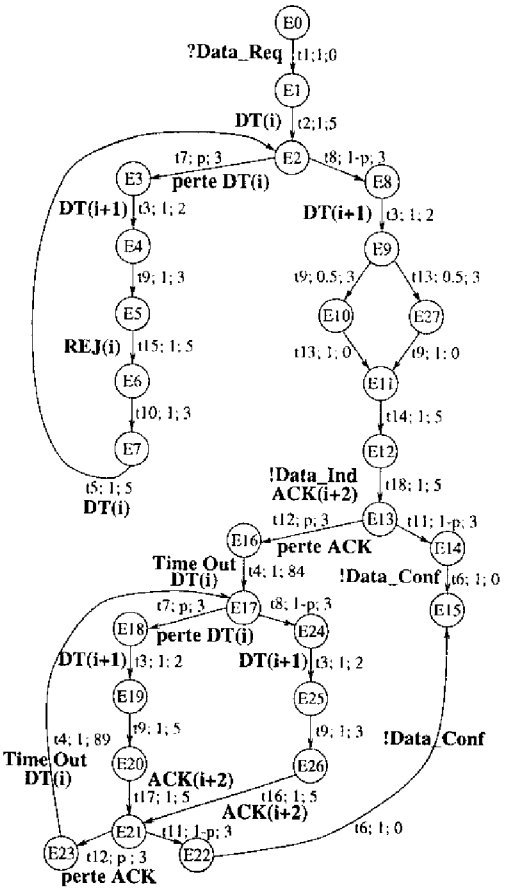


FIG. IV.22 – *Graphe d'états probabilisé moyen du modèle de la figure IV.18*

en fonction de la probabilité de perte et de la largeur de la fenêtre.

A partir de la table des valeurs de θ_1, on peut facilement paramétrer l'automate quotient quantifié représentant le service (Data_Req, Data_Ind) du protocole en fonction de la probabilité de perte et de la largeur de la fenêtre que l'on veut. Si on trace la courbe de variation de θ_1 en fonction de p_p pour chaque largeur de fenêtre, on peut par la suite connaître immédiatement la valeur de θ_1 pour n'importe quelle valeur de la probabilité de perte p_p, sans avoir à construire un graphe et à faire une mesure. Cette remarque met en exergue le pouvoir d'abstraction de l'automate quotient quantifié, à la fois en terme qualitatif (toujours le même automate) et en terme quantitatif (valeur de θ_1 paramétrée par une courbe).

L'automate quotient quantifié nous donne le comportement quantifié du modèle du système relativement à un ensemble d'événements. Mais il ne nous permet pas de nous focaliser sur une mesure quantitative particulière. Il peut donc être nécessaire d'effectuer une deuxième passe d'analyse sur cet automate, à l'aide des règles de réduction de Beizer,

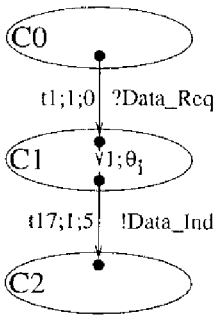


FIG. IV.23 - *Automate quotient quantifié du protocole Go Back N*

p_p	N=2	N=3	N=4
10^{-5}	18.00021	23.00026	28.0003
10^{-4}	18.0021	23.0026	28.003
10^{-3}	18.021	23.026	28.03
10^{-2}	18.21	23.26	28.3
10^{-1}	20.33	25.89	31.44

FIG. IV.24 - *Valeurs du temps de séjour θ_1*

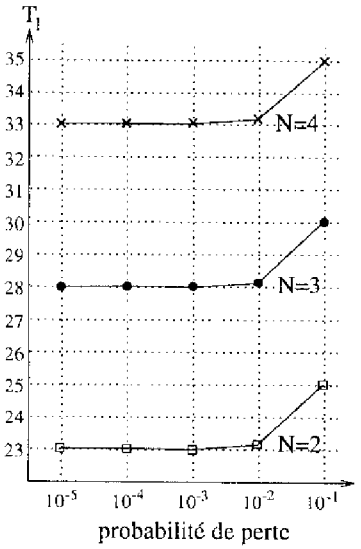


FIG. IV.25 - *Variation de T_1 en fonction de la probabilité de perte*

p_p	pente
N=2	21
N=3	26
N=1	30

FIG. IV.26 - *Pente de la droite T_1 en fonction de la largeur de la fenêtre*

pour pouvoir tirer une mesure quantitative significative du comportement du système.

Si on veut, par exemple, calculer le temps moyen T_1 entre l'occurrence de la primitive Data_Req et l'occurrence de la primitive Data_Ind, nous devons appliquer la règle de réduction série de Beizer sur l'automate quotient quantifié entre l'état d'entrée de la classe C_1 et l'état d'entrée de la classe C_2. De cette seconde passe d'analyse, on tire la valeur de T_1 qui est égale à $\theta_1 + 5$.

Il est clair que cette seconde passe d'analyse ne peut être automatisée. Elle dépend de la mesure que l'on souhaite effectuer.

La courbe de la figure IV.25 montre la variation de T_1 en fonction de la probabilité de perte, pour chaque largeur de fenêtre considérée. T_1 croit avec la probabilité p_p, et avec

la largeur de messages. Plus il y a de pertes, plus il y a de ré-émissions et plus le temps de réception de tous les fragments est long. Plus la fenêtre est grande, et plus il y a de fragments à envoyer, et donc plus le temps de réception de tous ces fragments est long.

Notons que l'axe des abscisses de la figure IV.25 est à l'échelle logarithmique (équidistance entre les puissances de 10) alors que l'axe des ordonnées est à une échelle normale. Les courbes de variation de T_1 sont en fait des droites (allure d'exponentielles ici). La variation de T_1 en fonction de p_p est donc uniforme. Les pentes de ces droites sont données sur la table de la figure IV.26. On remarque que plus la largeur de la fenêtre est grande, plus la pente est élevée. Dans ce protocole, plus la fenêtre est grande, et plus le temps de service est sensible aux pertes.

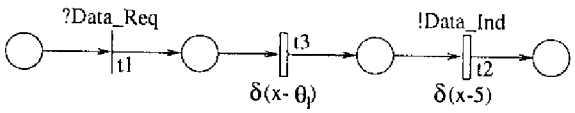


FIG. IV.27 – *Réseau de Petri équivalent à l'automate qualitatif quantifié*

On peut tirer de l'automate quotient de la figure IV.23 un réseau de Petri Temporisé Stochastique équivalent du point de vue du service (Data_Req, Data_Ind) au transfert de donnée. Chaque état apparaissant dans les classes de l'automate devient une place. Chaque arc entre deux états devient une transition entre deux places. On peut avoir deux types de transitions : les transitions immédiates, qui correspondent à des arcs de l'automate de durée nulle, et les transitions temporisées, qui correspondent à des arcs de l'automate de durée non nulle. Si ces transitions temporisées ne font pas partie d'une structure de choix (plusieurs transitions possibles à partir d'un état) dans l'automate, elle ont pour densités de probabilité une densité discrète de la forme $\delta(x - \theta_i)$. Nous verrons sur un exemple plus loin dans ce chapitre quel type de densité est associée à des transitions participant à un choix.

IV.6.5.2 Étude du service (Data_Req, Data_Conf)

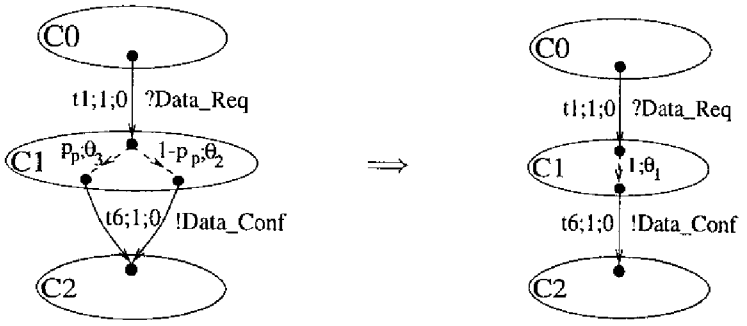


FIG. IV.28 – *Automate quotient quantifié du protocole Go Back N (service (Data_Req, Data_Conf))*

p_p		10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
θ_1	N=2	26.0012	26.012	26.12	27.2	39.5
	N=3	31.0013	31.013	31.13	32.32	45.61
	N=4	36.0014	36.014	36.14	37.42	51.72
θ_2	N=2	26.00021	26.0021	26.021	26.21	28.33
	N=3	31.00026	31.0026	36.026	31.26	33.89
	N=4	36.00031	36.0031	36.031	36.31	39.44
θ_3	N=2	126.0013	126.013	126.13	127.21	140
	N=3	136.0014	136.014	136.14	137.37	151.11
	N=4	146.0015	146.015	146.15	147.5	162.22

TAB. IV.1 *Valeurs des temps de séjour θ_1, θ_2 et θ_3*

Nous reprenons les mêmes analyses que celles qui ont été faites pour le service (Data_Req, Data_Ind) : à partir des mêmes graphes d'états probabilisés, nous construisons un automate quotient quantifié pour chaque valeur de la probabilité de perte et chaque largeur de fenêtre, en se focalisant sur le service (Data_Req, Dat_Conf), aussi appelé service local. La structure de ces automates est identique (automate de gauche de la figure IV.28), seules les valeurs des temps de séjour θ_2 et θ_3 dans la classe diffèrent (table IV.1). On remarque que la transition t_6 (Data_Conf) peut être tirée à partir de deux états différents de la classe C_2. Ces deux états symbolisent l'occurrence ou pas de la perte du PDU ACK (deux durées différentes entre l'occurrence des deux primitives).

Notons que comme il n'y a qu'un seul état d'entrée dans la classe C_1 et un seul état d'entrée dans la classe C_2, on peut simplifier cet automate en appliquant les règles de Beizer, pour calculer le temps et la probabilité entre ces deux états. On obtient alors un seul arc entre ces deux états, qui porte trois étiquettes : une étiquette qualitative (!Data_Conf) et deux étiquettes quantitatives (une probabilité de 1 et un temps $\theta_1 = p_p\theta_3 + (1 - p_p)\theta_2$). Afin de rendre l'automate plus explicite du point de vue du service, on transforme cet arc en deux arcs successifs, le premier portant les étiquettes quantitatives et le second l'étiquette qualitative (automate de droite de la figure IV.28). Nous insistons sur le fait que :

- le passage de l'automate de gauche à l'automate de droite de la figure IV.28 se fait à l'aide des règles de Beizer, mais aussi à l'aide d'une interprétation de l'automate obtenu par ces réductions. Les deux automates ne sont donc pas équivalents au sens de Beizer.

- les réductions de Beizer appliquées découlent d'une interprétation du premier automate (de gauche sur le figure IV.28), et ne peuvent pas elles non plus être automatisées.

Le temps moyen T_2 entre l'occurrence de la primitive Data_Req et l'occurrence de la primitive Data_Conf est égal à $\theta_1 = p_p\theta_3 + (1 - p_p)\theta_2$ (d'après l'automate de droite de la figure IV.28). Les courbes donnant T_2 en fonction de la probabilité de perte, pour chaque largeur de fenêtre, sont données sur la figure IV.29. Comme pour T_1 dans le cas du service (Data_Req, Data_Ind), ces courbes sont des droites (échelle logarithmique des abscisses) dont les pentes sont données par la table de la figure IV.30. La variation de T_2 en fonction

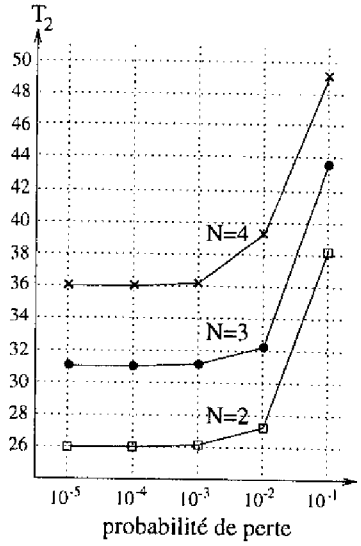


FIG. IV.29 – *Variation de T_2 en fonction de la probabilité de perte*

p_p	pente
N=2	121
N=3	130
N=4	140

FIG. IV.30 – *Pente de la droite T_2 en fonction de la largeur de la fenêtre*

de la probabilité de perte p_p est uniforme. La sensibilité de T_2 à p_p est d'autant plus grande que la largeur de la fenêtre est grande.

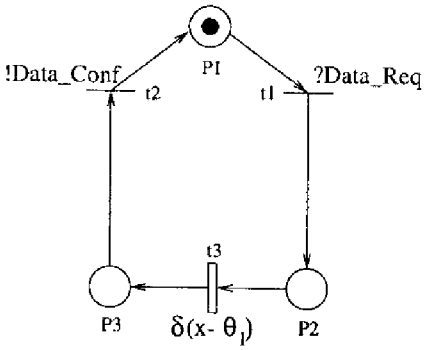


FIG. IV.31 – *Réseau de Petri équivalent du service (Data_Req, Data_Conf)*

On peut tirer de l'automate de droite de la figure IV.28 un réseau de Petri équivalent du transfert de donnée du point de vue du service (Data_Req, Data_Conf). Ce réseau de Petri est donné sur la figure IV.31. La place marquée P_1 représente l'entité de communication. Notons que l'entité émettrice, après avoir reçu une requête (tir de la transition t_1), ne redeviendra disponible qu'après l'envoi de la confirmation (tir de la transition t_2). Entre temps, l'entité est indisponible, ce qui est modélisé par le fait que la place P_1 est démarquée.

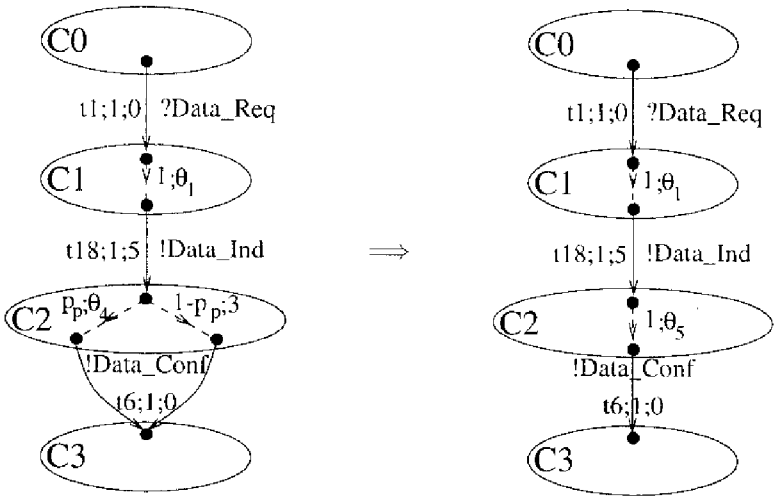


FIG. IV.32 – *Automate quotient quantifié du protocole Go Back N (service (Data_Req, Data_Ind, Data_Conf))*

p_p		10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
θ_4	N=2	103.001	103.01	103.1	104.06	114.67
	N=3	108.0011	108.011	108.11	109.11	120.22
	N=4	113.00115	113.0115	113.115	114.15	125.78
θ_5	N=2	3.001	3.01	3.1	4.01	14.17
	N=3	3.0011	3.011	3.11	4.06	14.72
	N=4	3.0011	3.011	3.11	4.11	15.28

FIG. IV.33 – *Valeurs des temps de séjour θ_4 et θ_5*

IV.6.5.3 Étude du service (Data_Req, Data_Ind, Data_Conf)

Cette fois, nous construisons, à partir des graphes d'états probabilisés, l'automate quotient quantifié sur la totalité du service modélisé, à savoir (Data_Req, Data_Ind, Data_Conf). Cet automate a toujours la même structure (partie de gauche de la figure IV.32), et seules les valeurs des temps de séjour dans les classes varient (la table IV.33 donne les valeurs de θ_4; les valeurs de θ_1 sont données dans la table de la figure IV.24).

Notons que nous pouvons, à partir de cette vue, calculer les valeurs de T_1 et de T_2 : T_1 est le temps moyen entre l'état de sortie de la classes C_0 et l'état d'entrée de la classe C_2, alors que T_2 est le temps entre l'état de sortie de la classe C_0 et l'état puits de la classe C_3. L'application des règles de réduction de Beizer nous permettrait de retrouver les mêmes résultats que ceux présentés sur les courbes IV.25 et IV.29. Cette vue du service global nous donne un automate plus complexe que ceux obtenus pour les vues locale (Data_Req, Data_Conf) et distribuée (Data_Req, Data_Ind), mais nous offre les mêmes possibilités en

termes d'analyses quantitatives.

On peut tirer de l'automate de droite de la figure IV.32 un réseau de Petri équivalent (figure IV.34). Ce réseau de Petri donne une vue abstraite du service offert par le transfert de donnée. Les places P_1 et P_2 représentent les deux entités communicantes. Ce réseau est la vue abstraite la plus simple et la plus réduit du service offert par le protocole Go Back N.

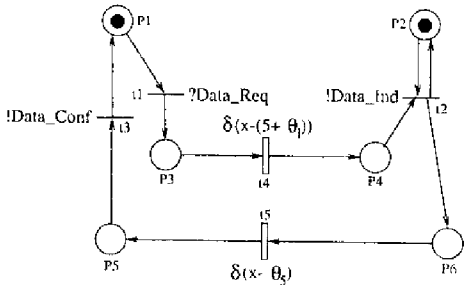


FIG. IV.34 – *Réseau de Petri équivalent du service (Data_Req, Data_Ind, Data_Conf)*

IV.7 Application 3 : modélisation ascendante d'une architecture multicouches

IV.7.1 L'architecture considérée

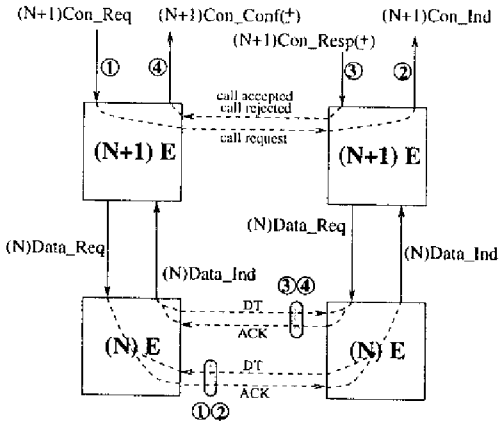


FIG. IV.35 – *Architecture de communication*

Considérons la couche (N) d'une architecture de communication (figure IV.35) qui fournit un service (N) de transfert de données unidirectionnel. Ce service, fourni à la couche (N+1), est constitué de deux primitives : (N)Data_Req et (N)Data_Ind. Le transfert de données est basé sur le protocole « stop and wait » [BSW69] et une fragmentation des données.

IV.7.2 Modélisation de la couche (N)

La figure IV.36 donne le réseau de Petri modélisant la couche (N). La transition t_{11} modélise la réception et le découpage en N fragments (DT(1), ... DT(N)) de la donnée par l'entité émettrice sur réception de la primitive (N)Data_Req (un fragment dans la place P_{10} prêt à être émis, N-1 fragments dans la place P_{11} en attente d'émission). Le marquage de la place P_{11} donne à tout instant le nombre de fragments restant à émettre. La transition t_1 modélise l'émission d'un fragment. La réception de l'accusé de réception d'un fragment de donnée (PDU ACK(i)) en provenance de l'entité réceptrice se fait au niveau de la place P_4. S'il reste encore des fragments à émettre (marquage de P_{11} non nul), l'entité émettrice se prépare à émettre un nouveau fragment (transition t_2). Sinon elle se replace en attente de la prochaine requête (transition t_{12}).

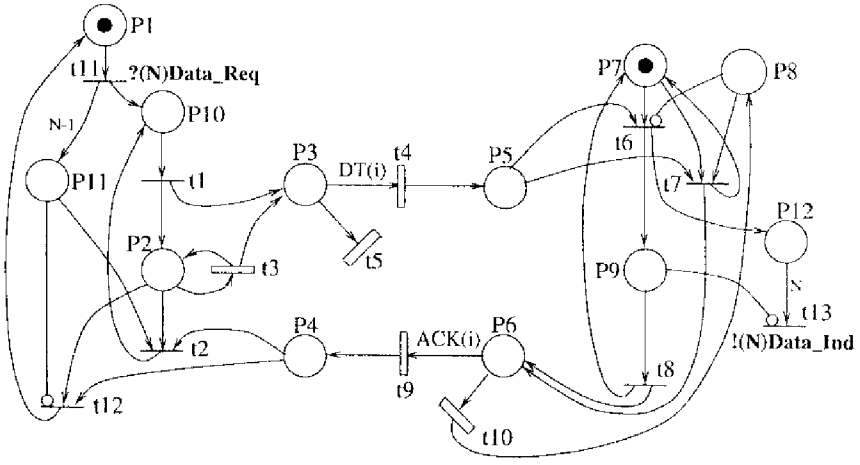


FIG. IV.36 – *Modèle de la couche (N)*

La transition t_6 représente la réception d'un fragment de donnée par l'entité réceptrice. La transition t_8 permet à cette dernière d'envoyer un accusé de réception à l'entité émettrice, et de se placer en attente d'un nouveau fragment (place P_7 marquée). A chaque arrivée d'un nouveau fragment (transition t_6), un jeton est ajouté dans la place P_{12}. Lorsque le marquage de cette place est égal à N (N fragments arrivés), la donnée est réassemblée et délivrée à l'entité supérieure (envoi de la primitive (N)Data_Ind lors du tir de t_{13}). Notons que l'arc inhibiteur arrivant sur la transition t_{13} permet de ne délivrer la donnée reconstituée qu'après avoir envoyé l'accusé de réception du dernier fragment reçu (place P_9 vide).

La transmission des données par le médium est modélisée par les transitions t_4 et t_9. Notons que le médium peut perdre des PDUs (transitions t_5 et t_{10}). De plus, la perte du PDU ACK(i) (t_{10}) est signalée à l'entité réceptrice (place P_8 marquée par le tir de t_{10}).

Un mécanisme de ré-émission (TIME OUT) est mis en place au niveau de l'entité émettrice (transition temporisée t_3), et permet de ré-émettre un fragment si son accusé de réception n'est pas reçu au bout d'un certain temps. Si l'entité réceptrice reçoit un fragment (place P_5 marquée) après que le médium lui ait signalé la perte d'un accusé de

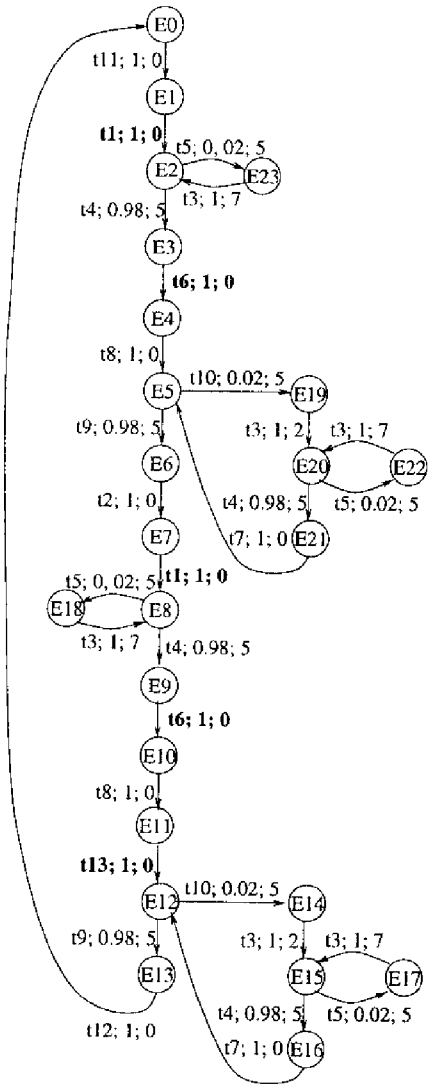


FIG. IV.37 - *Graphe d'états probabilisé moyen du modèle de la couche (N)*

densité de probabilité	transitions
$\delta(x)$	t_1, t_2, t_6, t_7 t_8, t_{11}, t_{12}, t_{13}
$\delta(x-5)$	t_4, t_9
$2p_p\delta(x-5)+\frac{1-2p_p}{45}$ $(5<x<50)$	t_5, t_{10}
$\delta(x-12)$	t_3

FIG. IV.38 - *Densités de probabilités associées aux transitions*

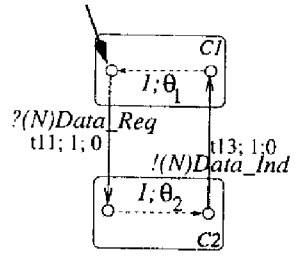


FIG. IV.39 - *Vue abstraite qualitative quantifiée de la couche (N)*

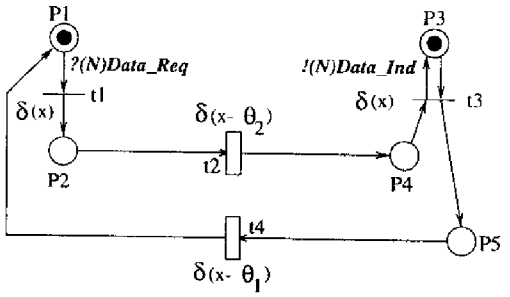


FIG. IV.40 - *Modèle équivalent de la couche (N)*

réception (place P_8 marquée), cet accusé est émis à nouveau (transition t_7) sans que le fragment ne soit comptabilisé comme nouveau fragment (pas de jeton ajouté dans la place P_{12}).

IV.7.3 Analyse de la couche (N)

Le graphe d'états probabilisé moyen issu de ce modèle, pour un nombre de fragments N=2, est donné sur la figure IV.37. Le nombre d'états de ce graphe, pour des valeurs de

N de 1 à 5, est donné dans la table IV.41. Les densités de probabilité associées à chaque transition du modèle sont données dans la table de la figure IV.38. La probabilité de perte d'un PDU (modélisée comme nous l'avons indiqué dans le paragraphe IV.6.3) issue de ces densités de probabilité est $p_p = 2.10^{-2}$.

Calculons la vue abstraite qualitative quantifiée de ce graphe d'états probabilisé en nous focalisant sur les primitives de service, à savoir (N)Data_Req (transition t_{11}) et (N)Data_Ind (transition t_{13}). L'automate obtenu est décrit sur la figure IV.39. La flèche en gras indique l'état initial. Les temps θ_1 et θ_2, temps de séjour dans les classes C_1 et C_2, dépendent du nombre de fragments N et de la probabilité de perte p_p. Nous donnons dans la table IV.41 les valeurs de ces temps en fonction du nombre de fragments, pour une probabilité de perte $p_p = 2.10^{-2}$.

Nombre de fragments	1	2	3	4	5
Nombre d'états	13	24	35	46	57
θ_1	5.25	5.25	5.25	5.25	5.25
θ_2	5.24	15.74	26.23	36.73	47.22

FIG. IV.41 - *Nombres d'états du graphe d'états et valeurs des temps θ_1 et θ_2 en fonction du nombre de fragments*

De cet automate, on peut tirer un modèle Réseau de Petri Temporisé Stochastique équivalent à la couche (N) du point de vue de son service (on utilise la même technique que celle décrite dans le paragraphe IV.6.5.1 pour le protocole Go Back N). Ce réseau de Petri est donné sur la figure IV.40. Notons que la structure de ce modèle ne dépend pas de la valeur de la probabilité de perte.

IV.7.4 Modèle de la couche (N+1)

Supposons que l'entité (N+1) doive ouvrir une connexion en utilisant le service de transfert de données de la couche (N) que nous venons de décrire. La figure IV.42 donne le réseau de Petri modélisant cette ouverture de connexion au niveau de la couche (N+1). Nous ne représentons que la logique de cette ouverture. Le traitement des données (encapsulation, ...) n'est pas notre propos ici.

Sur arrivée d'une requête de connexion, l'entité initiatrice émet une requête à la couche (N) (call request : transition t_{101}). En fonction de la réponse de l'entité réceptrice (call accepted : transition t_{103} ou call rejected : transition t_{102}), une confirmation positive ou négative ((N)Data_Conf(+) ou (N)Data_Conf(-)) est envoyée à l'entité supérieure, et la connexion est ouverte ou pas.

Du côté de l'entité réceptrice, l'arrivée de la demande de connexion est indiquée à l'entité supérieure par la primitive (N+1)Con_Ind (transition t_{104}). La réponse à cette primitive ((N)Data_Resp(+) ou (N)Data_Resp(-)) implique une requête à la couche (N) : soit un call rejected (transition t_{105}) en cas de réponse négative, soit un call accepted (transition t_{106}) en cas de réponse positive.

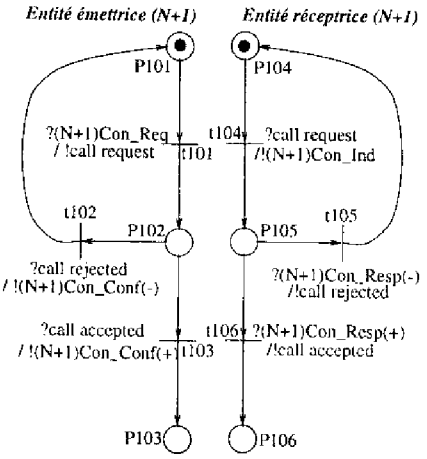


FIG. IV.42 – *Modèle de la couche (N+1)*

Le modèle réseau de Petri de la couche (N) utilisé pour le transfert de la requête *call request* est le modèle de la figure IV.36. Le modèle réseau de Petri de la couche (N) pour le transfert des deux requêtes *call accepted* et *call rejected* (figure IV.43) se déduit facilement du modèle présenté sur la figure IV.36.

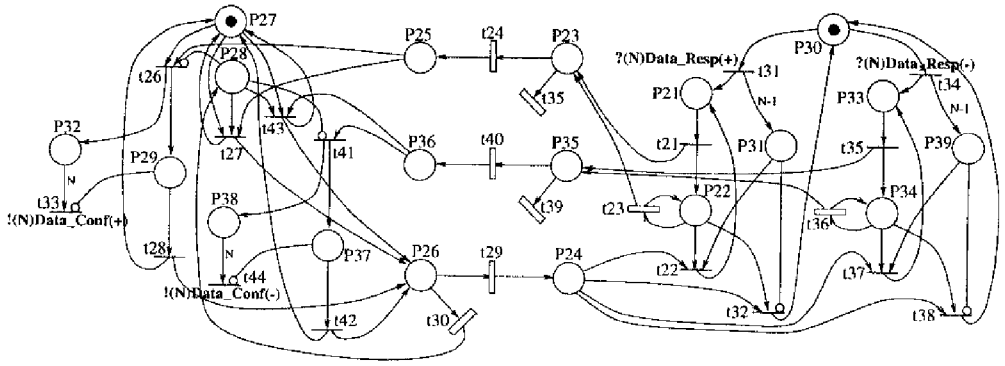


FIG. IV.43 – *Modèle de la couche (N) pour le transfert du call rejected et du call accepted*

IV.7.5 Méthodologies pour l'analyse de l'architecture (N)-(N+1)

IV.7.5.1 Description des méthodologies

L'étude du mécanisme d'ouverture de connexion peut être effectuée à partir de deux modèles :

- une méthode directe consiste à interconnecter par places partagées les réseaux de Petri modélisant les deux couches (N) et (N+1), pour obtenir un modèle global. Les analyses se font alors à partir de ce modèle global;

- une méthode de modélisation ascendante consiste, dans notre exemple, en trois phases:

  on calcule la vue abstraite qualitative quantifiée de la couche (N), en se focalisant sur le service (N). Ici, comme la couche (N) est modélisée par deux réseaux de Petri (un pour le transfert du call request, un pour le transfert du call accepted et du call rejected), on obtiendra deux automates différents;

  - on tire de cette vue le réseau de Petri équivalent de la couche (N). Ici, on aura donc deux réseaux de Petri équivalents;
  - on interconnecte par places partagées ces réseaux équivalents au modèle de la couche (N+1).

Le modèle obtenu est un modèle équivalent au couple couche (N)-couche (N+1). Les analyses se font à partir de ce modèle équivalent.

IV.7.5.2 Méthode directe

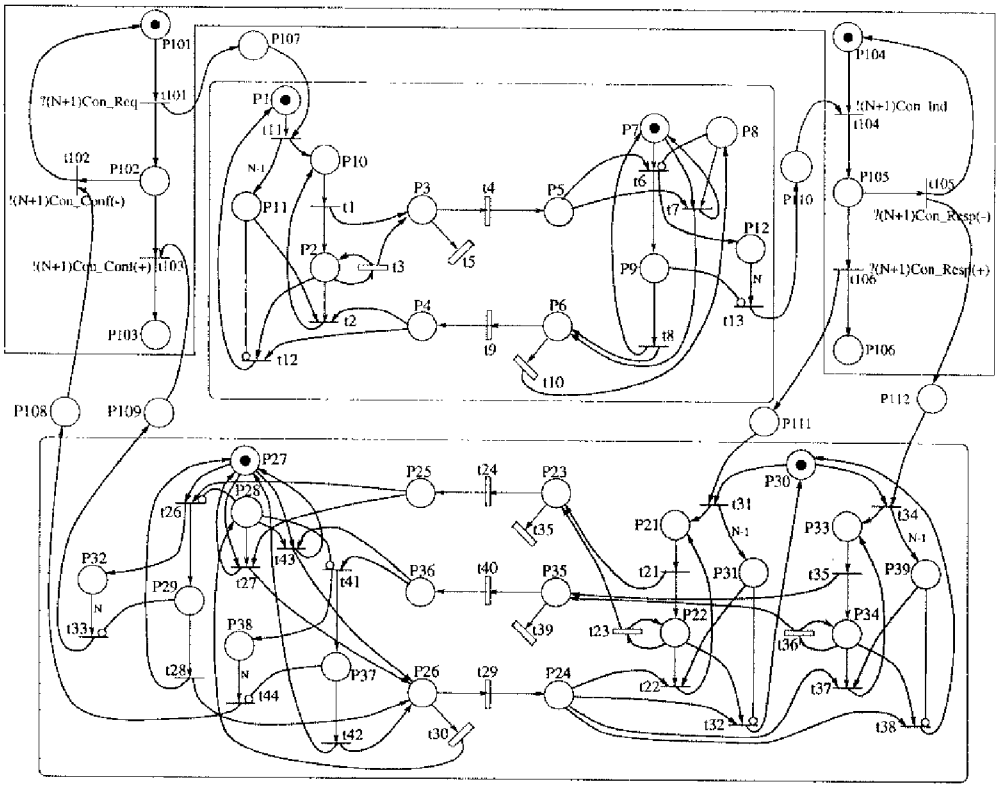


FIG. IV.44 – *Modèle global du couple couche (N)-couche (N+1)*

Le modèle global obtenu en interconnectant les différents modèles des couches (N) et (N+1) préserve toute la sémantique des mécanismes modélisés, puisqu'ils apparaissent explicitement sur le modèle d'analyse. On peut ainsi faire une étude détaillée du couple

couche (N)-couche (N+1), notamment en ce qui concerne les cas pires, en associant à chaque transition une règle de tir particulière. On doit néanmoins construire un graphe d'états probabilisé pour chaque scénario étudié. L'inconvénient majeur de cette méthode est l'explosion en nombre de places et de transitions au niveau du modèle, mais aussi en nombre d'états au niveau des graphes d'états probabilisés construits. Cette explosion peut être telle que les graphes d'états ne peuvent pas être construits (limitations « informatiques » et « humaines »), et donc qu'aucune analyse du comportement du système ne peut être faite. L'utilisation de cette technique est donc limitée à de petits modèles, avec par exemple seulement deux couches interconnectées.

La figure IV.44 donne le modèle global pour notre exemple. On voit facilement sur cet exemple que nous sommes déjà à la limite des dimensions raisonnables pour une bonne compréhension, alors que nous n'avons interconnecté que deux couches, en ne représentant qui plus est qu'une partie des mécanismes de la couche (N).

IV.7.5.3 Méthode ascendante

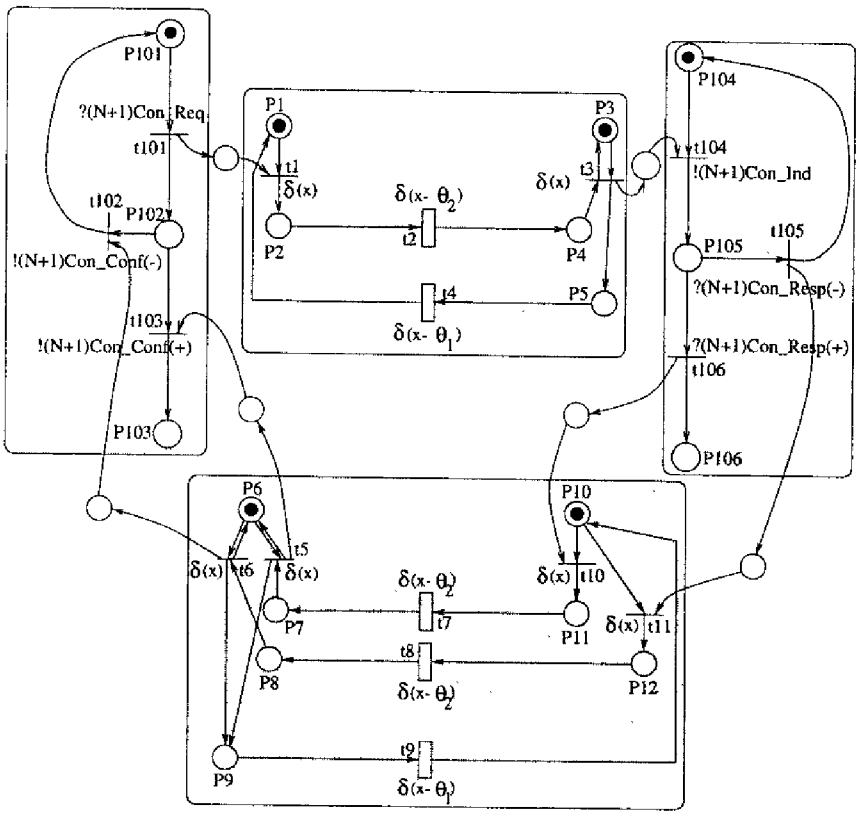


FIG. IV.45 – *Modèle global équivalent du couple couche (N)-couche (N+1)*

Le modèle global obtenu en interconnectant les modèles équivalents de la couche (N) au modèle de la couche (N+1) est beaucoup plus petit en nombre de places et de transitions. En effet, chaque abstraction d'une couche permet d'obtenir un réseau de Petri équivalent

de taille plus petite. Ce réseau modélise toutes les couches inférieures à la couche étudiée. Le gain en nombre de places et de transitions est alors très important. Le nombre d'états du graphe d'états probabilisé associé est lui aussi beaucoup plus petit, ce qui permet de faire des analyses plus facilement que sur des graphes de plus grande dimension. Il faut aussi noter que la structure de ce réseau équivalent ne dépend pas des valeurs temporelles choisies. On peut donc paramétrer facilement ce modèle pour pouvoir faire diverses analyses quantitatives (ce paramétrage a été étudié dans le paragraphe IV.6.5.1), notamment en ce qui concerne les cas pires. Mais il ne faut pas oublier que les réductions de Beizer permettent de calculer un temps et une probabilité moyens entre deux états. Même si on étudie un cas pire, ces réductions donneront un temps et une probabilité moyens du cas pire !

La figure IV.45 montre le réseau de Petri global équivalent obtenu à partir des vues abstraites qualitatives quantifiées des deux réseaux modélisant la couche (N).

IV.7.5.4 Comparaison des deux méthodes

nombre de fragments	2	3	4	5
méthode directe	698	931	1139	1374
méthode ascendante	26			

TAB. IV.2 – *Nombre d'états dans les modèles globaux en fonction du nombre de fragments*

La table IV.2 présente une comparaison du nombre d'états obtenus pour les graphes d'états probabilisé associés aux deux modèles globaux des figures IV.44 et IV.45. Le nombre d'états, dans le cas du modèle global équivalent, ne varie pas avec le nombre de fragments, puisque la structure des modèles équivalents de la couche (N) ne dépend pas de ce nombre. On constate que très rapidement, seul le modèle global équivalent offre un graphe d'états probabilisé tractable.

IV.8 Conclusion

Nous avons présenté dans ce chapitre la notion d'automate quotient quantifié, qui permet d'obtenir une vue abstraite à la fois qualitative et quantitative d'un graphe d'états initial avec étiquettes qualitatives et quantitatives. Nous l'appliquons ici dans le cadre du modèle Réseaux de Petri Temporisés Stochastiques, qui permet de construire de tels graphes d'états.

Il nous paraît important de mettre en exergue le pouvoir d'abstraction d'un tel automate, à la fois en terme qualitatif (pouvoir issu de la relation d'équivalence observationnelle de Milner) et en terme quantitatif (pouvoir issu des règles de réduction de Beizer), ainsi que le passage d'un automate quantifié à un réseau de Petri.

La première application de cette notion est l'obtention des réseaux de Petri équivalents d'une couche, du point de vue du service qu'elle offre à la couche supérieure. Nous avons

obtenu des vues équivalentes du service local (Data_Req, Data_Conf), du service distribué (Data_Req, Data_Ind) et du service global (Data_Req, Data_Ind, Data_Conf). Cette application peut être étendue à tout type de service.

La seconde application que nous avons proposée est la modélisation ascendante d'une structure en couches. Cette méthodologie peut être généralisée à tout système hiérarchisé, où chaque bloc feuille (de niveau le plus bas) sera modélisé et analysé séparément pour en tirer un réseau de Petri équivalent. Les modèles des blocs hiérarchiquement supérieurs utiliseront ces réseaux équivalents pour être eux mêmes analysés, et donner un réseau de Petri équivalent de leur comportement. Cette technique apporte beaucoup dans le contrôle des dimensions du modèle du système.

Par ses propriétés, l'automate quotient quantifié est un outil d'analyse indispensable dans l'analyse des systèmes temps-réel. Il permet de valider à la fois des propriétés qualitatives et quantitatives, même si la validation des propriétés quantitatives peut nécessiter une seconde phase d'analyse à partir de cet automate (pour tirer des mesures particulières des automates). Nous l'avons d'ailleurs utilisé dans le cadre de la modélisation et l'analyse du protocole embarqué temps-réel ARINC 629 CP, sujet du chapitre V de ce manuscrit.

Chapitre V

Application du modèle RdPTS : modélisation et analyse du protocole ARINC 629 CP

V.1 Introduction

Cette étude a été effectuée dans le cadre d'un contrat entre le groupe *Outils Logiciels pour la Communication* du LAAS-CNRS et *Aérospatiale* [GBJ96c, GBJ96a, GBJ96b, GJB96, Gal97]. Elle a pour but de montrer, sur exemple concret, l'adéquation du modèle RdPTS à la modélisation et l'analyse des systèmes distribués temps-réel.

La spécification ARINC 629 [Air91] définit un réseau embarqué temps critique pour le transfert de données entre systèmes avioniques. La sous-couche MAC définit dans cette norme permet de contrôler l'accès des stations au médium de communication. Elle est basée sur un mécanisme CSMA/CA (Carrier Sense Multiple Access Collision Avoidance). Les stations peuvent émettre deux types de messages: des messages périodiques et des messages apériodiques. Deux protocoles de la sous-couche MAC ont été définis pour permettre de faire passer les flux périodique et apériodique. Le premier, le protocole BP (*Basic Protocol*), fait basculer les stations d'un mode périodique à un mode apériodique. Les stations émettent les messages appropriés dans chaque mode. Le second, le protocole CP (*Combined Protocol*), permet de mélanger les deux types de flux sur le médium sans basculement de mode. C'est ce protocole que nous avons étudié.

Le but de l'étude que nous avons effectuée est de modéliser par Réseaux de Petri Temporisés Stochastiques le protocole CP, afin de pouvoir valider, d'une part, son comportement normal, et, d'autre part, son comportement vis à vis de situations exceptionnelles.

Ce chapitre se décompose en trois parties. Dans la première, nous donnons l'architecture et les principaux mécanismes du protocole CP. Dans la deuxième partie, nous dé-

taillerons les principaux modèles Réseaux de Petri Temporisés Stochastiques du protocole. La troisième partie est consacrée à l'analyse des comportements normaux et exceptionnels du protocole à partir de ces modèles. Cette analyse est basée principalement sur l'interprétation de vues abstraites obtenues à partir du graphe d'états probabilisé. Ce sont soit des vues abstraites qualitatives quantifiées, soit des vues abstraites qualitatives (ceci afin de ne pas trop alourdir la présentation).

Remarque : ce protocole est caractérisé « Collision Avoidance » car en fonctionnement normal, il n'y a pas de collision. Des collisions peuvent par contre apparaître, comme nous le verrons, lors de situations transitoires particulières.

V.2 Le protocole CP de la sous-couche MAC de la norme ARINC 629

V.2.1 Principaux mécanismes

Le point essentiel du protocole CP est le mécanisme d'écoute du bus : chaque station est en permanence à l'écoute des différents messages qui circulent sur le médium de communication. A partir des signaux BA (« Bus Active » : bus actif), BQ (« Bus Quiet » : bus silencieux) et BC (« Bus Clash » : collision), et l'utilisation de cinq temporisations sur chaque station, les stations peuvent se synchroniser, ou au contrainte être en compétition pour l'accès au bus.

L'évolution des stations sur le bus est basée sur différents mécanismes que nous allons étudier maintenant.

V.2.1.1 Concept de cycle-bus

Le concept principal du protocole CP est le concept de cycle-bus. Un cycle-bus défini une fenêtre temporelle dans laquelle les stations peuvent émettre leurs messages. Cette fenêtre est positionnée par une station particulière, appelée *station leader*.

Un cycle-bus est décomposé en quatre parties distinctes :

- la première partie correspond aux émissions périodiques des stations. Chaque station doit obligatoirement émettre un message périodique par cycle-bus. Cette partie est appelée niveau périodique, ou encore niveau L1 (« Level 1 »);
- la deuxième partie correspond aux émissions apériodiques urgentes des stations. Une station ne peut émettre qu'un message apériodique urgent dans un cycle-bus, mais ce n'est pas une obligation. Cette partie est appelée niveau apériodique urgent, ou encore niveau L2 (« Level 2 »);
- la troisième partie correspond aux émissions apériodiques non urgentes des stations. Une station peut émettre un ou plusieurs messages apériodiques non urgents dans un cycle-bus, mais ce n'est pas une obligation. Cette partie est appelée niveau apériodique non urgent, ou encore niveau L3 (« Level 3 »); Cette partie est divisée en

deux niveaux : le niveau L3 Backlog pour les messages de niveau L3 qui n'ont pu être émis dans le cycle-bus précédent, et le niveau L3 New pour les nouveaux messages de niveau L3. Le niveau L3 Backlog précède toujours le niveau L3 New, pour pouvoir émettre en priorité les messages les plus anciens. Si le temps le permet, le niveau L3 New peut se répéter pour les stations qui ont plusieurs messages apériodiques non urgents à émettre;

- la quatrième partie permet aux stations de se synchroniser avant de passer au cycle-bus suivant (pas d'émission de message, donc temps de silence sur le bus).

Le signal de début d'un nouveau cycle-bus est donné par la station leader, quand celle-ci émet son message périodique. Tout cycle-bus commence donc par l'émission périodique de la station leader. Dès qu'elles détectent cette émission (apparition de l'événement CE : « Concatenation Event »), les stations non leader débutent elles aussi un nouveau cycle-bus.

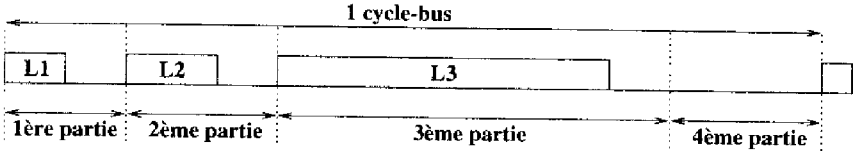


FIG. V.1 – *Décomposition d'un cycle-bus*

La station leader est désignée au moyen d'un mécanisme d'élection à la fin de chaque cycle-bus. Toute station peut devenir station leader. Un changement de station leader ne peut avoir lieu qu'à la suite d'une situation exceptionnelle (panne d'une station, ...). La figure V.1 illustre l'enchaînement des quatre parties dans un cycle-bus.

Les stations ne peuvent pas émettre plus d'un message par niveau. L'émission périodique dans le niveau L1 est obligatoire pour toutes les stations.

V.2.1.2 Les temporisations du protocole

On peut regrouper les cinq temporisations d'une station en trois groupes :

- la temporisation TG (« Terminal Gap ») permet de gérer la compétition d'accès au bus (elle implémente ce que nous avons appelé mécanisme de ségrégation d'accès au bus);

- les temporisations ASG (« Aperiodic Synchronization Gap ») et PSG (« Periodic Synchronization Gap ») permettent l'évolution de la station dans le cycle-bus. ASG permet aux stations de se synchroniser en fin de niveau, PSG permet aux stations de se synchroniser en fin de cycle-bus;

- les temporisations AT (« Aperiodic access Time-out ») et TI (« Transmit Interval ») sont relatives à la durée du cycle-bus;

Les temporisations TG, ASG et PSG ont un mode de fonctionnement basé sur l'activité du médium de communication : elles commencent à s'écouler dès que le bus est silencieux (signal BQ) et sont rechargées à leurs valeurs maximales quand le bus est actif (signal

BA). Les temporisations TI et AT ne dépendent pas du médium de communication : elles commencent à s'écouler à des instants particuliers du cycle-bus, et ne sont rechargées à leurs valeurs maximales qu'à la fin du cycle-bus.

Chaque temporisation peut être dans quatre états différents : l'état armé, l'état d'écoulement, l'état écoulé, l'état de dérive. Une temporisation est armée si elle est rechargée à sa valeur maximale et est prête à s'écouler. Les états d'écoulement et écoulé correspondent à l'écoulement et la fin d'écoulement de la temporisation. Enfin l'état de dérive correspond à une dérive excessive de la temporisation (nous étudierons ce cas lors de l'étude des situations exceptionnelles).

Le stemporisations sont les éléments les plus critiques du protocole. Nous étudions maintenant plus en détail leur fonctionnement.

La temporisation TG

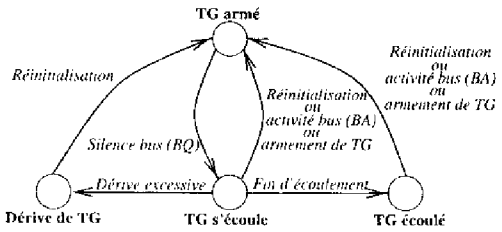


FIG. V.2 – *Machine à états de la temporisation TG*

Le rôle de la temporisation TG est de contrôler l'accès au médium de communication de la station, afin d'éviter toute collision de messages. C'est une temporisation de ségrégation. Quand une station a écoulé son TG, elle peut, si elle le désire, émettre un message. Une station ne peut pas émettre de message si son TG n'est pas écoulé.

La temporisation TG est rechargée à sa valeur maximale au début de chaque cycle-bus, au début de chaque niveau d'un cycle-bus, et à la fin de l'écoulement de la temporisation PSG. Puisque toutes les temporisations TG de toutes les stations ont le même mode de fonctionnement lié à l'activité sur le bus, elles s'écoulent en même temps (au temps de propagation près). Leurs valeurs respectives doivent donc être différentes pour pouvoir mettre en place une priorité d'accès au bus. La station qui a le plus petit TG sera la première à avoir accès au bus, la station qui a le plus grand TG sera la dernière à avoir accès au bus. D'après la spécification ARINC, afin d'être sûr qu'une seule station a accès au bus à la fois, la différence entre deux valeurs successives des TG doit être supérieure à deux fois le temps de propagation maximal :

$$TG_i - TG_j > 2.\tau_{max}$$

Cette différence assure que lorsqu'une station débute une émission, suite à la fin d'écoulement de son TG, le début de son message arrivera à temps sur toutes les autres stations

pour arrêter l'écoulement de leurs TG (activité bus) et les empêcher d'avoir accès au bus.

La figure V.2 donne la machine à état de la temporisation TG. La valeur de TG est fixée par la norme ARINC entre 3.7 et 127.7 μs. La plus petite différence entre deux TG est de 1 μs, ce qui correspond à une longueur maximale de bus entre les deux stations de 50 mètres.

La temporisation ASG

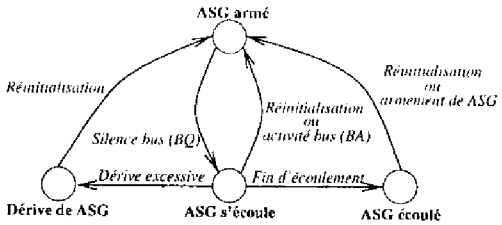


FIG. V.3 – *Machine à états de la temporisation ASG*

Le rôle de la temporisation ASG est de permettre une synchronisation de toutes les stations avant tout passage d'un niveau du cycle-bus au niveau suivant. La valeur de cette temporisation est identique sur toutes les stations, et est supérieure à la plus grande valeur des TG :

$$ASG > TG_{max}$$

Cette relation permet d'éviter la fin de l'écoulement de la temporisation ASG avant la fin de l'écoulement de la temporisation TG (les deux temporisations s'écoulent en même temps puisqu'elles dépendent toutes les deux de l'activité sur le bus), et donc de priver la station d'émission dans un niveau du cycle-bus.

Les débuts et fins d'émissions de messages sont des points de synchronisation des stations, qui arment et écoulent leurs ASG en même temps (au temps de propagation maximal près). La détection d'un temps de silence sur le bus égal à ASG (c'est à dire une écoulement complet de ASG) indique que plus aucune station ne veut émettre dans ce niveau, et que l'on peut passer au niveau suivant.

La temporisation ASG est rechargée à sa valeur maximale au début de chaque niveau d'un cycle-bus et au début de chaque cycle-bus.

La figure V.3 donne la machine à état de la temporisation ASG. Quatre valeurs d'ASG sont données par la norme ARINC : 16, 32, 64 ou 128 μs. La valeur choisie doit respecter la condition $ASG > TG_{max}$.

La temporisation PSG

Le rôle de la temporisation PSG est de permettre aux stations de détecter la fin du cycle-bus courant, et de synchroniser avant de changer de cycle-bus. La valeur de cette

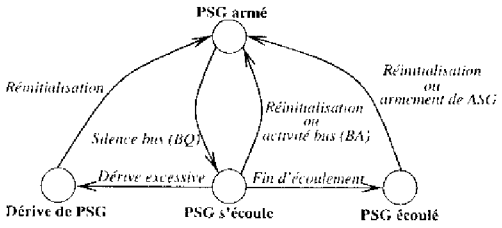


FIG. V.4 – *Machine à états de la temporisation PSG*

temporisation est la même sur toutes les stations, et est égale à :

$$PSG = 5.ASG$$

La valeur $5.ASG$ permet de s'assurer que quand PSG est écoulé, plus aucune station n'a de message à émettre dans ce cycle-bus : cette durée est supérieure au plus grand temps de silence qui peut apparaître sur le bus quand aucune station n'a émis de message dans les niveaux apériodiques (écoulement successif de quatre ASG).

De manière similaire à ASG, un temps de silence sur le bus égal à PSG indique que plus aucune station ne veut émettre de message dans ce cycle-bus, et que l'on peut passer au cycle-bus suivant.

La temporisation PSG est rechargée à sa valeur maximale au début du cycle-bus. Son écoulement est lié à l'activité sur le bus.

La figure V.4 donne la machine à état de la temporisation PSG. Quatre valeurs de PSG sont possibles, en fonction de la valeur de ASG choisie : 80, 160, 320 ou 640 μs.

La temporisation TI

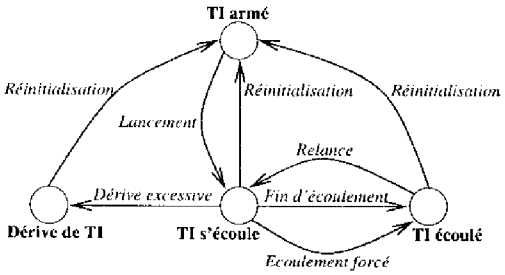


FIG. V.5 – *Machine à états de la temporisation TI*

La temporisation TI donne la durée d'un cycle-bus. C'est grâce à cette temporisation que la station leader sait quand elle doit changer de cycle-bus; elle permet donc d'assurer la périodicité du trafic périodique. En régime normal, seul le TI de la station leader a de l'importance pour le protocole. Néanmoins, cette temporisation s'écoule sur toutes les

stations pour que chacune d'entre elles soit prête à remplacer la station leader en cas de défaillance de cette dernière.

La temporisation TI est rechargée à sa valeur maximale au début de l'émission périodique de la station. Son écoulement se fait en permanence, et est indépendant de l'activité sur le bus. Cette temporisation peut soit arriver à son terme (c'est le cas à la fin de chaque cycle-bus pour la station leader en régime normal), soit être stoppée par le signal de changement de cycle-bus CE (c'est le cas pour les stations non leader à la fin d'un cycle-bus en régime normal). La valeur de TI est la même sur toutes les stations.

La figure V.5 donne la machine à état de la temporisation TI. La valeur de TI est fixée par la norme ARINC entre 0.5 et 64 ms.

La temporisation AT

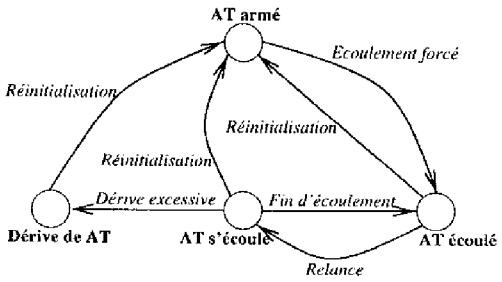


FIG. V.6 – *Machine à états de la temporisation AT*

La temporisation AT a pour rôle de donner le dernier instant possible de début d'émission de message apériodique dans le cycle-bus. La valeur de AT est la même sur toutes les stations. Cette temporisation commence à s'écouler chez la station leader au début de son émission périodique (en même temps que le TI), et chez les stations non leader sur l'occurrence du signal CE (au temps de propagation près, en même temps que sur la station leader). La valeur de cette temporisation est calculée pour que les stations aient le temps de se synchroniser en fin de cycle-bus et d'élire une nouvelle station leader avant la fin du cycle-bus courant, après l'émission du dernier message périodique :

$$AT = TI - MAL - PSG - ASG$$

où MAL correspond à la longueur maximale d'un message.

La figure V.6 donne la machine à état de la temporisation AT. La valeur de AT dépend fortement de la longueur maximale d'un message (257 mots) et du débit d'émission.

Résumé

La figure V.7 illustre le rôle des différentes temporisations du protocole sur un scénario représentant un cycle-bus du régime normal.

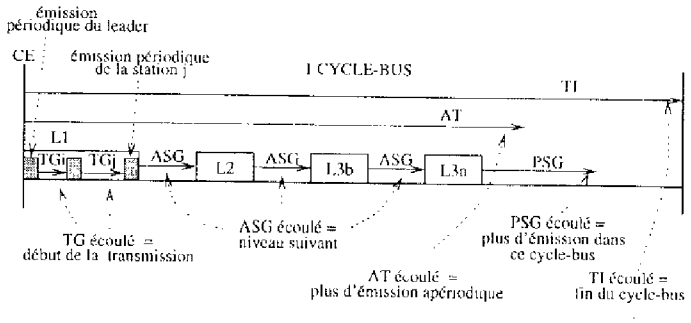


FIG. V.7 - *Rôle des temporisations dans un cycle-bus du régime normal*

V.2.1.3 Élection de la station leader

L'élection de la station leader est effectuée à la fin du cycle-bus courant, quand les stations ont écoulé leur PSG (synchronisation). Cette élection est distribuée sur l'ensemble des stations, et dépend de conditions locales à chacune d'elles. Une station se considère comme station leader si elle vérifie la condition :

TI, PSG et TG écoulés

sachant que la temporisation TG est relancée à la fin de PSG (l'explication de cette relance sera donnée dans l'étude de phénomène de surcharge). Chaque station s'affecte donc en fin de cycle-bus l'attribut leader ou l'attribut non leader. Dans le fonctionnement normal en régime permanent, seule une seule station détecte la condition de leader à la fois. Mais rien n'interdit dans le protocole que plusieurs stations détectent cette condition simultanément, et donc que plusieurs stations se déclarent leader en même temps. Ce cas provoque une collision entre ces stations. Nous reviendrons sur ce scénario lors de l'étude des situations exceptionnelles du protocole.

Dès qu'une station a détecté la condition de leader, elle change de cycle-bus et commence à émettre son message périodique. L'arrivée de ce message sur les autres stations provoque l'apparition de l'événement CE (qui signifie que ces stations sont non leader), et le changement de cycle-bus. Leur émission périodique pourra être effectuée dès que leur temporisation TG (accès au bus) sera écoulée.

V.2.2 L'initialisation

L'initialisation des stations sur le bus est une procédure particulière du protocole. Son but est de permettre aux stations qui viennent d'être mises sous tension de pouvoir s'introduire correctement sur le bus, en provoquant un minimum de perturbation pour les stations déjà initialisées.

Deux cas d'initialisation sont à envisager :

- une ou plusieurs stations s'initialisent sur un bus vide, c'est-à-dire un bus sur lequel aucune autre station n'est déjà initialisée; cette phase peut soit provoquer une

initialisation correcte, avec élection d'un leader parmi les stations qui cherchent à s'initialiser, soit provoquer une collision entre plusieurs stations à la fin de leur initialisation;

- une ou plusieurs stations s'initialisent sur un bus non vide, c'est-à-dire un bus sur lequel d'autres stations sont déjà initialisées; ces stations, à la fin de l'initialisation, s'intègrent dans le train des stations déjà sur le bus en tant que stations non leader.

La procédure d'initialisation d'une station est la suivante : après sa mise sous tension, une station doit écouler son PSG pour se synchroniser en fin de cycle-bus avec les stations (s'il y en a) déjà présentes sur le bus. Tant que PSG ne s'est pas écoulé, la station ne peut pas s'introduire sur le bus. A la fin de cet écoulement, elle cherche à écouler son TI. Ici, deux cas sont possibles :

- s'il y a déjà des stations sur le bus, l'une d'entre elle, la station leader, va commencer à émettre son message périodique avant que le TI de la station qui s'initialise ne s'achève. Cette dernière voit alors apparaître le signal CE, qui signifie l'arrêt de l'écoulement de son TI, la fin de son initialisation, et son entrée sur le bus en tant que station non leader;

- s'il n'y a pas de station déjà sur le bus, le TI arrive à son terme. A la fin de cet écoulement, la station sait qu'il n'y a pas d'autre station déjà sur le bus. Mais d'autres stations peuvent être en train de s'initialiser en même temps qu'elle. La procédure d'élection de la station leader (écoulement du TG) débute alors pour désigner parmi les stations qui s'initialisent la station qui sera leader du premier cycle-bus.

Dans le second cas, plusieurs stations peuvent détecter simultanément (au temps de propagation près) la condition de leader, car l'initialisation est une phase totalement asynchrone, où les stations qui s'initialisent ne se synchronisent pas entre elles (l'écoulement du PSG ne permet une synchronisation qu'avec les stations déjà sur le bus : ici, le bus est vide). Une collision est donc possible entre les stations qui détectent cette condition. Cette collision sera étudié dans le paragraphe V.4.1.3 consacré à l'analyse de la phase de récupération de la collision.

Notons enfin que durant la phase d'initialisation, une variable « Init Set » est activée dans la station, afin de mémoriser qu'elle est en phase d'initialisation. Sa remise à zéro se fait soit sur l'occurrence d'un CE après la fin du PSG, soit lors du lancement du TG à la fin du TI.

V.2.3 Récupération de la collision

Une collision correspond à l'émission simultanée de plusieurs messages par plusieurs stations sur le bus. Grâce au mécanisme CSMA/CA, les stations qui entrent en collision détectent cette dernière sur l'occurrence de l'événement BC (« Bus Clash »). Cet événement n'apparaît que sur les stations qui entrent en collision.

Deux types de collision peuvent être distinguées :

- les collisions que nous qualifierons de *collisions normales*, qui sont issues d'une phase d'initialisation (régime normal du protocole),

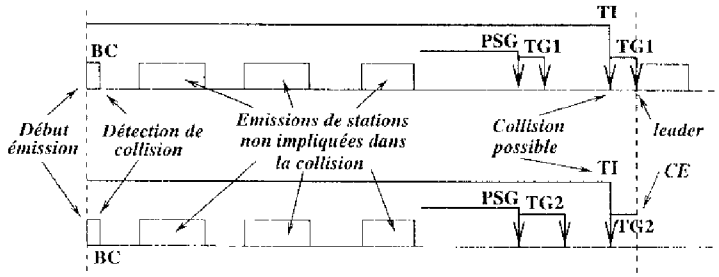


FIG. V.8 *Récupération de collision*

les collisions que nous qualifierons de *collisions anormales*, qui sont issues d'un régime particulier (transitoire) du protocole : fonctionnement défectueux d'une station (surdité, ...), apparition de perturbations sur le bus (foudre, ...), surcharge, ...

Quand une station voit apparaître le signal BC, elle arrête immédiatement son émission, et se place en attente de fin de cycle-bus (PSG écoulé). Deux cas sont alors à envisager :

- parmi les stations entrées en collision se trouve la station leader (obligatoire pour les collisions normales, possible pour les collisisons anormales); si la collision a eu lieu sur le message périodique, cela veut dire que les stations en collision ont détecté la condition de changement de cycle-bus en même temps (au temps de propagation près). Aussi, elles ont toutes démarré leurs TI en même temps (au temps de propagation près). Elles vont donc à nouveau détecter la condition TG, PSG et TI écoulés en même temps à la fin de ce cycle-bus, et entrer à nouveau en collision. Pour éviter cette répétition de collisions, le mécanisme de ségrégation (écoulement des TG) est appliqué à la fin de l'écoulement des TI. Ainsi, une seule station parmi celles qui sont entrées en collision détectera la condition de changement de cycle-bus. Les autres stations entreront dans le cycle-bus sur l'apparition du signal CE (non leader);

  la station leader ne fait pas partie des stations entrées en collision (obligatoirement une collision anormale). Les stations en collision attendent alors la fin du cycle-bus (PSG écoulé), et entrent dans le nouveau cycle-bus sur l'apparition du signal CE en tant que non leader.

Notons que pour savoir qu'elle doit écouler son TG après la fin de son TI, la station possède une variable « BC Mem» qui est marquée tant que le cycle-bus où a eu lieu la collision n'est pas fini. La remise à zéro de cette variable est faite soit lors du lancement du TG à la fin du TI, soit sur l'apparition du signal CE.

La figure V.8 illustre la récupération d'une collision entre deux stations qui se sont considérées leader ensemble (cas de l'initialisation de ces deux stations sur un bus vide).

V.2.4 Les situations particulières

Nous distinguons deux grands types de situations particulières (c'est-à-dire des situations autres que la récupération de collision et que l'initialisation) :

- des situations auxquelles le protocole est transparent, c'est à dire des situations dans lesquelles le protocole n'est pas perturbé et continue son régime normal : mutisme d'une station (elle ne peut plus émettre), disparition d'une station (sourde et muette), retrait/introduction d'une station, ...,
- des situations qui induisent un régime transitoire pour le protocole (dans lequel il peut y avoir des collisions) : surdité d'une station (elle n'entend pas l'activité sur le bus), les transmissions fantômes (activités parasites sur le bus, dues par exemple à la foudre), la dérive excessive d'une temporisation (qui provoque une ré-initialisation de la station), la surcharge (les émissions de messages débordent du cycle-bus), les pertes de messages (certaines stations n'entendent pas ce message), les fluctuations du signal BQ (apparitions de ce signal au milieu d'une réception d'un message), ...

Nous décrivons la surcharge et la dérive excessive d'une temporisation dans la suite de ce paragraphe. D'autres situations particulières seront abordées dans la partie analyse du protocole, à la fin de ce chapitre.

V.2.4.1 La surcharge

On appelle station en surcharge toute station qui voit son TG, lancé à la fin de son PSG, s'achever après la fin de l'écoulement de son TI. Dans ce cas, le TG n'étant pas écoulé à la fin du TI, la station ne change de cycle-bus qu'à la fin de son TG.

L'écoulement de TG à la fin du PSG, dont nous avons déjà parlé lors de l'étude du régime normal, se justifie ici : si le TI s'achève avant le PSG, et que le TG n'est pas relancé à la fin du PSG, toutes les stations en surcharge détecteraient la condition de changement de cycle-bus à la fin du PSG (en même temps au temps de propagation maximal près), et provoqueraient donc une collision. L'écoulement du TG à la fin du PSG permet de n'élire dans ce cas qu'une seule station leader.

Une surcharge peut être principalement provoquée par un flux périodique qui déborde du cycle-bus : les stations devant émettre obligatoirement un message périodique par cycle-bus, la longueur d'un message étant variable, si les messages périodiques sont trop longs, il se peut que leur émission se prolonge au delà de la fin du cycle-bus. La temporisation TI s'achève alors avant que PSG et TG ne soient écoulés.

Une surcharge peut aussi provenir d'une monopolisation du bus par une station en panne (émission continue de la station, station qui ne détecte plus l'activité sur le bus, ...).

La première station à être en état de surcharge est la station leader. En effet, comme elle est la première à émettre dans le cycle-bus, elle est la première à débuter l'écoulement de son TI, et donc la première à voir la fin de cet écoulement. La conséquence immédiate de cette propriété est que la périodicité du cycle-bus est perdue en cas de surcharge. La seconde conséquence est une collision ou un changement de leader possible, uniquement si

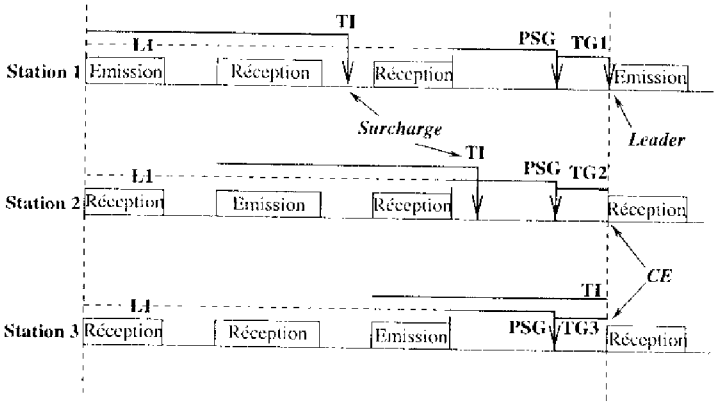


FIG. V.9 – *Exemple de surcharge*

la station qui était le leader n'a pas la plus petite valeur du TG (collision entre la station leader et une station non surchargée).

Aucune procédure n'est mise en place pour résoudre le problème d'une surcharge. La première station qui détecte la condition classique de changement de cycle-bus est le nouveau leader.

La figure V.9 illustre la surcharge de deux stations, alors qu'une troisième station n'est pas surchargée.

V.2.4.2 La dérive excessive d'une temporisation

Le protocole CP est essentiellement basé sur l'écoulement des cinq temporisations TG, ASG, PSG, TI et AT sur chaque station. Au niveau de l'implémentation de ces temporisations, il est possible que l'horloge qui permet de mesurer la valeur de la temporisation dérive, ce qui provoque un raccourcissement ou un allongement de la valeur de cette temporisation.

Pour pallier à ce problème, chaque temporisation est implémentée deux fois sur chaque station. Un mécanisme de surveillance permet de détecter un trop grand écart entre les deux valeurs données par ces deux implémentations : dès que l'une des deux temporisations s'achève, un décompte est effectué pour mesurer le temps entre la fin de la première temporisation et la fin de la seconde. Dès que cette valeur dépasse un certain seuil (seuil de tolérance), un message est envoyé à la partie contrôle du protocole, signifiant la dérive excessive de la temporisation. Dans les machines à états du protocole, ce message est noté « TG discrepancy » pour TG, « ASG discrepancy » pour ASG, ...

Une situation de dérive est appelée « Monitoring Window ». La variable Monitoring Window de la partie contrôle du protocole mémorise l'occurrence d'une dérive d'une temporisation de la station. La procédure de résolution appliquée est une réinitialisation complète de la station. Si la station était en train d'émettre, elle arrête son émission. Toutes les temporisations sont remises dans leur état d'initialisation, puis la station entame la procédure d'initialisation que nous avons décrite dans le paragraphe V.2.2, comme si elle

venait d'être mise sous tension.

V.2.5 Architecture d'une station

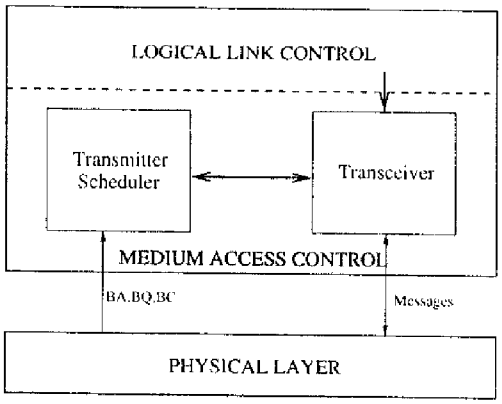


FIG. V.10 – *Architecture d'une station*

L'architecture de l'entité Liaison de données est donnée sur la figure V.10. La sous-couche MAC est scindée en deux modules : le module « Transceiver » et le module « Transmitter Scheduler ».

Le module Transceiver est chargé de l'émission et de la réception des messages sur le bus. Les instants où il peut émettre un message lui sont signifiés par le Transmitter Scheduler.

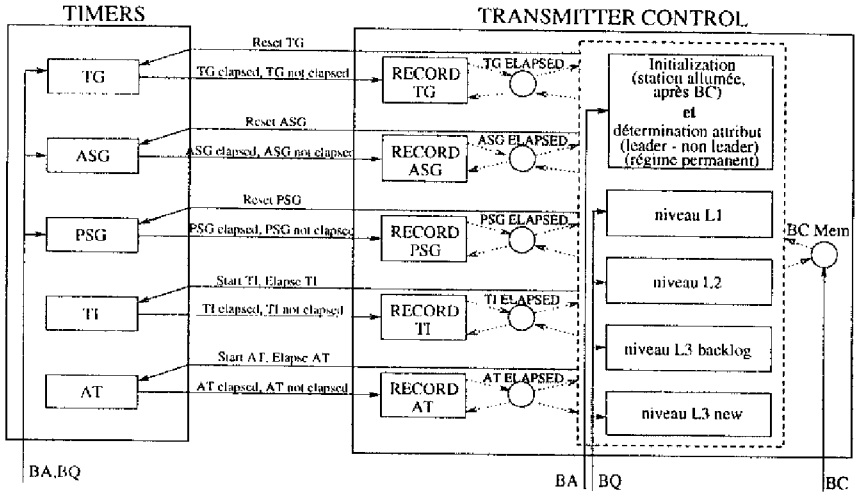


FIG. V.11 – *Architecture du « transmitter scheduler »*

Le Transmitter Scheduler est le cœur du protocole. Il permet à la station de s'initialiser, de se déplacer dans les différents niveaux du cycle-bus, de participer à l'élection de la

station leader, et donne les ordres de début d'émission et d'arrêt (en cas de panne ou de collision) des messages périodiques et apériodiques au Transceiver. Il se base pour cela sur l'état des temporisations de la station et sur les signaux en provenance de la couche physique, à savoir BA, BQ et BC, tel que le montre la figure V.11.

La représentation donnée sur cette figure est une structuration améliorée du Transmitter Scheduler : les parties « Record TG », « Record ASG », ..., ont été ajoutées pour faciliter la modélisation du protocole. Un Record permet au Transmitter Scheduler de mémoriser l'état de la temporisation. On peut considérer qu'il s'agit d'une variable qui est mise à jour par l'intermédiaire des messages « elapsed » et « not elapsed », émis par la temporisation lors de ses changements d'états.

La variable « BC Mem », qui mémorise l'occurrence d'une collision dans le cycle-bus courant, est positionnée sur l'apparition du signal BC. Elle est remise à zéro à la fin du cycle-bus (voir le mécanisme de résolution d'une collision du paragraphe V.2.3).

V.3 Modélisation par Réseaux de Petri Temporisés Stochastiques du protocole ARINC 629 CP

V.3.1 Méthodologie

La modélisation du protocole CP s'est faite en deux phases :

1. modélisations séparées des temporisations, des niveaux du Transmitter Scheduler, et de la couche physique (modèles locaux);
2. interconnexion des modèles locaux (obtention du modèle global).

Nous avons de plus utilisé deux types de transitions dans ces modèles :

- les transitions immédiates, qui ont servis à modéliser les évolutions logiques du protocole,
- les transitions discrètes avec mémoire de la dernière sensibilisation, qui ont permis de modéliser les écoulements des temporisations et les durées des émissions.

L'interconnexion entre les différents modèles s'est faite sur la base de deux principes :

- le mécanisme de places partagées. Chaque passage d'un module à un autre est effectué par place partagée, ainsi que la communication entre le Transmitter Control et les différents modules Record;
- le mécanisme de fusion de transitions. Cette technique a été utilisée notamment pour éliminer des transitions les étiquettes représentant une communication avec une temporisation (étiquettes $?X$ et $!X$).

Le modèle global obtenu comporte beaucoup de places et de transitions. Pour pouvoir l'analyser, nous avons, pour chaque propriété à valider, essayé de réduire au maximum ce modèle pour diminuer les problèmes d'explosion combinatoire du nombre d'états dans les graphes générés, en ne conservant que les parties nécessaires du modèle.

Nous nous proposons maintenant d'étudier les différents modèles locaux que nous avons construits.

V.3.2 Modélisation des temporisations

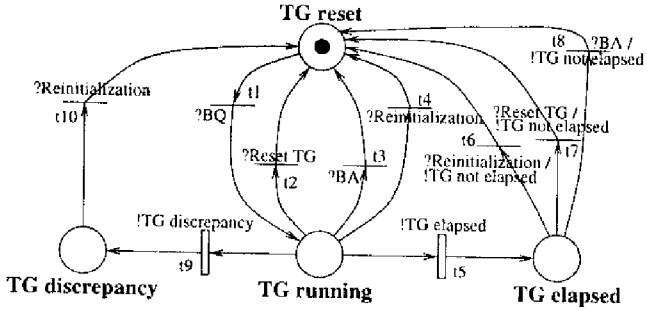


FIG. V.12 - *Modèle de la temporisation TG*

Nous ne présentons ici que le modèle Réseaux de Petri Temporisés Stochastiques de la temporisation TG. Les modèles des autres temporisations se déduisent très facilement des machines à états que nous avons donné dans le paragraphe V.2.1.2.

Chaque place du modèle de la figure V.12 correspond à un état de la machine à états de TG (paragraphe V.2.1.2), chaque transition correspond à une transition entre états.

Seules deux transitions sont temporisées: la transition qui modélise la durée de la temporisation (t_5), et la transition qui modélise l'occurrence d'une dérive excessive de la temporisation (t_9). Ces deux transitions ont les mêmes attributs temporels: une densité discrète avec un dirac à une date égale à la durée de la temporisation. Ces deux transitions sont exclusives (le tir de l'une désensibilise l'autre) et sensibilisées en même temps (par le marquage de la place *TG running*). Ces transitions permettent donc de simuler la possibilité, à la fin de l'écoulement de la temporisation, d'une dérive excessive ou pas (utilisation du non déterminisme des Réseaux de Petri).

Les autres transitions sont toutes immédiates, puisqu'elles sont tirées sur la réception des messages en provenance du Transmitter Scheduler (*?Reset TG*: transitions t_2 et t_7, *?Reinitialization*: transitions t_4, t_6 et t_{10}) et de la couche physique (*?BA*: transitions t_3 et t_8, *?BQ*: transition t_1).

Notons enfin que sur ce modèle apparaissent les messages qui permettent au Transmitter Scheduler de mettre à jour la variable qui lui donne l'état de la temporisation : le message *?TG not elapsed* est émis lors du tir des transitions t_6, t_7, t_8 qui permettent de quitter l'état *TG elapsed*, le message *?TG elapsed* est émis lors du tir de la transition t_5 qui permet d'arriver dans l'état *TG elapsed*. Ces deux messages sont utilisés par la partie *Record TG* du Transmitter Scheduler pour mémoriser l'état de la temporisation TG.

V.3.3 Quelques modèles du Transmitter Control

Pour des raisons de place, nous ne présentons dans ce paragraphe que certains modèles Réseaux de Petri du protocole CP. Les autres modèles peuvent se déduire très facilement des exemples donnés. De plus, pour des raisons de complexité des modèles, nous ne tenons

pas compte des signaux « Monitoring Window » (dérive excessive d'une temporisation).

Les modèles que nous allons présenter sont ceux : de l'initialisation et de l'élection de la station leader; du niveau périodique L1; du niveau apériodique urgent L2.

Dans ces modèles, les places *TG elapsed*, *ASG elapsed*, *PSG elapsed*, *TI elapsed* et *AT elapsed* donnent l'état des temporisations mémorisés par les Record (ces places sont partagées par les Record et le Transmitter Control, comme nous pouvons le voir sur la figure V.11).

V.3.3.1 L'initialisation et l'élection de la station leader

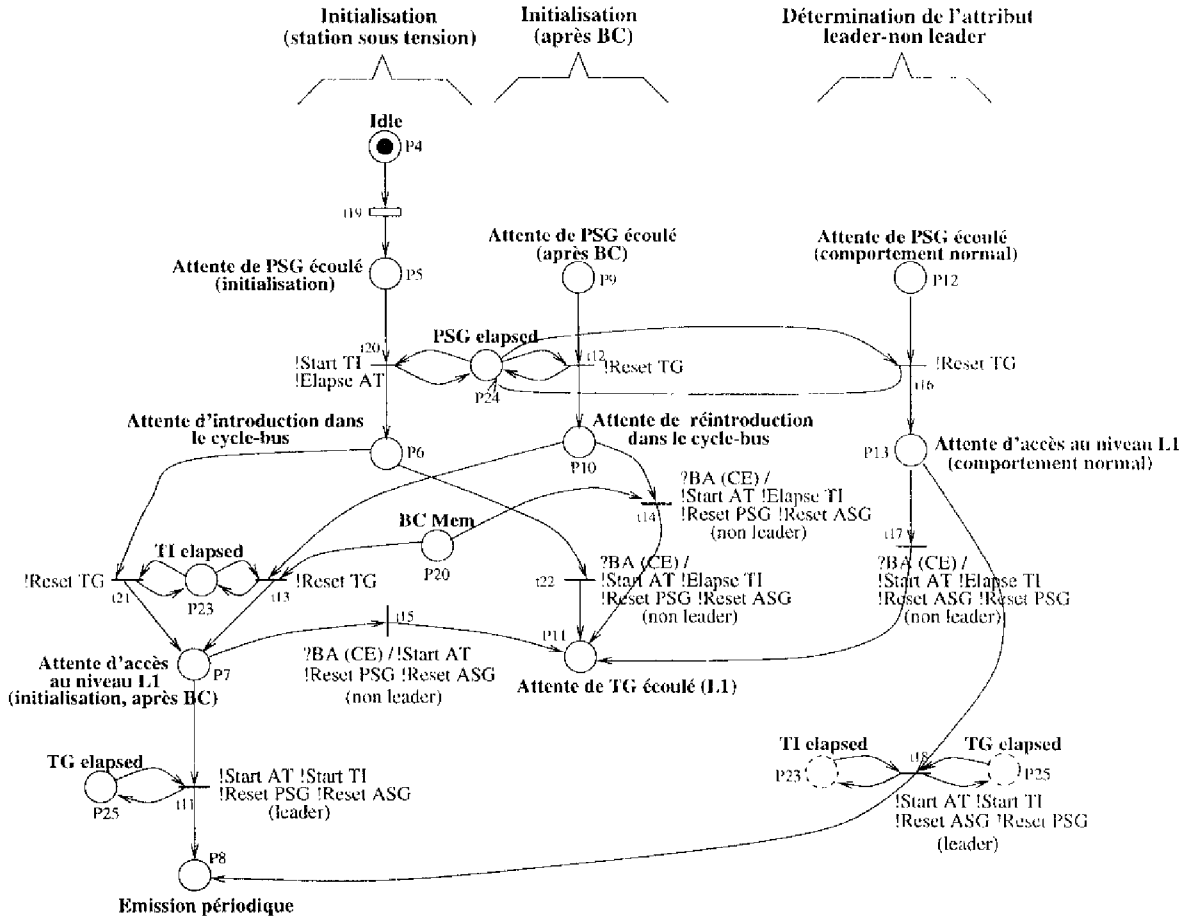


FIG. V.13 – *Modèle de l'initialisation et de l'élection de la station leader*

La figure V.13 peut être découpée en trois parties distinctes :

- la partie gauche de la figure représente l'initialisation de la station après sa mise sous tension. La transition temporisée t_{19} permet de positionner l'instant de mise sous tension de la station (elle permet de décaler les stations entre elles quand

plusieurs stations s'initialisent en même temps). Le tir de la transition t_{20} ne peut être effectué que si la temporisation PSG est écoulée (place P_{24} marquée). Lors de ce tir, la temporisation TI commence à s'écouler (*!Start TI*) et la temporisation AT est positionnée dans l'état écoulé (*!Elapse AT*). La transition t_{21} est tirée à la fin de l'écoulement de la temporisation TI, et permet de relancer TG (*!Reset TG* qui provoque le passage de la temporisation à l'état *Running* si le bus est silencieux). La transition t_{22} est tirée si une activité bus (CE) apparaît pendant l'écoulement du TI. La station a alors terminé son initialisation et entre dans son premier cycle-bus en tant que station non leader, après avoir positionné ses temporisations : TI est forcé à l'état écoulé (*!Elapse TI*) pour pouvoir être démarré sur l'émission périodique de la station, AT est démarré immédiatement (*!Start AT*), ASG et PSG sont armés (*!Reset ASG, !Reset PSG*) pour être prêts à s'écouler.

Si le TI est allé à son terme, le TG est relancé (transition t_{21}). Si une activité bus (CE) apparaît pendant cet écoulement (transition t_{15}), alors la station finit son initialisation et entre dans son premier cycle-bus comme station non leader (mêmes actions que lors du tir de t_{22}, si ce n'est que TI est déjà écoulé). Sinon, le TG s'achève. La station se considère alors comme leader, et commence son premier cycle-bus par une émission périodique (transition t_{11}); les actions effectuées lors du tir de t_{11} sont les mêmes que celles du tir de t_{22}, si ce n'est que TI est démarré immédiatement (*!Start TI*).

- la partie du milieu de la figure représente l'élection de la station leader pour une station qui vient de provoquer une collision dans le cycle-bus courant (place « BC Mem » marquée). Les mécanismes mis en œuvre sont strictement identiques à ceux que nous venons de décrire pour la phase d'initialisation classique. Notons que la place « BC Mem » est démarquée soit lors de l'occurrence d'un CE alors que le TI n'est pas encore écoulé (transition t_{14}), soit lors du lancement du TG à la fin du TI (mécanisme de récupération de la collision : transition t_{13}).
- la partie de droite de la figure représente l'élection de la station leader en comportement normal. A la fin du PSG (transition t_{16}), la temporisation TG est relancée (*!Reset TG*). La station attend alors soit l'occurrence d'un CE (transition t_{17}) qui lui permet d'entrer dans le cycle-bus suivant en tant que non leader (mêmes actions effectuées que lors du tir de t_{22}), soit la fin de ses temporisations TI et TG (transition t_{18}) qui lui permet de détecter la condition de changement de cycle-bus et de se considérer comme la nouvelle station leader (mêmes actions effectuées que lors du tir de t_{11}).

Notons que les places P_8 (« Émission périodique ») et P_{11} (« Attente de TG écoulé (L1) »), qui correspondent respectivement à l'entrée dans le cycle-bus suivant en tant que leader et en tant que non leader, sont communes au modèle Réseaux de Petri du niveau périodique L1. Les places P_9 (« Attente de PSG écoulé (après BC) ») et P_{12} (« Attente de PSG écoulé (comportement normal) ») sont communes à tous les niveaux du cycle-bus.

V.3.3.2 Le niveau périodique L1

Le niveau périodique L1 est le premier niveau du cycle-bus. Ce modèle est connecté par les places P_8 (« Émission périodique »), P_{11} (« Attente de TG écoulé (L1) ») et P_9

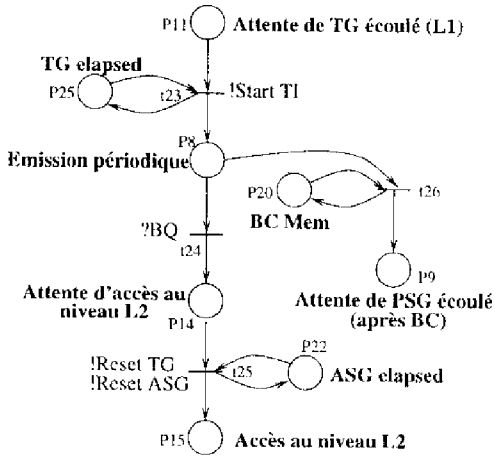


FIG. V.14 – *Modèle du niveau périodique L1*

(« Attente de PSG écoulé (après BC) »), au modèle de l'initialisation et d'élection de la station leader.

La transition t_{23} représente le début d'émission périodique de la station. Notons que cette transition n'est franchie que si la station est non leader (arrivée dans le niveau par la place P_{11}) et si la temporisation TG est écoulée.

Le tir de la transition t_{24} modélise la fin d'émission de la station, sur occurrence de l'événement BQ en provenance de la couche physique. La station se place alors en attente de changement de niveau (place P_{14} marquée). Dès que la temporisation ASG est écoulée, la station change de niveau sur le tir de la transition t_{25}. Lors de ce tir, un ordre d'armement est envoyé aux temporisations TG et ASG, afin de les positionner correctement dans le nouveau niveau (*!Reset TG*, *!Reset ASG*). La place P_{15} est alors marquée. Cette place est commune avec le module du niveau L2.

Notons que si une collision a lieu pendant la transmission de la station (place P_8 marquée pour la transmission, place P_{20} marquée pour mémoriser l'occurrence de la collision), la transition t_{26} peut alors être tirée, modélisant l'arrêt de l'émission suite à une collision. La station se retrouve alors en attente de fin de cycle-bus (place P_9 marquée, commune avec le module d'initialisation et d'élection du leader).

V.3.3.3 Le niveau apériodique urgent L2

Le modèle du niveau L2 (figure V.15) ne diffère du niveau du modèle L1 que par deux points :

- lors de l'arrivée dans le niveau (place P_{15} marquée), la station décide si elle veut émettre ou pas un message dans ce niveau. Ce choix est effectué en fonction de la file de messages à émettre: si un message apériodique urgent est en attente d'émission lors de l'entrée dans le niveau L2, la transition t_{27} sera tirée (la station se place alors en attente d'accès au bus); si aucun message n'est en attente d'émission, la transition

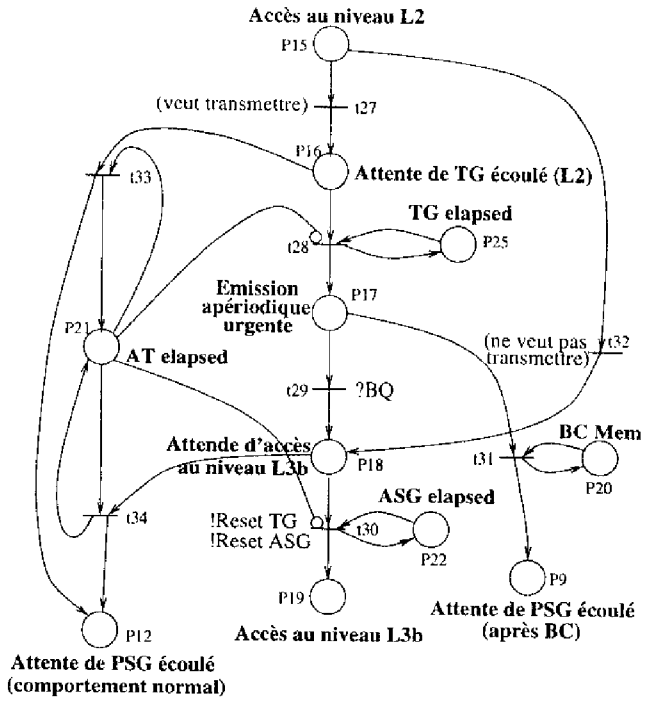


FIG. V.15 - *Modèle du niveau apériodique urgent L2*

t_{32} est tirée (la station se place alors en attente de fin du niveau). Cela implique que tout message apériodique urgent arrivant dans la file d'attente d'émission après le début du niveau ne pourra pas être émis avant le cycle-bus suivant;

- le niveau peut être interrompu par la fin de l'écoulement de la temporisation AT, qui indique que plus aucune émission apériodique ne peut débuter dans ce niveau. Ainsi, si AT s'achève alors que la station est en attente d'accès au bus (place P_{16} marquée) ou de changement de niveau (place P_{18} marquée), la station passe en attente de fin de cycle-bus (tir de la transition t_{33} ou t_{34} : place P_{12} marquée, commune au module d'initialisation et d'élection du leader).

Le reste du modèle reprend exactement les mécanismes déjà décrits pour le niveau L1. La place P_{19} (*Accès au niveau L3b*) est commune avec le modèle du niveau L3 backlog.

V.3.4 Modélisation de la couche physique

La couche physique ne fait pas vraiment partie des mécanismes que nous devons étudier dans le protocole ARINC 629 CP. Il est néanmoins nécessaire, dans le cadre d'une validation des mécanismes de la sous-couche MAC, de modéliser une vision abstraite de cette couche, pour notamment générer les signaux BA, BQ et BC, et pouvoir spécifier les temps de propagation entre les stations.

La modélisation est effectuée en deux parties. La première partie modélise la couche

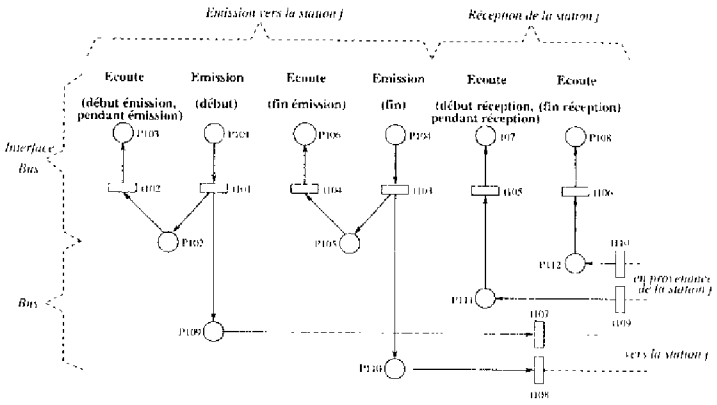


FIG. V.16 *Ecoute et transmission dans la couche physique*

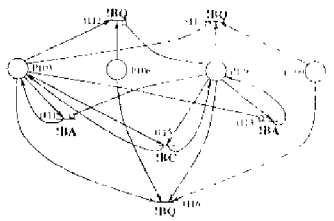


FIG. V.17 *Génération des signaux BA, BQ et BC*

physique, et plus particulièrement les temps de propagation entre les stations. La deuxième partie modélise l'interface entre la couche physique et la sous-couche MAC, interface qui génère les signaux BA, BQ et BC.

La modélisation de la couche physique (figure V.16) est basée sur deux éléments essentiels :

- nous considérons, depuis une station i, autant de connexions qu'il y a de stations sur le bus (une connexion par couple de stations), comme le montre la partie *bus* de la figure V.16;

  la transmission d'un message est représentée par le début et la fin de la transmission; chaque connexion entre une station i et une station j est donc composée de deux voies, l'une correspondant à la propagation du début du message (transition t_{107}) et l'autre à la propagation de la fin du message (transition t_{108}).

La station n'étant pas physiquement directement connectée au médium de communication, un temps de propagation existe entre la station et le bus. Ce temps est modélisé par les transitions t_{101}, t_{102}, t_{103}, t_{104}, t_{105} et t_{106}.

Enfin, nous distinguons l'écoute du bus, de la transmission sur le bus : les places P_{101} et P_{104} modélisent le début et la fin d'émission d'un message. Les places P_{103} et P_{106} modélisent l'écoute de sa propre émission, les places P_{107} et P_{108} l'écoute d'un message d'une autre station.

La modélisation de l'interface entre la couche physique et la sous-couche MAC est donnée sur la figure V.17. Les places de ce modèle sont partagées avec le modèle de la couche physique. Elles représentent l'écoute du bus par la station. Ici, nous ne représentons que la partie correspondante à la connexion de la station i avec la station j.

Les transitions de ce modèle permettent de générer les signaux BA, BQ et BC :

le signal BA est généré si la station entend son propre début de message sans qu'il y ait de réception d'un autre message (transition t_{111}) et si la station reçoit un début de message sans qu'elle n'en émette (transition t_{113});

- le signal BC est généré si la station reçoit à la fois le début de sa propre émission et le début d'un autre message (transition t_{115});
- le signal BQ est généré si la station reçoit la fin de sa propre émission sans qu'aucun autre message ne soit émis (transition t_{112}), si elle reçoit la fin d'un autre message sans qu'elle ne soit en train d'émettre (transition t_{114}), et si elle reçoit la fin de tous les messages en collision après la détection d'une collision (transition t_{116}).

Ces signaux sont exploités par les modules du Transmitter Control et les modules des temporisations, pour faire évoluer ces différents modules.

V.4 Analyse du protocole ARINC 629 CP

Les propriétés que nous avons validées tendent toutes à démontrer le bon comportement du protocole, en régime normal ou en situation exceptionnelle. Nous ne présentons ici principalement que certains des résultats qualitatifs et quantitatifs que nous avons obtenu. Pour des raisons de place, nous ne pouvons présenter toutes les analyses du protocole CP. La totalité de nos travaux se trouve dans [GBJ96c, GBJ96a, GBJ96b, GJB96, Gal97, GJB97].

V.4.1 Étude du régime normal

Les modèles construits donnant des graphes trop grands pour être présentés ici, nous nous sommes contraints à l'étude des deux premiers niveaux du protocole, à savoir le niveau périodique L1 et le niveau apériodique urgent L2.

Nous allons étudier trois phases différenets du protocole : le niveau L1, le niveau L2, et la collision suite à une initialisation sur un bus vide.

V.4.1.1 Cycle-bus composé du niveau L1 uniquement

Nous considérons dans cette étude deux stations sur le bus, la station 1 et la station 2, la station 1 ayant le plus petit TG. De plus, nous considérons que seul le niveau L1 est accessible par les stations dans le cycle-bus. A l'issu de l'initialisation, la station 1 est leader.

	TG	ASG	PSG	AT	TI
station 1	20.7	64	320	5516	8500
station 2	40.7	64	320	5516	8500

FIG. V.18 – *Configuration temporelle choisie*

Le but de cette analyse est de montrer le bon comportement du protocole par rapport au trafic périodique en régime normal. La configuration temporelle choisie est donnée dans la table de la figure V.18. L'unité de temps est la microseconde. La longueur des messages périodiques est supposée constante et égale à 500 μs. Le temps de propagation entre les deux machines est de 4.5 μs. La longueur maximale d'un message est $MAL = 2600\mu s$.

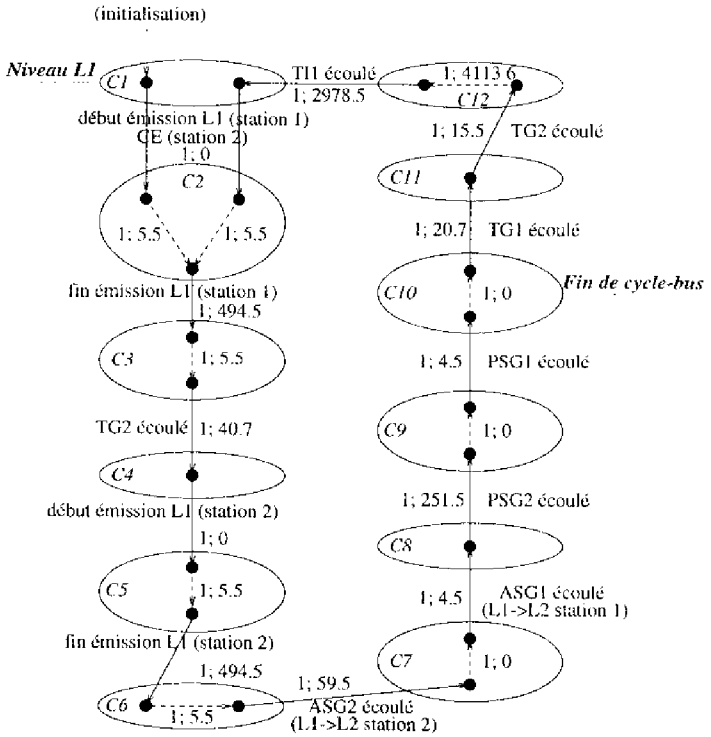


FIG. V.19 – *Automate quotient quantitatif quantifié relatif au niveau L1*

La figure V.19 montre la vue abstraite qualitative quantifiée relative aux événements « début et fin d'émission », « TG elapsed », « ASG elapsed », « PSG elapsed », « TI elapsed » des deux stations, effectuée sur le graphe d'états probabilisé obtenu à partir de la configuration temporelle précédente. Cet automate se décompose en deux parties, qui forment un cycle : le niveau L1 (de l'état 1 à l'état 9) et la fin du cycle-bus (de l'état 9 à l'état 1).

Le niveau L1 est composé des émissions périodiques des deux stations. Il commence par l'émission du message de la station leader (station 1) de la classe 1 à la classe 3. Notons que le début d'émission de ce message est interprété par la station 2 comme le signal de changement de cycle-bus (CE), et que le TI de la station 1 commence à s'écouler. Ensuite, les temporisations TG des deux stations s'écoulent en parallèle (temps de silence sur le bus). Le TG de la station 2 s'achève (classe 4 à classe 5), autorisant le début de son émission périodique (classe 5 à classe 6) et le début d'écoulement de son TI. A la fin de cette émission (classe 6 à classe 7), les deux stations commencent à écouler leurs temporisations ASG. Notons ici que ce début d'écoulement ne se fait pas simultanément sur les deux stations : la station 2, qui est la dernière à émettre, détecte dès la fin de son émission un temps de silence sur le bus, et commence à écouler son ASG. La station 1 ne détecte ce même silence que plus tard, après propagation de la fin du message de la station 2. Le décalage entre le lancement des deux ASG est donc égal à un temps de propagation. Les deux ASG étant identiques, c'est la station 2 qui verra son ASG s'achever en premier,

avant celui de la station 1 (un temps de propagation entre les deux), ce que nous vérifions sur la transition entre la classe 7 et la classe 8.

A la fin de chaque ASG, la station concernée change de niveau. Cela implique que les stations ne changent pas simultanément de niveau (puisque les ASG ne s'achèvent pas en même temps), mais dans un laps de temps de durée égale à un temps de propagation. Notons que cette remarque est aussi valable pour le changement de cycle-bus, et pour le début de tous les écoulements des temporisations liées à l'activité bus. Dans le protocole CP, un point de synchronisation est en fait un intervalle temporel de durée égale à un temps de propagation, dans lequel toutes les stations évoluent de manière identique.

La partie fin de cycle-bus débute quand les temporisations PSG s'achèvent. Comme pour la fin des ASG, la fin des PSG des stations se fait dans un intervalle temporel de durée égale au temps de propagation. La dernière station à avoir émis est la première à voir son PSG s'achever : ici la station 2 (classe 9 à classe 10). A la fin de ce PSG, la station 2 commence à écouler son TG. Un temps de propagation plus tard, le PSG de la station 1 s'achève (classe 10 à classe 11), et le TG de cette station est lancé.

Dans la configuration temporelle que nous avons choisie, le TG de la station 1 est plus petit que celui de la station 2. Or, ici, le TG de la station 2 a été lancé un temps de propagation avant le TG de la station 1. La différence d'au moins deux fois le temps de propagation maximal entre les valeurs des deux TG implique que le TG de la station 1 s'achève avant celui de la station 2 (transition entre a classe 10 et la classe 12).

Enfin, le TI de la station 1 ayant commencé avant celui de la station 2, il s'achève en premier (classe 13 à classe 14). A cet instant, la station 1 détecte la condition de changement de cycle-bus, à savoir TI, TG et PSG écoulés. Elle se considère alors leader et entame le nouveau cycle-bus par son émission périodique (classe 1 à classe 2). Le début de cette émission sera interprété comme le signal de changement de cycle-bus de la station 2 (CE) avant que son TI ne soit écoulé.

Les étiquettes quantitatives apparaissant sur cet automate nous permettent, par exemple, de mesurer le temps de cycle : $T_c = 8500\mu s$. D'autre analyses quantitatives pourront être trouvées dans [GBJ96a].

Les principales propriétés vérifiées ici sont :

- un cycle-bus a une durée égale à TI;
- en régime normal, le leader ne change pas d'un cycle-bus à un autre;
- le leader est la première station à émettre dans le cycle-bus;
- les stations accèdent au niveau L1 périodiquement (à plus ou moins δ_i en fonction de la variation des messages);
- la condition d'accès au bus est : TG écoulé.

V.4.1.2 Cycle-bus composé des niveaux L1 et L2

Nous considérons dans cette étude deux stations sur le bus, la station 1 et la station 2, la station 1 ayant le plus petit TG. De plus, nous considérons que seul les niveaux L1 et

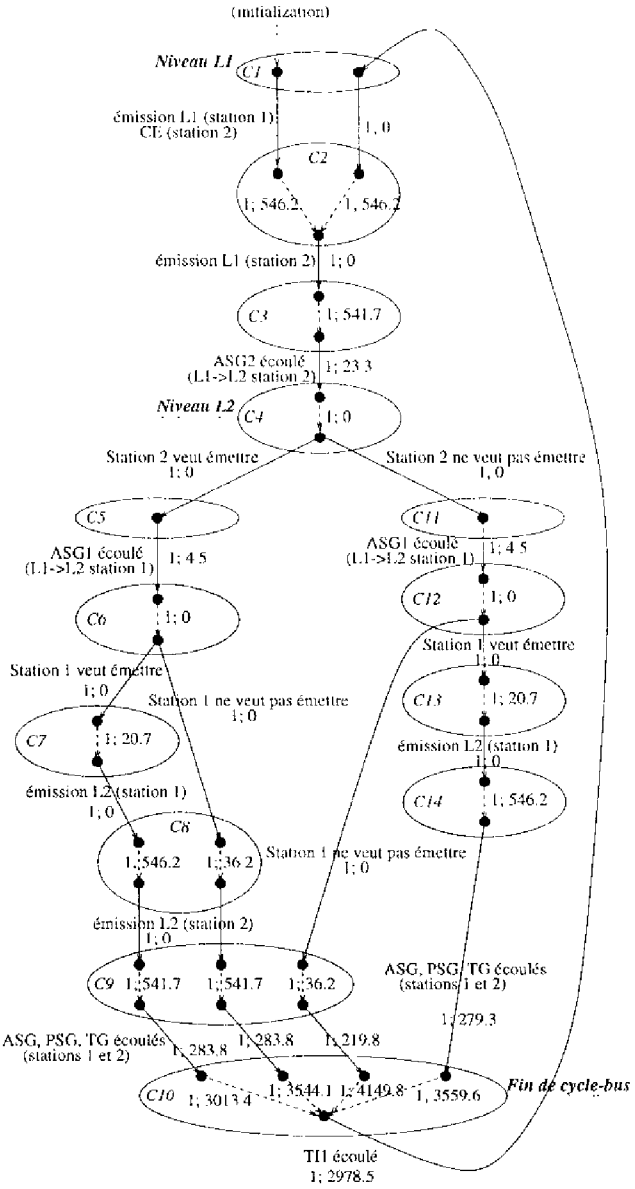


FIG. V.20 – *Cycle-bus composé des niveaux L1 et L2 (automate quotient qualitatif quantifié)*

L2 sont accessibles par les stations dans le cycle-bus. A l'issu de l'initialisation, la station 1 est leader.

Le but de cette analyse est de montrer le bon comportement du protocole par rapport aux trafics périodique et apériodique urgent en régime normal. La configuration temporelle choisie est la même que celle utilisée pour l'étude du niveau périodique L1 (table de la

figure V.18).

Le graphe d'états probabilisé correspondant à cette configuration temporelle comporte 196 états. Nous donnons sur la figure V.20 la vue abstraite qualitative quantifiée de ce graphe relative aux événements « émission L1 », « émission L2 », « choix d'émission en L2 », « choix de non émission en L2 », « TG écoulé », « ASG écoulé », « PSG écoulé », « TI écoulé » des deux stations.

Cette vue peut être découpée en trois parties: le niveau L1 (de la classe C_1 à la classe C_4), le niveau L2 (entre les classes C_4 et C_{10}) et la fin du cycle-bus (de la classe C_{10} à la classe C_1).

Le niveau L1 voit l'émission des deux stations 1 et 2 (état 1 à état 3), avant que les ASG des deux stations ne s'écoulent pour passer au niveau L2.

La première station qui voit son ASG s'achever est la station 2 (classe C_3 à classe C_4), puisqu'elle est la dernière à avoir émis. A cet instant, la station 2 change de niveau, et passe au niveau L2. Instantanément, elle teste son buffer d'émission pour voir si une requête apériodique urgente est en attente: si il y a une telle requête, alors la station cherche à écouler son TG pour avoir accès au bus, et émettre ce message (transition de la classe C_4 vers la classe C_5); sinon, elle cherche à écouler son ASG pour changer de niveau (transition de la classe C_4 à la classe C_{11}).

Un temps de propagation plus tard, la temporisation ASG de la station 1 s'achève à son tour. La station passe alors du niveau L1 au niveau L2. Comme pour la station 2, elle teste son buffer d'émission pour savoir si elle a un message à émettre ou pas.

Le reste du niveau est occupé par les émissions apériodiques urgentes des stations, après écoulement des TG (ces écoulements ne sont pas représentés sur la figure V.20), en fonction des choix faits par les deux stations en début de niveau. Après cette phase d'émission, les stations passent en attente de fin d'écoulement des ASG. Comme il n'y a pas d'autre niveau dans le cycle-bus, la fin des ASG est suivie par la fin des PSG. Les TG alors relancés s'achèvent (arrivée dans la classe C_{10}), et la station qui a lancé son TI en premier (station 1) détecte en premièr la condition de changement de cycle-bus, et devient leader du nouveau cycle-bus (transition de la classe C_{10} à la classe C_1).

Les principales propriétés qualitatives et quantitatives vérifiées ici sont:

- la durée d'un cycle-bus est de 8500 μs, et est égale à TI, quel que soient les choix d'émission des deux stations dans le niveau L2;
- les choix d'émission ou de non émission en L2 ne perturbe pas le bon déroulement du cycle-bus; en particulier, il n'y a pas de changement de leader;
- il y a séquencement des niveaux, sans mélange entre eux;
- dans un niveau apériodique, les émissions se font dans l'ordre croissant des TG des stations qui veulent émettre;
- dans le niveau L2, chaque station émet au plus un message;

D'autres propriétés ont été vérifiées dans nos travaux sur le protocole, et notamment l'effet de l'arrivée d'un message apériodique urgent dans le buffer L2 d'émission d'une

station, alors qu'elle est déjà entrée le niveau. Dans ce cas, nous avons pu vérifier que le message ne peut être émis qu'au cycle-bus suivant. Le reste de nos analyses peuvent être trouvées dans [GBJ96a, GBJ96b].

V.4.1.3 Situation de collision

Le but de cette analyse est de montrer comment le protocole permet de résoudre une situation de collision, et d'éviter les répétitions de collisions.

Pour cela, nous étudions un cycle-bus composé de trois stations, dont les TG sont tels que $TG_1 < TG_2 < TG_3$. Comme nous étudions la situation de collision, nous considérons que les stations n'ont accès qu'au niveau L1 dans les cycles-bus. La configuration temporelle est celle donnée dans la table V.21 (en μs). Les dates de mise sous tension des stations sont telles qu'elles compensent les différences entre les TG. Si nous appelons ces dates τ_1 pour la station 1, τ_2 pour la station 2, τ_3 pour la station 3, on a : $\tau_1 - \tau_3 = TG_3 - TG_1$ et $\tau_1 - \tau_2 = TG_2 - TG_1$, à plus ou moins un temps de propagation près. Nous avons choisi $\tau_1 - \tau_3 = 20.2\mu s$ et $\tau_1 - \tau_2 = 10.1\mu s$. La station 3 est donc la première a être mise sous tension, puis la station 2, et enfin la station 1.

	TG	ASG	PSG	AT	TI
station 1	10.7	64	320	5516	8500
station 2	20.7				
station 3	30.7				

FIG. V.21 – *Configuration temporelle choisie*

Le graphe d'états probabilisé correspondant à cette configuration comprend 234 états. La figure V.22 nous donne la vue abstraite qualitative quantifiée de ce graphe d'états probabilisé, relative aux événements « TG écoulé », « TI écoulé », « début et fin d'émission périodique (leader ou non leader) », « détection de collision ».

Dès leur mise sous tension, les trois stations cherchent à écouler leur PSG, puis leur TI. C'est dans l'ordre de mise sous tension des stations (station 3, puis station 2 et enfin station 1) que les TI s'achèvent (classe C_1 à classe C_4).

A la fin des TI, les stations lancent leur TG. Ces dates de début d'écoulement des TG sont décalées entre elles comme les dates de mise sous tension (puisque un temps égal pour les trois stations s'est écoulé depuis ces mises sous tension). Les TG s'achèvent donc tous dans un intervalle temporel inférieur au temps de propagation (il y a un temps de $0.2\mu s$ de la classe C_5 à classe C_9), puisque les décalages à la mise sous tension compensent les différences entre les TG. Ceci implique que chaque station commence à émettre en tant que leader son message périodique avant d'entendre le début d'émission d'une des autres stations. Dès que cette écoute commence, les stations détectent une collision et s'arrête d'émettre (classe C_{10} à classe C_{13}).

La résolution d'une collision nécessite l'écoulement d'un TI plus l'écoulement d'un TG. Ici, les TI des stations ont tous été lancés lors du début d'émission périodique des stations, c'est à dire presque en même temps (décalage de $0.1\mu s$ entre la station 3 et la station 2, décalage de $0.6\mu s$ entre la station 2 et la station 1). Ils s'achèvent donc tous avec les mêmes

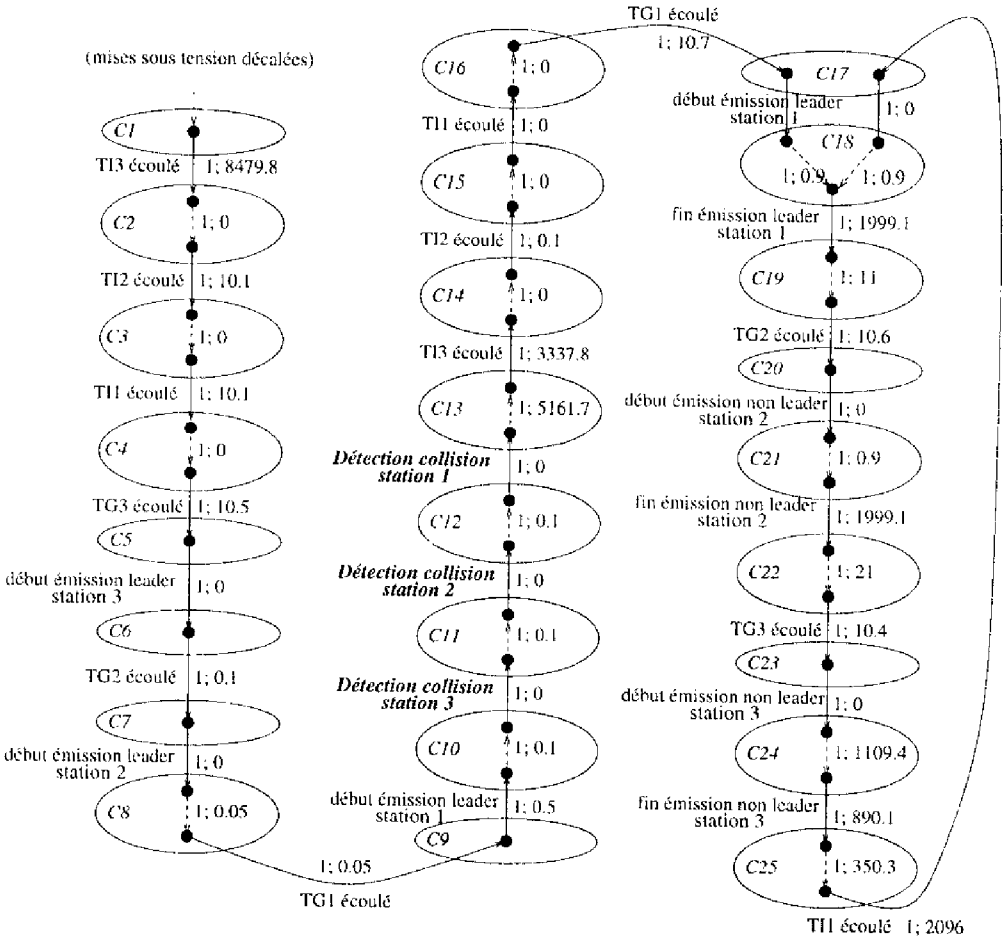


FIG. V.22 – *Collision à l'initialisation (automate quotient qualitatif quantifié)*

décalages (classe C_{13} à classe C_{16}).

Les stations, à la fin de l'écoulement de leur TI, relancent leur TG. La différence entre deux TG étant supérieure au décalage entre les stations (la différence entre deux TG est supérieure à deux fois le temps de propagation maximal), c'est la station qui a le plus petit TG qui verra ce dernier s'achever en premier (ici la station 1 : classe C_{17}). La station 1 se considère alors comme station leader, et débute son émission périodique.

Ce début d'émission va forcément arriver sur les autres stations avant que leurs TG s'achèvent. En effet, le décalage maximal, dans notre scénario, entre deux TG est d'un temps de propagation maximal. Or, la différence entre deux TG est au moins égale à deux temps de propagation maximal. Cela implique qu'entre la fin du plus petit TG et la fin du TG suivant, il y a au moins un temps de propagation maximal. Donc, le début du message périodique de la station 1 a le temps d'arriver sur les autres stations avant qu'elles n'aient fini d'écouler leurs TG. Cette différence entre les TG permet d'assurer qu'il n'y a pas de répétition de collision dans le protocole CP.

La suite du graphe (de la classe C_{17} à la classe C_{25}) est un régime normal du protocole,

où la station 1 est leader.

Les principales propriétés validées par cette analyse sont :

- une initialisation sur un bus vide peut générer une collision entre plusieurs stations;
- la répétition de collisions entre les mêmes stations n'est pas possible;
- le temps de récupération de la collision est inférieur à $TI + TG_{max}$;
- la collision ne perturbe le bus que pendant un cycle-bus.

D'autres propriétés ont été analysées pour la collision. Nous avons notamment montré qu'une collision ne perturbe que les stations impliquées dans cette collision (les autres stations ne sont pas perturbées). La totalité de nos travaux sur les collisions se trouvent dans [GBJ96b, GJB96].

Remarque : lors de l'étude de l'initialisation, nous avons démontré que la répétition de collisions n'était impossible dans le protocole que si l'on assurait une différence entre deux TG d'au moins *quatre fois* le temps de propagation maximal, contrairement au deux temps de propagation maximal proposés par la norme ARINC.

V.4.2 Étude de situations particulières

Nous allons étudier ici quelques régimes transitoires du protocole ARINC 629 CP, et sa réaction a de telles situations. Ces études sont primordiales dans un contexte temps critique, puisqu'elles permettent de valider la robustesse du protocole.

Le but dans cette étude est de montrer que le protocole, suite à un régime transitoire, revient toujours à un régime normal. C'est pourquoi nous ne présentons dans ce qui suit que des analyses qualitatives (mis à part pour la situation de collision, qui nécessite une analyse qualitative quantifiée pour bien comprendre les causes de la collision). Les analyses quantitatives effectuées pour ces régimes transitoires seront trouvées dans [GBJ96b, GJB96, Gal97].

Nous commencerons par étudier la surcharge, qui peut se produire sans qu'aucune panne n'apparaisse sur une station. Nous étudierons ensuite quelques situations caractéristiques du monde de l'avionique.

V.4.2.1 Situation de surcharge

Une situation de surcharge correspond à une durée du niveau périodique plus grande que la durée du cycle-bus. Ceci est rendu possible par la longueur variable des messages périodiques.

Pour mettre en place un scénario de surcharge, et voir ses conséquences sur le premier cycle-bus normal qui suit la surcharge, nous avons modifié nos réseaux de Petri de façon

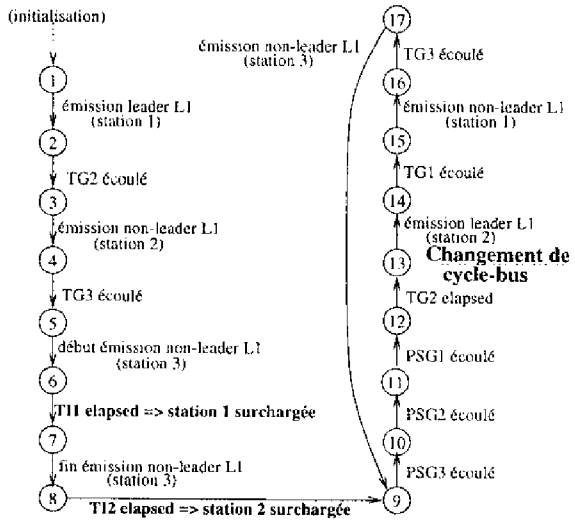


FIG. V.23 – *Situation de surcharge (automate quotient qualitatif)*

à émettre de longs messages dans le premier cycle-bus, puis des messages beaucoup plus courts dans les cycles-bus suivants.

La figure V.23 montre ce scénario de surcharge et ses conséquences possibles. Trois stations sont initialisées sur le bus, la station 1 étant leader. L'ordre des TG est donné par $TG_2 < TG_1 < TG_3$.

De l'état 1 à l'état 6, on voit un niveau L1 tout à fait normal, avec les émissions successives des stations 1, 2 et 3. Notons que la station 1, même si elle n'a pas le TG le plus petit, émet la première dans ce niveau puisqu'elle est la station leader.

Pendant l'émission périodique de la station 3, le TI de la station 1 s'achève (état 6 à état 7). On considère alors que la station 1 est en surcharge, car elle n'est plus capable d'assurer la périodicité du cycle-bus (son TI est écoulé avant qu'elle ne puisse changer de cycle-bus). Plus précisément, on dira qu'une station est en surcharge si le TG qu'elle lance à la fin de son PSG s'achève *après* son TI. Notons qu'une situation de surcharge n'influe pas directement sur le déroulement du cycle-bus, puisqu'aucune action particulière n'est accomplie par une station surchargée.

Après la fin d'émission de la station 3 (état 7 à état 8), et pendant l'écoulement des ASG pour changer de niveau, le TI de la station 2 s'achève (état 8 à état 9), ce qui indique que la station 2 est en surcharge. A la fin des ASG, comme il n'y a pas de temps pour émettre des messages apériodiques, on attend la fin des PSG. Ils s'achèvent dans un intervalle temporel d'un temps de propagation, et dans un ordre dicté par le positionnement des stations sur le bus. A la fin de chaque PSG, le TG de la station concerné est relancé.

Le TG le plus petit est celui de la station 2. C'est donc lui qui va s'achever en premier (état 12 à état 13). La station 2 est donc désignée leader du nouveau cycle-bus. Le reste

du cycle-bus est tout à fait classique.

On sort de cette analyse trois conséquences principales d'une surcharge :

une situation de surcharge ne concerne pas forcément l'ensemble des stations sur le bus; le leader est forcément la première station en surcharge (puisqu'elle est la première station à voir son TI s'achever);

il y a perte de périodicité du cycle-bus, puisque le leader est la première station touchée par la surcharge;

si le leader n'est pas la station qui a le plus petit TG, il peut y avoir changement de leader.

Un autre scénario aurait pu nous montrer qu'une collision est possible à la sortie de la surcharge entre la station qui a le plus petit TG parmi les stations en surcharge, et la station qui a le plus petit TG parmi les stations non en surcharge. Nous ne développerons pas cette situation ici. Le lecteur trouvera cette analyse dans [GBJ96b].

V.4.2.2 Panne et réparation

Le but de cette analyse est de montrer que la disparition d'une station non leader du cycle-bus et sa réapparition n'influent pas sur le bon déroulement de ce dernier.

Le scénario étudié comprend trois stations, avec comme ordre des TG : $TG_1 < TG_2 < TG_3$. La station 1 est leader du cycle-bus, et on suppose que seul le niveau périodique est accessible. La station 2 disparaît pendant un cycle-bus avant de réapparaître.

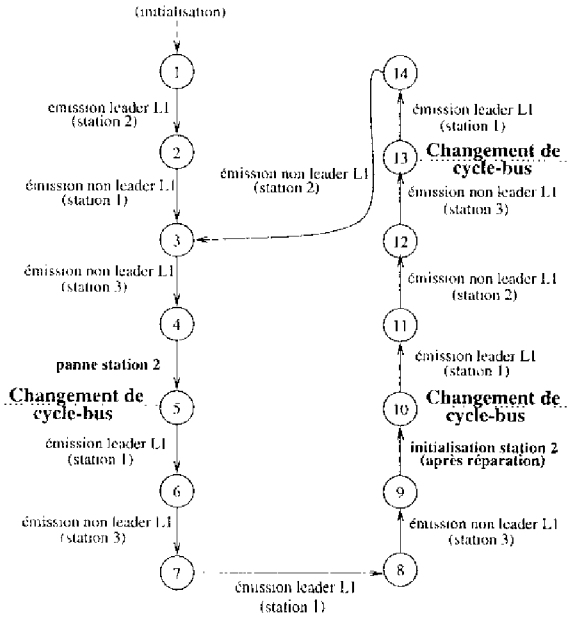


FIG. V.24 *Situation de panne et de réparation (automate quotient qualitatif)*

La figure V.24 donne une vue réduite du graphe d'états probabilisé obtenu. Nous n'avons représenté sur ce graphe que les émissions périodiques des stations, la panne et la réapparition de la station 2.

La panne de la station 2 a lieu à la fin du premier cycle-bus, après les émissions périodiques des trois stations (état 4 à état 5). Dans le cycle-bus qui suit, où la station 2 n'apparaît plus, aucune perturbation ne vient gêner le bon déroulement du cycle-bus : les stations 1 et 3 continuent à émettre normalement. A la fin de ce cycle-bus, et après une phase de réinitialisation complète, la station 2 tente de s'introduire dans le cycle-bus. Elle y arrive lors de l'émission périodique de la station leader (station 1), qui lui permet d'être introduite comme station non leader. Son TG étant plus petit que celui de la station 3, elle reprend la place qu'elle occupait avant son arrêt.

Notons que si la panne se produit pendant l'émission de la station, le message de cette station sera incompréhensible pour les autres stations, mais le déroulement du cycle-bus n'est pas affecté.

Les principales propriétés vérifiées sur ce scénario sont :

- une panne sur une station n'affecte pas l'état des autres stations;

  lors de sa réintroduction, la station s'intègre dans le train des stations déjà sur le bus sans provoquer de perturbation;

V.4.2.3 Transmissions fantômes

Nous nous proposons d'étudier ici l'apparition d'activités intempestives sur le bus, que nous appellerons messages fantômes. Ces activités peuvent apparaître sur le bus, par exemple, quand la foudre tombe sur un avion en vol. Le but de cette étude est de montrer que le protocole, après une phase transitoire particulière, retrouve bien un fonctionnement normal.

Nous considérons ici que deux stations sont sur le bus. La station 1, leader, a le plus petit TG. Nous étudions l'effet de l'apparition de trois messages fantômes dans un régime normal.

La figure V.25 donne une vue réduite du graphe d'états probabilisé obtenu. Après un premier cycle-bus classique (état 1 à état 5), une transmission fantôme apparaît sur le bus après la fin de l'écoulement des PSG des deux stations, mais avant la fin de l'écoulement du TI de la station leader.

Cette activité est considérée par les deux stations comme étant le message périodique d'une troisième station (que nous appellerons station fantôme), et donc les deux stations changent de cycle-bus (apparition de l'événement CE sur les deux stations : état 6 à état 8), et se considèrent non leader. On entre donc dans un cycle-bus particulier, où il n'y a pas de station leader.

A la fin de cette transmission (état 9), les deux stations entrent en compétition pour avoir accès au bus. La station 1 qui a le plus petit TG voit ce dernier s'achever en premier,

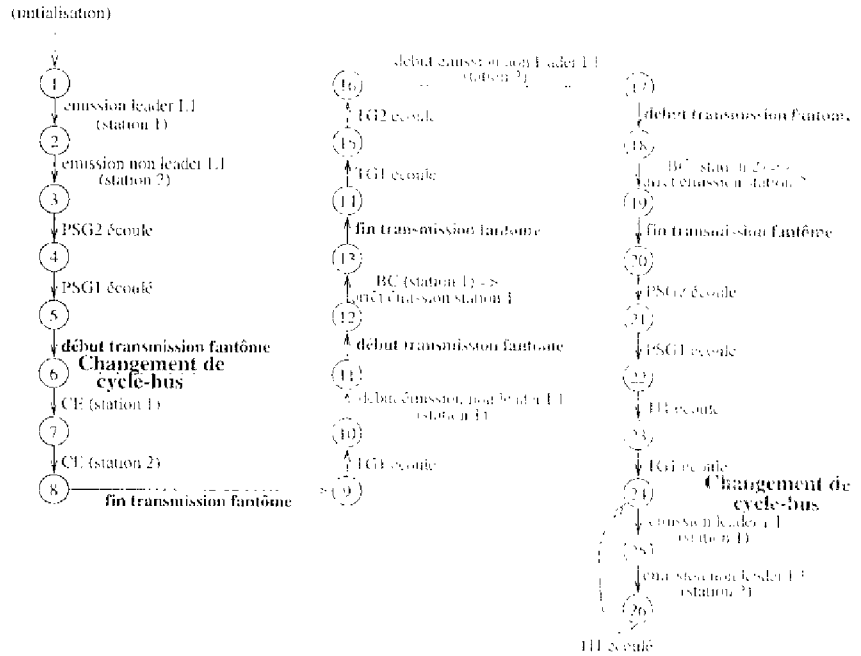


FIG. V.25 *Influence de transmissions fantômes sur le cycle-bus (automate quotient qualitatif)*

et commence à émettre son message périodique (état 9 à état 11). Pendant cette émission, un second message fantôme apparaît sur le bus. Ce message brouille le message de la station 1, qui se considère alors en collision avec la station fantôme (état 11 à état 13). Elle arrête donc son émission et passe en attente de fin de cycle-bus.

A la fin de cette deuxième activité fantôme, les stations 1 et 2 écoulent leurs TG. L'écoulement du TG de la station 1 ne provoque rien, puisque la station attend la fin du cycle-bus (attente de fin d'écoulement de PSG). La station 2, à la fin de son TG, peut émettre son message périodique (état 15 à état 17), puisqu'elle n'a pas encore émis et qu'elle n'a pas été impliquée dans la collision.

Durant cette émission, un troisième message fantôme apparaît sur le bus, provoquant une collision avec la station 2. Celle-ci s'arrête d'émettre, et se place en attente de fin de cycle-bus, comme la station 1.

Après la fin de l'activité fantôme, les deux stations écoulent leur PSG (état 20 à état 22). L'écoulement des TG après les PSG n'est pas représenté sur cette figure.

La première station à voir son TI s'achever est la station 1, puisqu'elle avait été la première à émettre en début de cycle-bus. A la fin de ce TI (état 23), la récupération de collision implique que la station doit écouler son TG. A la fin de cet écoulement (état 24), la station se considère comme leader, et entame un nouveau cycle-bus par l'émission de son message périodique. La station 2, sur l'apparition du CE, change elle aussi de cycle-bus, et se considère comme non leader. On est à partir de cet instant revenu dans un régime

normal.

Les principales propriétés vérifiées sur ce scénario sont :

- après disparition des messages fantômes, une des stations reprend l'attribut leader pour reprendre un régime normal;

  il y a perte de périodicité quand le leader est touché par la perturbation;

V.4.3 Remarque sur les régimes transitoires

Nous n'avons pas donné ici de résultats quantitatifs pour des raisons de place insuffisante. Néanmoins, nous avons pu vérifier que le protocole, suite à une perturbation, retrouvait un comportement normal dans un temps inférieur à $2.TI$ après la fin de cette perturbation.

V.5 Conclusion

La but de ce chapitre était double :

montrer l'adéquation du modèle Réseaux de Petri Temporisés Stochastiques à l'étude des systèmes distribués temps-réel,

utiliser les outils d'analyse du graphe d'états probabilisé pour valider les principales propriétés du protocole ARINC 629 CP.

La modélisation du protocole a nécessité un découpage en module. Chaque module, après avoir été modélisé, communique de manière asynchrone avec les autres modules.
Les différents mécanismes temporels mis en œuvre dans le protocole CP, et notamment les cinq temporisations de chaque station, ont pu être facilement représentés par le modèle Réseaux de Petri Temporisés Stochastiques.

La validation du protocole s'est essentiellement faite par l'intermédiaires d'analyses qualitatives et qualitatives quantifiées. Mais des études quantitatives ont été décrites dans [GBJ96c, GBJ96a, GBJ96b, GJB96] qui permettent de mesurer, par exemple, la durée des cycles-bus ou le temps de récupération après une perturbation. Ici encore, les outils d'analyse associés au modèle Réseaux de Petri Temporisés Stochastiques se sont révélés tout à fait adaptés à l'analyse des systèmes distribués temps-réel.

Ces différentes analyses ont montré tout l'intérêt de ce protocole, à savoir des caractéristiques temps-réel et des caractéristiques de tolérance aux pannes.

Notons encore qu'après avoir validé le protocole ARINC 629 CP, nous avons donné un guide de passage des Réseaux de Petri Temporisés Stochastiques vers LDS, de manière à ce que nos résultats puissent être intégrés dans les ateliers d'Aérospatiale.

Conclusion et Perspectives

Le travail présenté dans ce mémoire concerne des extensions et des applications du modèle Réseaux de Petri Temporisés Stochastiques, modèle de spécification formelle des systèmes distribués temps-réel, sur la base d'une première définition donnée dans [Ata94].

Nous résumons maintenant les contributions et principaux résultats que nous avons obtenus, et donnons quelques perspectives.

Contributions

Du point de vue du pouvoir d'expression du modèle RdPTS :

plusieurs règles de tir des transitions ont été définies, et notamment les règles MIN, moyen et MAX. Elles permettent d'étudier des scénarios particuliers du comportement dynamique du système étudié. Ces règles permettent l'analyse des cas pires, ce qui, dans un contexte temps-réel, et plus particulièrement dans le domaine des réseaux de communication temps critique, permet de valider le bon comportement de tels systèmes;

l'utilisation de différentes politiques de mémoire temporelle a été autorisée, ce qui permet d'augmenter le domaine d'application du modèle : la mémoire de la dernière sensibilisation permet de modéliser des mécanismes de type « time-out »; la mémoire de toutes les sensibilisations permet de modéliser les mécanismes de préemption. L'utilisation de la préemption avec des densités de probabilité de type uniforme n'avait, à notre connaissance, jamais été faite.

Du point de vue du pouvoir d'analyse du modèle RdPTS :

- nous avons défini la notion d'automate quotient qualitatif quantifié. Cette notion est basée sur la relation d'équivalence observationnelle de Milner, qui permet d'obtenir un automate quotient du graphe des classes d'états relatif à un sous-ensemble d'étiquettes qualitatives, et les règles de réduction de Beizer, qui permettent de quantifier les classes d'états de l'automate quotient qualitatif.

Cet automate permet de faire des analyses à la fois qualitatives et quantitatives des graphes d'états probabilisés à partir d'un seul et même objet. Nous l'avons utilisé dans la modélisation de service, où il permet d'obtenir une vue équivalente simple quantifiée de ce dernier. Un autre avantage de cet automate est qu'il permet, par exemple dans une modélisation ascendante d'une architecture protocolaire en couches, de combattre l'explosion combinatoire (à la fois au niveau de la structure du Réseau de Petri équivalent (nombre de places et de transitions) et au niveau du graphe d'états probabilisé (nombre d'états)).

Une réflexion sur les modèles RdPTS et RdPT a permis :

de positionner les graphes d'états probabilisés MIN, moyen et MAX au sein de la référence comportementale qu'est le graphe des classes d'états (qui donne le comportement dynamique des RdPT). Ce positionnement permet d'obtenir des points de vue quantifiés du comportement du système;

de montrer, à travers deux exemples, qu'il est impossible, par construction, d'effectuer certaines analyses qualitatives sur le graphe des classes d'états (hétérogénéité comportementale, inclusion et intersection de classes d'états). Nous avons défini, pour des cas particulier de Réseaux de Petri, la notion de graphe des sous classes d'états, qui permet d'éliminer l'hétérogénéité comportementale du graphe des classes d'états.

Nous avons montré l'adéquation du modèle RdPTS et de ses extensions à la modélisation et l'analyse des systèmes distribués temps-réel à travers l'analyse et la validation du protocole de bus embarqué temps critique ARINC 629 CP. Les aspects temps critique (cinq temporisations par station) et de tolérance aux pannes de ce protocole ont été mis en exergue;

Ces extensions ont toutes été implémentées dans un prototype logiciel de simulation des Réseaux de Petri Temporisés Stochastiques (RPTS), sur la base d'un premier outil développé dans le groupe OLC [Ata93].

Perspectives

- Au niveau du pouvoir d'expression du modèle RdPTS, d'autres extensions doivent être étudiées de manière à augmenter le domaine d'application du modèle. Jusqu'à présent, les densités de probabilités associées aux transitions sont affectées de manière statique. Il serait intéressant de pouvoir avoir des densités de probabilités dynamiques, qui dépendraient de l'état du système. On peut citer à titre d'exemple le transfert de la densité de probabilité d'une transition tirée à une autre transition (ceci permettrait, entre autre, de modéliser des mécanismes comme la temporisation THT du protocole du bus à jeton), ou la dépendance vis-à-vis du marquage des taux de densités exponentielles (cette extension est déjà utilisée dans d'autres modèles, comme SPN et GSPN);

- Lorsque les systèmes étudiés sont complexes, leur représentation en Réseau de Petri Temporisé Stochastique peut conduire à des modèles de grande taille, et par conséquent difficilement exploitables (notamment à cause de la taille du graphe d'états

probabilisé correspondant). Le concept de Réseaux de Petri Stochastiques Bien Formés [CDFH91] permet de définir un graphe symbolique qui représente le comportement du système. La combinaison entre les concepts de graphe des marquages symbolique et graphe d'états probabilisé a été initiée dans [Ata94]. Cet axe de recherche doit être poursuivi pour permettre de prendre en compte les symétries des systèmes modélisés.

- Concernant la relation entre le modèle RdPTS et le modèle RdPT, d'autres études doivent être menées pour éliminer du graphe des classes d'états l'hétérogénéité comportementale et les problèmes d'inclusion et d'intersection de classes. Les concepts de graphes MIN et MAX du modèle RdPTS doivent pouvoir être utilisés pour quantifier ce nouveau graphe, et augmenter ainsi le pouvoir d'analyse du modèle RdPT;

- Dans un cadre applicatif, l'augmentation du pouvoir d'expression du modèle RdPTS doit permettre désormais de l'utiliser dans d'autres domaines que celui des protocoles de communication, comme par exemple les Systèmes de Production.

Références bibliographiques

[**Air91**] Airlines Electronic Engineering Committee. *Multi-Transmitter Data Bus. ARINC Specification 629-2 : Part 1, Technical Description*, Aeronautical Radio Inc. edition, october 1991.

[**AJ95**] Y. Atamna and G. Juanole. Methodology for Obtaining Abstract Views of State Graphs Labeled with Probabilities and Times : an Example of Application to a Communication Protocol. In *MASCOTS'95, the Third International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 299–306, Durham, North Carolina, january 1995.

[**ALL83**] J. F. ALLEN. Maintaining knowledge about temporal intervals. In *Communications of the ACM*, volume 126. december 1983.

[**Ata93**] Y. Atamna. RPTS, a tool for Stochastic Timed Petri Nets. In *Fifth International Workshop on Petri Nets and Performance Models, Tools Exhibition*, Toulouse, France, 1993.

[**Ata94**] Y. Atamna. *Réseaux de Petri Temporisés Stochastiques Classiques et Bien Formés : Définition, Analyse et Application aux Systèmes Distribués Temps Réel.* PhD thesis, Université Paul Sabatier de Toulouse (FRANCE), october 1994.

[**BB93**] H. Boucheneb and G. Berthelot. Towards a simplified building of Time Petri nets reachability graph. In *5rd Interntional Workshop on Petri Nets and Performance Models PNPM'93*, 1993.

[**BCFMR87**] G. Balbo, G. Chiola, G. Francheschinis, and G. Molinear Roet. On the efficient construction of the tangible reachability graph of Generalized Stochastic Petri Nets. In *PNPM'87, 2nd International Workshop on Petri Nets and Performance Models*, Madison, USA, August 1987.

[**Bei71**] B. Beizer. *The Architecture and Engineering of Digital Computer Complexes*, volume 1. Plenum Press, 1971.

[**Bla89**] A. Blakemore. The Cost of Elimination Vanishing Markings from Generalized Stochastic Petri Nets. In *3rd Interntional Workshop on Petri Nets and Performance Models PNPM'89*, Kyoto, Japan, December 1989.

[BM82] B. Berthomieu and M. Menasche. A state enumeration approach for analysing time Petri nets. In *3rd European Workshop on Applications and Theory of Petri Nets*, 1982.

[BM83] B. Berthomieu and M. Menasche. An Enumerative Approach for Analysing Time Petri Nets. In *IFIP Congress 1983*, Paris, France, Septembre 1983.

[Bra83] G. W. Brams. *Réseaux de Petri: théorie et pratique*, volume 1 et 2. Masson Editeur, 1983.

[BSW69] K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on realible full-duplex transmission over half-duplex links. In *Commun. Ass. Comput. Mach.*, volume 12. may 1969.

[CDFH91] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic Well-Formed Coloured Petri Nets and multiprocessor modelling application. In K. Jensen and G. Rozenberg, editors, *High-Level Petri Nets. Theory and Application*. Springer Verlag, 1991.

[CGL93] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. In *PNPM'93, 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, october 1993.

[Chi85] G. Chiola. A Software Package for the Evaluation of Stochastic Petri Net Models. In *TPN'85, the First International Workshop on Time Petri Nets*, Torino, Italia, July 1985.

[Cia93] G. Ciardo. Generalized and Deterministic Petri Nets (GSPN et DSPN). In *Tutorials of PNPM'93, 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, october 1993.

[CKT93] H. Choi, V.G. Kilkarni, and K.S. Trivedi. Transient Analysis of Deterministic and Stochastic Petri Nets. In *14th European Workshop on Application and Theory of Petri Nets*, Chicago, USA, june 1993.

[CL93] G. Ciardo and C. Lindemann. Analysis of Deterministic and Stochastic Petri Nets. In *PNPM'93, 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, october 1993.

[DB91] M. Diaz and B. Berthomieu. Modeling and Verification of Time Dependant Systems Using Time Petri Nets. *IEEE Transactions on Software Engineering*, 1991.

[DeHA89] R. David et H. Alla. *Du Grafcet aux réseaux de Petri*. Traité des Nouvelles Technologies, série Automatique, 1989.

[Don97] S. Donatelli. An introduction to Timed Petri Nets and Generalized Stochastic Petri Nets. In *Introductory Tutorial of ICATPN'97, the 18th International Conference on Application and Theory of Petri Nets*, Toulouse, France, june 1997.

[DTGN84] J.B. Dugan, K.S. Trivedi, R.M. Geist, and V.F. Nicola. Extended Stochastic Petri Nets: Applications and Analysis. In *10th International Symposium on Computer Performance*, Paris, december 1984.

[Flo85] G Florin. *Réseaux de Petri Stochastiques: théorie et techniques de calcul.* PhD thesis, Université Pierre et Marie Curie (Paris VI), 1985.

[FN85] G. Florin and S. Natkin. Les Réseaux de Petri Stochastiques. *TSI*, 4(1):143-160, february 1985.

[Gal97] L. Gallon. Modélisation par Réseaux de Petri Temporisés Stochastiques et Analyse du protocole ARINC 629 CP : guides de passage des Réseaux de Petri Temporisés Stochastiques à LDS. Technical Report 97022, LAAS-CNRS, january 1997.

[GBJ96a] L. Gallon, I. Blum, and G. Juanole. Modélisation par Réseaux de Petri Temporisés Stochastiques et Analyse du protocole ARINC 629 CP : le fonctionnement normal. Technical Report 96328, LAAS-CNRS, august 1996.

[GBJ96b] L. Gallon, I. Blum, and G. Juanole. Modélisation par Réseaux de Petri Temporisés Stochastiques et Analyse du protocole ARINC 629 CP : 1) les fonctionnements transitoires en régime normal et suite à l'occurrence de surcharges ou de pannes, 2) le phénomène de collision, ses causes et son contrôle. Technical Report 96411, LAAS-CNRS, october 1996.

[GBJ96c] L. Gallon, I. Blum, and G. Juanole. Présentation du protocole ARINC 629 CP et du cadre formel prévu pour son étude. Technical Report 96144, LAAS-CNRS, may 1996.

[GJB96] L. Gallon, G. Juanole, and I. Blum. Modélisation par Réseaux de Petri Temporisés Stochastiques et Analyse du protocole ARINC 629 CP : des phénomènes particuliers : surdiré d'une station, pertes de messages sur le bus, fluctuation du signal BQ, transmissions fantômes, répétition de collisions. Technical Report 96427, LAAS-CNRS, november 1996.

[GJB97] L. Gallon, G. Juanole, and I. Blum. Modelling and Analysis of the ARINC 629 CP MAC layer protocol. In *WFCS'97, IEEE International Worshop on Factory Communication Systems*, Barcelone, Spain, october 1997.

[HV87] M.A. Holliday and M.K. Vernon. A Generalized Timed Petri Net Model for Performance Analysis. *IEEE Transactions on Software Engineering*, 1987.

[Kle75] L. Kleinrock. *Queuing systems.* Jonh Wiley and Sons, 1975.

[KS60] J.G. Kennedy and J.L. Snell. *Finite Markov Chains.* Norstrand, V and Princeton, 1960.

[Liè97] L Lièvre. Rpts, outil logiciel de modélisation et d'analyse des systèmes à évènements discrets : Développement d'une Interface Graphique, Application à la Modélisation d'Algorithmes d'Ordonnancement de Tâches et de Messages.

Technical report, mémoire de fin d'étude et rapport de DEA, INSA Toulouse, 1997.

[MBAB+89] M.A. Marsan, G. Balbo, Andrea Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15(7):832-846, july 1989.

[MBB+85] M.A.. Marsan, G. Balbo, A. Bobbio, G. Conte, G. Chiola, and A. Cumani. On Petri Nets with Stochastic Timing. In *TPN'85, the First International Workshop on Time Petri Nets*, Torino, Italia, 1985.

[MBC86] M.A. Marsan, G. Balbo, and G. Conte. Generalized stochastic Petri nets revisited : random switches and priorities. In *Int. Workshop on Petri Nets and Performance Models (PNPM'86)*, pages 44–53, Madisson, USA, 1986.

[MBCC84] A.M. Marsan, G. Balbo, G. Conte, and G. Chiola. A class of Generalized Stochastic Petri Nets for the performance analysis of multiprocessor systems. In *ACM TOCS*, volume 2, may 1984.

[MC86] A.M. Marsan and G. Chiola. On Petri Nets with Deterministic and Exponential transition firing time. In *7th European Workshop on Application and Theory of PetriNets*, Oxford, june 1986.

[Men82] M. Menasche. *Analyse des Réseaux de Petri Temporisés et Application aux Systèmes Distribués*. PhD thesis, Université Paul Sabatier de Toulouse, 1982.

[MF76] P.M. Merlin and D.J. Farber. Recoverability of communication protocols, implications of a theorical study. *IEEE Trans. on Communications*, 24, 1976.

[Mil80] R. Milner. *A Calculus of Communication Systems*, volume 92. Springer Verlag, Berlin Heidelberg, 1980.

[Mol81] M.K. Molloy. *On the integration of Delay and throughput Measures in Distributed Processing Models*. PhD thesis, University of California, Los Angeles, 1981.

[Mol85] M.K. Molloy. Discrete Time Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, 11(4), april 1985.

[Mur89] T. Murata. Petri Nets : Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4), april 1989.

[Nat80] S. Natkin. *Les Réseaux de Petri Stochastiques*. PhD thesis, (Docteur-Igénieur) CNAM, Paris, 1980.

[NOR94] *IEC SC65C/WG6 Digital data communications for measurement and control : data link protocol specifications (DRAFT)*, 1994.

[Ram74] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. PhD thesis, MIT, Project MAC TR-120, Février 1974.

[RH80] C.V. Ramamoorthy and G.S. Ho. Performance evaluation of asynchronous consurrent systems using Petri Nets. *IEEE Transactions on Software Engineering*, SE-6, 1980.

[RJ87] J.L. Roux and G. Juanole. Functional and Performance Analysis Using Extended Time Petri Nets. In *PNPM'87, the Second International Workshop on Petri Nets and Performance Models*, pages 14–23, Madison, Wisconsin, august 1987.

[Rou85] J.L. Roux. *Modélisation et analyse des systèmes distribués par les réseaux de Petri Temporels.* PhD thesis, Institut National des Sciences Appliquées de Toulouse, 1985.

[RP84] R.R. Razouk and C.V. Phelps. Performance Analysis using Timed Petri Nets. In *International Conference on Parallel Processing and Performance Models*, august 1984.

[Sif77] J. Sifakis. Use of Petri Nets for performance evaluation. In J. Beilner and North Holland E. Gelende, editors, *3rd Int. Symposium on Modelling and Performance Evaluation of Computer Systems*, 1977.

[Tou97] J. Toussaint. *Modélisation d'applications temps-réel réparties pour la validation de propriétés temporelles. Méthodologie de construction de modèles et algorithmes de validation.* PhD thesis, Institut National Polytechnique de Lorraine, octobre 1997.

[Vas96] F. Vasques. *Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC de réseaux locaux temps-réel.* PhD thesis, Université Paul Sabatier, Toulouse, 1996.

[Ver89] F. Vernadat. *Vérification Formelle d'Applications Réparties. Caractérisation Logique d'une Equivalence de Comportement.* PhD thesis, Université Paul Sabatier de Toulouse (FRANCE), 1989.

[Zub80] W.W. Zuberek. Timed Petri Nets and Preliminary Performance Evaluation. In *7th Annual Symposium on Computer Architecture*, may 1980.

RESUME

Cette thèse s'inscrit dans le cadre général de l'utilisation des techniques formelles lors de la phase conceptuelle des systèmes distribués temps-réel, et concerne plus précisément l'extension des pouvoirs d'expression et d'analyse du modèle Réseaux de Petri Temporisés Stochastiques. Ce modèle, qui appartient à la classe des Réseaux de Petri Stochastiques, associe à chaque transition un intervalle de temps, et une distribution de probabilité sur cet intervalle (ceci permet de représenter des caractéristiques temporelles variées et en particulier des contraintes)
En ce qui concerne le pouvoir d'expression, nous avons introduit, d'une part, la notion de mémoire temporelle de toutes les sensibilisations (ceci est important pour l'étude des phénomènes de préemption que l'on rencontre dans des algorithmes d'ordonnancement et dans des problèmes de sûreté de fonctionnement) et, d'autre part, plusieurs règles de tir de transitions (en particulier les règles MIN et MAX qui permettent des études des cas pires, aspect important dans les systèmes temps-réels). Les graphes ainsi obtenus ont été également situés par rapport à la référence comportementale que constitue le graphe des classes d'états (obtenu à partir des Réseaux de Petri Temporels).
En ce qui concerne le pouvoir d'analyse, nous avons défini le concept d'automate quotient quantifié (concept basé, à la fois, sur la relation d'équivalence de Milner et les règles de réduction de Beizer), qui a deux qualités importantes : il permet, d'une part, d'obtenir des modèles, à la fois qualitatifs et quantitatifs de services de communication ; et, d'autre part, de contrôler les dimensions des modèles à traiter lors de la modélisation d'architectures de communication multicouches.
Ce modèle a été appliqué à un exemple industriel, le protocole embarqué temps-réel ARINC 629 CP. L'étude faite a mis en évidence l
es propriétés temps-réel et de tolérance aux pannes de ce protocole.

MOTS CLES : systèmes distribués temps-réel, technique formelle, Réseaux de Petri Temporisés Stochastiques, validation, évaluation

ABSTRACT

This thesis is submitted within the framework of formal description techniques used during the design process of real-time distributed systems. More precisely, it concerns the extension of the expression and analysis powers of the Stochastic Timed Petri Nets model (STPN). This model associates to each transition of a Petri Net a time interval and a density probability function on this interval (a lot of time characteristics can thus be represented and, in particular, time constraints).
The expression power improvement concerns the introduction, on the one hand, of the age memory concept (which allows to study preemption mechanisms and is very useful in scheduling algorithms and dependability analysis) and, on the other hand, of several transition firing rules (in particular, MIN and MAX rules which allows o study worst cases, very important aspect in real-time systems). The different graphs which are obtained with this rules have been situated with respect to the behavioural reference which is the state class graph (considering only time intervals).
The analysis power improvement concerns the definition of the Quantified Abstract Quotient Automaton concept (based on the Milner equivalence relation and Beizer rules) : it provides abstract models which are both qualitative and quantitative; il allows to control the size of the models to consider when modelling multilayer communication architectures.
This model has been applied to the study of the communication protocol ARINC 629 CP (for plane systems). This study has demonstrated the real-time and fault tolerance properties of the protocol.

KEYWORDS : Real-time distributed systems, Formal technique, Stochastic Timed Petri Net, Validation, Evaluation