

**Le Service de la couche MAC
pour l'Anneau à Jeton
Modélisation et validation par
réseaux de Petri Réguliers**

**Bernard COUSIN
Jean-Michel COUVREUR**

Laboratoire MASI - CNRS 818 - PRC C³
Université Pierre et Marie CURIE
4, place Jussieu
75252 - PARIS cedex 05

Résumé

Cette étude porte sur le service du niveau Mac en phase de transfert de données. Le protocole du niveau Mac est une sous-couche de la couche Liaison, conformément aux normes internationales, gérant les accès à un réseau local. L'anneau à jeton constitue pour nous le médium supportant notre protocole MAC.

Nous construisons un modèle du service en Réseau de Petri Régulier. Notre travail met en évidence la puissance de modélisation et les méthodes d'analyse qui justifient l'intérêt des réseaux réguliers. Nous vérifions des propriétés générales (vivacité, état d'accueil) et des propriétés spécifiques au protocole :

- Un adaptateur émetteur ne traite qu'un transfert à la fois. Il commence par une demande d'émission de trame et finit par une confirmation positive ou négative.
- Une confirmation positive indique que l'adaptateur destinataire a reçu la trame.
- Une confirmation négative indique une perturbation dans la transmission, mais la trame peut avoir été reçue.
- Les trames ne sont pas déséquencées.

PLAN

1.- Présentation générale.....	p.5
2.- Interface Plasma / Mac.....	p.7
3.- Mécanisme de transfert de données.....	p.9
3.1- Processus émetteur.....	p.9
3.2- Processus récepteur.....	p.11
3.3- Processus de transfert.....	p.12
4.- Modèle du service.....	p.15
4.1- Les classes.....	p.15
4.2- Les places.....	p.15
4.3- Les transitions.....	p.18
4.4- L'état initial.....	p.19
5.- Validation du modèle.....	p.20
5.1- Introduction.....	p.20
5.2- Calcul d'une base de flots.....	p.21
5.3- Vivacité.....	p.30
5.4- Propriétés spécifiques du modèle.....	p.40
6.- Conclusion	p.47

Figures

Fig 1.1	Architecture fonctionnelle.....	p.5
Fig 1.2	Architecture générale.....	p.6
Fig 2.1	Interface du Service MAC.....	p.7
Fig 2.2	Mode d'utilisation des primitives MAC.....	p.8
Fig 3	Mécanisme de transfert de données du service MAC.....	p.10
Fig 3.1	Processus émetteur.....	p.10
Fig 3.2	Processus récepteur.....	p.11
Fig 3.3	Processus de transfert.....	p.14
Fig 4	Modèle du service.....	p.17
Fig 5.2.1	Etapes de calcul d'une base de flots.....	p.21
Fig 5.2.2	Arbre de transformation d'un réseau régulier.....	p.22
Fig 5.2.3	Le Flot du processus Emetteur.....	p.26
Fig 5.2.4	Le Flot du processus Récepteur.....	p.27
Fig 5.2.5	Le Flot du processus Jeton.....	p.28
Fig 5.2.6	Le Flot du processus Transfert.....	p.29
Fig 5.3.2	Construction d'une séquence transformant un état M en Mo.....	p.32
Fig 5.3.3	Graphe des marquages réduits.....	p.39
Fig 5.4.1.1	Enchaînement des transitions : MA_Req, MA_Conf(+) et MA_Conf(-).....	p.40
Fig 5.4.1.2	Graphe associé aux transitions : MA_Req, MA_Conf(+) et MA_Conf(-).....	p.42
Fig 5.4.2.1	Les trois scénarios d'un transfert de données.....	p.43
Fig 5.4.2.2	Graphe associé aux transitions : MA_Req, MA_Conf(+), MA_Conf(-), Pert_message, Pert_transf, TR_Ind ...	p.44

Références Bibliographiques

- [IEEE 802.5] *Token ring access methode and Physical layer specification*. ANSI/IEEE Standard 802.5 ,1985.
- [TMS 380] *TMS380 Adaptater Chipset, User's Guide*, 1986.
- [Cousin 87] B.Cousin, P. Estraillier, *PLASMA : A multi-medium Protocol for the link layer*. Rapport de recherche MASI n° 194, Septembre 1987.
- [Cousin 88] B.Cousin, P.Estraillier, J.M.Couvreur , *The virtual layer concept : Validation of the PLASMA protocol*. Soumis à ICC-88, TelAviv, october 1988.
- [Couvreur 88] J.M.Couvreur, S.Haddad, *Toward a general and powerful computation of flows for parametrized coloured nets*. Soumis au 9th European Workshop on Theory and Application of Petri Nets, Venise, June 1988.
- [Haddad 87] S. Haddad .*Une catégorie régulière de réseau de Petri de haut niveau: Définition, propriétés et réductions*. Thèse de docteur de l'université PARIS 6 , Juin 1987.

Abstract

We study the service of the **MAC sublayer** during its data transfer phase. The MAC protocol is a sublayer of the Link layer for Local Area Network. We use **Token ring** as medium.

We modelize the MAC service with Regular Petri Net. This example shows the power of modelization and validation of the Regular Petri Net. We verify general properties of the model (liveness, home state), and specific properties of the protocol (interface primitives, sequentiality).

Mots clefs

Réseau local, Anneau à jeton, niveau MAC, couche Liaison, phase de transfert de données, Modélisation, Validation, Réseau de Petri.

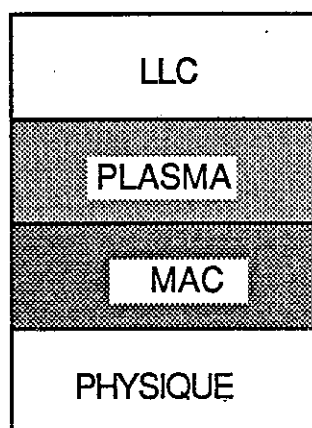
1.- PRESENTATION GENERALE

D'après la normalisation [IEEE 802.5], le protocole du **niveau MAC** gère les liaisons d'un **anneau à jeton**. Le niveau MAC forme avec le niveau LLC les deux sous-couches de la couche Liaison gérant les protocoles des réseaux locaux. Notre étude porte uniquement sur le service du niveau MAC en phase de transfert de données. On ne tiendra pas compte des mécanismes internes propres à la gestion de l'anneau:

- Transmission, supervision et maintenance du jeton;
- Insertion, retrait et contrôle des stations;
- etc...

Ces mécanismes sont autonomes et internes au protocole. Ils ne sont pas perçus par le niveau supérieur.

Cette étude constitue un pré-requis pour la modélisation du protocole PLASMA. Le protocole PLASMA permet de gérer de manière transparente aux couches supérieures (LLC) un double anneau à jeton [Cousin 87]. Pour l'intégration dans l'étude du protocole PLASMA, seules nous intéressent les primitives exprimant le service de transfert de données rendu par le niveau MAC.

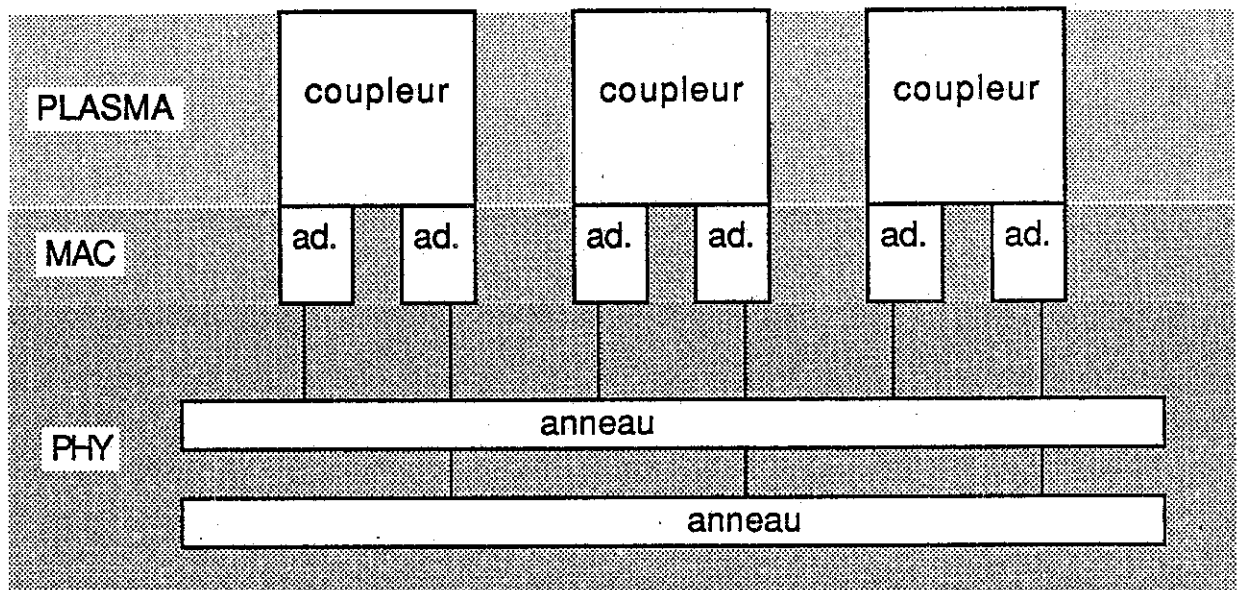


-Fig 1.1 - Architecture fonctionnelle-

Le Service de la couche MAC pour l'anneau à jeton

Le niveau MAC est supporté par un ensemble d'adaptateurs TMS380 connectés sur deux anneaux.

A chaque coupleur du niveau PLASMA sont associés deux adaptateurs, chacun connecté un anneau. Ainsi tout coupleur peut communiquer avec les autres coupleurs en utilisant les services offerts par ces deux adaptateurs.



-Fig 1.2 - Architecture générale-

Ce document est divisé en quatre parties formées par les chapitres suivants:

- Le chapitre 2 définit le service offert par le niveau MAC.
- Le chapitre 3 analyse les mécanismes de transmission.
- Le chapitre 4 présente le modèle du service.
- Le chapitre 5 analyse et valide le modèle.

2.- INTERFACE PLASMA / MAC

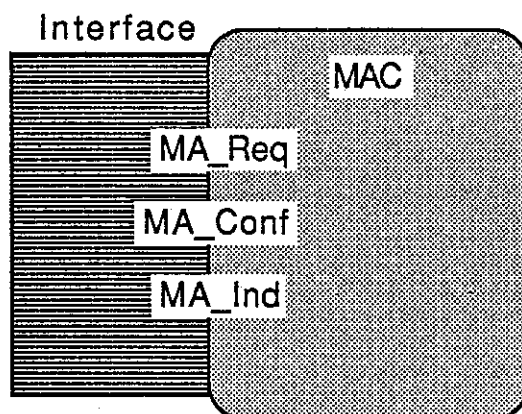
Le niveau MAC, conformément à la norme, communique avec le niveau PLASMA à travers un interface. De par la contrainte de transparence du niveau PLASMA, cet interface correspond très exactement à l'interface MAC / LLC standard.

Le transfert de données s'effectue en utilisant les trois primitives suivantes:

- MA_Data_Request : Demande d'émission des données,
- MA_Data_Indication : Réception de données,
- MA_Data_Confirmation : Confirmation d'une réception de données.

Dans tous nos modèles, nous utilisons pour les primitives les abréviations:

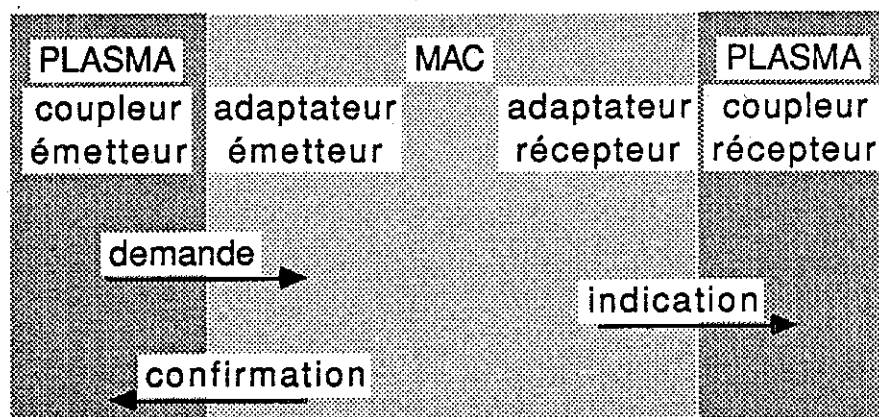
- MA_Data_Request : MA_Req,
- MA_Data_Indication: Ma_Ind,
- MA_Data_Confirmation: MA_Conf.



-Fig 2.1 - L'interface du service MAC-

Nous appelons "message" le triplet <adaptateur-émetteur, adaptateur-récepteur, données>. Il représente les trames qui circulent à travers les adaptateurs et dans l'anneau au cours d'un transfert de données.

Le schéma ci-après donne les principes d'utilisation de ces primitives :



-Fig 2.2 - Mode d'utilisation des primitives MAC-

La primitive *confirmation* est générée systématiquement par l'adaptateur émetteur en réponse à la *demande*.

Nous distinguons deux types de *confirmation* :

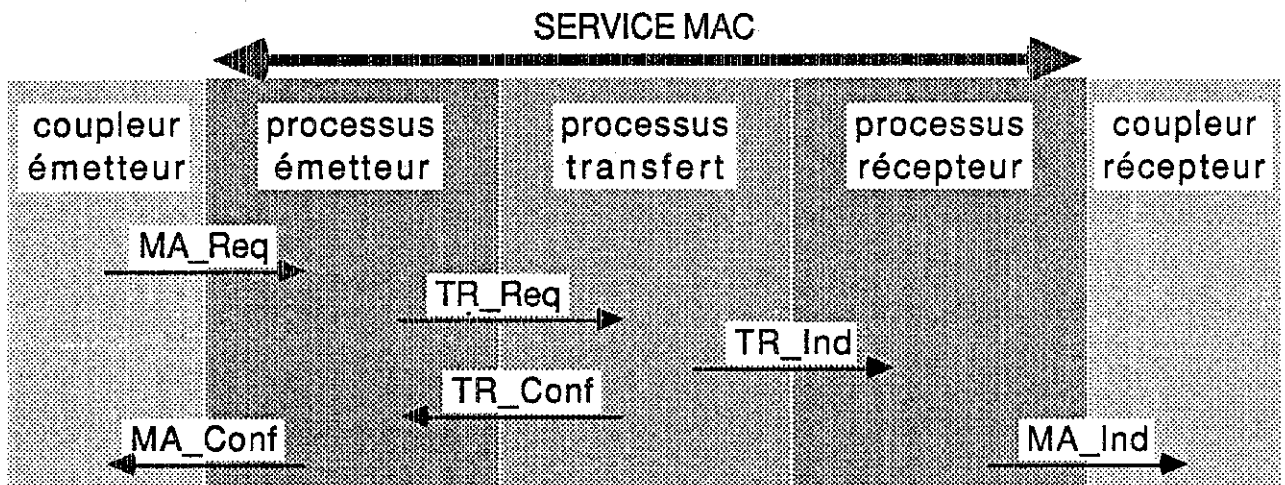
- *confirmation(+)* indique que l' adaptateur destinataire a reçu le message.
- *confirmation(-)* indique une perturbation (temporaire ou durable) au cours de la transmission à l'aller (de l'émetteur vers le destinataire) ou au retour (du destinataire vers l'émetteur). Attention, une confirmation n'est pas un acquittement, le message peut avoir ou ne pas avoir été reçu.

3. - MECANISME de TRANSFERT de DONNEES

Nous structurons le service de transfert du niveau MAC en trois processus:

- Le processus "émetteur" gère le transfert de données sur l'adaptateur émetteur.
- Le processus "récepteur" gère le transfert de données sur l'adaptateur récepteur.
- Le processus "transfert" gère le transfert de données sur l'anneau à jeton entre le processus émetteur et le processus récepteur.

Le schéma ci-après décrit le mécanisme d'un transfert de données et donne les primitives associées aux processus.



-Fig 3 - Mécanisme de transfert de données du service MAC -

3.1 - PROCESSUS EMETTEUR

A chaque adaptateur est associé un processus émetteur. Ce processus est défini par un automate ayant les états suivants :

Repos

Dans cet état, l'adaptateur attend l'arrivée d'une demande *MA_Req* du coupleur. Le processus d'émission passe de l'état "Repos" à l'état "A émettre" à la suite d'une requête *MA_Req* en provenance du coupleur associé à l'adaptateur.

A émettre

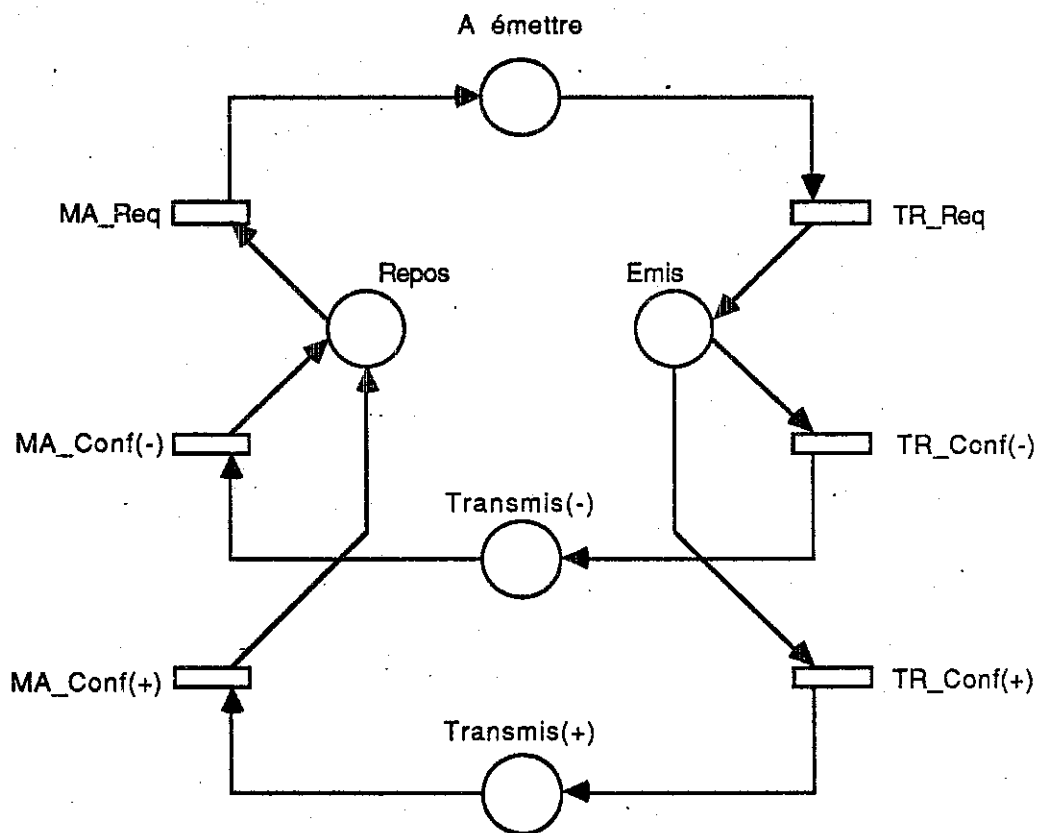
L'adaptateur possède dans son "buffer d'émission" un message prêt à transmettre sur l'anneau à jeton. Cette transmission est effectuée dès que l'adaptateur prend possession du jeton de l'anneau. Cette prise du jeton est réalisée par la requête *TR_Req*. Le processus passe à l'état "Emis".

Emis

L'adaptateur possède un message circulant sur l'anneau à jeton. Il attend que le message effectue le tour de l'anneau. Cet événement sera signalé par le "processus transfert" grâce à la primitive *TR_Conf*.

TR_Conf(+) indique que le message a été reçu par l'adaptateur récepteur". Le processus passe à l'état "Transmis(+)".

TR_Conf(-) indique qu'une perturbation a eu lieu durant le transfert du message. Le processus passe alors à l'état "Transmis(-)".



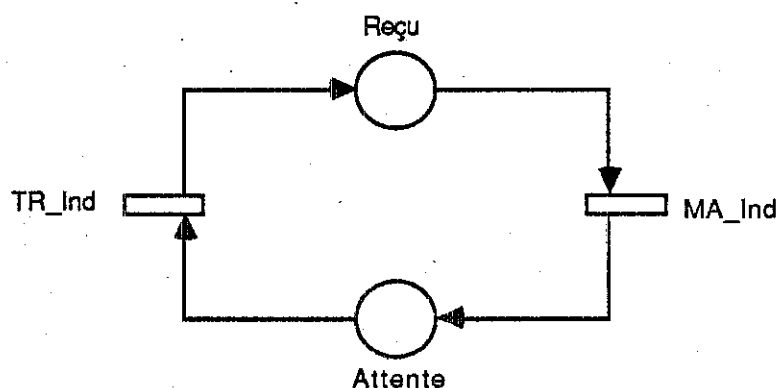
-Fig 3.1 - Processus émetteur-

Transmis(+) et Transmis(-)

L'adaptateur émet le résultat de la transmission du message . Il communique ce résultat au "coupleur émetteur" par la primitive *MA_Conf* en délivrant comme paramètre "+" ou "-". Le processus émetteur revient alors à l'état "Repos".

3.2 - PROCESSUS RECEPTEUR

Un processus de réception est associé à chaque adaptateur. Ce processus est défini par l' automate suivant :



-Fig 3.2 - Processus récepteur-

Attente

Le "buffer de réception" de l'adaptateur est vide. La réception d'un message par l'adaptateur sur l'anneau est signalée par la primitive *TR_Ind*. Le "processus de réception" passe alors à l'état "Reçu".

Reçu

Le "buffer de réception" de l'adaptateur est occupé. Dans cet état, l'adaptateur ne peut plus recevoir de message de l'anneau (cas de congestion potentielle de l'adaptateur). La réception du message par le coupleur en utilisant la primitive *MA_Ind* remet l'état du processus à "Attente".

Hypothèse: Le "buffer de réception" d'un adaptateur ne peut contenir qu'un seul message.

3.3 - PROCESSUS de TRANSFERT

Le processus de transfert gère le transfert de messages entre processus émetteur et récepteur associés à des adaptateurs connectés sur le même anneau. A chacun de ces anneaux est associé un processus de transfert décrivant les états d'un transfert sur l'anneau à jeton.

Dans un premier temps, nous donnons les principes du protocole à circulation de jeton et dans un second temps, l' automate décrivant un processus de transfert.

L' anneau à jeton

Le principe de ce protocole consiste à faire circuler séquentiellement un jeton entre les adaptateurs d' un anneau. Lorsqu' un adaptateur désire émettre un message, il prend possession du jeton, puis émet le message. Ce message passe d'adaptateur en adaptateur. L'adaptateur récepteur capture le message, il lui affecte du même coup l'indication "copié", puis le passe à son adaptateur successeur. Lorsque le message revient à l'adaptateur émetteur, le jeton est libéré.

Durant la circulation du message sur l'anneau, ce message peut être perdu ou dégradé. Un mécanisme interne non décrit ici permet de contrôler ces événements.

Processus de transfert

L' automate suivant décrit le fonctionnement d' un processus de transfert :

Jeton

Dans cet état, un jeton circule sur l'anneau. Lorsque un adaptateur prend possession du jeton par la primitive *TR_Req* , un message est inséré sur l'anneau. Le processus de transfert passe à l'état "Message".

Message

Un message vient d'être inséré sur l'anneau et circule vers l'adaptateur destinataire. Plusieurs scénarios peuvent alors se produire :

- Le message arrive sur l'adaptateur récepteur, puis il est copié. La primitive

TR_Ind est effectivement activée. Le processus de transfert passe à l'état Transfert(+).

- Dans les autres cas, nous considérons qu'il y a échec de l'émission . Nous représentons cet échec par l'événement *Pert_message* (perturbation du message). Le processus de transfert passe à l'état Transfert(-). Ces cas de perturbation sont les suivants :

- Le message arrive sur l'adaptateur récepteur mais ne peut être copié (congestion du récepteur, panne de la station, etc...).

- Une perturbation de l'anneau provoque la perte ou une dégradation du message (incohérence due à l'insertion ou au retrait d'une station, rupture d'une liaison, etc...).

Transfert(+)

Le message vient d'être copié par l'adaptateur récepteur. Il circule sur l'anneau vers l'adaptateur émetteur avec l'indication "copié". Nous pouvons avoir les scénarios suivants:

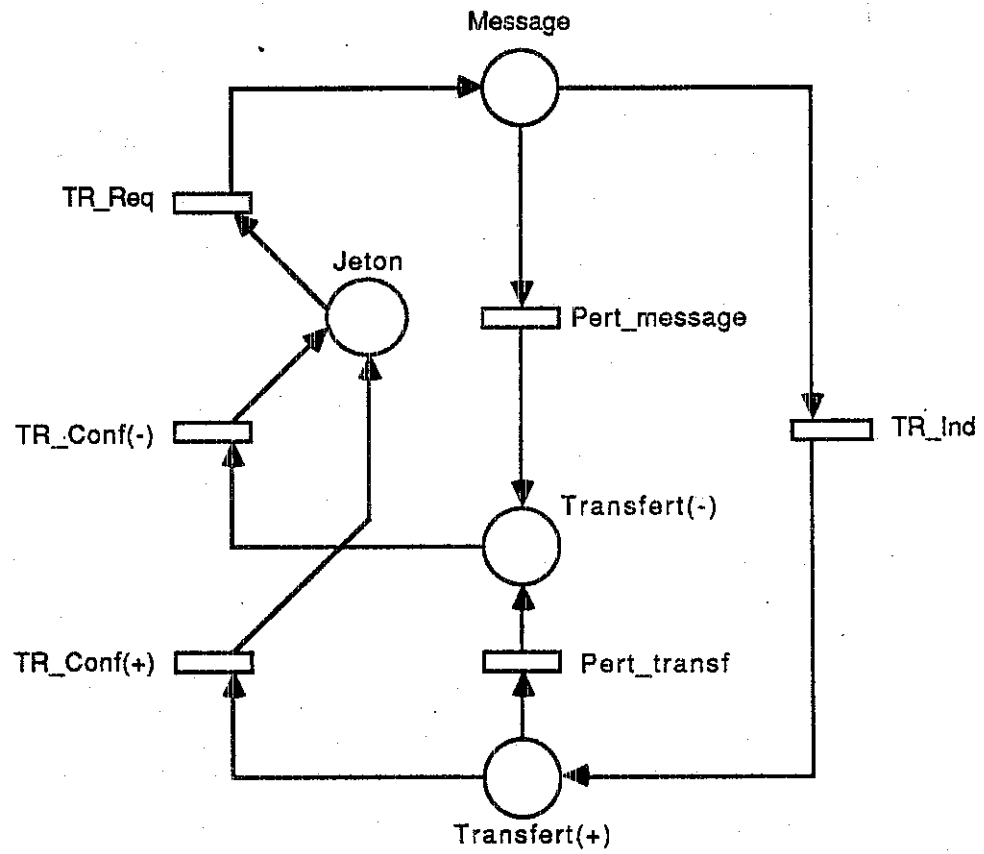
- Le message arrive sur l'adaptateur émetteur avec l'indication copié. L'adaptateur émetteur est prévenu. La primitive *Tr_conf(+)* est ainsi activée qui libère le jeton. Le processus de transfert revient alors à l'état "Jeton".

- Une perturbation de l'anneau provoque la perte ou une falsification du message. Nous représentons cette événement par *pert_transf*. Le processus de transfert passe à l'état "Transfert(-)".

Transfert(-)

Une perturbation est survenue au cours du transfert. L'adaptateur émetteur est prévenu de ce défaut de transmission par la primitive *TR_Conf(-)* . L'état du processus de transfert revient alors à "Jeton".

LeService de la couche MAC pour l'anneau à jeton



-Fig 3.3 - Processus de Transfert-

4.- MODELE du SERVICE

Le modèle du service que nous présentons est décrit en réseau de Petri Régulier. [Haddad 87]. Il est obtenu en fusionnant les transitions communes aux trois automates modélisant le comportement des processus.

4.1- LES CLASSES

Les classes utilisées dans la définition du réseau sont :

- <a> : Anneau $\in \{1, 2\}$
- <s> : Emetteur $\in \{1, \text{Nb_émetteur}\}$
- <d> : Récepteur $\in \{1, \text{Nb_récepteur}\}$
- <x> : Message $\in \{1, \text{Nb_données}\}$

Ces classes nous permettent de représenter les entités qui interviennent dans un transfert de données:

- <a,s> : L' adaptateur connecté sur l'anneau <a> associé au coupleur émetteur <s>;
- <a,d> : L' adaptateur connecté sur l'anneau <a> associé au coupleur récepteur <d>;
- <a,s,d,x>: Un message en provenance de l'adaptateur émetteur <a,s>, pour l'adaptateur récepteur <a,d> contenant la donnée <x>.

4.2 - LES PLACES

A chaque place d'un réseau régulier est associé un type représentant le format des marques que peut contenir la place. Ainsi il est possible de modéliser les états de nos processus émetteur, récepteur et de transfert.

Une marque <a,s,d,x> dans une place P indique:

>Si c'est une place des processus émetteurs, alors le processus émetteur <a,s> est dans l'état P et traite le message <a,s,d,x>.

>Si c'est une place des processus récepteurs, alors le processus récepteur <a,d> est dans l'état P et traite le message <a,s,d,x>.

>Si c'est une place des processus transferts, alors le processus transfert <a> est dans l'état P et traite le message <a,s,d,x>.

Nous retrouvons trois autres types de marque dans notre modèle:

>La marque <a,s> dans la place "Repos" indique le processus émetteur est en attente d'émission.

Le Service de la couche MAC pour l'anneau à jeton

>La marque <a> dans la place "Jeton" indique qu'il n'y a pas de message circulant dans l'anneau (le jeton est libre).

>La marque <a,d> dans la place "Attente" indique le processus récepteur est en attente de réception.

Repos (Anneau X Emetteur)

Une marque <a,s> dans Repos indique que le processus émetteur <a,s> est en attente d'émission.

A émettre, Emis, Transfert(-), Transfert(+)

(Anneau X Emetteur X Récepteur X Message)

Une marque <a,s,d,x> dans A émettre (resp. Emis, Transmis(-), Transmis(+)) indique que le processus émetteur <a,s> est en cours d'émission (resp. en attente de réception d'une confirmation, ou en cours de traitement d'une confirmation positive ou négative) d'un message <a,s,d,x> .

Jeton (Anneau)

Une marque <a> dans Jeton indique que le jeton de l'anneau <a> est libre.

Message, Transfert(-), Transfert(+)(Anneau X Emetteur X Récepteur X Message)

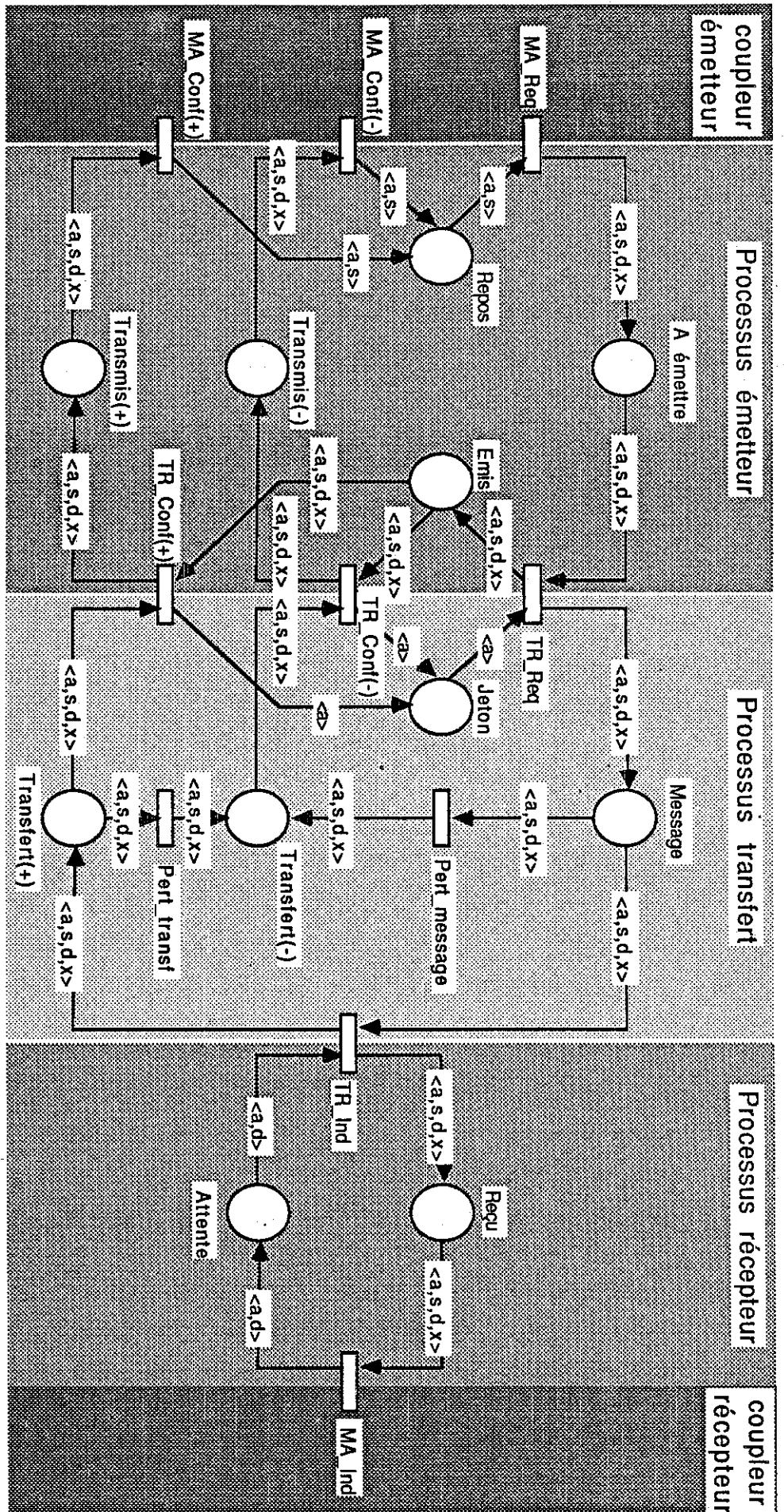
Une marque <a,s,d,x> dans Message (resp. Transfert(-), Transfert(+)) indique que le processus de transfert <a> transporte un message <a,s,d,x> (resp. transporte une confirmation négative ou positive).

Attente (Anneau X Récepteur)

Une marque <a,d> dans Attente indique que le processus récepteur <a,d> est inactif.

Reçu (Anneau X Emetteur X Récepteur X Message)

Une marque <a,s,d,x> dans Reçu indique que le processus récepteur <a,d> a reçu un message <a,s,d,x>.



-Fig 4 - Modale du service MAC -

4.3- LES TRANSITIONS

A chaque transition d'un réseau de Petri régulier est affecté un n-uplet $\langle a,s,d,x \rangle$ construit sur le produit des ensembles Anneau, Emetteur, Récepteur, Message. Une transition du modèle modélisé:

- Soit une primitive du processus émetteur de l'adaptateur $\langle a,s \rangle$;
- Soit une primitive du processus récepteur de l'adaptateur $\langle a,d \rangle$;
- Soit une primitive du processus transfert de l'anneau $\langle a \rangle$.

MA_Req[a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus émetteur $\langle a,s \rangle$ est dans l'état "Repos" (une marque $\langle a,s \rangle$ dans la place "Repos").

Par son franchissement, le processus émetteur $\langle a,s \rangle$ passe de l'état "Repos" à l'état "A émettre" pour le message $\langle a,s,d,x \rangle$ (la marque $\langle a,s \rangle$ de la place "Repos" est retirée, une marque $\langle a,s,d,x \rangle$ est ajoutée dans la place "A émettre").

TR_Req[a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus émetteur $\langle a,s \rangle$ est dans l'état "A émettre" pour le message $\langle a,s,d,x \rangle$ et le processus transfert $\langle a \rangle$ dans l'état "Jeton".

Par son franchissement, le processus émetteur $\langle a,s \rangle$ passe de l'état "A émettre" à l'état "Emis" pour le message $\langle a,s,d,x \rangle$ et le processus transfert $\langle a \rangle$ de l'état "Jeton" à l'état "Message" pour le message $\langle a,s,d,x \rangle$.

TR_Conf(-)[a,s,d,m], TR_Conf(+)[a,s,d,m]

La transition TR_Conf(-)[a,s,d,m] (resp. TR_Conf(+)[a,s,d,m]) est franchissable pour $\langle a,s,d,x \rangle$ si le processus émetteur $\langle a,s \rangle$ est dans l'état "Emis" pour le message $\langle a,s,d,x \rangle$ et le processus transfert $\langle a \rangle$ dans l'état "Transfert(-)" (resp "Transfert(+)).

Par son franchissement, le processus émetteur $\langle a,s \rangle$ passe de l'état "Emis" à l'état "Transmis(-)" (resp "Transmis(+))" pour le message $\langle a,s,d,x \rangle$ et le processus transfert $\langle a \rangle$ de l'état "Transfert(-)" (resp "Transfert(+))" à l'état "Message" pour le message $\langle a,s,d,x \rangle$.

Pert_message [a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus transfert $\langle a \rangle$ est dans l'état "Message" pour le message $\langle a,s,d,x \rangle$.

Par son franchissement, le processus transfert $\langle a \rangle$ passe de l'état "Message" à l'état "Transmis(-)".

Pert_transf [a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus transfert $\langle a \rangle$ est dans l'état "Transfert(+)" pour le message $\langle a,s,d,x \rangle$.

Par son franchissement, le processus transfert $\langle a \rangle$ passe de l'état " Transfert(+)" à l'état "Transmis(-)".

TR_Ind[a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus récepteur $\langle a,d \rangle$ est dans l'état "Attente" et le processus transfert $\langle a \rangle$ dans l'état "Message" pour le message $\langle a,s,d,x \rangle$.

Par son franchissement, le processus récepteur $\langle a,d \rangle$ passe de l'état "Attente" à l'état "Reçu" pour le message $\langle a,s,d,x \rangle$ et le processus transfert $\langle a \rangle$ de l'état "Message" pour le message $\langle a,s,d,x \rangle$ à l'état "Transfert(+)" pour le message $\langle a,s,d,x \rangle$.

MA_Ind[a,s,d,m]

Cette transition est franchissable pour $\langle a,s,d,x \rangle$ si le processus récepteur $\langle a,d \rangle$ est dans l'état "Reçu" pour le message $\langle a,s,d,x \rangle$.

Par son franchissement, le processus récepteur $\langle a,d \rangle$ passe de l'état "Reçu" pour le message $\langle a,s,d,x \rangle$ à l'état "Attente".

4.4- L'ETAT INITIAL

L'état initial M_0 représente l'état où tous les processus sont inactifs :

- Tous les processus émetteurs sont dans l'état Repos,
$$M_0(\text{Repos}) = \sum_{a,s} \langle a,s \rangle$$
- Tous les processus de transfert sont dans l'état Jeton,
$$M_0(\text{Jeton}) = \sum_a \langle a \rangle$$
- Tous les processus récepteurs sont dans l'état Attente,
$$M_0(\text{Attente}) = \sum_{a,d} \langle a,d \rangle$$
- Les autres places sont vides.

5.- VALIDATION du MODELE

5.1 - INTRODUCTION

Ce chapitre présente la validation de notre modèle du Service du niveau MAC. Cette validation va nous permettre d'illustrer de nombreuses techniques de preuves adaptées aux réseaux de Petri Réguliers.

Pour ce faire, nous allons, dans un premier temps, utiliser l'outil ARP afin d'extraire automatiquement du modèle ses invariants, qui sont autant de propriétés de base. Puis, nous démontrons que ce modèle possède une propriété majeure : il est vivant, c'est à dire qu'il est capable d'assurer son service sans blocage. Enfin, nous prouvons un ensemble de propriétés spécifiques de ce modèle concernant la sémantique des primitives d'interface avec le niveau Plasma.

PLAN

5.1 - INTRODUCTION

5.2 - CALCUL d'une BASE de FLOTS

5.2.1 - Description de l'algorithme

5.2.2 - Décomposition du réseau

5.2.3 - Calcul des flots pour les sous-réseaux

5.2.4 - Calcul d'une base de flots

5.3 - VIVACITE

5.3.1 - Introduction

5.3.2 - L'état Initial M_0 est un état d'accueil

5.3.3 - Le modèle est Quasi-vivant

5.3.4 - Conclusion

5.4 - PROPRIETES SPECIFIQUES DU MODELE

5.4.1 - Un adaptateur émetteur ne traite qu'un transfert à la fois

5.4.2 - Propriétés de la primitive confirmation

5.4.3 - Les messages ne sont pas déséquencés

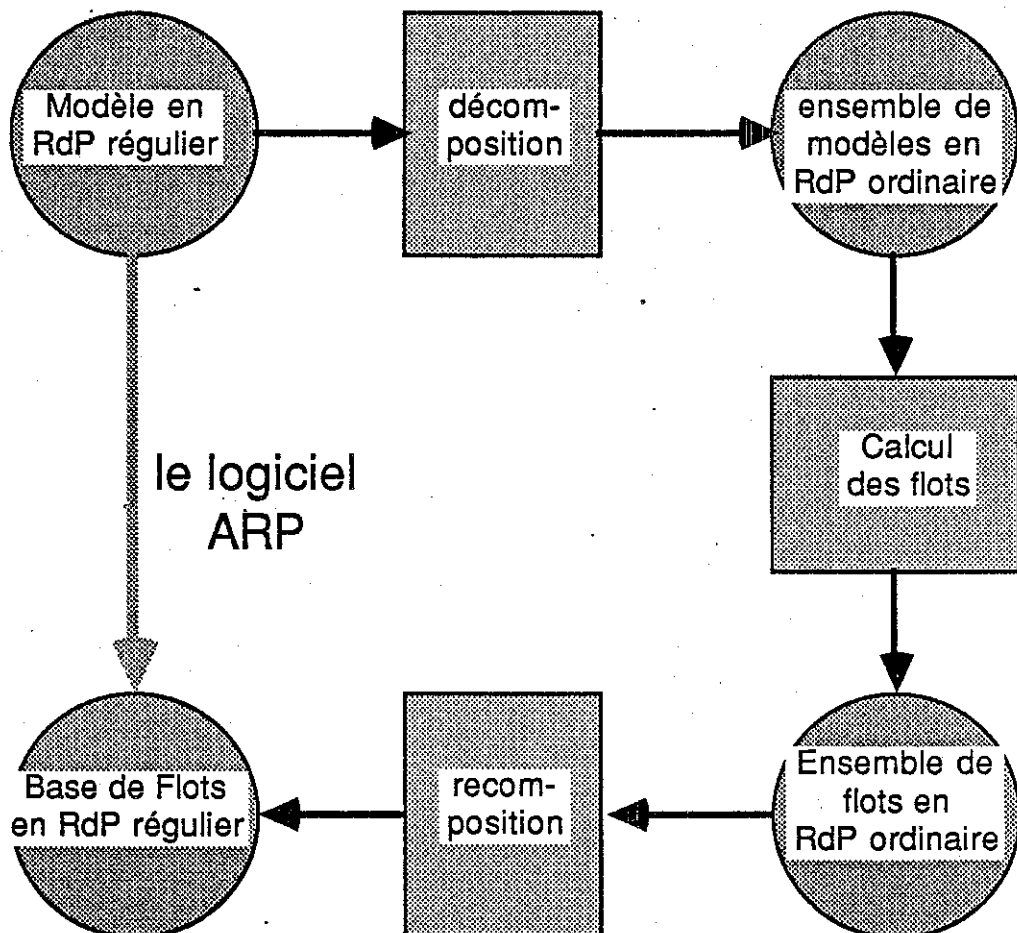
5.2 CALCUL d'une BASE de FLOT

5.2.1-Description de l'algorithme

Le calcul d'une base de flots d'un modèle décrit en Réseau Régulier s'établit en trois étapes (Figure 5.2.1) :

- **La décomposition** permet d'obtenir un ensemble de modèles décrit en RdP ordinaire équivalent au modèle initial;
- **Le calcul d'une base de flots** à partir de l'ensemble des modèles en RdP ordinaire;
- **La recombposition** permet d'interpréter la précédente base de flots sur le modèle initial exprimé en RdP régulier.

Le fondement théorique de cette démarche peut être trouvé dans la thèse du créateur des réseaux réguliers colorés [Haddad 86], document qui est muni de nombreux exemples. L'outil ARP, implémenté sous le système UNIX et intégré dans un atelier de validation graphique et automatisé, exécute l'algorithme permettant d'effectuer ces différentes étapes.



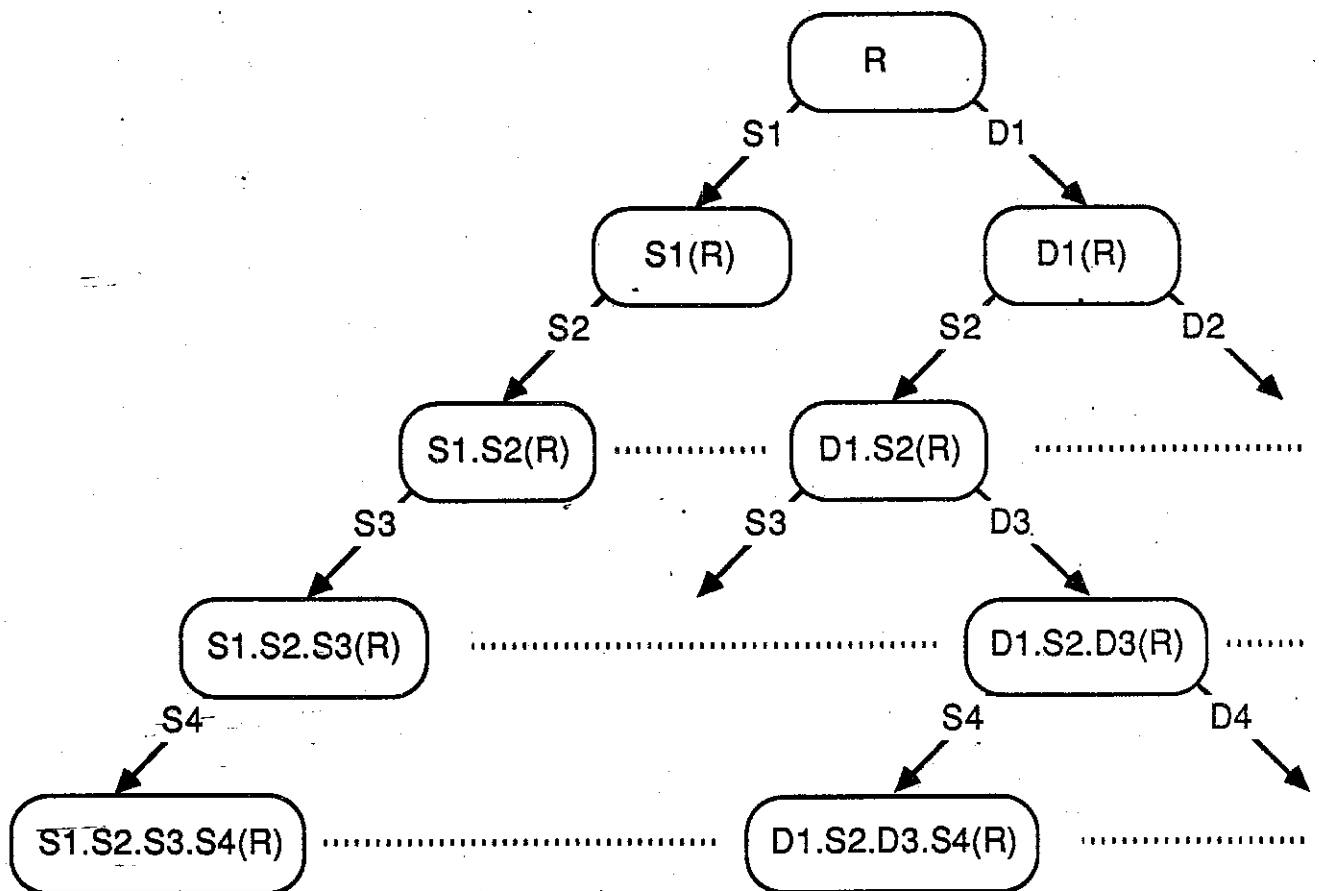
-Figure 5.2.1 - Etapes de calcul d'une base de flots-

5.2.2-Décomposition du réseau

On effectue la décomposition du modèle initial R, décrit en RdP régulier, par rapport à une classe C_i à l'aide des deux transformations S (homogénéisation) et D (différenciation). Ces deux transformations simplifient le modèle en supprimant les champs relatifs à la classe C_i des arcs du modèle, tout en conservant les informations essentielles à la constitution de la base de flots. Chacune de ces deux transformations définissent un nouveau modèle $S_i(R)$, et $D_i(R)$. On itère le processus pour une autre classe C_j sur les deux nouveaux modèles $S_i(R)$ et $D_i(R)$. On obtient, alors, quatre sous-modèles $S_j.S_i(R)$, $D_j.S_i(R)$, $S_j.D_i(R)$, $D_j.D_i(R)$.

En appliquant ce processus sur toutes les classes de couleurs constituant le modèle initial, on aboutit, finalement, à un ensemble de modèles sans classe de couleurs, donc décrit en RdP ordinaire.

Le modèle du service MAC est constitué d'uplets construits à partir de quatre classes élémentaires (C_1, C_2, C_3, C_4). Nous obtenons, donc, $2^4 (= 16)$ sous-modèles. Ces transformations se résument à l'aide de l'arbre de transformation suivant (Figure 5.2.2) :



- Figure 5.2.2 - Arbre de transformation d'un réseau régulier -

Chacune des feuilles de l'arbre est un réseau de Petri ordinaire.

On note $DI(R)$, la transformation "D" (différentielle) suivant la classe "Ci" appliquée au modèle "R".

On note $SI(R)$, la transformation "S" (homogénéisation) suivant la classe "Ci" appliquée au modèle "R".

5.2.3 - Calcul des flots pour les sous-réseaux

La décomposition du "modèle du service MAC définit seize réseaux de Petri ordinaires. Ils possèdent les flots suivants :

Flots des réseaux $S1.S2.S3.S4(R)$ et $D1.S2.S3.S4(R)$:

Repos+A_émettre+Emis+Transmis(+)+Transmis(-)
Attente+Reçu
Emis+Jeton
-Emis+Message+Transfert(+)+Transfert(-)

Flots des réseaux $S1.D2.S3.S4(R)$ et $D1.D2.S3.S4(R)$:

Repos+A_émettre+Emis+Transmis(+)+Transmis(-)
-Emis+Message+Transfert(+)+Transfert(-)

Flots des réseaux $S1.S2.D3.S4(R)$ et $D1.S2.D3.S4(R)$:

Attente+Reçu
-Emis+Message+Transfert(+)+Transfert(-)

Flots des réseaux

$S1.S2.S3.D4(R)$, $D1.S2.S3.D4(R)$,
 $S1.S2.S3.D4(R)$, $S1.S2.D3.D4(R)$,
 $S1.D2.S3.D4(R)$, $S1.D2.D3.D4(R)$,
 $D1.S2.S3.D4(R)$, $D1.S2.D3.D4(R)$ et
 $D1.D2.S3.D4(R)$, $D1.D2.D3.D4(R)$:

-Emis+Message+Transfert(+)+Transfert(-)

5.2.4 - Calcul d'une base de flots

On peut remarquer que les différents sous-réseaux possèdent des flots identiques. La recomposition consistera à regrouper l'ensemble des sous-réseaux ayant le même

flot.

En fait, nous n'avons que quatre flots. Nous pouvons décrire les sous-réseaux auxquels ils appartiennent de manière factorisée, ce qui rend leur interprétation plus aisée.

Le Flot 1 obtenu à partir des sous-réseaux (S1,D1).(S2,D2).S3.S4(R) décrit le comportement du processus émetteur (figure 5.2.3) :

$\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}$

$$\text{Repos } [a,s] + \sum_{d,x} A_émettre [a,s,d,x] + \sum_{d,m} \text{Emis } [a,s,d,x] + \sum_{d,x} \text{Transmis}(+) [a,s,d,x] + \sum_{d,x} \text{Transmis}(-) [a,s,d,x] = 1$$

Interprétation : Tout processus Emetteur défini par l'uplet $[a,s]$ a : soit une marque de type $[a,s]$ dans la place Repos, soit une marque de type $[a,s,d,x]$ dans une des places A_émettre, Emis, Transmis(+), Transmis(-).

Corollaire: Tout processus Emetteur soit n'a rien à émettre (place Repos), soit attend le jeton pour émettre (place A_émettre), soit est en cours d'émission (place Emis), soit a réussi son émission (place Transfert(+), soit ne l'a pas réussie (place Transfert(-)).

Le Flot 2 obtenu à partir des sous-réseaux (S1,D1).S2.(S3,D3).S4(R) décrit le comportement du processus récepteur (figure 5.2.4) :

$\forall a \in \text{Anneau}, \forall d \in \text{Destinataire}$

$$\text{Attente } [a,d] + \sum_{s,x} \text{Reçu } [a,s,d,x] = 1$$

Interprétation : Tout processus Récepteur défini par l'uplet $[a,d]$ a : soit une marque de type $[a,d]$ dans la place Attente, soit une marque de type $[a,s,d,x]$ dans la place Reçu.

Corollaire : Tout processus Récepteur est soit en attente de réception de message (place Attente), soit traite la réception d'un message (place Reçu).

Le Flot 3 obtenu à partir des sous-réseaux (S1,D1).S2.S3.S4(R) décrit le comportement du processus associé à l'anneau (figure 5.2.5) :

$\forall a \in \text{Anneau}$

$$\sum_{s,d,x} \text{Emis}[a,s,d,x] + \text{Jeton } [a] = 1$$

Interprétation : Tout processus Anneau définit par l'uplet $[a]$ a : soit une marque de type $[a,s,d,m]$ dans la place Emis, soit une marque de même type $[a]$ dans la place-Jeton.

Corollaire: Chaque anneau véhicule soit une trame comportant le jeton (place Jeton), soit une trame comportant un message (place Emis).

Le Service de la couche MAC pour l'anneau à jeton

Le Flot 4 obtenu à partir des sous-réseaux $(S1,D1).(S2,D2).(S3,D3).(S4,D4)(R)$ décrit le comportement du processus transfert (figure 5.2.6) :

$\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}, \forall d \in \text{Destinataire}, \forall x \in \text{Données}$

$\text{Message}[a,s,d,x] + \text{Transfert}(+)[a,s,d,x] + \text{Transfert}(-)[a,s,d,x] = \text{Emis}[a,s,d,x]$

Interprétation : Pour tout message défini par l'uplet $[a,s,d,x]$, si une marque de type $[a,s,d,x]$ est dans la place Emis, alors il existe une marque de même type $[a,s,d,x]$ dans une des places Message, Transfert(+), Transfert(-).

Corollaire: Quand le processus Emetteur a émis un message (place Emis), le message est soit véhiculé dans la partie de l'anneau émetteur/destinataire (place Message), soit dans la partie de l'anneau destinataire/émetteur et le message a été copié (place Transfert(+), soit dans la partie de l'anneau destinataire/émetteur et le message n'a pas été copié ou est illicite (place Transfert(-)).

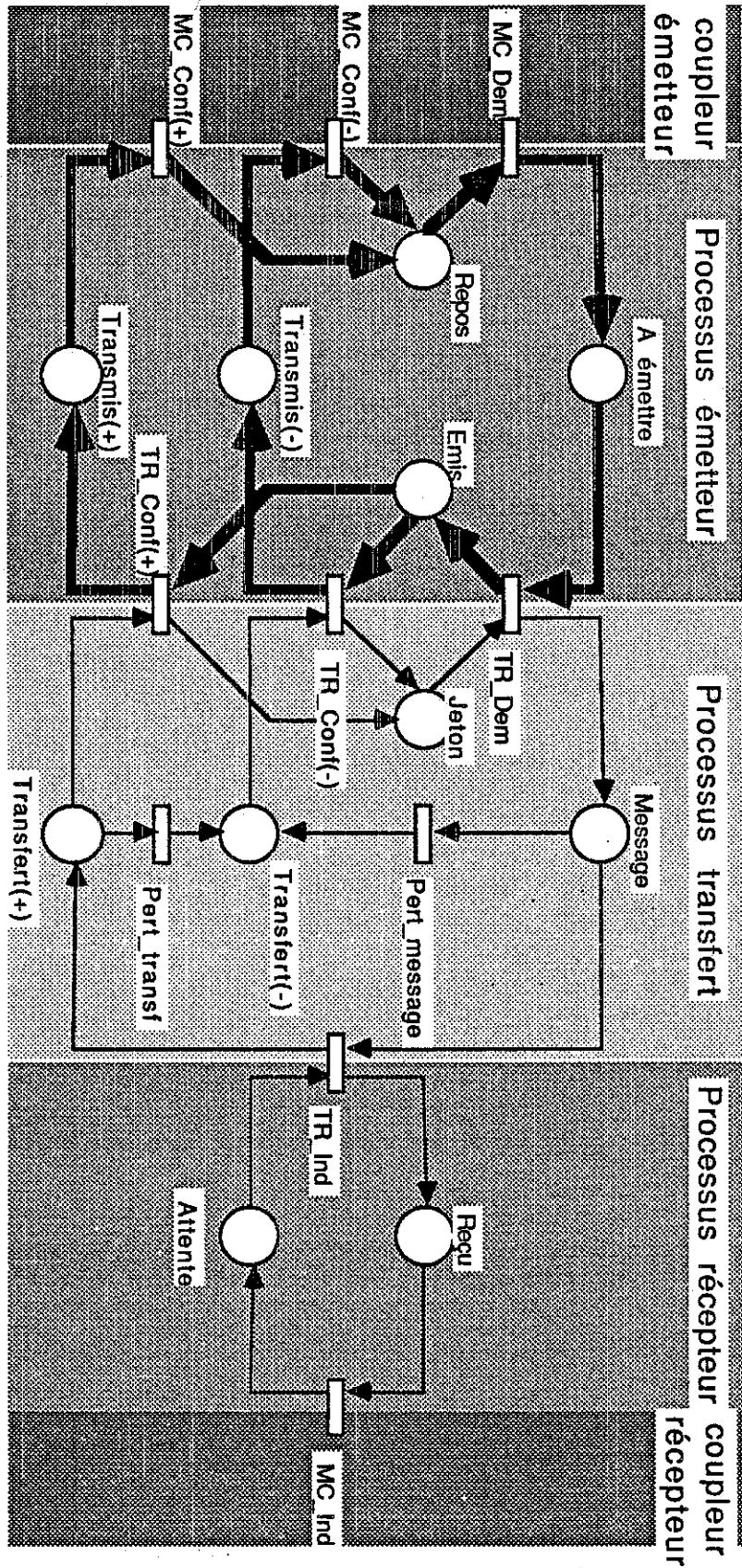
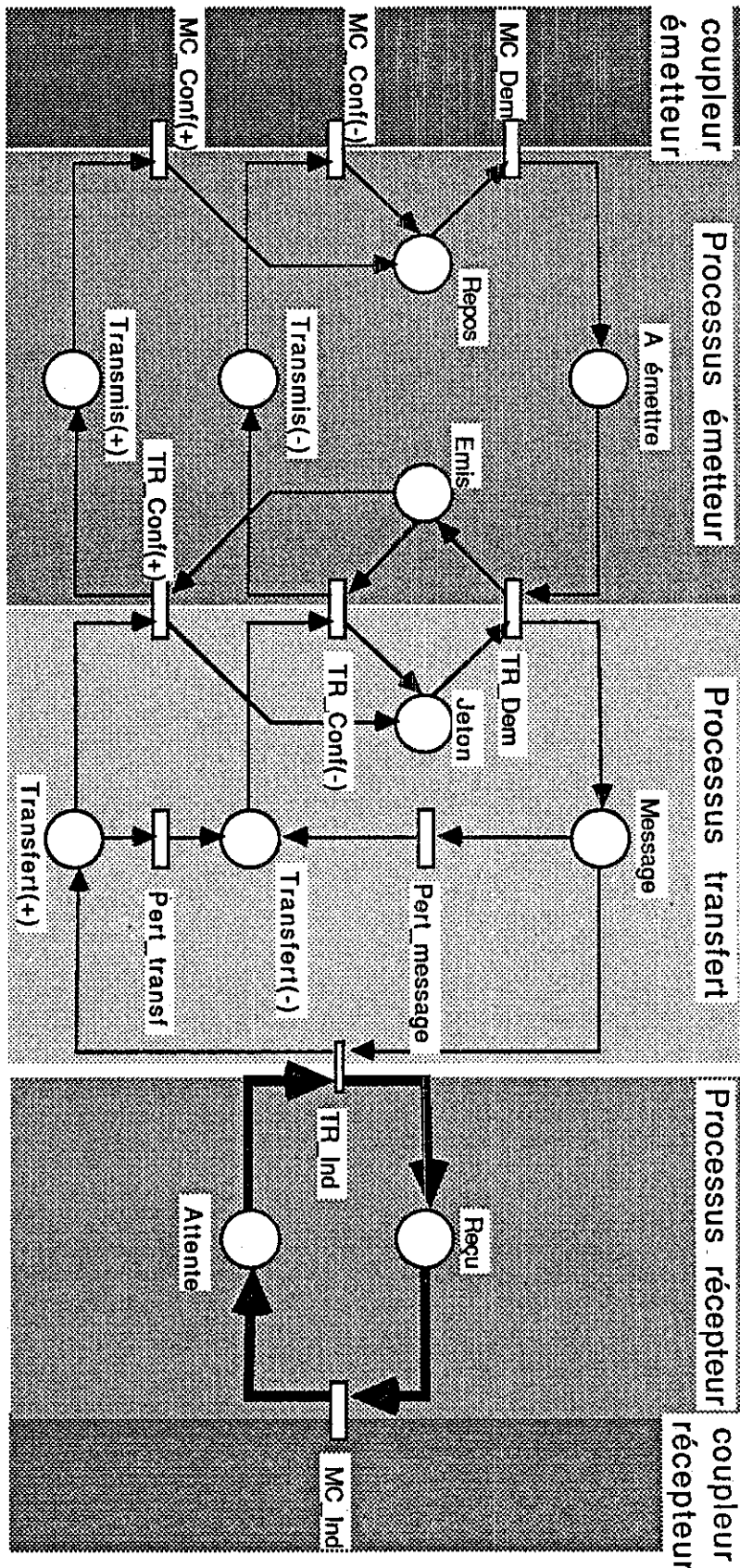
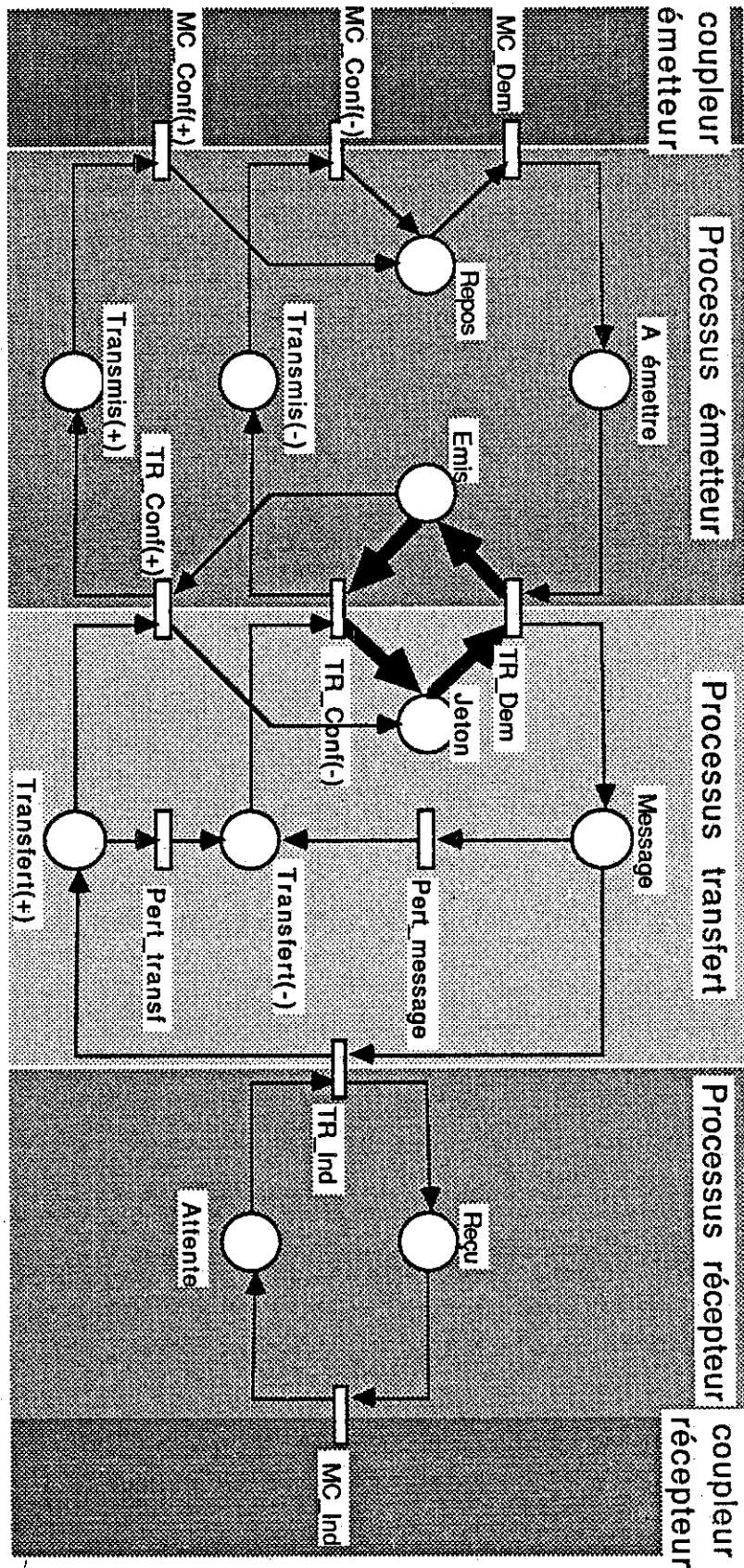


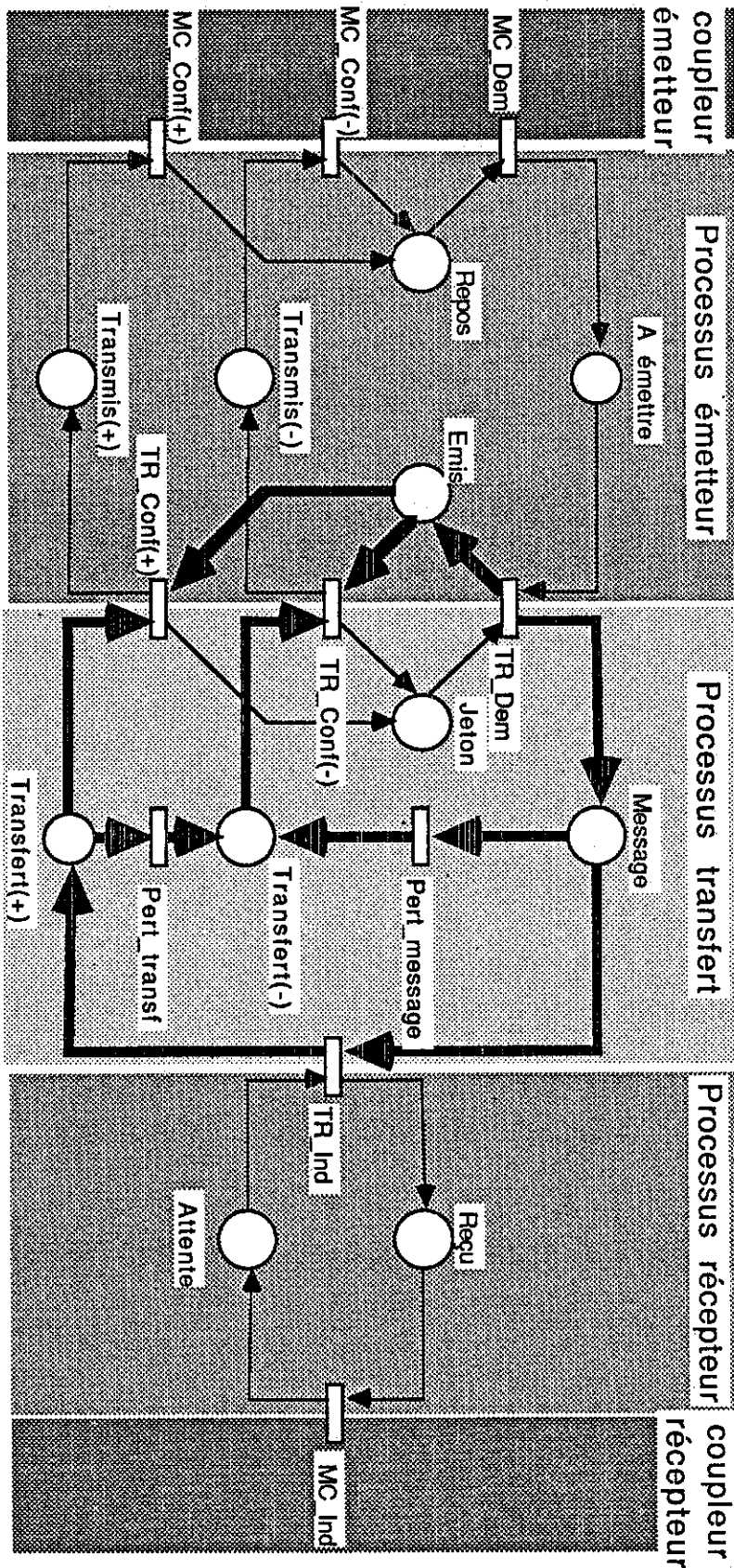
Figure 5.2.3- Flot associé au processus émetteur



- Figure 5.2.4 - Flot associé au processus récepteur-



- Figure .5.2.5 - Flot associé au jeton -



- Figure 5.2.6 - Flot associé au processus transfert -

5.3- VIVACITE

5.3.1- Introduction

La vivacité d'un réseau de Petri est une propriété qui permet d'obtenir des résultats immédiats sur le comportement du réseau, en particulier son non blocage. Pour qu'un réseau soit vivant, il faut et il suffit que pour tout état accessible M et pour toute transition t, il soit toujours possible de franchir à partir de l'état M, la transition t donnée du modèle en prolongeant la séquence de franchissement.

Nous allons montrer en deux étapes que "le modèle du service MAC" est vivant :

- L'état initial Mo est un état d'accueil.
- Le modèle est quasi-vivant (A partir de l'état initial Mo, il existe pour chaque transition t, une séquence de franchissement contenant la transition t).

5.3.2- L'état initial Mo est un état d'accueil

Un état d'accueil est un état qui est accessible à partir de n'importe quel état accessible du réseau. Pour que Mo soit un état d'accueil, il faut et il suffit que pour tout état $M \in A(R)$ (Ensemble des états accessibles), il existe une séquence f franchissable à partir de M qui transforme l'état M en l'état initial Mo. Autrement dit:

$$\forall M \in A(R), \exists f \in T^* / M \xrightarrow{f} Mo$$

Pour un état accessible quelconque, la construction de la séquence f permettant d'aboutir à l'état initial Mo s'établit en construisant les trois séquences f1f2f3:

f1> Mettre tous les processus récepteurs à l'état "Attente". En d'autres termes, nous construisons une séquence f1 transformant l'état quelconque M en un état M1. Le marquage des places "Reçu" et "Attente" pour le marquage M1 sont:

$$M1[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M1[\text{Reçu}] = 0$$

f2> **Mettre tous les processus transferts à l'état "Jeton"**. Nous construisons donc une séquence f2 transformant l'état M1 en un état M2. L'état M2 est caractérisé par:

$$M2[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M2[\text{Reçu}] = 0$$

$$M2[\text{Jeton}] = \sum_a \langle a \rangle$$

$$M2[\text{Message}] = M2[\text{Transfert}(-)] = M2[\text{Transfert}(+)] = 0$$

f3> **Mettre tous les processus émetteurs dans l'état "Repos"**. Nous construisons une séquence f3 transformant l'état M2 en l'état initial Mo.

$$Mo[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

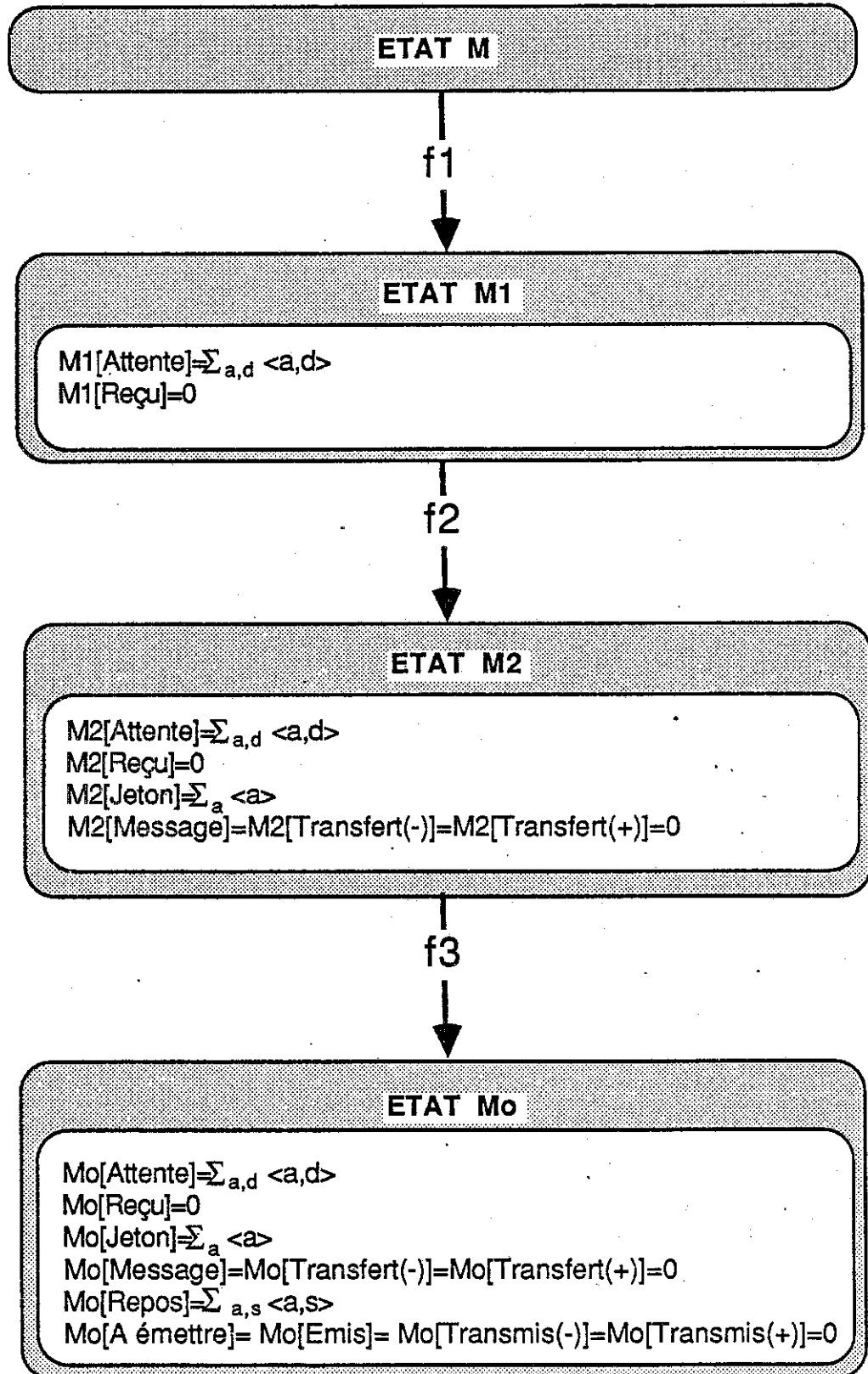
$$Mo[\text{Reçu}] = 0$$

$$Mo[\text{Jeton}] = \sum_a \langle a \rangle$$

$$Mo[\text{Message}] = Mo[\text{Transfert}(-)] = Mo[\text{Transfert}(+)] = 0$$

$$Mo[\text{Repos}] = \sum_{a,s} \langle a,s \rangle$$

$$Mo[\text{A émettre}] = Mo[\text{Transmis}(-)] = Mo[\text{Transmis}(+)] = 0$$



-Figure 5.3.2-Construction d'une séquence transformant un état M en Mo-

Nous allons prouver maintenant l'existence des 3 séquences f1, f2 et f3.

5.3.2.1 - f1> Mettre les processus récepteurs à l'état "Attente"

La transition MC_Ind est franchissable dès que la place "Reçu" est marquée. Donc pour tout état accessible M, il est possible de franchir la transition MC_Ind jusqu'à obtenir un état M1 où la place "Reçu" est vide. La séquence f1 est donc:

D'après le flot 2: $\forall M \in A(R), \forall a \in \text{Anneau}, \forall d \in \text{Récepteur}$:

$$M(\text{Attente}[a,d]) + \sum_{s,x} M(\text{Reçu}[a,s,d,x]) = 1,$$

nous obtenons alors pour l'état M1, le résultat:

$$\forall a \in \text{Anneau}, \forall d \in \text{Récepteur} \quad M1(\text{Attente}[a,d]) = 1$$

ceci prouve bien que:

$$M1[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M1[\text{Reçu}] = 0$$

5.3.2.2 - f2> Mettre les processus transferts à l'état "Jeton"

En partant de état M1, nous vidons les places "Message", "Transfert(-)" et "Transfert(+)" en franchissant des transitions du modèle. Chacune de ces séquences ne modifie pas pour autant les places "Reçu" et "Attente". Ainsi nous aurons construit une séquence qui transformera l'état M1 en un état M2.

Vidons la place "Message"

La transition $Pert_message$ est franchissable dès que la place "Message" est marquée. En franchissant la transition $Pert_message$ autant de fois que possible, nous obtenons à partir de l'état M1, un état M1a où la place "Message" est vide. La transition $Pert_message$ ne change que les marquages des places "Message" et "Transfert(-)". Donc l'état M1a a les caractéristiques suivantes:

$$M1a[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M1a[\text{Reçu}] = 0$$

$$M1a[\text{Message}] = 0.$$

Vidons la place "Transmis(+)"

La transition $Pert_transf$ est franchissable dès que la place "Transfert(+)" est marquée. En franchissant la transition $Pert_transfert$ autant de fois que possible, nous obtenons à partir de l'état $M1a$, un état $M1b$ où la place "Transfert(+)" est vide. La transition $Pert_transf$ ne change que les marquage des places "Transfert(+)" et "Transfert(-)". Donc l'état $M1b$ a les caractéristiques suivantes:

$$M1b[Attente] = \sum_{a,d} \langle a,d \rangle$$

$$M1b[Reçu] = 0$$

$$M1b[Message] = M1b(Transfert(+)) = 0.$$

Vidons la place "Transmis(-)"

La transition $TR_Conf(-)$ est franchissable dès que les places "Emis" et "Transfert(-)" contiennent des marques $\langle a,s,d,x \rangle$ identiques.

D'après le flot 4:

$$\forall M \in A(R), \forall a \in \text{Anneau}, \forall s \in \text{Emetteur}, \forall d \in \text{Récepteur}, \forall x \in \text{Données},$$

$$M(\text{Emis}[a,s,d,m]) =$$

$$M(\text{Message}[a,s,d,x]) + M(\text{Transfert}(-)[a,s,d,x]) + M(\text{Transfert}(+)[a,s,d,x])$$

Nous obtenons pour l'état $M1b$:

$$M1b(\text{Emis}[a,s,d,x]) = M1b(\text{Transfert}(-)[a,s,d,x])$$

Il nous est donc possible de franchir la transition $TR_Conf(-)$ autant de fois qu'il y a de marques dans "Transfert(-)". Nous obtenons ainsi un état $M2$.

L'état $M2$ est caractérisé par:

$$M2[Attente] = \sum_{a,d} \langle a,d \rangle$$

$$M2[Reçu] = 0$$

$$M2[Message] = M2[Transfert(+)] = M2[Transfert(-)] = 0.$$

D'après le flot 4: $M2(\text{Emis}[a,s,d,x])=0$,

D'après le flot 2: $\forall M \in A(R), \forall a \in \text{Anneau}$

$$M(\text{Jeton}[a]) + \sum_{s,d,x} M(\text{Emis}[a,s,d,x]) = 1,$$

d'où $M2(\text{Jeton}[a])=1$.

Ceci prouve que le marquage de la place "Jeton" est le suivant:

$$M2[\text{Jeton}] = \sum_a \langle a \rangle$$

Conclusion

Nous avons construit une séquence $f2$ qui transforme $M1$ en $M2$. $M2$ vérifie bien les caractéristiques prévues:

$$M2[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M2[\text{Reçu}] = 0$$

$$M2[\text{Jeton}] = \sum_a \langle a \rangle$$

$$M2[\text{Message}] = M2[\text{Transfert}(+)] = M2[\text{Transfert}(-)] = 0.$$

5.3.2.3 - $f3$ > Mettre les processus émetteurs à l'état "Repos"

En partant de état $M2$, nous vidons les places "A émettre", "Transmis(-)" et "Transmis(+)" en franchissant des transitions du modèle. Ces séquences ne modifient pas pour autant les places liées aux processus transferts et récepteurs. Ainsi nous aurons construit une séquence $f3$ qui transformera l'état $M2$ en l'état initial $M0$.

Vidons la place "A émettre"

La séquence de transition $TR_Dem, Pert_message, TR_Conf(-)$ est franchissable dès que la place "Jeton" contient une marque $\langle a \rangle$ et la place "A émettre" une marque $\langle a,s,d,m \rangle$. En itérant cette séquence pour chaque marque de la place "A émettre" de l'état $M2$, nous obtenons un état $M2a$ pour lequel la place "A émettre" est vide.

Les caractéristiques de $M2a$ sont les suivantes

$$M2a[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M2a[\text{Reçu}] = 0$$

$$M2a[\text{Jeton}] = \sum_a \langle a \rangle$$

$$M2a[\text{Message}] = M2a[\text{Transfert}(+)] = M2a[\text{Transfert}(-)] = 0$$

$$M2a[\text{A émettre}] = 0.$$

Vidons la place Transmis(-)

La transition $MC_Conf(-)$ est franchissable dès que la place "Transmis(-)" possède une marque. En itérant le franchissement de la transition $MC_Conf(-)$, nous obtenons à partir de l'état $M2a$ un état $M2b$ pour lequel la place "Transmis(-)" est vide.

Les caractéristiques de $M2b$ sont les suivantes

$$M2a[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M2a[\text{Reçu}] = 0$$

$$M2a[\text{Jeton}] = \sum_a \langle a \rangle$$

$$M2a[\text{Message}] = M2a[\text{Transfert}(+)] = M2a[\text{Transfert}(-)] = 0$$

$$M2a[\text{A émettre}] = M2b[\text{Transmis}(-)] = 0.$$

Vidons la place Transmis(+)

La transition $MC_Conf(+)$ est franchissable dès que la place "Transmis(+)" possède une marque. En itérant le franchissement de la transition $MC_Conf(+)$, nous obtenons à partir de l'état $M2b$ un état $M3$ pour lequel la place "Transmis(+)" est vide.

Les caractéristiques de $M3$ sont les suivantes

$$M3[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$M3[\text{Reçu}] = 0$$

$$M3[\text{Jeton}] = \sum_a \langle a \rangle$$

$$M3[\text{Message}] = M3[\text{Transfert}(+)] = M3[\text{Transfert}(-)] = 0$$

$$M3[\text{A émettre}] = M3[\text{Transmis}(-)] = M3[\text{Transmis}(+)] = 0.$$

D'après le flot 1:

$$\forall M \in A(R), \forall a \in \text{Anneau}, \forall s \in \text{Emetteur},$$

$$M(\text{Repos}[a,s]) + \sum_{s,x} M(A_émettre[a,s,d,x]) + \sum_{s,x} M(\text{Emis}[a,s,d,x])$$

$$\sum_{s,x} M(\text{Transmis}(-)[a,s,d,x]) + \sum_{s,m} M(\text{Transmis}(+)[a,s,d,x]) = 1$$

donc pour M3, nous obtenons: $M3(\text{Repos}[a,s]) = 1$.

Le marquage de la place "Repos" est donc:

$$M3[\text{Repos}] = \sum_{a,d} \langle a,d \rangle$$

Conclusion

L'état M3 ainsi obtenu est en réalité l'état initial Mo. Nous avons construit une séquence f3 qui transforme l'état M2 en l'état Mo.

$$Mo[\text{Attente}] = \sum_{a,d} \langle a,d \rangle$$

$$Mo[\text{Reçu}] = 0$$

$$Mo[\text{Jeton}] = \sum_a \langle a \rangle$$

$$Mo[\text{Message}] = Mo[\text{Transfert}(+)] = Mo[\text{Transfert}(-)] = 0$$

$$Mo[\text{Jeton}] = \sum_{a,d} \langle a,d \rangle$$

$$Mo[A \text{ émettre}] = Mo[\text{Transmis}(-)] = Mo[\text{Transmis}(+)] = 0.$$

Ainsi à partir d'un état accessible quelconque M, nous avons construit une séquence f1f2f3 qui transforme l'état M en l'état initial Mo. Mo est donc un état d'accueil.

5.3.3 - Le modèle est Quasi-vivant

Pour que notre modèle soit quasi vivant, il faut et il suffit que pour toute transition t et tout n-uplet $\langle a,s,d,x \rangle$, il existe une séquence franchissable à partir de l'état initial Mo contenant $t[\langle a,s,d,x \rangle]$.

Pour une valeur quelconque de $\langle a,s,d,m \rangle$, nous développons un graphe des marquages accessibles où seules les transitions correspondant à cette couleur sont franchies. Nous obtenons le graphe de la figure 5.3.3.

Le Service de la couche MAC pour l'anneau à jeton

Dans ce graphe, un état du modèle est représenté par:

>l'état du processus émetteur <a,s>

>l'état du processus de transfert <a>

>l'état du processus récepteur <a,d>

Il est facile de vérifier que toutes les transitions du modèle sont franchies. Nous avons ainsi prouvé qu'à partir de l'état initial, il est possible de construire une séquence permettant le franchissement de toute transition $t[a,s,d,x]$. Le modèle est donc quasi-vivant.

5.3.4-Conclusion

Pour tout état accessible M , il existe une séquence f transformant l'état M en l'état M_0 (M_0 est un état d'accueil). De plus, pour toute transition t et tout n -uplet $\langle a,s,d,x \rangle$, il existe une séquence f' franchissable à partir de M_0 contenant la transition $t[a,s,d,x]$. Nous en concluons que ff' est franchissable à partir de M et que de plus ff' contient la transition $t[a,s,d,x]$. Notre modèle est donc vivant.

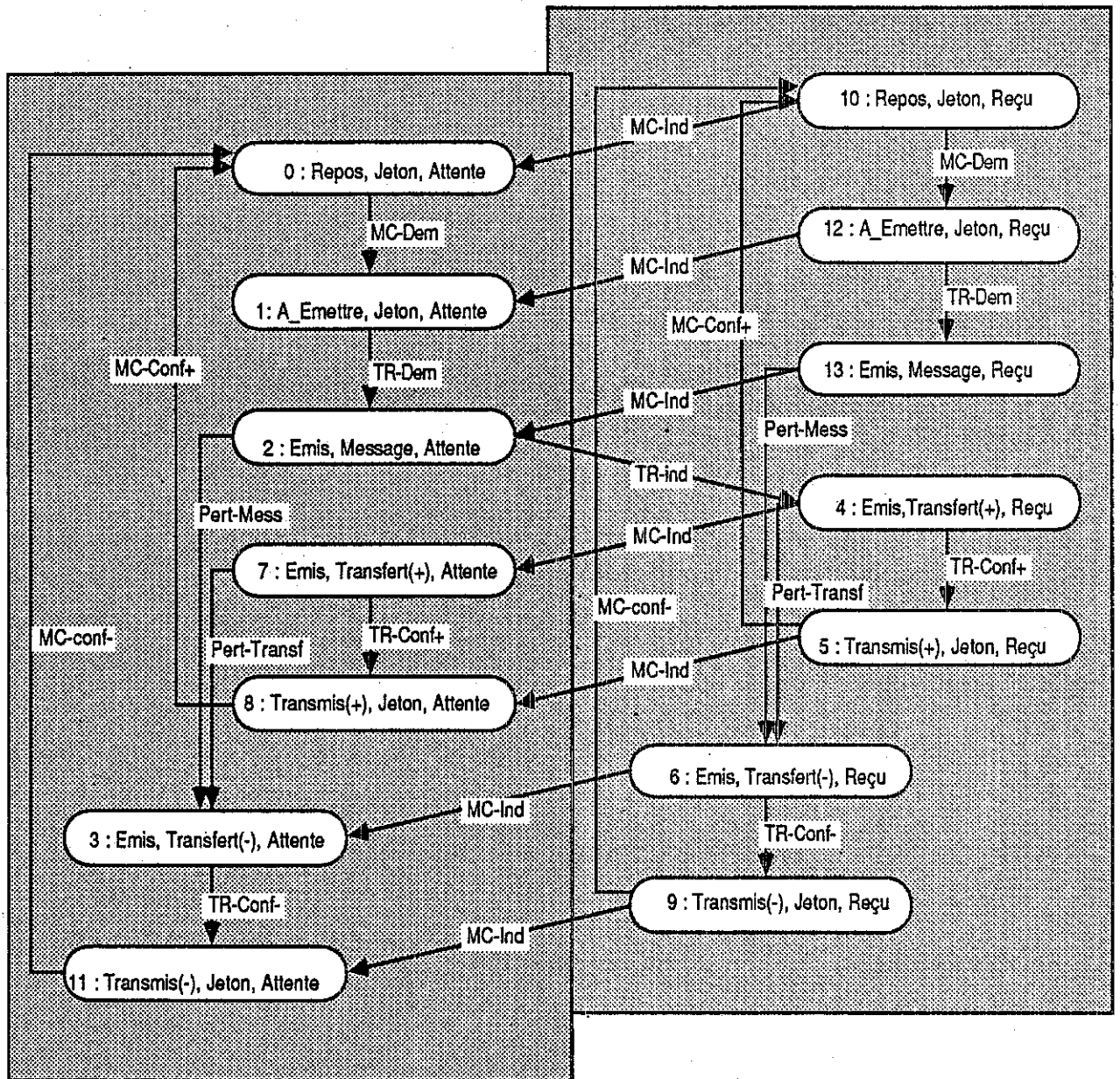
En appliquant cette propriété au "modèle du service MAC" pour les transitions modélisant les primitives d'interface, nous obtenons les résultats suivants:

MC_Dem : Il est toujours possible d'envoyer un message $\langle s,d,x \rangle$ sur l'anneau $\langle a \rangle$ à partir de l'émetteur $\langle a,s \rangle$ vers le récepteur $\langle a,d \rangle$.

MC_Ind : Il est toujours possible de recevoir un message $\langle s,d,x \rangle$ sur l'anneau $\langle a \rangle$ du récepteur $\langle a,d \rangle$ venant de l'émetteur $\langle a,s \rangle$.

MC_Conf(+) (*resp MC_Conf(-)*) : Il est toujours possible de recevoir une confirmation positive (respectivement négative) d'un message $\langle s,d,x \rangle$ sur l'anneau $\langle a \rangle$ à partir de l'émetteur $\langle a,s \rangle$ du récepteur $\langle a,d \rangle$.

Ces propriétés montrent que notre modèle est toujours apte à remplir son service.



-Figure 5.3.3 - Graphe de marquage réduit pour une couleur <a,s,d,x>-

5.4 - PROPRIETES SPECIFIQUES DU MODELE

Pour valider complètement notre modèle, il convient de vérifier des propriétés concernant la sémantique des primitives d'interface avec le niveau supérieur :

<P1> Un adaptateur émetteur ne traite qu'un transfert à la fois. Le transfert commence par une demande MA_Req et finit par une confirmation positive MA_Conf(+) ou négative MA_Conf(-).

<P2> Une confirmation(+) indique que l'adaptateur récepteur a reçu le message,

<P3> Une confirmation(-) indique une perturbation dans la transmission,

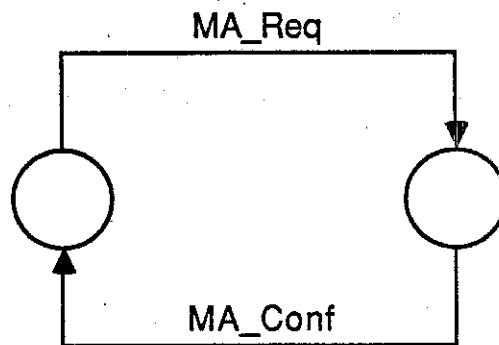
<P4> Les messages ne sont pas déséquencés.

5.4.1 - Un adaptateur émetteur ne traite qu'un transfert à la fois

Définition:

Un adaptateur émetteur ne traite qu'un transfert à la fois. Le transfert commence par une demande MA_Req et finit par une confirmation positive MA_Conf(+) ou négative MA_Conf(-).

La propriété P1 définit l'enchaînement des franchissements des transitions MA_Req, MA_Conf(+) et MA_Conf(-) au niveau de chaque processus émetteur <a,s>. Il est constitué d'une suite de séquence MA_Req, MA_Conf.



-Figure 5.4.1.1-Enchaînement des transitions MA_Req et MA_Conf-

Preuve

Nous considérons un processus émetteur <a₀,s₀> quelconque.

D'après le flot 1: $\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}$

$$\text{Repos}[a,s] + \sum_{d,x} A_{\text{émetteur}}[a,s,d,x] + \sum_{d,m} \text{Emis}[a,s,d,x] + \sum_{d,x} \text{Transmis}(+)[a,s,d,x] + \sum_{d,x} \text{Transmis}(-)[a,s,d,x] = 1$$

Nous pouvons donc partitionner l'ensemble des marquages accessibles par les parties:

$$R = \{ M \in A(R) / M[\text{Repos}(a_0, s_0)] = 1 \}$$

$$E_{d,x} = \{ M \in A(R) / M[A \text{ émettre}(a_0, s_0, d, x)] + M[\text{Emis}(a_0, s_0, d, x)] + \\ + M[\text{Transmis}(-)(a_0, s_0, d, x)] + M[\text{Transmis}(-)(a_0, s_0, d, x)] = 1 \}$$

Nous considérons maintenant les transitions

- MA_Req[a₀,s₀,d,x],
- MA_Conf(-)[a₀,s₀,d,x],
- MA_Conf(+)[a₀,s₀,d,x].

Nous constatons que ces transitions sont les seules transformant un état d'une partie en un état d'une autre partie. Les autres transitions ne provoquent que des transformations intraparties.

Pour les états M de la partie R, nous avons d'après le flot 1:

- > M[Repos(a₀,s₀)] = 1
- > $\forall d \in \text{Recepteur}, \forall x \in \text{Message}$
M[Transmis(-)(a₀,s₀,d,x)] = M[Transmis(-)(a₀,s₀,d,x)] = 0

Ainsi :

- > Les transitions MA_Req [a₀,s₀,d,x] sont franchissables. Après franchissement d'une transition MA_Req [a₀,s₀,d,x], nous obtenons un état de la partie E_{d,x}.
- > Les transitions MA_Conf(-)[a₀,s₀,d,x], MA_Conf(+)[a₀,s₀,d,x] sont infranchissables.

Pour les états d'une partie E_{d,x}, nous avons d'après le flot 1 :

- > M[Repos(a₀,s₀)] = 0
- > $\forall d' \in \text{Récepteur}, \forall x' \in \text{Message}$ tel que (d',x') ≠ (d,x)
M[Transmis(-)(a₀,s₀,d',x')] = M[Transmis(-)(a₀,s₀,d',x')] = 0
- > M[A émettre(a₀,s₀,d,x)] + M[Emis(a₀,s₀,d,x)] +
M[Transmis(-)(a₀,s₀,d,x)] + M[Transmis(-)(a₀,s₀,d,x)] = 1

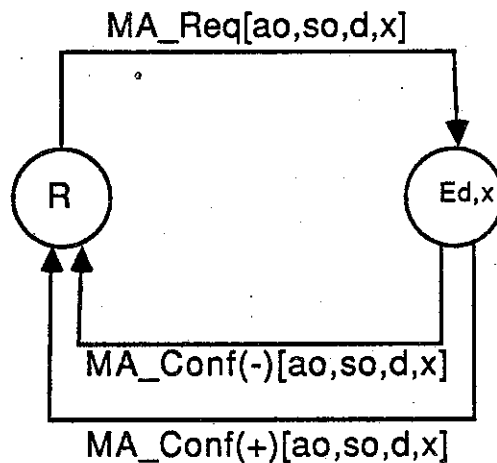
Ainsi :

- > Les transitions MA_Req [a₀,s₀,d,x] sont infranchissables.
- > Pour (d',x') ≠ (d,x), les transitions MA_Conf(-)[a₀,s₀,d,x], MA_Conf(+)[a₀,s₀,d,x] sont infranchissables.
- > La transition MA_Conf(-)[a₀,s₀,d,x] est franchissable si M[Transmis(-)(a₀,s₀,d,x)] = 1. Pour un tel état M de E_{d,x}, nous obtenons par le franchissement un état de R.
- > La transition MA_Conf(+)[a₀,s₀,d,x] est franchissable si

Le Service de la couche MAC pour l'anneau à jeton

$M[\text{Transmis}(+(a_0, s_0, d, x))]=1$. Pour un tel état M de $E_{d,x}$, nous obtenons par le franchissement un état de R .

Nous obtenons finalement le graphe suivant:



-Figure 5.4.1.2- Graphe associé aux transitions:
MA_Req, MA_Conf(+) et Ma_Conf(-) -

Ce graphe définit les contraintes de séquençement des transitions $MA_Req[a_0, s_0, d, x]$, $MA_Conf(-)[a_0, s_0, d, x]$, $MA_Conf(+)[a_0, s_0, d, x]$. Pour qu'une transition $MA_Req[a_0, s_0, d, x]$ soit franchissable, il faut que le modèle soit dans un état M de la partie R . Une fois la transition franchie, le modèle passe dans un état M' de la partie $E_{d,x}$. Pour refranchir à nouveau une transition $MA_Req[a_0, s_0, d, x]$, il faut franchir avant une des transitions $MA_Conf(-)[a_0, s_0, d, x]$, $MA_Conf(+)[a_0, s_0, d, x]$.

Nous avons donc prouvé qu'une demande $MA_Req[a_0, s_0, d, x]$ est toujours suivie d'une confirmation négative $MA_Conf(-)[a_0, s_0, d, x]$ ou positive $MA_Conf(+)[a_0, s_0, d, x]$. Ainsi nous voyons qu'un transfert de données commence par une demande $MA_Req[a_0, s_0, d, x]$ et finit par une confirmation $MA_Conf(-)[a_0, s_0, d, x]$ ou $MA_Conf(+)[a_0, s_0, d, x]$. D'autre part, il apparaît clairement que le processus émetteur ne traite qu'un transfert à la fois. La propriété $P1$ est donc vérifiée.

5.4.2 - Propriétés de la primitive confirmation

Définitions:

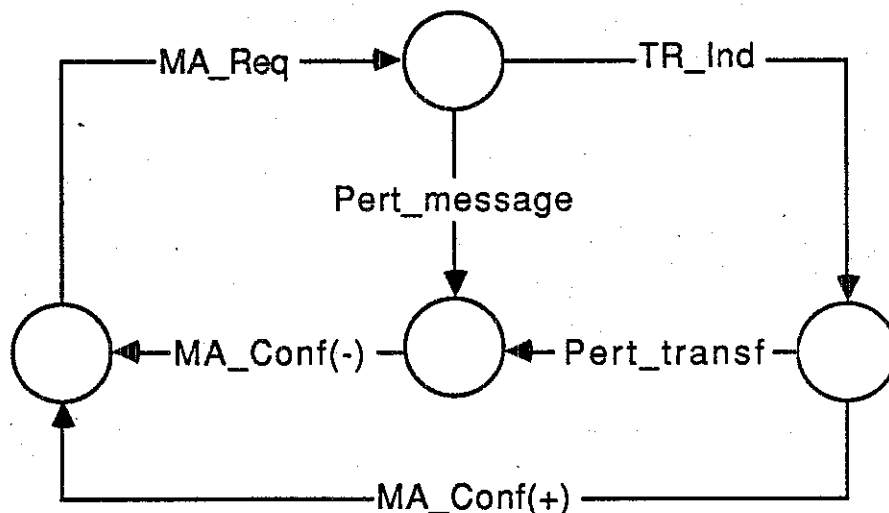
<P2> La primitive MA_Conf(+) indique que l'adaptateur receteur a reçu le message par la primitive TR_Ind,

<P3> La primitive MA_Conf(-) indique une perturbation dans la transmission. Le message peut être ou ne pas être reçu par l'adaptateur receteur.

Les propriétés P2 et P3 permettent de redéfinir tous les scénarios possibles d'une transmission de données:

- <S1> MA_Req, TR_Ind, MA_Conf(+) : Message reçu et confirmation positive,
- <S2> MA_Req, MA_Conf(-) : Perte message et confirmation négative,
- <S3> MA_Req, TR_Ind, MA_Conf(-) : Perte accusé et confirmation négative.

L'automate suivant décrit les trois scénarios :



-Figure 5.4.2.1-Les trois scénarios d'un transfert de données-

Preuve :

Nous considérons un processus émetteur <ao,so>

Le flot1: $\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}$

$$\text{Repos} [a,s] + \sum_{d,x} A_émettre [a,s,d,x] + \sum_{d,m} \text{Emis} [a,s,d,x] + \sum_{d,x} \text{Transmis}(+) [a,s,d,x] + \sum_{d,x} \text{Transmis}(-) [a,s,d,x] = 1$$

et le flot 4: $\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}, \forall d \in \text{Récepteur}, \forall x \in \text{Message}$

$$\text{Message}[a,s,d,x] + \text{Transfert}(+)[a,s,d,x] + \text{Transfert}(-)[a,s,d,x] = \text{Emis} [a,s,d,x]$$

nous permettent d'obtenir un nouveau flot 5 en remplaçant les valeurs de Emis dans le flot 1 par celles définies dans le flot 4.

Le flot5: $\forall a \in \text{Anneau}, \forall s \in \text{Emetteur}$

$$\text{Repos}[a,s] + \sum_{d,x} A_émettre[a,s,d,x] + \sum_{d,x} \text{Message}[a,s,d,x] + \sum_{d,x} \text{Transfert}(+)[a,s,d,x] + \sum_{d,x} \text{Transfert}(-)[a,s,d,x] + \sum_{d,x} \text{Transmis}(+) [a,s,d,x] + \sum_{d,x} \text{Transmis}(-) [a,s,d,x] = 1$$

Nous pouvons donc partitionner l'ensemble des marquages accessibles par les parties:

$$R = \{ M \in A(R) / M[\text{Repos}(a_0, s_0)] = 1 \}$$

$$A_{d,x} = \{ M \in A(R) / M[A \text{ émettre}(a_0, s_0, d, x)] + M[\text{Message}(a_0, s_0, d, x)] = 1 \}$$

$$B_{d,x} = \{ M \in A(R) / M[\text{Transfert}(-)(a_0, s_0, d, x)] + M[\text{Transmis}(-)(a_0, s_0, d, x)] = 1 \}$$

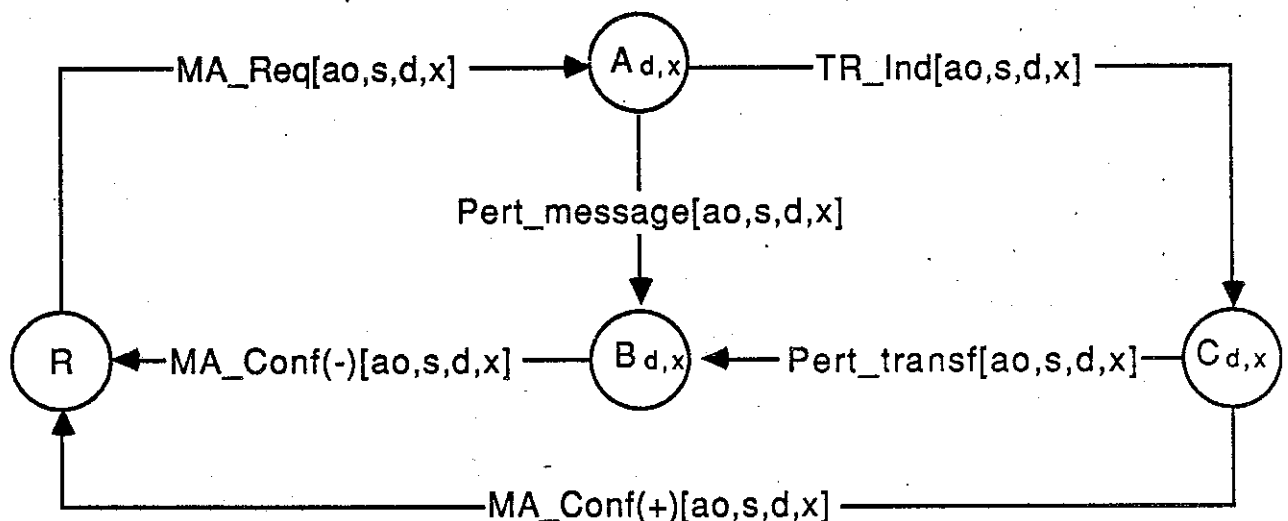
$$C_{d,x} = \{ M \in A(R) / M[\text{Transfert}(+)(a_0, s_0, d, x)] + M[\text{Transmis}(+)(a_0, s_0, d, x)] = 1 \}$$

Nous considérons maintenant les transitions :

- MA_Req[a_0, s_0, d, x],
- MA_Conf(-)[a_0, s_0, d, x],
- MA_Conf(+)[a_0, s_0, d, x].
- TR_Ind[a_0, s_0, d, x],
- Pert_message[a_0, s_0, d, x],
- Pert_trans[a_0, s_0, d, x].

Nous constatons que se sont les seules transitions pouvant provoquer une transformation d'états entre deux parties distinctes. Les autres transitions ne provoquent que des transformations intraparties.

Nous obtenons à partir de ces transitions et de ces parties le graphe suivant:



-Figure 5.4.2.2- Graphe associé aux transitions:

MA_Req , MA_Conf(+), MA_conf(-), Pert_message, Pert_trans , TR_Ind -

Il nous permet de conclure qu'il ne peut exister que trois scénarios possibles pour un transfert de données:

Scénario 1:

- MA_Req[a_o, s_o, d, x],
- TR_Ind[a_o, s_o, d, x],
- MA_Conf(+)[a_o, s_o, d, x].

Scénario 2:

- MA_Req[a_o, s_o, d, x],
- Pert_message[a_o, s_o, d, x],
- MA_Conf(-)[a_o, s_o, d, x],

Scénario 3:

- MA_Req[a_o, s_o, d, x],
- TR_Ind[a_o, s_o, d, x],
- Pert_trans[a_o, s_o, d, x].
- MA_Conf(-)[a_o, s_o, d, x],

Il apparaît clairement qu'une confirmation positive indique que le message a été reçu et qu'une confirmation négative indique une perturbation de la transmission. Nous avons donc prouvé les propriétés P2 et P3.

5.4.3 - Les messages ne sont pas déséquencés

Définition:

<P4> Les messages ne sont pas déséquencés.

Pour deux messages de données $\langle a_o, s_o, d_o, x_a \rangle$ et $\langle a_o, s_o, d_o, x_b \rangle$ émis par l'adaptateurs émetteur $\langle a_o, s_o \rangle$ et et reçu par l'adaptateur récepteur $\langle a_o, d_o \rangle$, les demandes MA_Req[a_o, s_o, d_o, x_a] et MA_Req[a_o, s_o, d_o, x_b] sont franchies dans le même ordre que les indications MA_Ind[a_o, s_o, d_o, x_a] et MA_Ind[a_o, s_o, d_o, x_b].

Preuve:

Soit un adaptateur émetteur $\langle a_o, s_o \rangle$ et un adaptateur récepteur $\langle a_o, d_o \rangle$.

Soit deux messages de données $\langle a_o, s_o, d_o, x_a \rangle$ et $\langle a_o, s_o, d_o, x_b \rangle$ franchissant les transitions MA_Req[a_o, s_o, d_o, x_a], MA_Req[a_o, s_o, d_o, x_b], MA_Ind[a_o, s_o, d_o, x_a],

Le Service de la couche MAC pour l'anneau à jeton

$MA_Ind[a_o, s_o, d_o, x_b]$. Nous savons que la transition $MA_Req[a_o, s_o, d_o, x_a]$ est franchie avant $MA_Req[a_o, s_o, d_o, x_b]$.

D'après le graphe associé aux transitions MA_Req , $MA_Conf(+)$, $MA_conf(-)$, $Pert_message$, $Pert_transf$, TR_Ind , nous constatons que la transition $TR_Ind [a_o, s_o, d_o, x_a]$ est franchie avant la transition $TR_Ind [a_o, s_o, d_o, x_b]$.

De plus, il est facile de voir que si la transition $TR_Ind [a_o, s_o, d_o, x_a]$ est franchie avant la transition $TR_Ind [a_o, s_o, d_o, x_b]$ alors la transition $MA_Ind[a_o, s_o, d_o, x_a]$ est franchie avant la transition $MA_Ind[a_o, s_o, d_o, x_b]$. En effet ces transitions apparaissent dans une séquence de franchissement dans l'ordre $TR_Ind[a_o, s_o, d_o, x_a]$, $MA_Ind[a_o, s_o, d_o, x_a]$, $TR_Ind [a_o, s_o, d_o, x_b]$, $MA_Ind[a_o, s_o, d_o, x_b]$.

Nous avons donc bien prouvé la propriété P4.

6.- CONCLUSION

Nous avons modélisé le service du niveau Mac en phase de transfert de données. Le support de communication du réseau local est un anneau à jeton. Nous utilisons comme modèle formel de description les réseaux de Petri Réguliers.

Notre modèle vérifie des propriétés générales aux réseaux de Petri (état d'accueil, vivacité) et spécifiques au protocole étudié. Deux techniques automatiques pour obtenir ces propriétés ont été employées, une recherche d'une base d'invariants linéaires, et une exploration du graphe d'états engendrés par le modèle. Grâce à la régularité de la classe de réseaux employés, ces preuves sont indépendantes du nombre d'adaptateurs et d'anneaux et du type de trames véhiculées. Les propriétés du Service du niveau MAC sont respectées :

<P1> Un adaptateur émetteur ne traite qu'un transfert à la fois. Le transfert commence par une demande d'émission de trame MA_Req et finit par une confirmation positive MA_Conf(+) ou négative MA_Conf(-).

<P2> Une confirmation positive indique que l'adaptateur destinataire a reçu la trame.

<P3> Une confirmation négative indique une perturbation dans la transmission.

<P4> Les messages ne sont pas déséquencés.

Ce qui prouve la parfaite adéquation des réseaux de Petri Réguliers pour la description de protocole, et la puissance des outils d'analyse automatiques développés dans notre laboratoire.

Le modèle du service MAC que nous venons de construire et valider s'intègrera au modèle d'un protocole original, le niveau Plasma [Cousin 88]. Il nous suffira de fusionner les transitions associées aux primitives d'interface communes aux deux modèles.

