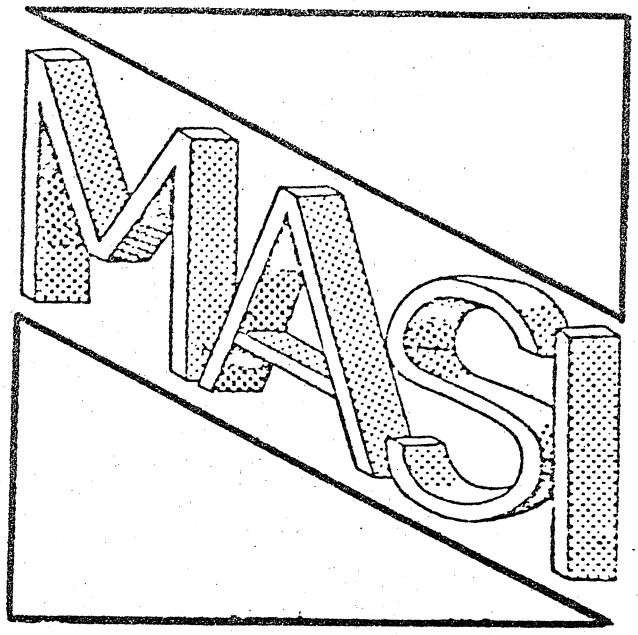
OCTOBRE 1985 N° 73



METHODOLOGIE
&
ARCHITECTURE
DES SYSTEMES
INFORMATIQUES

VALIDATION FONCTIONNELLE DE RESEAUX A PREDICATS : APPLICATION AU MODELE D'UN PROTOCOLE DE COMMUNICATIONS

B. COUSIN, P. ESTRAILLIER

UPMC

CNRS

UNIVERSITE P. ET M. CURIE
Institut de Programmation

CNRS
U.A. 818

4 Place JUSSIEU 75230 PARIS cedex 05 FRANCE

VALIDATION FONCTIONNELLE de RESEAUX a PREDICATS

Application au modele d'un protocole de communications

par Bernard COUSIN
et Pascal ESTRAILLIER

INSTITUT de PROGRAMMATION - CNRS/ MASI(UA 818)
4 , place JUSSIEU
75230 PARIS cedex 05

Resume

Nous avons developpe une methodologie permettant de verifier l'adequation d'un modele avec les specifications d'un systeme parallele. Cette methodologie est basee, non pas sur l'analyse des mecanismes internes, mais sur la construction d'abstractions permettant d'integrer les differentes fonctionnalites du systeme etudie.
La methodologie proposee est particulierement adaptee a l'etude des systemes dont l'architecture est hierarchique, tels que les protocoles de telecommunication. En particulier, nous l'avons appliquee a la modelisation et la preuve du service de la couche Reseau, conformement aux specifications des normes ISO.
Le modele est exprimee a l'aide de reseaux de Petri a predicats.

PLAN
======

1. INTRODUCTION

De nombreux efforts de recherche sont desormais diriges vers la modelisation de processus paralleles. Sans se contenter d'etablir un modele, il est necessaire d'evaluer la qualite de leurs comportements pendant leurs phases d'interactions. On peut ainsi valider les regles de synchronisation qui ont ete etablies.

Malheureusement, la complexite des mecanismes de fonctionnement des systemes paralleles rendent de telles demarches difficiles et souvent laborieuses. En effet, ces travaux reposent sur une specification detaillee et complete des mecanismes mis en oeuvre. Cette specification doit integrer l'ensemble (parfois tres important) des fonctions realisees.

Une analyse partielle de ces protocoles et, trop souvent, l'introduction d'hypotheses par trop simplificatrices ne sont succeptibles d'etre satisfaisantes que si l'on est assure du bien fonde de ces simplifications.

Nous proposons une methodologie originale de modelisation du comportement de systemes paralleles, non pas basee sur l'analyse (exhaustive ?) des mecanismes internes mis en oeuvre, mais au contraire sur la construction d'abstractions permettant d'integrer les differentes fonctionnalites du systeme etudie.

La methodologie proposee est particulierement adaptee a l'etude des systemes dont l'architecture est hierarchique. On peut alors realiser une locale de chacun de ces niveaux en ne prenant en compte comme environnement externe que les interfaces qu'il possede avec les niveaux adjacents

L'application d'une telle methodologie permet l'obtention d'un resultat final constitue par l'ensemble des modeles, dont chacun represente l'abstraction d'un des niveaux hierarchiques. Ainsi l'abstraction de la fonctionnalite d'un niveau integre celle du niveau hierarchiquement inferieur, sans qu'il soit necessaire d'en representer tous ses fondements internes .

Dans cet article, nous appliquons cette methodologie a la modelisation, (au moyen des reseaux de Petri a Predicats [G.W. BRAMS]) des services offerts par la couche RESEAU pendant la phase de transfert des donnees. Ces services ont ete definis dans une norme internationale [ISO].

Toute analyse d'un protocole repose sur une etude prealable de son milieu d'execution. En raison de la structuration en couches des systemes de telecommunication, la hierarchie des protocoles peut etre vue comme une hierarchie de milieux d'execution. Le milieu associe au protocole d'une couche est alors caracterise par l'ensemble des services rendus par le protocole de couche directement inferieure.

Apres avoir detaille les etapes de notre methodologie, nous proposons trois modeles des services de la couche RESEAU. Ces modeles sont construits progressivement a partir de specifications de plus en plus precises.

Le dernier modele est ensuite prouve fonctionnellement, dans la mesure ou les assertions qui lui correspondent sont en parfaite adequation avec les proprietes qui definissent le service.

Un tel modele peut alors etre utilise dans la modelisation de la couche superieure (TRANSPORT) qui integre ainsi, sous une forme tres reduite, les services de la couche RESEAU.

Cette etude originale permet une description precise et complete du contexte d'execution des protocoles de communications, en se degageant des particularites d'implementation.

2. METHODOLOGIE DE VALIDATION FONCTIONNELLE

Toute etude d'un systeme peut donner lieu a plusieurs modelisations differentes quant a leur forme, leur precision et leur fidelite par rapport au systeme modelise.

Pour valider le modele propose, il est necessaire d'evaluer sa conformite par rapport aux specifications du systeme. La methodologie que nous proposons est divisee en cinq etapes successives:

- La specification des proprietes caracterisant le comportement des processus que l'on veut etudier.
  Cette etape repose sur une etude complete du fonctionnement des processus, dans laquelle les fonctionnalites du systeme doivent etre degagees.

- La modelisation des processus qui contituent le systeme.
  Nous utilisons les reseaux de Petri a Predicats qui sont parfaitemen adaptes a la description comportementale des systemes paralleles.
  Cette modelisation est basee sur la specification etablie durant l'etape precedente.
  L'application de cette demarche permet d'obtenir, comme modele, une abstraction des caracteristiques du systeme et non pas une description de ses composantes internes. Nous sommes ainsi parvenus a nous degager de toute implementation particuliere et a definir ce qui est le modele des proprietes du systeme.

- L'etablissement des assertions qui caracterisent le fonctionnement du modele.
  Une bonne connaissance du modele et de la specification du systeme permettent d'effectuer une selection dans l'ensemble des assertions possibles. En effet, seulement certaines d'entre-elles seront neces-saires a la preuve fonctionnelle du modele.

- La validation des assertions.
  Elle est realisee par la preuve des theoremes et des lemmes qui permettent de les exprimer. On notera que la methodologie proposee n'impose aucune methodologie particuliere de preuve. Celle-ci est choisie en fonction du type de resultats desires et de la complexite du modele.
  Nous avons employe la methode de preuve par induction qui est bien adaptee au nombre peu important de transitions que comporte notre modele.

- Preuve de l'adequation entre les assertions relatives au modele et les proprietes du systeme etudie.
  Cette etape permet de montrer formellemnt que les assertions definies dans la troisiemme etape induisent les proprietes introduites a la premiere etape. On en deduit ainsi que le modele propose a la deuxieme etape est une abstraction equivalente, d'un point de vue fonctionnel, au systeme etudie.

Dans le cadre d'un systeme hierarchise, et en particulier dans les systemes de communications, on peut donc, grace a notre methodologie, etudier le comportement de chacun des niveaux de maniere independante et en proposer un modele valide fonctionnellement.

3. Le SERVICE RESEAU

3.1 PRESENTATION

La normalisation en matiere d'interconnexion d'ordinateurs, suite a l'emergence d'un consensus au sein des organismes internationaux pour une architecture a 7 couches [Zimmermann], a fait de rapide progres.

Parmi ces couches, la couche de niveau 4, appelee Transport, doit assurer un service universel de transport de donnees. Ce transport, dont la qualite est controlee, doit etre transparent et optimise [Transport,ECMA].

La couche inferieure (de niveau 3), appelee Reseau, autorise l'acheminement des donnees structurees en paquet au travers d'un large reseau afin de permettre a deux abonnes de dialoguer [Transpac].

Le service Reseau n'est qu'une vue externe des fonctionnalites de la couche Reseau. De ce fait nous ne modeliserons pas tous les mecanismes internes a cette couche (routage, retransmission, gestion de fenetres,...), car ils n'apparaissent pas a l'exterieur.

Notre propos est de definir les proprietes, relatives au transport des donnees, que garantit la couche Reseau. C'est pourquoi nous limiterons notre etude au service de la couche Reseau en phase de transfert de donnees.

Nous porterons un interet particulier a la gestion des pannes de noeuds du reseau de transmission. Ces pannes produisent la desynchronisation de la transmission (perte de paquets). Le traitement de reprise genere des paquets de reinitialisation. Ils sont destines a la couche superieure, qui sera ainsi prevenue de l'incident. Les paquets de reinitialisation transitent dans le reseau au meme titre que les autres paquets (en particulier, ils peuvent aussi etre perdus).

3.2 SPECIFICATIONS

Le modele doit respecter les proprietes definies dans la norme [ECMA], qui caracterisent le service. Notamment, le service Reseau doit assurer la transmission bidirectionnelle des donnees emises et veiller au respect des directives qui definissent le comportement specifique du Reseau face aux desynchronisations :

- Propriete 0 : le service Reseau est toujours susceptible de transferer des paquets (il n'est jamais en situation de blocage irremediable).

- Propriete 1 : non-desordonnancement des paquets durant le transfert;

- Propriete 2 : non-duplication des paquets durant leur transfert;

- Propriete 3 : gestion des desynchronisations.
  "- L'indication de reinitialisation est transmise aux deux extremites de la connexion.
  - Les donnees soumises au reseau avant le signal de reinitialisation sont, soit delivrees avant le signal, soit detruites.
  - Les donnees soumises au reseau apres le signal sont delivrees a l'autre extremite, apres que celle-ci ait recu le signal."

Nous appelons signal de reinitialisation l'evenement traduit par l'arrivee d'un paquet de reinitialisation a l'une des entites exterieures.

4. Le MODELE

Nous presentons, en premier lieu, un modele des services offerts par la couche Reseau: le transfert des paquets sur une voie de communication unidirectionnelle dans un contexte non defaillant.

Puis en incluant les fonctionnalites definies aux paragraphes precedents, nous modelisons les traitements des desynchronisations et le transfert bidirectionnel des donnees.

Notations:

Dans les figures presentees, les noms des places sont prefixes par la lettre P et le nom de transition par la lettre T.

"Mo" denote la fonction de marquage initial.

4.1 Le TRANSFERT de PAQUETS

La couche reseau, en plus de la connexion et de la deconnexion de liaison, assure le transfert de paquets entre deux stations integrees a un systeme d'interconnexion.

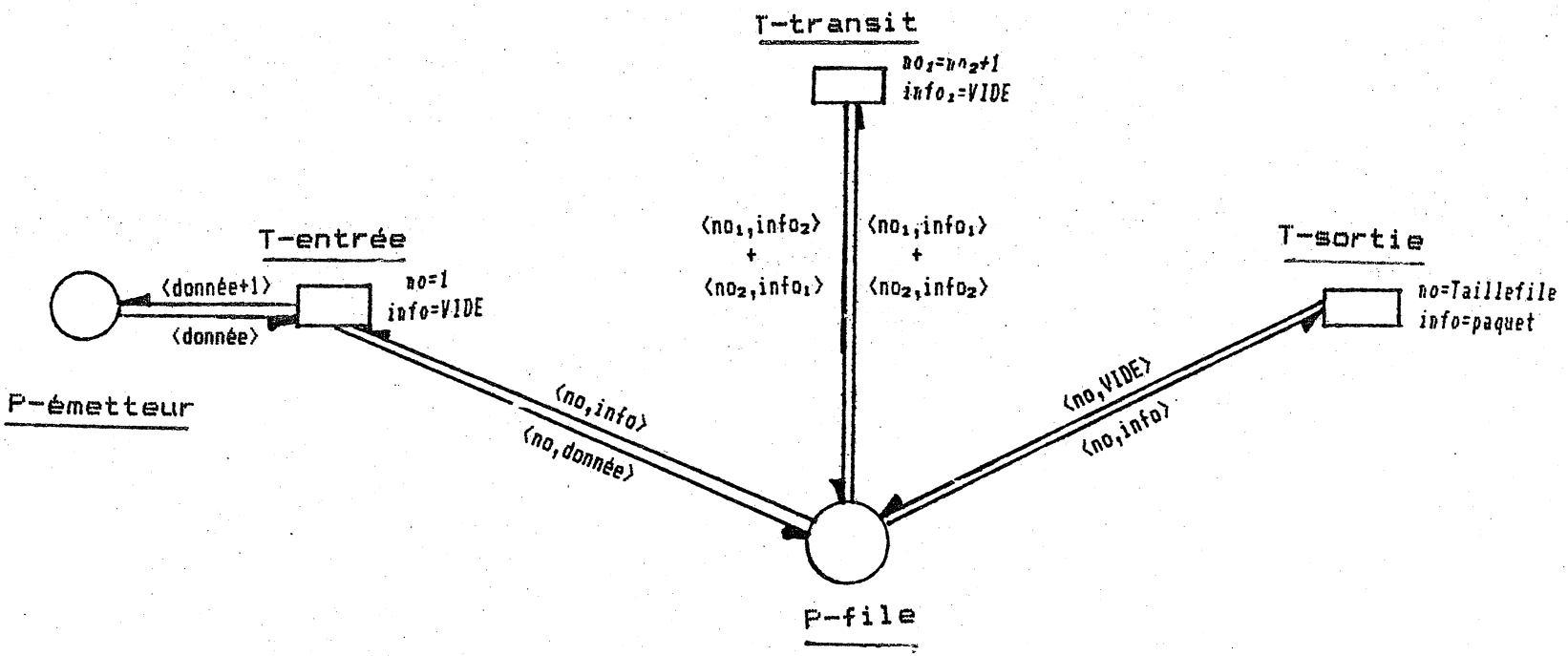
4.11 Fonctionnalites

D'un point de vue fonctionnel, le transfert de paquets pour les reseaux de type Transpac (a contrario des reseaux de type Datagramme), peut etre assimile a la gestion d'une file FIFO dont chaque emplacement correspond a la zone de memorisation d'un paquet traite par une station du systeme de telecommunication. Les paquets d'informations transitent sur des stations intermediaires avant d'etre delivres a leur destinataire. Dans ce modele, la relation d'ordre, induite par la file, est l'interpretation de la chronologie des arrivees des paquets sur la connexion reseau.

Le modele de la file FIFO a ete deja largement etudie dans [Berthelot81]. Nous proposons une modelisation equivalente sous une forme tres condensee (1 place, 3 transitions) propice a notre demarche.

Chaque emplacement est caracterise par son numero d'ordre et par l'eventuel paquet qu'il contient. La file contient un nombre fini d'emplacements. Cette limite correspond au nombre maximum de paquets pouvant circuler simultanement sur chaque connexion reseau.

figure 1


T-transit
no₂=no₂+1
info₂=VIDE
⟨no₁,info₂⟩
+
⟨no₂,info₁⟩
⟨no₁,info₁⟩
+
⟨no₂,info₂⟩
T-entrée
⟨donnée+1⟩
⟨donnée⟩
no=1
info=VIDE
T-sortie
no=Taillefile
info=paquet
P-émetteur
⟨no,info⟩
⟨no,donnée⟩
⟨no,VIDE⟩
⟨no,info⟩
P-file

4.12 Modelisation (figure 1) :

Un paquet entre dans le reseau a l'initiative de la couche transport. La donnee qu'il contient permet l'identification des paquets qui circulen sur le reseau. Cette identification est realisee à l'aide d'un compteur gere par la couche transport (P-emetteur).

Ce compteur est incremente apres chaque nouvelle emission. Un paquet de donnee transitant sur le reseau est ainsi identifie par un numero uniq Le plus grand numero correspond au dernier message emis par la couche TRANSPORT. Ce compteur est uniquement ajoute au modele afin d'exprimer les proprietes du protocole, sans toutefois en perturber le fonctionnemen

Le paquet peut etre depose (T-entree) en debut de la file des paquets (P-file) des que l'emplacement numero 1 est libre.

Le paquet progresse ensuite (T-transit) dans la file en fonction des emplacements qui se liberent.

Il peut enfin etre delivre (T-sortie) a son destinataire lorsqu'il atteint l'emplacement numero TAILLEFILE (le dernier de la file). Le paquet est alors transmis au recepteur et l'emplacement qui le contenait est desormais vide.

Structure des uplets:

. La marque de la place P-emetteur est le uplet <donnee>:
  - donnee : represente la valeur du compteur,
    . donnee $\in \mathbb{N}$.

. Chaque marque de la place P-file est un emplacement defini par un 2-uplet <no,info> :
  - no : represente le numero d'emplacement,
    . no $\in$ [1..TAILLEFILE] ;
  - info : represente l'information contenue par l'emplacement,
    . info $\in$ {VIDE,donnee},
    . donnee $\in \mathbb{N}$.

Marquage initial:

- $\forall$ ui $\in$ Mo(P-file) info=VIDE
  la place P-file ne contient que des emplacements vides (il n'y a aucun paquet circulant sur le reseau)
- Mo(P-emetteur) = { $\emptyset$ }
  La place P-emetteur contient une marque de valeur $\emptyset$ (la couche Transport n'a emis encore aucune donnee).

4.2 La GESTION de le DESYNCHRONISATION

Nous completons le modele precedent, afin de gerer les phenomenes dus aux pertes de synchronisation (propriete 3 restreinte a une seule voie de communication).

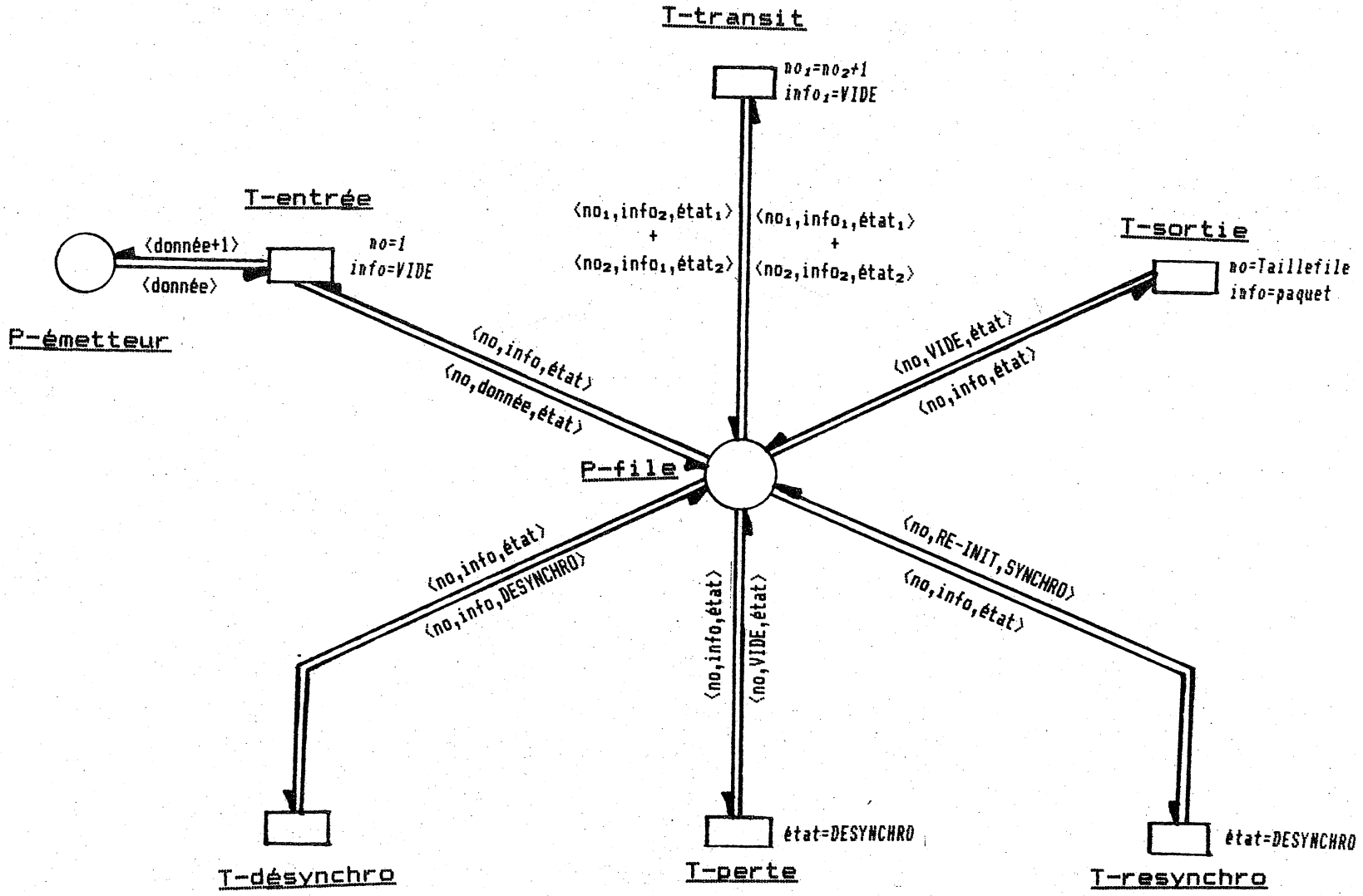
4.21 Fonctionnalites

Au cours de la phase de transfert, un incident (panne de station, etat incoherent du protocole,...) peut provoquer une desynchronisation qui perturbe eventuellement le transit des paquets.

Les zones de memorisation des paquets peuvent alors contenir des informations incoherentes. Les emplacements qui leur correspondent sont, soit dans l'etat synchronise (c'est le fonctionnement normal de la station),soit dans l'etat desynchronise en cas d'erreur. Chaque emplacement est donc caracterise, en plus des elements precedemment decrits, par son etat.

Le service Reseau integre la perte de paquets de tous types (donnee,reinitialisation...), mais il assure qu'apres une phase de desynchronisation (pendant laquelle des paquets ont ete perdus) survient une phase de resynchronisation . Cette phase permet de prevenir les deux extremites receptrices par l'intermediaire de signaux de reinitialisation.

figure 2


T-transit
no₁=no₂+1
info₁=VIDE
T-entrée
⟨no₁,info₂,état₁⟩
+
⟨no₂,info₁,état₂⟩
⟨no₁,info₁,état₁⟩
+
⟨no₂,info₂,état₂⟩
T-sortie
no=Taillefile
info=paquet
⟨donnée+1⟩
⟨donnée⟩
no=1
info=VIDE
P-émetteur
⟨no,info,état⟩
⟨no,donnée,état⟩
⟨no,VIDE,état⟩
⟨no,info,état⟩
P-file
⟨no,info,état⟩
⟨no,info,DESYNCHRO⟩
⟨no,info,état⟩
⟨no,VIDE,état⟩
⟨no,RE-INIT,SYNCHRO⟩
⟨no,info,état⟩
état=DESYNCHRO
état=DESYNCHRO
T-désynchro
T-perte
T-resynchro

4.22 Modelisation (figure 2) :

Toute perturbation du comportement d'une station provoque eventuellement la desynchronisation (T-desynchro) au niveau des emplacements correspondants qui passent d'un etat synchronise a l'etat desynchronise.

A partir de cet instant, chaque paquet transitant par cet emplacement peut eventuellement etre perdu (T-perte). L'emplacement redevient alors vide.

Des qu'il detecte la desynchronisation, le service reseau, apres un traitement de synchronisation, emet (T-resynchro) le paquet de reinitialisation RE-INIT. L'emplacement retourne alors a l'etat synchronise.

On note qu'une fois emis, le paquet de reinitialisation subit le traitement de tout autre paquet. De ce fait, il progresse dans la file de maniere sequentielle et peut eventuellement disparaitre lors d'une panne de noeud.

La perte d'un paquet de reinitialisation peut survenir pendant son transfert sur le Reseau. Toutefois, du fait de la recursivite du phenomene de resynchronisation, cette perte regenere inevitablement (ulterieurement) un paquet de reinitialisation sur la voie de communication.

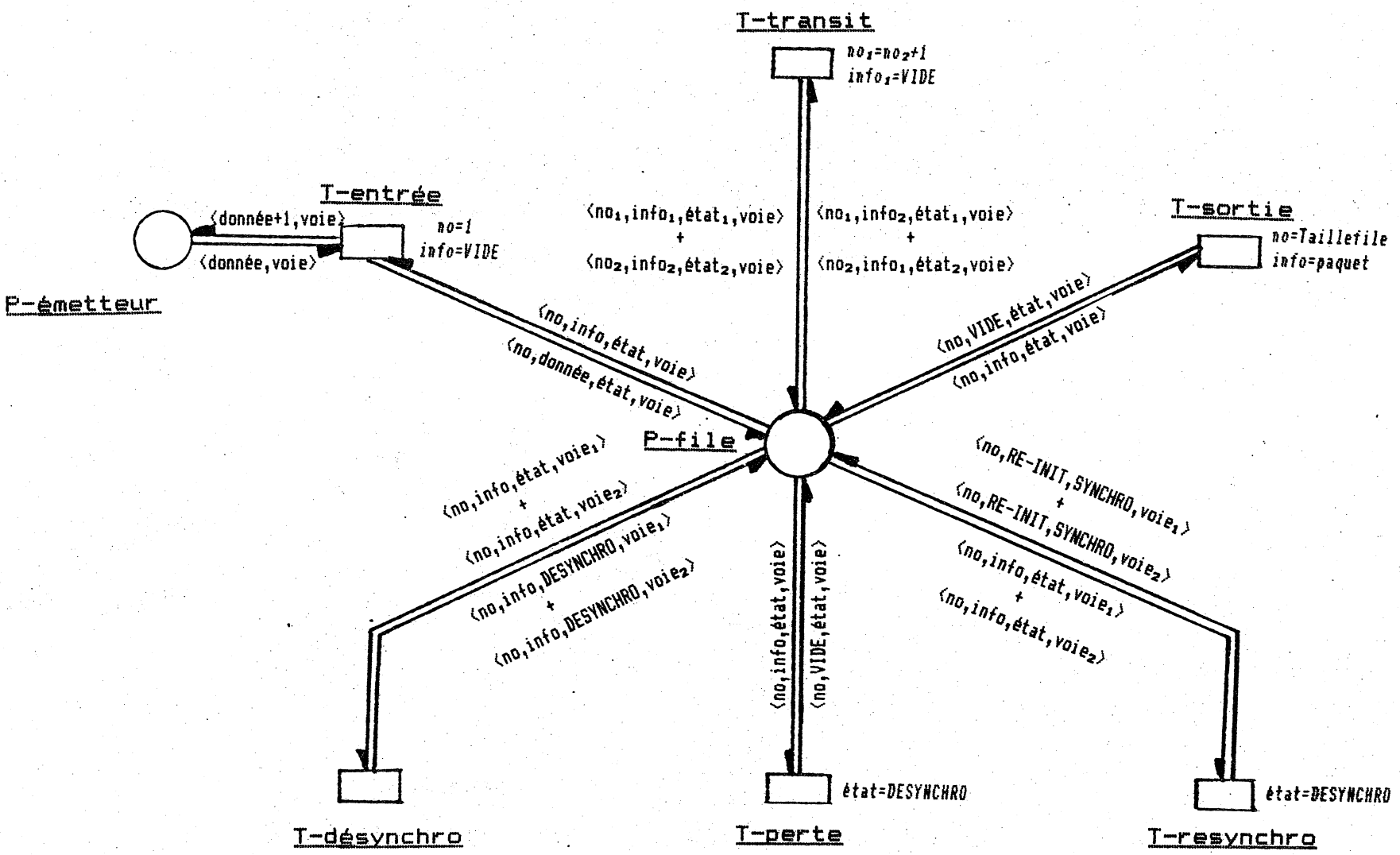
Structure des uplets:

- La marque de la place P-emetteur est le uplet <donnee>:
  - donnee : represente le compteur,
    - donnee $\in \mathbb{N}$.
- Chaque marque de la place P-file est un emplacement defini par un 3-uplet <no,info,etat> :
  - no : represente le numero d'emplacement ,
    - no $\in$ [1..TAILLEFILE] ;
  - info : represente l'information contenue par l'emplacement, on a
    - info $\in$ {VIDE,paquet},
    - paquet $\in$ {RE-INIT,donnee},
    - donnee $\in \mathbb{N}$;
  - etat : represente l'etat de l'emplacement,
    - etat $\in$ {SYNCHRO,DESYNCHRO} .

Marquage initial:

- $\forall$ ui $\in$ Mo(P-file) info=VIDE, ETAT=DESYNCHRO
  (la place P-file ne contient que des emplacements vides et synchronis
  il n'y a donc aucun paquet circulant sur le reseau, et aucune panne de station: le reseau est dans un etat sain).
- Mo(P-emetteur) = { $\emptyset$ }
  (la couche transport n'a emis encore aucune donnee)

figure 3


T-transit
no₁=no₂+1
info₁=VIDE
T-entrée
⟨donnée+1,voie⟩
⟨donnée,voie⟩
no=1
info=VIDE
P-émetteur
⟨no₁,info₁,état₁,voie⟩ + ⟨no₂,info₂,état₂,voie⟩
⟨no₁,info₂,état₁,voie⟩ + ⟨no₂,info₁,état₂,voie⟩
T-sortie
no=Taillefile
info=paquet
⟨no,info,état,voie⟩
⟨no,donnée,état,voie⟩
⟨no,VIDE,état,voie⟩
⟨no,info,état,voie⟩
P-file
⟨no,info,état,voie₁⟩ + ⟨no,info,état,voie₂⟩
⟨no,info,DESYNCHRO,voie₁⟩ + ⟨no,info,DESYNCHRO,voie₂⟩
⟨no,info,état,voie⟩
⟨no,VIDE,état,voie⟩
⟨no,RE-INIT,SYNCHRO,voie₁⟩ + ⟨no,RE-INIT,SYNCHRO,voie₂⟩
⟨no,info,état,voie₁⟩ + ⟨no,info,état,voie₂⟩
T-désynchro
état=DESYNCHRO
T-perte
état=DESYNCHRO
T-resynchro

4.3 Le TRANSFERT BIDIRECTIONNEL des PAQUETS

Nous etendons le modele de la figure 2 en le dupliquant de maniere logique de sorte a decrire le comportement des deux voies de communication.

4.31 Fonctionnalites

Le service de transmission est bidirectionnel. La description d'un emplacement doit donc etre completee par la caracterisation de sa voie de transfert (voie1,voie2).

Chaque station du systeme de telecommunication est donc modelisee par un double emplacement, qui represente l'organe de stockage de chacun des deux sens de transmission.

4.32 Modelisation (figure 3) :

Dans leur phase de transmission de donnees, les voies de communication sont asynchrones. Sur ces voies, les traitements d'entree (T-entree), de transit (T-transit) et de sortie (T-sortie) des paquets sont independant

Le mauvais fonctionnement d'une station est repercute (T-desynchro) au niveau des deux connexions par la desynchronisation des emplacements qui correspondent a la station. Les emplacements d'emission et de reception passent alors simultannement dans l'etat desynchronise .

De maniere independante a chacun des deux sens de transfert, les eventuelles pertes (T-perte) des paquets qui transitent sur la station detruisent le contenu des emplacements.

A la detection de la desynchronisation, le service reseau, emet (T-resynchro) un paquet de resynchronisation simultanement sur chacun des deux sens de communication (propriete 3).

Structure des uplets:

. Les deux marques de la place P-emetteur sont definies par le 2-uplet <donnee,voie> :
  - donnee : represente la valeur du compteur d'emission,
    . donnée $\in \mathbb{N}$;
  - voie : represente L'appartenance du compteur a l'un des sens de transmition,
    . voie $\in$ {VOIE1,VOIE2} .

. Chaque marque de la place P-file est un emplacement defini par un 4-uplet <no,info,etat,voie> :
  - no : represente le numero d'emplacement ,
    . no $\in$ [1..TAILLEFILE] ;
  - info : represente l'information contenue par l'emplacement, on a
    . info $\in$ {VIDE,paquet},
    . paquet $\in$ {RE-INIT,donnee},
    . donnee $\in \mathbb{N}$;
  - etat : represente l'etat de l'emplacement,
    . etat $\in$ {SYNCHRO,DESYNCHRO};
  - voie : represente l'appartenance de l'uplet a l'un des sens de transmition,
    . voie $\in$ {VOIE1,VOIE2}.

Marquage initial:

- $\forall$ ui $\in$ Mo(P-file) info = VIDE, etat=SYNCHRO
  (il n'y a aucun paquet circulant sur le reseau, qui est dans un etat sans panne).
- $\forall$ ui $\in$ Mo(P-emetteur) donnee = $\emptyset$
  (sur chacune des voies de transmission, la couche TRANSPORT n'a encore emis aucune donnee).

5. EVALUATION du MODELE

Apres avoir modelise le service de la couche Reseau, nous demontrons que le modele respecte les proprietes definies dans la norme, qui caracterisent ce service.

Nous introduisons d'abord l'ensemble des notations necessaires a notre preuve, puis nous proposons un ensemble d'assertions,exprimees a l'aide de lemmes et de theoremes, qui concretisent, au niveau du modele propose, les proprietes du service Reseau.

Apres avoir explique la demarche suivie pour obtenir les preuves des theoremes, nous exprimons les trois proprietes a partir des theoremes et des lemmes.
Nous prouvons ainsi que le modele du service est conforme a la norme, et qu'il est equivalent, dans ce sens, au modele du protocole.

5.1 Les Assertions

5.11 Les Notations

Afin de manipuler aisement les uplets du modele, nous definissons les projections suivantes :

- NOEMPL(ui): la fonction rend le numero d'emplacement associe a l'uplet ui.
  Dans le modele : pour ui=<noi,infoi,etati,voiei> ,
  noempl(ui) = noi;

- TYPAQUET(ui): la fonction rend le type du paquet associe a l'uplet ui.
  Dans le modele : pour ui=<noi,infoi,etati,voiei> ,
  typaquet(ui) = if infoi=VIDE then VIDE
  else if infoi=RE-INIT then RE-INIT
  else DONNEE;

- NODONNEE(ui): la fonction rend le numero de donnee du paquet associe a l'uplet ui.
  Dans le modele : pour ui=<noi,infoi,etati,voiei> ,
  nodonnee(ui) = if typaquet(ui)=DONNEE then infoi
  else INDETERMINEE;

- ETAT(ui): la fonction rend l'etat de l'uplet (emplacement) ui.
  Dans le modele : pour ui=<noi,infoi,etati,voiei> ,
  etat(ui)=etati ;

- SENS(ui): la fonction rend le voie de transmission auquel appartient l'uplet ui.
  Dans le modele : pour ui=<noi,infoi,etati,voiei> ,
  voie(ui) = voiei ;

Nous utiliserons les fonctions suivantes:

- Suivant(ui,P): la fonction rend l'uplet contenant un paquet suivant l'uplet ui parmi l'ensemble P.

  Dans le modele :

  Soit S l'ensemble de tout paquet emis apres le paquet ui:

  $S = \{uk/uk \in P$, typaquet(uk)/=VIDE et noempl(uk)<noempl(ui)

  Suivant(ui,P)=uj Ssi l'uplet uj est le plus proche de l'uplet u

  $\forall$ uk $\in$ S noempl(uj)>=noempl(uk).

- Dernier(P): la fonction rend l'uplet contenant le paquet le plus proche de l'emetteur (il comporte le numero d'emplacement le plus peti parmi l'ensemble P).

  Dans le modele :

  Dernier(P)=ui ssi ui $\in$ P, typaquet(ui)=DONNEE et

  $\forall$uj $\in$ P telque typaquet(uj)=DONNEE, noempl(ui)<=noempl(uj)

- Premier(P): la fonction rend l'uplet contenant le paquet le plus proche du recepteur (il comporte le numero d'emplacement le plus grand de l'ensemble P).

  Dans le modele :

  Premier(P)=ui ssi ui $\in$ P, typaquet(ui)=DONNEE et

  $\forall$uj $\in$ P telque typaquetuj)=DONNEE, noempl(ui)>=noempl(uj).

- Voie1(P): la fonction rend l'ensemble des uplets appartenant a la voie de transmission 'VOIE1' de l'ensemble P.

  Dans le modele :

  Voie1(P)= { ui $\in$ P telque voie(ui)=VOIE1}.

- Voie2(P): la fonction rend l'ensemble des uplets appartenant a la voie de transmission 'VOIE2' de l'ensemble P.

  Dans le modele :

  Voie2(P)= { ui $\in$ P telque voie(ui)=VOIE2}.

3.12 Les Theoremes

Nous allons demontrer, que le modele de la figure 3, verifie les quatre propositions suivantes :

- le modele de la couche Reseau est vivant.
- la couche Reseau conserve la sequentialite des paquets.
- la couche Reseau ne duplique pas les paquets.
- toute desynchronisation est correctement detectee.

Soit A l'ensemble des fonctions de marquage accessible a partir du modele.
Soit T l'ensemble des transitions du modele.

Theoreme 0: le modele est vivant.

T0: $\forall M \in A$, $\forall t \in T$, $\exists s \in T^*$ telque $M(st>$.

Interpretation: toutes les transitions sont toujours franchissables.

Theoreme 1: le modele conserve la sequentialite des paquets.

T1: $\forall M \in A$, $\forall$ Voie $\in$ (Voie1,Voie2) ,
$\forall ui \in$ Voie(M(P-file)) , $\forall uj \in$ Voie(M(P-file)) avec
typaquet(ui)=DONNEE et typaquet(uj)=DONNEE

si noempl(ui)>=noempl(uj) alors nodonnee(ui)<=nodonnee(uj).

Interpretation: Soient 2 emplacements contenant des paquets de donnees,le paquet le plus pres du recepteur (de numero d'emplacement le plus grand) est le paquet ayant ete emis le plus tot (de numero de paquet le plus petit).

Theoreme 2: le modele ne duplique pas les paquets.

T2: $\forall M \in A$, $\forall$ Voie $\in$ (Voie1,Voie2) ,
$\forall ui \in$ Voie(M(P-file)) , $\forall uj \in$ Voie(M(P-file)) avec
typaquet(ui)=DONNEE et typaquet(uj)=DONNEE ,

si noempl(ui)/=noempl(uj) alors nodonnee(ui)/=nodonnee(uj).

Interpretation: Soient deux emplacements contenant des paquets de donnees, s'ils sont differents alors ils contiennent des donnees differentes).

Theoreme 3: toute desynchronisation est correctement detectee.

T3: $\forall M \in A$, $\forall$ Voie $\in$ (Voie1,Voie2) ,
$\forall ui \in$ Voie(M(P-file)) avec typaquet(ui)=DONNEE,
si $\exists uj \in$ Voie(M(P-file)) tel que Suivant(ui,Voie(M(P-file)))=uj ,
typaquet(uj)=DONNEE et nodonnee(ui)/=nodonnee(uj)-1 alors :

- T3.1: soit $\exists uk \in$ Voie(M(P-file)) tel que typaquet(uk)=RE-INIT et noempl(uk)<noempl(ui);
- T3.2: soit $\exists uk \in$ Voie(M(P-file)) tel que etat(uk)=DESYNCHRO et noempl(uk)<noempl(ui);

Interpretation: soient deux emplacements consecutifs contenant deux paquets de donnee emis non-successivement (il y a eu une perte entre ces deux paquets):

- soit il existe un paquet de reinitialisation dans un des emplacements suivants;
- soit un des emplacements suivants est encore desynchronise .

5.13 Les Lemmes

Pour la demonstration de ces theoremes, les lemmes suivants doivent etre ennonces.

Lemme 1 : Aucun emplacement ne contient un paquet de donnee de numero de paquet superieur au compteur d'emission.

L1: $\forall M \in A$, $\forall Voie \in \{Voie1, Voie2\}$,
$\forall ui \in Voie(M(P\text{-}file))$,
si typaquet(ui)=DONNEE alors nodonnee(ui) < Voie(M(T-P-emetteur)).

Lemme 2 : Toute phase de desynchronisation (perte) est determinee par la presence d'un emplacement desynchronise.

L2: $\forall M \in A$, si M(T-perte) = M' alors
$\exists ui \in M'(P\text{-}file)$ tel que etat(ui)=DESYNCHRO .

Lemme 3 : Toute phase de resynchronisation se concretise par l'emission dans chaque voie de transmission d'un paquet de reset.

L3: $\forall M \in A$, si M(T-resynchro) = M' alors
$\exists u1 \in Voie1(M'(P\text{-}file))$ telque typaquet(u1)=RE-INIT ,
$\exists u2 \in Voie2(M'(P\text{-}file))$ telque typaquet(u2)=RE-INIT .

Lemme 4 : Chaque marque de la place P-file comporte un numero unique 1..Taillefile qui definit le rang de l'emplacement correspondant dans la file modelise par la place.

L4: $\forall M \in A$, $\forall Voie \in \{Voie1, Voie2\}$,
{noempl(uk) tel que uk $\in$ Voie(M(P-file))} = [1..Taillefile].

Lemme 5 : Toute desynchronisation du dernier paquet emis est detectee.

L5: $\forall M \in A$, $\forall Voie \in \{Voie1, Voie2\}$,
si $\exists ui \in Voie(M(P\text{-}file))$ tel que Dernier(Voie(M(P-file)))=ui
et nopquet(ui)/=Voie(M(P-T-emetteur))-1 alors :
- L5.1: soit $\exists uk \in Voie(M(P\text{-}file))$ tel que typaquet(uk)=RE-INIT et noempl(uk)<noempl(ui);
- L5.2: soit $\exists uk \in Voie(M(P\text{-}file))$ tel que etat(uk)=DESYNCHRO et noempl(uk)<noempl(ui);

Interpretation: Soit le paquet le plus pres de l'emetteur, s'il comporte un numero de paquet non inferieur de un a la valeur du compteur d'emission (il ne vient pas d'etre emis):
- soit il existe un paquet de reinitialisation entre lui et l'emetteur;
- soit il existe un emplacement en etat de desynchronisation entre lui et l'emetteur.

Lemme 6: Les deux organes de stockage d'une meme station sont toujours dans le meme etat.

L6: $\forall M \in A$, $\forall u1 \in Voie1(M(P\text{-}file))$, $\forall u2 \in Voie2(M(P\text{-}file))$,
si noempl(u1)=noempl(u2) alors etat(u1)=etat(u2).

Interpretation: Soient deux uplets de chacune des voies de transmission, s'ils ont meme numero d'emplacement (s'ils appartiennent a la meme station), ils ont meme etat.

5.2 Validation du modele

Nous ne presentons ici, pour des raisons de concision, qu'un succedane de cette preuve. La preuve exhaustive du dernier modele est proposee dans [Cousin].

5.21 Demarche de preuve

L'ensemble des lemmes definis au paragraphe precedent sont principalement utilises pour demontrer les theoremes.
Nous allons introduire la demarche de preuve suivie pour la demonstration des theoremes (T0,T1,T2,T3).

Theoreme 0 : Nous montrons la vivacite du modele en deux etapes successives.

Etape 1 : Le modele possede un ensemble E d'etats d'accueil

Nous definissons l'ensemble E par le marquage des deux places du modele tel que :

- Les uplets de la place P-file sont vides de contenu, leur etat synchrone (sans erreur):
  $\forall M \in E , \forall u \in M(P\text{-file})$ etat(u)=SYNCHRO , typaquet(u)=VIDE.
- Le marquage de la place P-emetteur peut etre quelconque.

On note que l'etat initial fait partie de l'ensemble d'accueil.

Pour prouver que E est un ensemble d'accueil, il faut montrer qu'a partir d'un etat quelconque de l'ensemble A des etats accessibles, il est toujours possible d'atteindre un des etats de l'ensemble d'accueil. Cette demonstration est effectuee en deux sous-etapes :

Sous-etape 1 : Il est possible de delivrer aux recepteurs (s'il en exist l'ensemble des paquets presents dans la place P-file (l'ensemble de emplacements devient vide);

Nous vidons les emplacements de la place P-file.(i.e. on veut que $\forall u$ typaquet(u)=VIDE).
On effectue les actions suivantes pour la VOIE1, puis pour la VOIE2 Cela est licite, car toutes les transitions que nous avons a franch a mettent en cause qu'une seule voie de transmission.

Tant que la place P-file contient encore des paquets (i.e. tant que $\forall u \in M(P\text{-file})$ tel que typaquet(u)/=VIDE),
soit p l'uplet contenant le paquet le plus proche du recepteur (.i.e p=Premier(M(P-file))),
tant que l'uplet p n'est pas en bout de file (cote recepteur) (i.e tant que noempl(p)/=Taillefile),
on fait progresser l'uplet p dans le reseau.
C'est possible, car par construction l'uplet p etant le premier paquet encore contenu par le reseau, les emplacements precedents sont vides.
Donc la transition T-transit est franchissable et provoque l'echange entre p et l'emplacement vide le precedant. Le paquet p progresse d'un emplacement, et il conserve son statut de premier paquet (d'apres le libelle des arcs et predicats de la transition : noempl(p)=noempl(p)+1).
Le premier paquet p arrive, donc, en bout de reseau (i.e. p=premier(M(P-file)), typaquet(p)=paquet et noempl(p)=Taillefile): la transition T-sortie devient franchissable, ses predicats sont verifies. Le reseau contient un paquet de moins.
Et ce, jusqu'a delivrance de tous les paquets contenus par le reseau.

Sous-etape 2 : Il est possible de resynchroniser (si necessaire) les emplacements desynchronises de la place P-file (l'ensemble des emplacements devient synchrone).

Nous supprimons les emplacements en etat de desynchronisation (on veut que $\forall u \in M(P\text{-file})$ etat(u)=SYNCHRO).

Tant qu'il existe des emplacements en etat de desynchronisation (i.e. $\forall u \in M(P\text{-file})$ tel que etat(u)=DESYNCHRO), soit d l'emplacement de numero maximun en etat de desynchronisation (i.e. noempl(d)>=noempl(u), $\forall u \in \{uk \in M(P\text{-file})/etat(uk)=DESYNCHR$
D'apres le lemme L6, les deux emplacements d1 et d2, de meme numero (appartenant a la meme station de transport) sont dans le meme etat desynchronisation (i.e. noempl(d1)=noempl(d2) et etat(d1)=etat(d2)).
La transition T-resynchro est donc franchissable, ce qui produit sur chacune des deux voies de transmission un paquet de reinitialis (i.e. $\exists r1 \in Voie1(M(P\text{-file}))$ et $\exists r2 \in Voie2(M(P\text{-file}))$ tel que typaquet(r1)=typaquet(r2)=RE-INIT (lemme L3)).
Le reseau comporte alors un emplacements de moins en etat de desynchronisation.
Nous pouvons faire progresser ces paquets jusqu'a leur extremite receptrice de facon similaire a la premiere etape. En leur faisant franchir la transition T-sortie, le reseau ne contient, a nouveau, plus aucun paquet.
Et ce, jusqu'a disparition totale de tous les emplacements en etat de desynchronisaton.

Ceci nous permet de montrer que l'ensemble E est bien un ensemble d'etats d'accueil.

Etape 2 : Dans un etat quelconque de l'ensemble E d'accueil, le modele est quasi-vivant.

Pour prouver cette propriete, nous devons de franchir successivement toutes les transitions du modele.
Dans un etat quelconque de l'ensemble des etats d'accueil, la place P-file ne contient, que des emplacements vides. la transition T-entree est franchissable et provoque l'apparition d'un paquet de donnee dans l'emplacement numero 1.
La transition T-desynchro est toujours franchissable. Nous la declenchons sur l'emplacement 1, qui devient desynchronise.
De ce fait, la transition T-perte peut etre franchie pour l'emplacement 1. Le paquet de donnee est donc perdu.
Nous decidons de franchir la transition T-resynchro, ce qui provoque l'apparition d'un paquet de reinitialisation sur chacune des deux voies de transmission.
La transtion T-transit permet de les propager a travers le reseau jusqu'a leur entite destinatrice.
La transition T-sortie devient alors franchissable.

Pour demontrer les autres lemmes et theoremes, nous avons utilise la methode de preuve par induction :
Nous demontrons qu'ils sont vrais dans l'etat initial et, etant vrais, qu'ils restent vrais apres le franchissement de n'importe quelle transition du modele.

Theoremes 1 et 2 : Les demonstrations des deux premiers theoremes T1 et T2 procedent de la meme demarche. La preuve a ete realisee en deux etapes :

Etape 1 : Assimilation des proprietes du modele a celles d'une file FIFO.
Dans le modele (figure 3), on remarque que les champs <etat, voie> ne figurent dans aucun predicat associe aux transitions T-entree, T-transit et T-sortie. On peut alors projeter les uplets du modele, restreint a ces transitions, sur le uplet <no, info>. Le modele obtenu est celui decrit dans la figure 1 qui represente une file FIFO sans perte.

On en conclue que le modele de la figure 3, restreint a ces trois transitions, possede les proprietes d'une file FIFO (sequentialite et non duplication):

Etape 2 : Preservation de ces proprietes dans la gestion de la desynchronisation.
Les proprietes de sequentialite et de non duplication sont exprimees par les champs <no,info> des uplets.
(ui et uj sont tels que typaquet(ui) = typaquet(uj) = DONNEE)

Sequentialite :
si noempl(ui)>=noempl(uj) alors nodonnee(ui)<=nodonnee(uj)
Non duplication:
si noempl(ui)/=noempl(uj) alors nodonnee(ui)/=nodonnee(uj)

. Sur les arcs des transitions T-desynchro, T-perte et T-resynchro il n'existe aucun couple de uplet echangeant leurs champs d'information.
. La transition T-desynchro ne modifie que le champ <etat> des uplets (elle ne gere aucun nouvel uplet). Les champs <nØ, info> ne sont donc pas affectes et les proprietes sont conservees.
. La transition T-perte efface le contenu du paquet sans modifier le numero d'emplacement. Le paquet est donc vide et n'intervient plus dans le theoreme.
. La transition T-resynchro remplace l'information contenue par un paquet de reinitialisation sans modifier le numero d'emplacement et ainsi n'intervient plus dans le theoreme.

Theoreme 3 : Ce theoreme sera verifie pour chaque transition du modele.

. La transition T-entree provoque l'apparition d'un nouveau paquet de donnee ayant pour numero de donnee la valeur du compteur P-emetteur.
Le paquet insere peut etre le suivant d'un autre paquet, qui etait, avant le franchissement de la transition, le premier paquet de donnee de la file. Donc il verifiait le Lemme 5 : "toute desynchronisation au debut de la file est detectee".
Jointes au lemme 1, les proprietes du lemme 5 sont applicables a ces deux paquets et permettent alors de verifier le theoreme 3.

. La transition T-transit provoque la propagation des paquets en les echangeant avec des emplacements vides.
Etant donne que :
- La transition T-transit ne modifie pas les informations contenues par les paquets se propageant dans les emplacements, et ne modifie pas l'etat de ces emplacements;
- Les emplacements vides n'interviennent pas dans le libelle du theoreme 3;
- Chaque numero d'emplacement identifie de maniere unique son uplet porteur (lemme 4);

On conclue que la transition T-transit conserve le theoreme 3.

. Les transitions T-sortie et T-perte suppriment les paquets en les echangeant par des emplacements vides de meme numero. Comme ces emplacements vides n'interviennent pas dans le theoreme 3, il conserve sa veracite.

. Le franchissement de la transition T-desynchro fait passer un emplacement de chaque voie de transmission dans l'etat desynchronise. De ce fait, cet emplacement verifie la proposition T3.2 (toute desynchronisation est detectee) du theoreme 3.

. Le franchissement de la transition T-resynchro remplace pour chaque voie de transmission un emplacement vide en etat desynchronise par un emplacement contenant un paquet de reinitialisation en etat synchronise. Ces emplacements verifiaient la proposition T3.2 du theoreme 3 (detection des desynchronisations) et ils verifient, apres le declenchement de la transition, la proposition T3.1 (toute desynchronisation est corrigee).

5.22 Adequation fonctionnelle

Nous allons montrer que l'ensemble des theoremes joints aux lemmes donnes ci-dessus, induisent les quatre proprietes definies par la specification du service Reseau.

. La propriete Ø est directement obtenue par la demonstration du theoreme Ø sur la vivacite du modele.

. La propriete 1 est issue de la preuve du theoreme 1.

. La propriete 2 est apportee par la preuve du theoreme 2.

. La propriete 3 caracterise le comportement du service Reseau face aux desynchronisations. Nous prouvons, a l'aide des lemmes 2, 3 et des trois derniers theoremes que le modele est conforme a sa specification.

Nous savons que toute phase de desynchronisation est concretisee par la perte de paquet circulant sur le Reseau.

D'apres le Lemme 2, la transition T-perte n'est franchissable que s'il existe un emplacement en etat de desynchronisation.

D'apres le Lemme 3, la phase de resynchronisation (transition T-Resynchro) insere un paquet de reinitialisation dans chacune des voies de communication.

Le theoreme 3 confirme que ces insertions sont effectuees apres toute desynchronisation.

L'ordre relatif des paquets est conserve jusqu'a leur delivrance a l'extremite receptrice (theoreme 1), et ce, sans duplication (theoreme 2).

5.3 Conclusion

Nous avons propose, dans cet article, un modele des services de la couche Reseau en phase de transfert de donnees.

Nous avons prouve que ce modele fonctionnel possede l'ensemble des proprietes definies par la norme.

Un tel modele peut desormais etre integre a la modelisation du protocole de la couche Transport.

Les assertions relatives au modele de la couche Reseau sont conservees dans le modele de la couche Transport. Elles pourront etre utilisees dans la demonstration des proprietes de la couche Transport.

6. CONCLUSION

Nous avons presente une methodologie permettant de definir un modele fonctionnel du comportement d'un ensemble de processus.
La methodologie developpee comprend les trois phases suivantes:

- Specification des proprietes caracterisant la fonctionnalite externe des processus etudies.
- Modelisation de ces fonctionnalites a l'aide de reseaux de Petri.
- Validation formelle du modele par rapport aux proprietes definies dans la premiere phase.

Le modele fonctionnel ainsi obtenu est minimal, car il regroupe, sous une forme tres reduite, les proprietes exigees.

Il est equivalent fonctionnellement a un modele decrivant l'ensemble des mecanismes internes et respectant les proprietes du service.

Il est integrable a tout modele ulterieur utilisant les fonctions definies par les proprietes.

Cette demarche originale de validation fonctionnelle d'un modele peut etre generalisee a la caracterisation de tout milieu d'execution hierarchisee.

BIBLIOGRAPHIE

Berthelot 81
G.Berthelot,R.Terrat 'Petri Nets theory for correctness of protocols' IEEE Trans. on Comm. Vol COM 30, no 12.

Berthelot 83
G.Berthelot 'Transformation et analyse de R.d.P : application aux protocoles'. These d'etat - Univ. PARIS VI - Juin 83

G.W Brams
G.W.Brams 'Reseau de Petri: Theorie et Pratique' Tome I et II ed MASSON, 82

Diaz
M.Diaz 'Modelling and analysis of communication and cooperation protocols using Petri Nets based models' Computer networks, Vol no 6, Dec 82

ECMA
' Standard ECMA Transport Protocol' ECMA/TC24/80/16

ISO
' Reference Model on Open Systems Interconnection' ISO/TC97/SC16/N227, juin 79

Transpac
' Transpac caracteristique technique d'utilisation des services - STUR -', oct 79

Transport
' Transport Norme ' ISO/Dp.8073

Zimmermann
M.Zimmermann 'OSI reference model- the ISO model of architecture for Open Systems Interconnection' IEEE Trans on Comm. Vol COM 28, Avril 80