La REQUISITION de CREDIT

du protocole TRANSPORT

Modélisation et Validation

Bernard COUSIN
Laboratoire MASI - CNRS UA 818
Tour 65-66 bureau 201
Université Pierre et Marie CURIE
4, place JUSSIEU
75252 PARIS cedex 05

RESUME

Le protocole de la couche Transport est le seul, parmi l'ensemble des protocoles normalisés, à autoriser le **réquisition de crédit**. Le crédit est le nombre de messages que l'émetteur est autorisé à envoyer par anticipation (sans recevoir préalablement d'acquittement). Ce nombre est établi dynamiquement suivant les possibilités de traitement du récepteur. La réquisition, c'est la décision unilatérale du récepteur de diminuer la valeur du crédit, alors qu'il l'avait auparavant accordé. Cette décision peut éminemment prêter à confusion vis-à-vis de l'émetteur.

Notre modèlisation se focalise tout particulièrement sur la description de la gestion du contrôle de flux durant la phase de transfert de données de la couche Transport. Nous prenons garde toutefois de toujours conserver au modèle l'ensemble des stratégies d'émission et de reception possibles, telles qu'elles sont définies par les spécifications du Protocole de la couche Transport.

En utilisant l'ensemble des techniques de preuves développées pour les réseaux de Petri à prédicats, nous validons le modèle, en prouvant qu'il offre bien les fonctionnalités attendues par la définition du Service de la couche Transport.

PLAN

ABSTRACT

The protocol of the Transport layer is the only one, among all the normalized protocol, which manages the reducing credit. The credit mechanism is defined allowing the reciver to inform the sender of the exact number of messages it is willing to receive. This number is computing according to the receiver treatment possibilities. The reduction is a unilateral decision of the reciver, whereas the credit was granted before, so it is eminently dangerous.

We are particularly interested by the modelizing of the flow control during the data transfert of the Transport layer. However, the model makes all the sending and receiving strategies allowed, according to the protocol specification of the Transport layer.

Using all the proof technics, we validate our model, proving that it gives the functionnality awaited by the service of the Transport layer.

1.INTRODUCTION

Dans le cadre de la norme internationnale pour l'Interconnexion des Systèmes Ouverts [Zimmerman 80], la classe 3 de la couche Transport [ISO 8072, ISO 8073] présente un mécanisme original : le contrôle de flux avec réquisition de crédit. Ce mécanisme permet de gérer le contrôle de flux des messages de manière très souple, il autorise la réquisition du crédit uni-latéralement par le récepteur, après qu'il l'ait déjà accordé. Un tel phénomène peut facilement occasionner un comportement incohérent.

C'est pourquoi, nous allons, après l'établissement d'un modèle conforme aux spécifications de la norme, nous attacher à valider le protocole. Le mécanisme du contrôle de flux n'entrant en jeu qu'en phase de transfert de données, nous y limiterons notre étude. Les phases de connexion et de déconnexion peuvent être considérées comme indépendantes. Elles présentent moins de difficultés, et ont déjà fait l'objet d'études [Berthelot 83].

Toutefois, toute modélisation de processus nécessite une modélisation préalable de son milieu d'éxécution. Dans le cas d'une modélisation de protocole de communication, le milieu est representé par le service de la couche sous-jacente, appelée Réseau [ISO 8348]. Du fait de la décomposition en couches des protocoles de télé-communication, le protocole d'une couche n'utilise que le service de la couche immédiatement inférieure. Nous avons déjà eu l'occasion d'étudier et de modéliser un tel service [Cousin 87a].

Notre sous-modèle aura le comportement suivant:
- Tout message lui étant transmis, après un délai de transmission, est, soit délivré séquentiellement à l'entité réceptrice, soit perdu.

Nous allons dans un premier temps décrire le protocole de la couche Transport, afin d'en tirer les propriétés caractéristiques. Notre description du protocole de Transport se focalise sur les problèmes essentiels de la phase de transfert des données de la couche Transport et notamment la gestion de la fenêtre avec réquisition de crédit.

Nous allons, ensuite, construire un modèle du protocole de Transport en phase de transfert de données, en lui adjoignant au fur à mesure l'ensemble de ses fonctionnalités. Ce modèle utilise les réseaux de Pétri à prédicats (RdPàP) [Genrich 79, Jensen 83], qui ont déjà prouvé leurs adéquation à la modèlisation et la validation de protocole de communication [Ayache85, Sunshine 82]. Ce modèle intègre un ensemble de fonctionnalités jamais encore modélisées.

Enfin, nous validons le modèle final pour établir l'adéquation du protocole avec son service. Pour ce faire, nous analysons le modèle pour obtenir les propriétés du protocole et mettre en évidence leurs concordances avec les propriétés attendues par la couche supérieure (niveau Session).

2. Le PROTOCOLE de la COUCHE TRANSPORT

2.1 La Fenêtre

Le mécanisme de la fenêtre, très utilisé par la plupart des protocoles de communication [Bochmann 77], possède trois avantages:
(1) Son émission par anticipation permet d'utiliser pleinement le débit de la connexion;
(2) C'est un moyen sûr et simple d'effectuer un contrôle de flux;
(3) Il autorise la détection et la récupération d'erreur par retransmission.

Les messages sont transmis en les numérotant à partir d'un compteur de l'émetteur s'incrémentant modulo "N." Ce numéro devient l'identité du message, qui permet aux entités Transport de le repérer de manière unique [Steining 76].

Le récepteur acquitte les messages reçus. L'émetteur est autorisé à envoyer "f" messages (f étant appelé largeur de la fenêtre) par anticipation à partir du numéro du dernier message acquitté. Le récepteur peut à tout moment émettre un message d'acquittement comportant le numéro du prochain message attendu. Cet acquittement prouve qu'il a reçu tous les messages de numéro strictement inférieur.

A la réception, un mécanisme de détection d'erreur vérifie la validité des messages reçus (leur syntaxe interne, leur adéquation avec l'état du protocole du récepteur, l'absence de corruption du contenu du message, etc...). Dans le cas où la détection s'avère positive, un rejet pur et simple du message s'opère (parfois, dans le cas où il y a désynchronisation entre le récepteur et l'émetteur, le protocole émet un message de rejet permettant une resynchronisation des deux partenaires).

A l'émission, un mécanisme de reprise d'erreurs provoque la retransmission par l'émetteur des messages situés à l'intérieur de la fenêtre d'émission. Ce mécanisme est déclenché après que le laps de temps, que l'on considère comme maximal pour la transmission d'un message et de son acquittement, soit écoulé, ou qu'un message provoque une resynchronisation. Ces deux mécanismes de détection et de récupération d'erreurs, bien que pris en compte dans la modélisation que nous faisons du protocole, ne sont pas modélisés de manière explicite. Seul l'est leur aspect externe, et ce, pour des raisons de contraintes temporelles difficiles à exprimer dans le modèle en RdPàP, et pour des choix d'implémentations difficiles à faire ou qui restreindraient trop le modèle. Ne pas faire de choix, nous permet d'offrir un modèle qui synthétise l'ensemble des politiques de détection et de récupération d'erreurs.

De même, la norme ne précise pas explicitement la méthode à employer quant à l'ordre de transmission et de retransmission des messages par l'émetteur. On peut comprendre entre les lignes et par simple bon sens, que l'on émette un message dès que possible, et qu'on retransmette la suite de messages dès qu'un message de rejet est reçu. Cela aurait pour conséquence de toujours transmettre les messages suivant un ordre croissant.

Nous n'avons pas choisi cette solution, trop dépendante à notre avis des choix d'implémentation. Nous avons préféré ne pas imposer d'ordre ni de transmission, ni de retransmission. Notre modèle prouve ainsi que le protocole ne depend pas de ces

choix, cependant il est clair qu'un choix judicieux est propice à de bonnes performances, mais ce n'est pas notre propos de les mesurer.

2.2 Le Crédit

On décide d'accorder une fonctionnalité supplémentaire au récepteur, en l'autorisant à déterminer lui-même la largeur de la fenêtre (appelée alors crédit) (Figure 1). Toutefois, le crédit doit rester dans l'intervalle créé par les bornes maximales de la fenêtre ($c \in [0,f]$). Un crédit nul étant un refus de recevoir un message quelconque, un crédit de valeur "c" permet à l'émetteur d'envoyer "c" messages par anticipation (sans attendre l'acquittement des messages prédécesseurs).

La valeur que le récepteur attribue au crédit dépend, principalement des contraintes de traitement et de stockage des messages. Ces contraintes dépendant de l'implémentation du protocole, nous ne pouvons les prendre en compte dans notre modèle. Notre modèle est donc capable d'engendrer et de traiter des variations de crédit, mais les causes de ces variations ne sont pas explicitées. Tout se passe comme si les variations étaient aléatoires. De ce fait, nous prouvons, ainsi, que toute stratégie d'allocation de crédit conserve au protocole un état valide, sans toutefois mesurer ses performances quant au débit réel de cette stratégie.

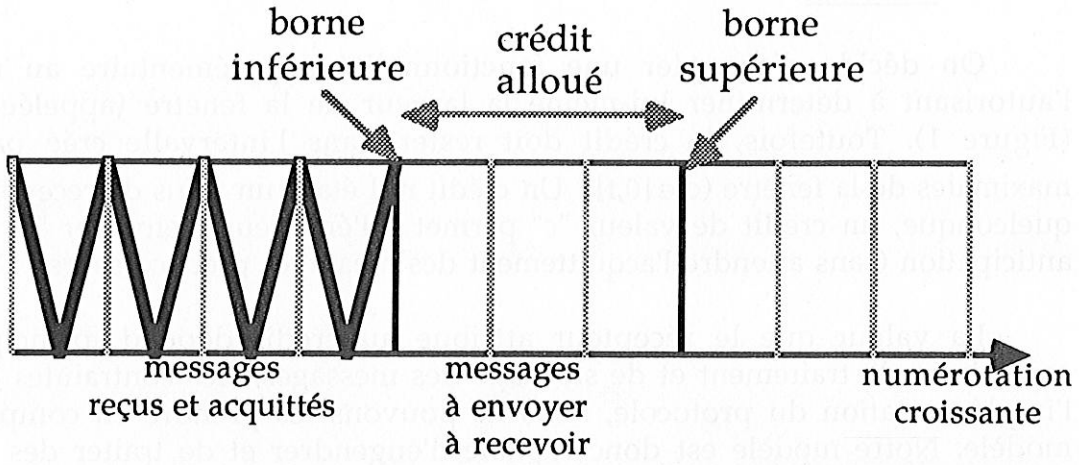
Nous étudions maintenant la fonctionnalité qui permet au récepteur de recevoir les messages dans un ordre différent de leur numérotation. Comme ils doivent être délivrés à la couche Session dans un ordre de numéro strictement croissant, il convient de mémoriser les messages déséquencés, et cependant contenus dans l'intervalle de la fenêtre de réception.

Au premier abord, on pourrait se poser des questions sur l'utilité de cette technique. Le médium ayant la propriété de ne pas déséquentialiser les messages, si l'émetteur les envoie toujours dans l'ordre des numéros croissants, on pourrait s'attendre à les recevoir dans ce même ordre. Cependant c'est sans compter sur les pertes susceptibles de survenir dans le médium. Les pertes pouvant supprimer n'importe quel message transitant dans le réseau, et le mécanisme de retransmission de l'émetteur, peuvent faire apparaître au récepteur une suite quelconque dans l'ordre des messages (Figure 2).

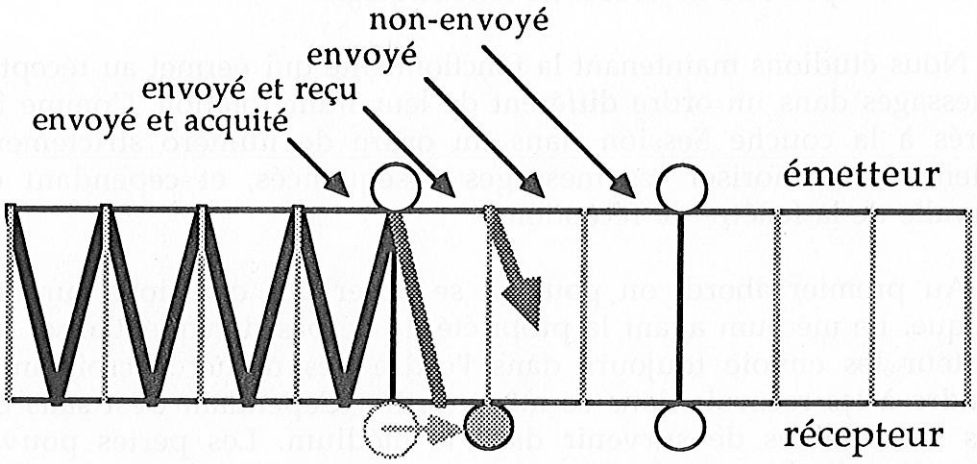
Les numéros des messages attendus par le récepteur peuvent donc se trouver dans trois états :

- non-attendu : le numéro n'est pas dans la fenêtre de réception (Le récepteur ne doit recevoir aucun message comportant un numéro de ce type).
- attendu et non-reçu : le numéro est dans la fenêtre de réception, mais le message correspondant n'a pas été encore reçu (Le récepteur mémorisera tout message comportant un numéro de ce type).
- attendu et reçu : le numéro est dans la fenêtre de réception, et le message a déjà été reçu par le récepteur (Le récepteur considère tout nouveau message reçu comportant un numéro de ce type comme un message dupliqué, de ce fait, il l'ignorera).

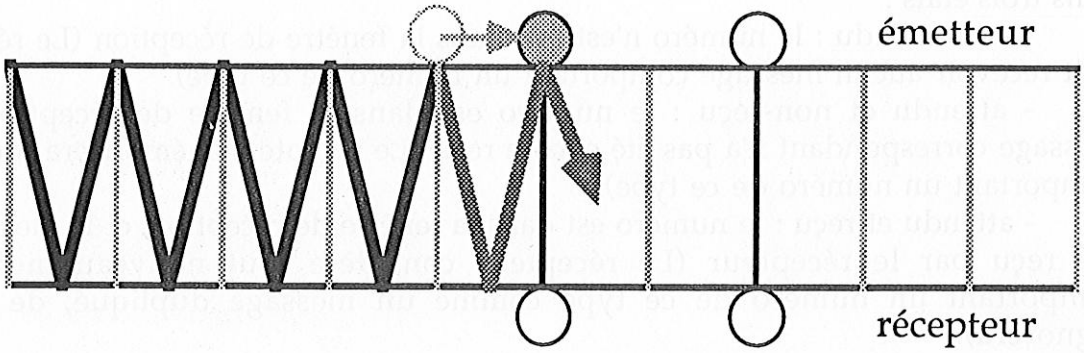
- Le contrôle de flux par crédit - | Figure 1



La fenêtre



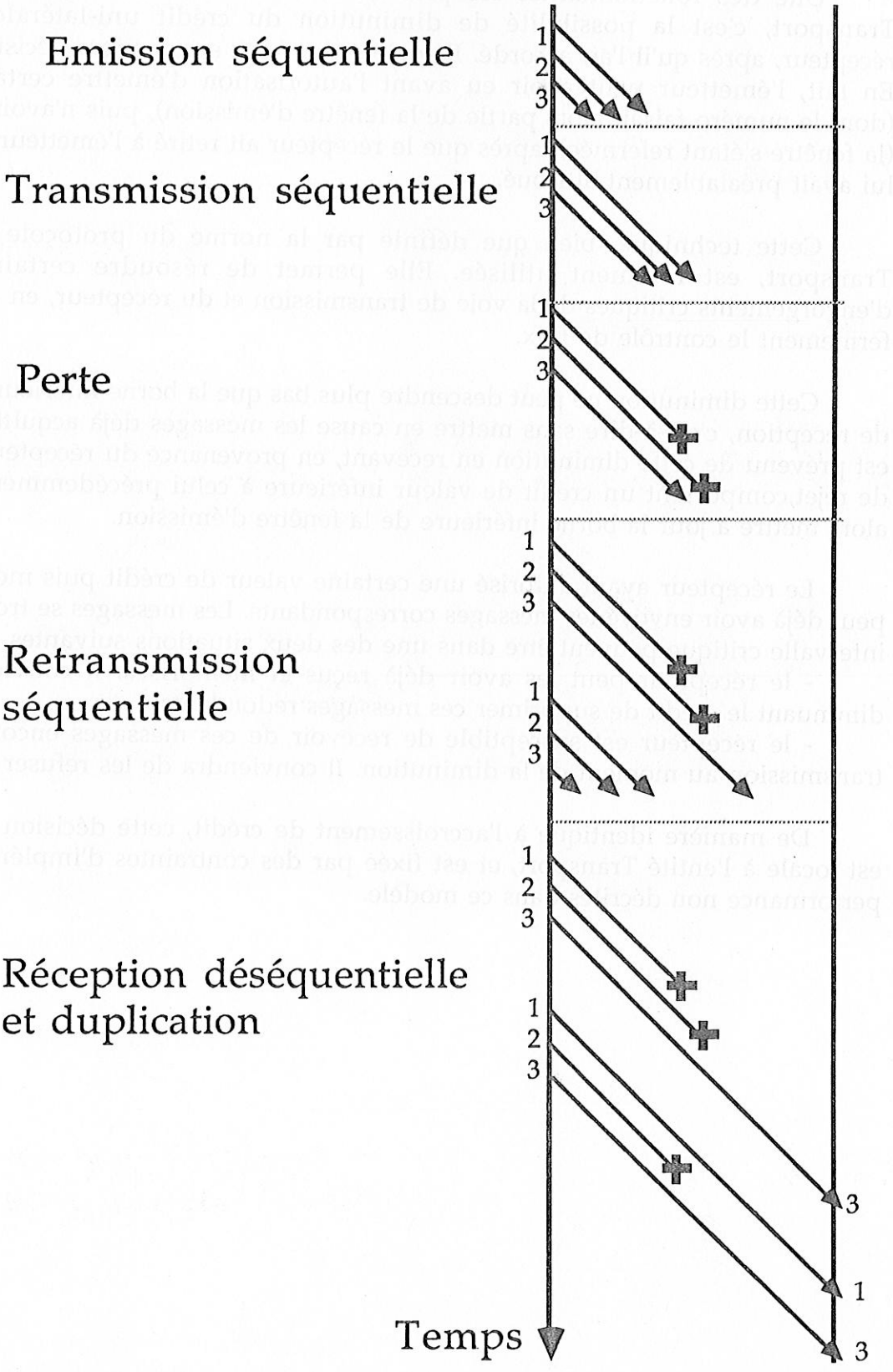
augmentation de la borne inférieure du récepteur



augmentation de la borne inférieure de l'émetteur

Réception désequentielle des messages

Figure 2



3 * SEQUENTIEL + PERTE = DESEQUENTIEL + DUPLICATION

2.3 La Réquisition

Une des fonctionnalités les plus intéressantes de la classe 3 de la couche Transport, c'est la possibilité de diminution du crédit uni-latéralement par le récepteur, après qu'il l'ait accordé. Il revient donc sur sa première décision (Figure 3). En fait, l'émetteur peut avoir eu avant l'autorisation d'émettre certains messages (dont le numéro faisait alors partie de la fenêtre d'émission), puis n'avoir plus ce droit (la fenêtre s'étant refermée), après que le récepteur ait retiré à l'émetteur le crédit qu'il lui avait préalablement attribué.

Cette technique, bien que définie par la norme du protocole de la couche Transport, est rarement utilisée. Elle permet de résoudre certains problèmes d'engorgements critiques de la voie de transmission et du récepteur, en contrôlant très fermement le contrôle de flux.

Cette diminution ne peut descendre plus bas que la borne inférieure de la fenêtre de réception, c'est-à-dire sans mettre en cause les messages déjà acquittés. L'émetteur est prévenu de cette diminution en recevant, en provenance du récepteur, un message de rejet,comportant un crédit de valeur inférieure à celui précédemment reçu. Il doit alors mettre à jour la borne inférieure de la fenêtre d'émission.

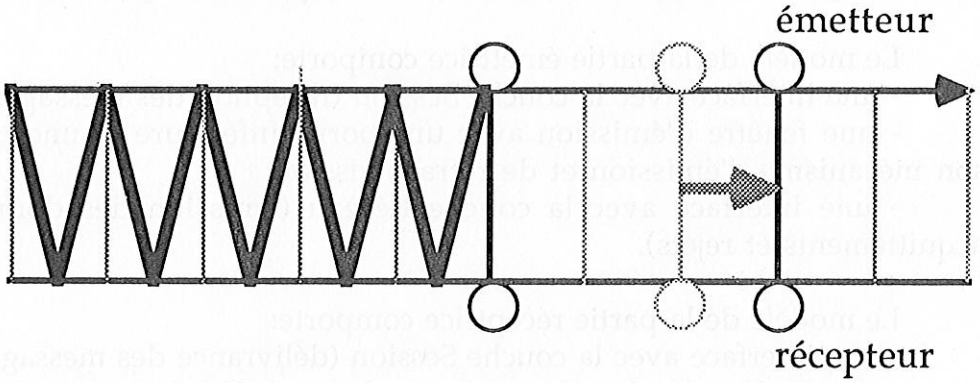
Le récepteur ayant autorisé une certaine valeur de crédit puis moins, l'émetteur peut déjà avoir envoyé les messages correspondants. Les messages se trouvant dans cet intervalle critique peuvent être dans une des deux situations suivantes :

- le récepteur peut les avoir déjà reçus et mémorisés. Il conviendra donc en diminuant le crédit de supprimer ces messages redondants.
- le récepteur est susceptible de recevoir de ces messages encore en cours de transmission au moment de la diminution. Il conviendra de les refuser à la réception.

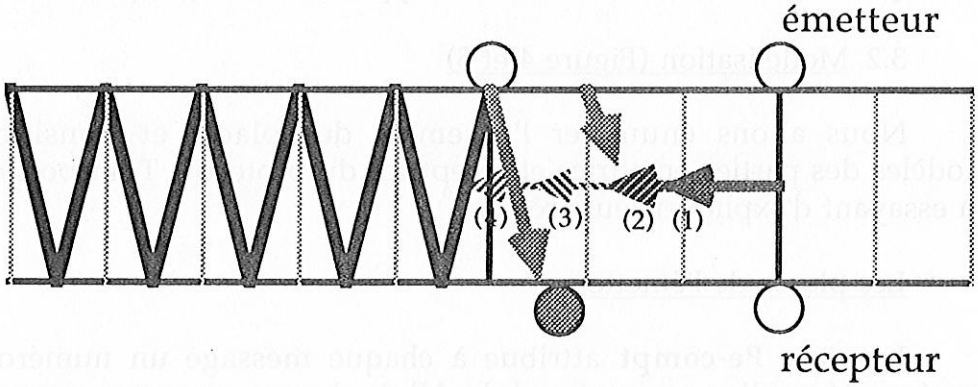
De manière identique à l'accroissement de crédit, cette décision de diminution est locale à l'entité Transport, et est fixée par des contraintes d'implémentation et de performance non décrites dans ce modèle.

Augmentation et Réquisition du crédit | **Figure 3**

Augmentation de la borne supérieure :
l'attribution de crédit



Diminution de la borne supérieure :
la réquisition de crédit !?!



(1) : message ni envoyé,ni reçu, ni acquitté (réduction autorisée);
(2) : message envoyé mais ni reçu, ni acquitté (réduction autorisée);
(3) : message envoyé et reçu mais pas acquitté (réduction interdite);
(4) : message envoyé , reçu et acquitté (réduction interdite);

3. Le MODELE

3.1 Présentation

Nous avons construit graphiquement le modèle de l'émetteur séparement de la partie réceptrice, bien qu'en fait, une connexion de Transport soit une liaison bidirectionnelle (Figure 4 et Figure 5). Ainsi toute entité Transport connectée au réseau est à la fois émettrice et réceptrice. Cependant, l'indépendance de chaque partie autorise notre démarche, ce qui facilite la conception, la lecture et la preuve du modèle. Pour construire un modèle dans sa totalité, il faut pour chaque entité communiquante: une partie émettrice, et une partie réceptrice.

Le modèle de la partie émettrice comporte:
- une interface avec la couche Session (réception des messages);
- une fenêtre d'émission avec une borne inférieure et une borne supérieure, et son mécanisme d'émission et de retransmission;
- une interface avec la couche Réseau (émission des données, réception des acquittements et rejets).

Le modèle de la partie réceptrice comporte:
- une interface avec la couche Session (délivrance des messages);
- une fenêtre de réception avec une borne inférieure et une borne supé- rieure, et son mécanisme d'acquittement;
- une interface avec la couche Réseau (réception des données, émission des acquittements et des rejets).

Tout au cours de notre modélisation nous avons tenté de faire apparaître le maximum de parallélisme de traitement, entre les parties émettrice et réceptrice, mais aussi, pour le traitement des différents types de messages puis de l'action engendrée.

3.2. Modélisation (Figure 4 et 5)

Nous allons énumérer l'ensemble des places et transitions composant les modèles des parties émettrice et réceptrice du protocole Transport et celle du médium, en essayant d'expliciter leurs rôles.

Les places de l'émetteur :

La place **Pe-compt** attribue à chaque message un numéro. Le compteur est incrémenté cycliquement (modulo N) à chaque message provenant de la couche Session.

La place **Pe-inf** est la borne inférieure de la fenêtre d'émission. Tous les messages de numéro strictement inférieur ont déjà été transmis et acquittés (nota: les numéros supérieurs sont soit à émettre, soit n'ont pas encore été attribués à un message).

La place **Pe-sup** modélise la borne supérieure de la fenêtre d'émission. Sa marque contient la valeur du crédit alloué par le récepteur. Tous les messages de numéro strictement supérieur n'ont pas encore été attribués à un message (nota: les

numéros inférieurs sont soit à émettre, soit ont déjà été transmis et acquittés).

La place **Pe-fenêtre** contient l'ensemble des messages contenus dans la fenêtre d'émission. Cette place modélise l'organe de stockage pour la ré-émission ultérieure des messages.

Les transitions de l'émetteur :

La transition **Te-Session** est franchie quand l'interface Session délivre à l'entité Transport un message à émettre (T-SDU-data.req). Ce message est placé dans l'organe de stockage (Pe-fenêtre), si celui-ci n'est pas surchargé. Le modèle maintient au maximun "f" messages, largeur maximale de la fenêtre d'emission.

La transition **Te-donnée** est franchie à chaque émission d'un message vers le récepteur. L'envoi de message est conditionné par le crédit d'émission. Le numéro du message à émettre doit être dans l'intervalle de la fenêtre d'émission (Pe-sup, Pe-inf). On insère, alors, le message dans le sous-modèle Réseau. Cependant la copie du message est conservée en vue d'une retransmission ultérieure.

La transition **Te-acq** est franchie par les messages d'acquittement issus du récepteur. L'arrivée de l'acquittement provoque la mise à jour des bornes inférieure et supérieure de la fenêtre d'émission (Pe-inf et Pe-sup), à l'aide du numéro d'acquittement et du crédit que contient le message d'acquittement. Le message d'acquittement est valide, si le numéro d'acquittement du message est compris dans les bornes de la fenêtre d'émission, et si le crédit du message est non-décroissant et inférieur à la largeur maximale de la fenêtre.

La transition **Te-rej** est franchie par les messages de rejet issus du récepteur. Le franchissement met à jour les bornes inférieure et supérieure de la fenêtre d'émission (Pe-inf et Pe-sup), à l'aide du numéro d'acquittement et du crédit que contient le message de rejet. Le message de rejet est valide, si le numéro d'acquittement du message est compris dans les bornes de la fenêtre d'émission, et si le crédit du message est inférieur à la largeur maximale de la fenêtre. On remarque donc que le crédit peut ici diminuer.

La transition **Te-erreur** est franchie par tous les messages ne pouvant pas franchir les autres transitions (Te-acq et Te-rej). Elle permet de recevoir du médium Réseau des paquets, qui ne sont ni des acquittements, ni des rejets (tous les paquets erronés).

Les places du récepteur :

La place **Pr-inf** mémorise la borne inférieure de la fenêtre de réception.

La place **Pr-sup** représente la borne supérieure de la fenêtre de réception.

La place **Pr-état** mémorise l'état de la connexion du protocole de Transport. L'état du protocole devient érroné sur tout évènement déclenché par la réception d'un message inattendu ou incorrect. Le protocole devra alors émettre un message de rejet, afin de se resynchroniser avec l'émetteur (Tr-rej).

La place **Pr-attendu** mémorise l'ensemble des numéros des messages attendus (dont les numéros sont dans l'intervalle de la fenêtre de réception), mais non encore reçus par le récepteur.

La place **Pr-recu** mémorise l'ensemble des numéros des messages attendus et reçus par le récepteur, mais non encore délivrés à l'entité de la couche supérieure Session. On note que l'union des numéros des uplets des deux places (Pr-reçu et Pr-attendu) représentent exactement l'intervalle qui s'inscrit entre les bornes inférieure et supérieure de la fenêtre de réception.

Les transitions du récepteur :

La transition **Tr-donnée** modélise la réception d'un message en provonance de l'émetteur via le service Réseau. Le message franchit effectivement la transition, uniquement si son numéro appartient bien à l'intervalle de la fenêtre de réception, et s'il n'a pas déjà été reçu (son numéro est contenu par la place Pr-attendu). Le message est mémorisé alors, pour savoir qu'on l'a reçu et pour le delivrer plus tard à la couche Session.

La transition **Tr-duplic** permet de traiter les messages appartenant à l'intervalle de la fenêtre de réception dont on s'aperçoit qu'ils ont déjà été reçus. Cependant ces messages n'ont pas encore été délivrés à la couche Session. Ces doubles sont détruits à la réception. La transition Tr-duplic empêche une double mémorisation des messages reçus en vérifiant leur présence dans l'organe de stockage du récepteur (Pr-recu).

La transition **Tr-hors** représente la réception et la détection d'un message déjà reçu ou non-attendu (hors des bornes de la fenêtre). Le message est supprimé.

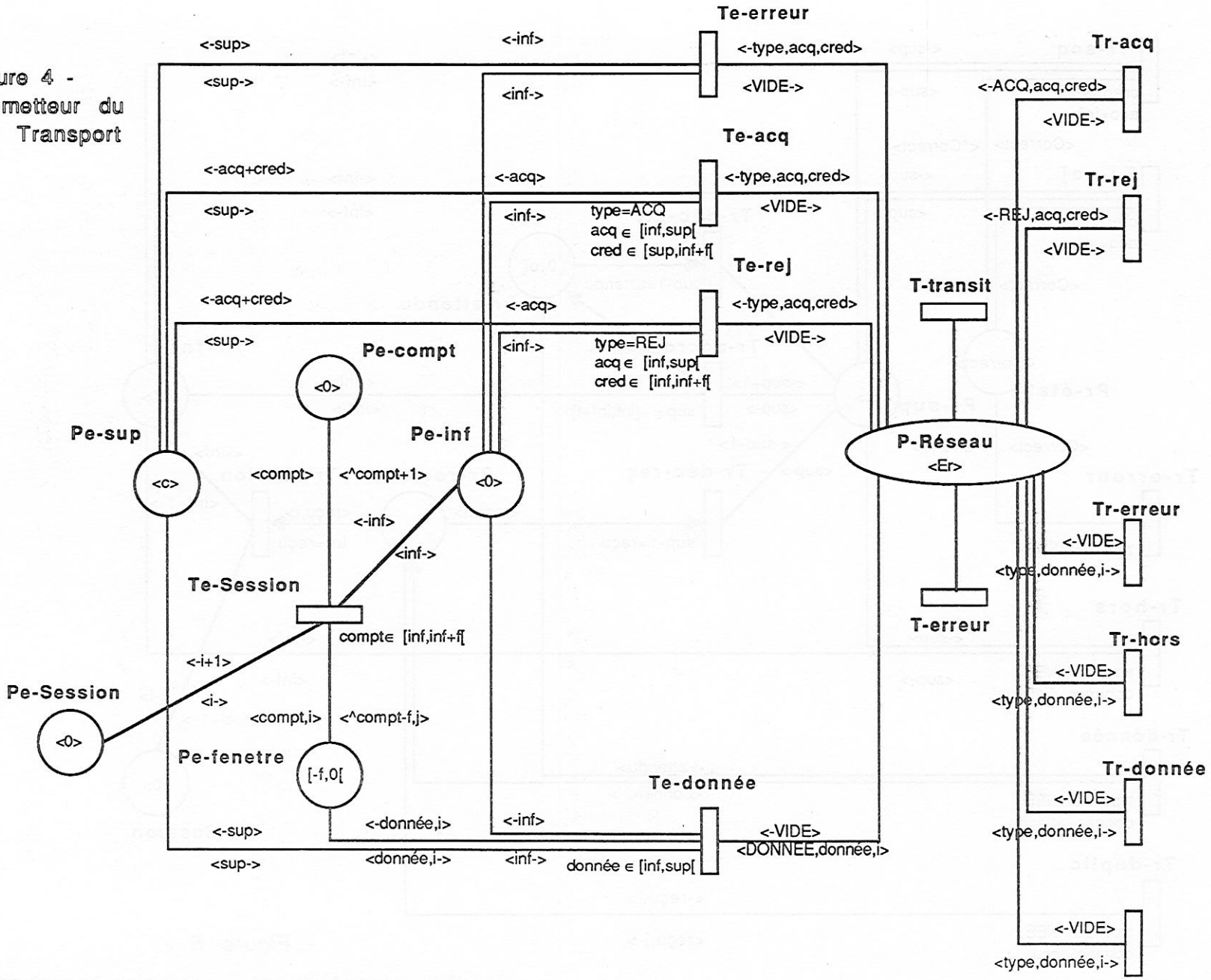
La transition **Tr-erreur** est franchie pour chaque message reçu, erroné inattendu, ou provenant d'une désynchronisation. Elle joue un rôle symétrique à la transition Te-erreur en acceptant du médium tous les messages inattendus ou erronés.

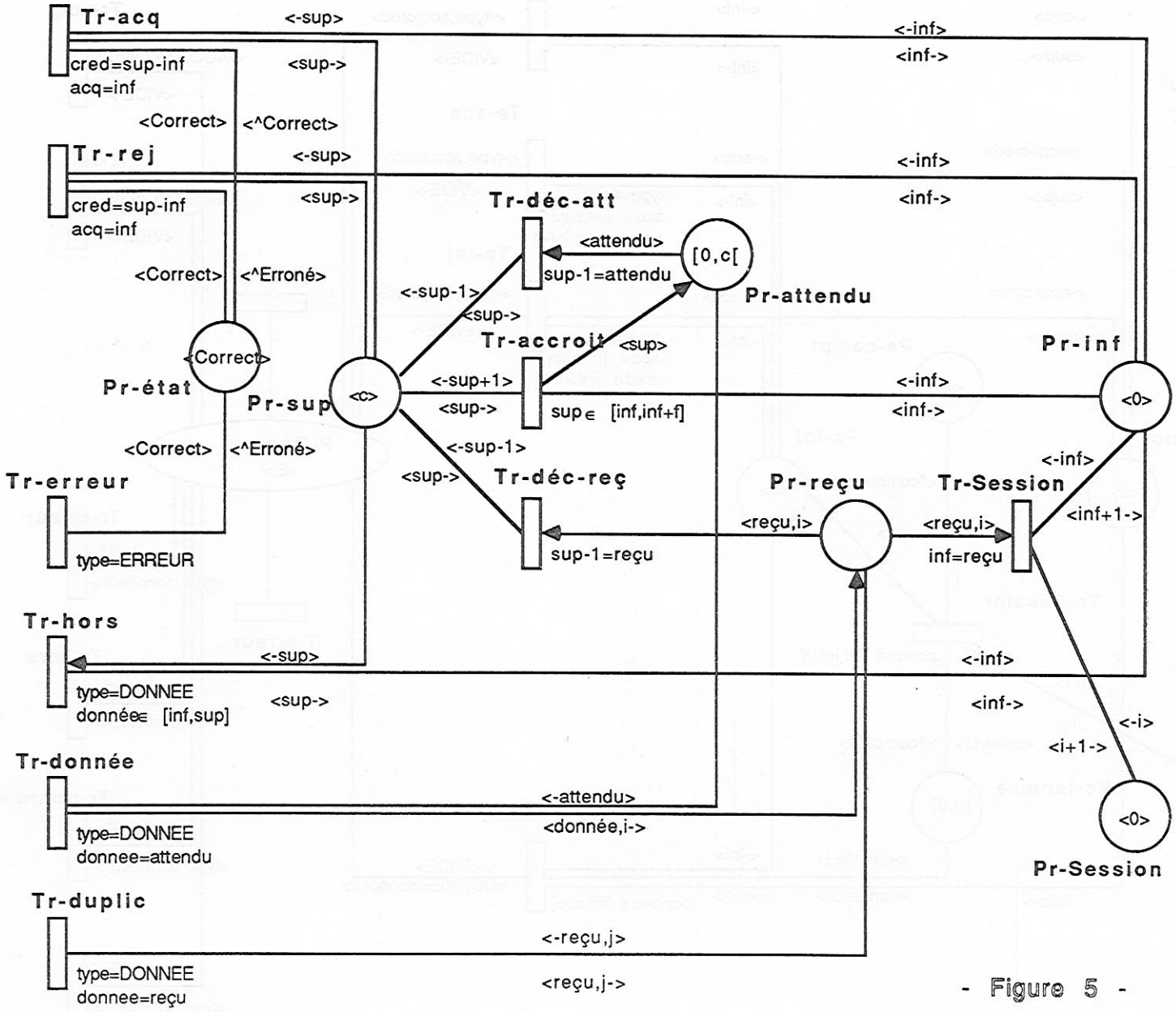
La transition **Tr-acq** délivre, pour transport par le Réseau, les messages d'acquittement. Le numéro du message acquitté et le crédit sont basés sur la valeur des bornes inférieure et supérieure de la fenêtre de réception (Pr-inf et Pr-sup).

La transition **Tr-rej** délivre, pour transport par le Réseau, des messages de rejet. Ces messages sont motivés par l'état erroné du protocole. Cette émission permet aux entités de revenir dans un état correct, et de le faire savoir à l'autre entité. Le numéro du message acquitté et le crédit sont basés sur la valeur des bornes inférieure et supérieure de la fenêtre de réception (Pr-inf et Pr-sup).

La transition **Tr-accroit** provoque l'accroissement de la borne supérieure de la fenêtre de réception (Pr-sup), tout en restant dans la limite maximale (Pr-inf + "f"). Cette augmentation de crédit provoque l'apparition dans la place Pr-attendu d'une marque contenant le nouveau numéro autorisé à la réception. L'accroissement du crédit est une décision locale, qui se fait d' une seule unité à chaque fois. Un accroissement de plusieurs unités de crédit s'effectue par un franchissement multiple de la transition Tr-accroit.

- Figure 4 -
Modèle émetteur du protocole Transport





- Figure 5 -

Modèle récepteur du protocole Transport

La transition **Tr-déc-reç** permet au récepteur de diminuer le crédit accordé à l'émetteur. Ayant déjà reçu les messages dont il veut refuser la transmission, il doit les supprimer de l'organe de stockage (Pr-reçu). De même supprimant la marque associée à ce message, il oublie sa réception. Le numéro associé sort de la fenêtre de réception.

La transition **Tr-déc-att** retire le droit de recevoir les messages hors crédit, en supprimant les marques ayant valeur hors de la fenêtre de réception nouvellement diminuée.

La transition **Tr-Session** délivre les messages à la couche supérieure Session(T-SDU-data.ind). Elle assure que la livraison respecte la séquentialité voulue (tout message transitant avant tout autre message au travers de l'interface émetteur Session-Transport, franchira de même l'interface récepteur Transport-Session avant ces dits messages). Chaque accroissement de la borne inférieure par délivrance d'un message à la couche Session, provoque le retrait de la place Pr-reçu du jeton de même numéro.

Le médium :

La place **P-réseau** modélise le circuit virtuel reliant les deux entités Transport communiquantes.

La transition **T-transit** permet de faire progresser les messages de l'émetteur vers le récepteur.

La transition **T-erreur** provoque des pertes aléatoires de messages transitant sur le réseau.

4. VALIDATION du protocole de Transport

Nous n'avons pas ici la place pour exhiber la preuve dans sa totalité, mais vous trouverez celle-çi dans [Cousin 87b].

4.1 La Vivacité

Nous avons prouvé la vivacité de notre modèle, preuve le protocole reste dans un état cohérent. Cette preuve est établie en deux temps : dans un premier temps, nous prouvons que le modèle possède un ensemble d'états d'accueil ET; puis nous prouvons, dans un deuxième temps, que le modèle est quasi-vivant à partir d'un état de ET.

En fait, nous définissons ET comme l'ensemble des états de repos du modèle: les fenêtres d'émission et de réception sont vides, ainsi que les organes de stockage.

Nous utilisons une norme D, qui permet de prouver que quelque soit l'état atteint par le modèle du protocole Transport, il est toujours possible de trouver une séquence de franchissements de transitions qui permet d'arriver à un des états de

l'ensemble d'accueil ET.

On sait que D est une norme pour ET [Keller 76] ,si et seulement si pour tout marquage M appartenant à l'ensemble A des marquages accessibles du modèle Transport :

- si le marquage M est élément de l'ensemble ET ,il faut alors que la norme de M soit nulle ($\forall M \in ET : D(M) = 0$) ;
- sinon il existe une séquence de franchissement qui permet de faire décroitre la norme ($\forall M \notin ET : \exists s \in T^*$ tel que $D(M) > D(M(s>)$) .

Nous définissons la norme D ainsi :

$\forall M \in A$, D(M) = ((M(Pr-inf)-M(Pe-inf)) + (M(Pr-sup)-M(Pe-sup)) + (M(Pr-sup)-M(Pr-inf)) + (M(Pe-compt)-M(Pe-inf)) + M(Pe-état)) * (M(Pe-compt))

La preuve, que D est bien une norme, est facilitée par notre corpus d'invariants et par les prédicats associés à chaque transition, qui restreignent l'ensemble des états atteignables et permettent d'obtenir aisémment une séquence de transition qui diminue la norme pour chaque état accessible du modèle.

La preuve, que notre modèle est quasi-vivant à partir de son état d'accueil ET, est apportée en exhibant une séquence de franchissements qui permet de déclencher toutes les transitions du modèle du protocole Transport à partir du marquage caractérisant l'état d'accueil.

4.2 La Séquentialité

Nous prouvons que le protocole Transport délivre séquentiellement au récepteur les messages qui lui sont transmis par l'émetteur.

Pour faciliter cette preuve nous ajoutons au modèle du protocole Transport deux places. La première place Pe-Session modélise la couche supérieure (appelé Session) côté émetteur, elle est connectée à la transition d'interface Te-Session. La deuxième place Pr-Session modélise la couche Session côté récepteur, elle est connectée à la transition Tr-Session. Ces deux places fonctionnent comme des compteurs constamment croissants, identifiant de manière unique les messages émis ou reçus franchissant Transport/Session.

C'est ainsi que les marques modélisant les messages véhiculés par le protocole Transport doivent comporter un champ supplémentaire. Ce champ modélise le contenu du message et permet de l'identifier, sa valeur lui est attribuée au franchissement de la transition Te-Session.

Ces adjonctions ne modifient pas le comportement du modèle initial, car elles respectent les conditions d'équivalence de comportement définies dans [Cousin 87b], aucun prédicat du réseau initial ne référence le nouveau champ. De plus les deux places Pe-Session et Pr-Session sont non-contraignantes pour le réseau.

Nous définissons la propriété de séquentialité de la manière suivante: Il faut que l'ordre de réception des messages à l'interface récepteur Transport/Session soit l'ordre

d'émission de ces mêmes messages à l'interface émetteurSession/Transport, sans omission ni duplication. Dans le modèle, la transition Te-Session (modélisant l'interface Session/Transport) attribue un numéro croissant à chaque message qui la franchit, il est alors facile de contrôler que les messages franchissant la transition Tr-Session (modélisant l'interface Transport/Session) ont un numéro qui suit cet ordre strictement croissant.

Nous exprimons la propriété ainsi:
si M(Tr-Session>M' et M'=M-u+u' alors mess(u)=M(Pr-Session).

Il est important de remarquer que cette propriété ne prouve pas que tous les messages sont délivrés, mais elle prouve que si un message (une marque) est délivré au destinataire (franchit la transition Tr-Session) alors il respecte la contrainte de séquentialité.

5. CONCLUSION

Après avoir étudié les comportements du protocole de la couche Transport, nous venons de construire un modèle de la phase de transfert du protocole de la couche Transport classe 3, en nous intéressant tout particulièrement à la gestion du contrôle de flux avec réquisition de crédit.

Ce modèle écrit à l'aide des Réseaux de Pétri à prédicats, utilise le service rendu par la couche Réseau.

Après une analyse du modèle, nous prouvons que le phénomène particulier de réquisition de crédit (en dehors de toute tentative de mesure de performance), géré comme la norme le spécifie, n'entraine aucun blocage ni dis-fonctionnement du protocole Transport, et permet de rendre le service nécessaire à la couche Session.

Nous avons employé de nombreuses techniques pour établir nos preuves, la plupart de ces techniques étant déjà exploités sur les RdP ordinaires, cependant nous les avons employées intensivement sur un modèle décrit par RdP à prédicats:

La technique assertionnelle consiste à démontrer la conservation d'une assertion dans tous les états accessibles du modèle, l'assertion devient un invariant du modèle. Cette technique est pratique, à défaut d'obtention automatique (comme pour les invariants linéaires), si l'on a une bonne connaissance de la sémantique du modèle (c'était le cas ici, nous avons conçu le modèle) et un petit nombre de transitions (dans notre cas, une dizaine). La preuve est rendue plus aisée, si les prédicats des transitions sont contraignants et si l'on dispose d'autres invariants.

Nous avons employé des propriétés et des résultats démontrés par ailleurs, notamment pour établir la vivacité : Extension de la notion d'état d'accueil pour les RdPàP (ensemble d'accueil); Application d'une norme pour atteindre l'ensemble d'accueil.

Nous avons utilisé une technique plus directe ou mathématique, pour démontrer la conservation de la séquentialité. La combinaison de l'ensemble des invariants précédemment établis, les propriétés de fonctionnement du modèle Réseau, et les propriétés des modulos ont permis d'obtenir cette dernière propriété.

Nous avons, aussi, employé notre propre résultat (montrant l'équivalence de comportement de deux réseaux issus l'un de l'autre par modification de la structure interne d'une classe d'uplet) pour transporter dans le médium de manière transparente les messages du protocole Transport.

Dans une démarche accumulative, les résultats (modèles et propriétés) pourront ultérieurement servir à l'établissement du protocole de couche supérieure (niveau Session).

BIBLIOGRAPHIE

[Ayache 85] J.M.Ayache, J.P.Courtiat, M.Diaz, G.Juanole 'Utilisation des Réseaux de Pétri pour la modélisation et la validation de protocoles' , T.S.I vol 4 n° 1 - 1985 .

[Berthelot 83] G.Berthelot, "Transformation et analyse de Réseaux de Pétri, application aux protocoles" , Thèse d'état - Univ. PARIS VI , Juin 1983.

[Bochmann 77] C.V.Bochmann, R.J.Chung, "A formalized description of HDLC classe of procedure" , I.E.E.E nat. Telecom. , Los Angeles , 1977.

[Cousin 87a] B.Cousin, P.Estraillier, "Etude de la resynchronisation d'un protocole de communication", TSI vol 6 n°3, 1987.

[Cousin 87b] B.Cousin, "Méthodologie de validation des systèmes structurés en couches, application au protocole Transport", Thèse de l'Université de PARIS VI, 1987.

[Genrich 79] H.J.Genrich, K.Lautenbach ,"The analysis of distributed systems by means of predicate/transitions nets" Semantics and concurrent computation, Lect notes in Computer Sciences n° 70, Springer Verlag , 1979.

[ISO 8072] " O.S.I - Transport Service Definition ", (DIS 8072) , 1984.

[ISO 8073] "O.S.I - Transport Protocol Specification", (DIS 8073), 1984.

[ISO 8348] " Network Service Definition ", (DIS 8348) , 1984.

[Jensen 83] K.Jensen, "High-Level Petri Nets" , Application and Theory of Petri Nets, Springer-Verlag - 1983 .

[Keller 76] R.M.Keller, "Formal Verification of Parrallel Programs" , Communication of ACM vol 19 n° 7 , pp371-384 , 1976.

[Stenning 76] N.V Stenning, "A data transfer protocol", Computer Network 1, 99-110, 1976 .

[Sunshine 82] C.A Sunshine "Specification and Verification of communication Protocol" , I.E.E.E Transactions , 1982 .

[Zimmerman 80] M.Zimmermann "The ISO reference model of architecture for Open Systems Interconnection", I.E.E.E Trans on Comm, vol 28 n°4, p425 ,1980.