

Juillet 2005

Mise en place d'un mécanisme de répétition sélective dans le framework ETP

VAN WAMBEKE NICOLAS
LAAS-CNRS
Groupe OLC

Version 1.0

SOMMAIRE

INTRODUCTION	2
I. PRÉSENTATION DU FRAMEWORK ETP	3
A. MOTIVATIONS	3
B. ARCHITECTURE GENERALE	3
C. LE PLAN DE CONTROLE	3
1. LE PLAN DE CONTROLE D'EMISSION (CO)	4
2. LE PLAN DE CONTROLE EN RECEPTION (CI)	4
D. LE PLAN DE GESTION	4
II. MODÉLISATION DU “SELECTIVE REPEAT”	5
A. MOTIVATIONS	5
B. MISE EN OEUVRE	5
1. MÉTHODOLOGIE	5
2. ARCHITECTURE ET ROLES	6
a) Le plan de contrôle d'émission (CO)	7
b) Le plan de contrôle de réception (CI)	8
c) Plan de gestion (MG)	11
C. SIMULATION & TRACES D'EXECUTION	20
1. MISE EN PLACE DE LA SIMULATION	20
2. TRACES D'EXECUTION	21
a) Pertes & déséquencements ponctuels	21
b) Importance des mécanismes temporisés	23
III. INTRODUCTION DU VECTEUR D'ACQUITTEMENT (SACK)	28
A. MODIFICATIONS D'ARCHITECTURE	29
1. AJOUT DU SACK AUX PDUS	29
2. DEFINITION DES MESSAGES DE CONTROLE LOCAUX	30
B. MODIFICATIONS COMPORTEMENTALES	30
1. MODIFICATIONS APPORTES AU PLAN DE CONTROLE.	30
2. MODIFICATIONS APPORTES AU PLAN DE GESTION.	32
a) Identification Initiale	32
b) Emission du vecteur d'acquittement sélectif	33
c) Réception d'un acquittement sélectif	34
3. NOUVEAUX MECANISMES AJOUTES	35
a) Evaluation du RTT	35
b) Contrôle d'erreur agressif	36
BIBLIOGRAPHIE	37
TABLE DES FIGURES	37

Introduction

La modélisation UML (« Unified Modeling Language ») est aujourd'hui devenue une étape inévitable du développement dans le monde du logiciel. Les avantages apportés par ce langage de modélisation permettent une meilleure compréhension des solutions que doit apporter le logiciel ainsi que des mécanismes sous-jacents à la mise en œuvre de ces solutions.

Dans le cadre de la construction de protocoles, il existe depuis quelques années un engouement tout aussi important que dans le monde du logiciel pour les techniques de modélisation. Bien que les besoins soient différents de ceux exprimés pour la modélisation d'un logiciel, la norme UML 2.0 intègre une grande part de mécanismes qui sont utilisables dans le cadre d'une modélisation de protocole comme le sont les signaux, les compteurs ainsi que les « timers ».

De plus, certains outils de modélisation UML intègrent une fonctionnalité de simulation [Telelogic] basée sur la description formelle des mécanismes sous forme de machines à états et diagrammes d'architecture visant à permettre une vérification de comportement ce qui est très utile pour la mise en place de protocoles.

Dans le but d'offrir une garantie d'une certaine Qualité de Service dans les réseaux dont l'architecture, les mécanismes déployés ou les circonstances d'utilisation ne permettent pas l'approche classique de la réservation de ressources, de nouveaux protocoles de transport ont été proposés.

Ces nouveaux protocoles mettent en œuvre des mécanismes dits à ordre et fiabilité partielle. C'est dans cette optique qu'a été proposé en 2001 dans [EXP03] le protocole FFTP (Fully Programmable Transport Protocol) comme une implémentation issue de la logique sous-jacente au « framework » ETP.

Dans une première partie de ce document, les fondements du « framework » ETP seront présentés afin de poser les bases de la description de la mise en place d'un nouveau mécanisme dans ce framework qui sera faite dans une seconde partie. Enfin, l'extension du mécanisme présenté sera envisagée et décrite dans une dernière discussion.

I. Présentation du Framework ETP

A. Motivations

Le framework ETP (Enhanced Transport Protocol) propose un ensemble d'éléments UML/SDL dans lequel l'utilisateur de cet outil peut facilement modéliser les différents services de la logique ETP.

La logique ETP propose une nouvelle approche, non seulement à la conception mais aussi à l'utilisation des protocoles de transport qui s'inscrit dans la logique des protocoles aux services dynamiquement composables. En effet, le but est de fournir à l'utilisateur une série de mécanismes qu'il pourra activer à sa guise et qui lui permettront de disposer, selon les différentes combinaisons, d'un service allant du simple multiplexage fourni par UDP à l'ordre et fiabilité totale de TCP. Il est certain néanmoins que les primitives d'accès à ces services devront être simples afin de permettre une utilisation aisée. Les tâches complexes relatives à l'interdépendance des mécanismes et leur composition ne seraient être non plus laissées à l'utilisateur.

B. Architecture Générale

Au delà de cette approche qui place l'utilisateur directement face aux mécanismes de transport, une approche plus orientée qualité de service est envisagée dans laquelle l'utilisateur spécifie simplement ses besoins à l'entité protocolaire (en utilisant par exemple XQoS décrit dans [EXP03a] et [EXP03]) qui ensuite se charge de la composition des mécanismes au besoin, éventuellement de façon dynamique pendant la communication.

Le framework ETP propose, sur le plan de la conception, un modèle tout objet. D'un point de vue conceptuel, il permet de séparer deux plans:

- Le plan de contrôle, lui même décomposé en deux sous plans. Respectivement, il s'agit du plan d'émission (Control Output) et de réception (Control Input).
- Le plan de gestion (Management)

Pour chacun de ces deux plans, les interfaces sont définies et extensibles à souhait. Il est également possible d'utiliser une série d'objets standards tels des applications émettrices et réceptrices ainsi que des médiums.

De plus, une série d'objets tels buffers, médiums complexes, applications et interfaces ont été modélisés dans le cadre du travail de Master Recherche de M. Armando présenté dans [ARM05].

C. Le plan de contrôle

Chargé de tous les échanges avec les applications ainsi que de tous les transferts de type données (DATA). Les traitements effectués par ce plan sont systématiques et doivent être menés pour chaque paquet entrant ou sortant.

1. Le plan de contrôle d'émission (CO)

En émission, le comportement du plan de contrôle est directement lié aux mécanismes activés. Indépendamment des traitements ultérieurs, la première étape consiste en la réception d'une donnée applicative et à son stockage éventuel dans des buffers. Vient ensuite une série de modules qui ont pour but d'effectuer les traitements relatifs aux services activés pour la communication en cours. Enfin, le traitement se termine par la remise à l'entité de niveau inférieure du PDU.

Bien évidemment, pendant les différentes étapes décrites ci-dessus, une étroite communication avec le plan de gestion est envisagée.

2. Le plan de contrôle en réception (CI)

En réception, le plan de contrôle commence toutes ses exécutions avec la réception d'un PDU remis par l'entité de niveau inférieur et son stockage éventuel dans des buffers. Viennent ensuite les traitements des différents modules optionnels. L'étape finale du traitement du plan de contrôle de réception consiste en la remise des différents ADUs à l'application.

Bien évidemment, pendant les différentes étapes décrites ci-dessus, comme pour l'émission, une étroite communication avec le plan de gestion est envisagée.

D. Le plan de Gestion

Contrairement au plan de contrôle, le plan de gestion n'est pas composé de sous entités. Ceci est justifié par le rôle central de ce plan. En effet, alors que les plans de contrôle se limitent à effectuer une série d'actions simples, le plan de gestion a pour vocation la prise de décision en fonction des événements qui lui sont signalés par le plan de contrôle.

Cette architecture vise à privilégier l'efficacité des traitements usuels vis à vis des algorithmes de prise de décision qui peuvent selon leur complexité prendre un certain temps.

II. Modélisation du “selective repeat”

Parmi les mécanismes de correction d'erreurs de type ARQ, il en existe un qui malgré son important degré de complexité de mise en oeuvre et l'importance des ressources requises à son bon fonctionnement, permet d'obtenir des résultats imbattables en terme d'efficacité.

Le principe du mécanisme est très simple. Dès que l'entité réceptrice détecte un déséquence, celle-ci signale explicitement les numéros des paquets supposés perdus à l'émetteur. C'est lui qui se chargera alors de les retransmettre au plus vite. Pendant ce temps, tous les paquets reçus hors séquence sont bufferisés par l'entité réceptrice. Sur réception des paquets demandés, l'entité réceptrice peut alors remettre tous les ADUs dont la séquence a été reconstituée à l'application et acquitter l'émetteur de cette réception.

A. Motivations

De par l'efficacité et la forte sélectivité du mécanisme, le « selective repeat » est un algorithme de choix dans le cadre d'une provision en qualité de service dans des environnements dont l'architecture, les mécanismes déployés ou les circonstances d'utilisation ne permettent pas l'approche classique de la réservation de ressources.

En effet, dans le cadre d'un flux de données multimédia par exemple, ce mécanisme permet d'avoir un niveau de sélectivité égal au paquet au lieu d'une sélectivité égale à la taille de la fenêtre d'émission obtenue avec le Go-Back-N. De plus, l'avantage du stockage des paquets hors séquence permet d'accroître l'efficacité vis-à-vis d'un mécanisme à acquittement cumulatif classique car l'entité émettrice n'a besoin de retransmettre que les paquets signalés comme perdus et non tous les paquets successifs à une perte, d'où un gain évident en souplesse et en fluidité.

Aussi, dans le cadre d'une garantie partielle de l'ordre et de la fiabilité, ce mécanisme permet non seulement de fixer un taux maximal de pertes admissibles ainsi qu'un délai d'acheminement critique mais il permet également de mettre en place un système de garanties différenciées basées sur la nature du flux véhiculé. On peut par exemple imaginer que dans le cadre d'un flux vidéo codée en H.263 et encapsulé dans RTP, une fiabilité totale sera requise pour les images de type I tout en conservant un délai maximal de 200ms, on pourra tolérer par exemple une fiabilité partielle pour les images P et B avec un taux de perte maximal de 50% et un délai maximal de 350ms.

B. Mise en oeuvre

1. Méthodologie

La mise en oeuvre de ce mécanisme protocolaire a été proposée dans le « FrameWork ETP » décrit précédemment. Afin d'effectuer une spécification progressive, une modélisation itérative a été choisie. A chaque itération de la

modélisation, certaines fonctionnalités du mécanisme sont rajoutées afin de permettre d'optimiser progressivement le mécanisme mis en œuvre.

Dans une première itération de ce cycle, les objectifs fixés ont été simplistes et visaient à acquérir progressivement une familiarité avec le logiciel utilisé (Télélogiciel TAU 2.4) ainsi que le fonctionnement du « framework » en lui-même. Les spécifications de cette première itération ont été les suivantes :

- Demande de répétition systématique des PDUs perdus
- Pas de congestion irréversible des buffers d'émission
- Garantie d'acheminement des demandes de répétition

Il est évident que cette approche simpliste du problème ne saurait être proposée comme une amélioration des mécanismes existants et déjà mis en œuvre, néanmoins, une amélioration de performances lors d'une comparaison avec d'autres mécanismes modélisés dans le « framework » devrait être notable.

Dans un deuxième temps, une autre amélioration a été proposée, il s'agit de la mise en place d'un vecteur d'acquittement envoyé par l'entité réceptrice du flux afin d'avoir une signalisation moins importante en augmentant la sémantique de chaque paquet transmis.

Enfin, une gestion de la taille limitée des buffers de stockage des PDUs arrivés hors séquence en réception a été envisagée. La mise en place de ce mécanisme dans le framework n'a pas été faite car les buffers d'émission étant eux même contraints en taille, la possibilité de voir les buffers de réception dépasser une taille limite est écartée.

2. Architecture et rôles

Comme tous les mécanismes du framework ETP, le « selective repeat » se décompose en deux plans qui sont le plan de contrôle ainsi que le plan de gestion. Compte tenu des différentes vocations de ces deux plans, les mécanismes mis en œuvre dans chacun d'entre eux devront être clairement définis.

Afin de mieux situer les différents plans dans l'entité protocolaire développée, le diagramme d'architecture UML se présente comme une alternative intéressante. La Figure 1 page 7 présente ce diagramme. On y voit apparaître clairement les différents éléments du « framework » ETP. Néanmoins, certains éléments non décrits dans le « framework » mais proposés dans les travaux de [ARM05] sont également présents. En effet, ces éléments permettent une meilleure concentration sur les problèmes liés à la modélisation du mécanisme protocolaire proprement dit, tout en faisant quasi-abstraction des problèmes de bufferisation aux interfaces.

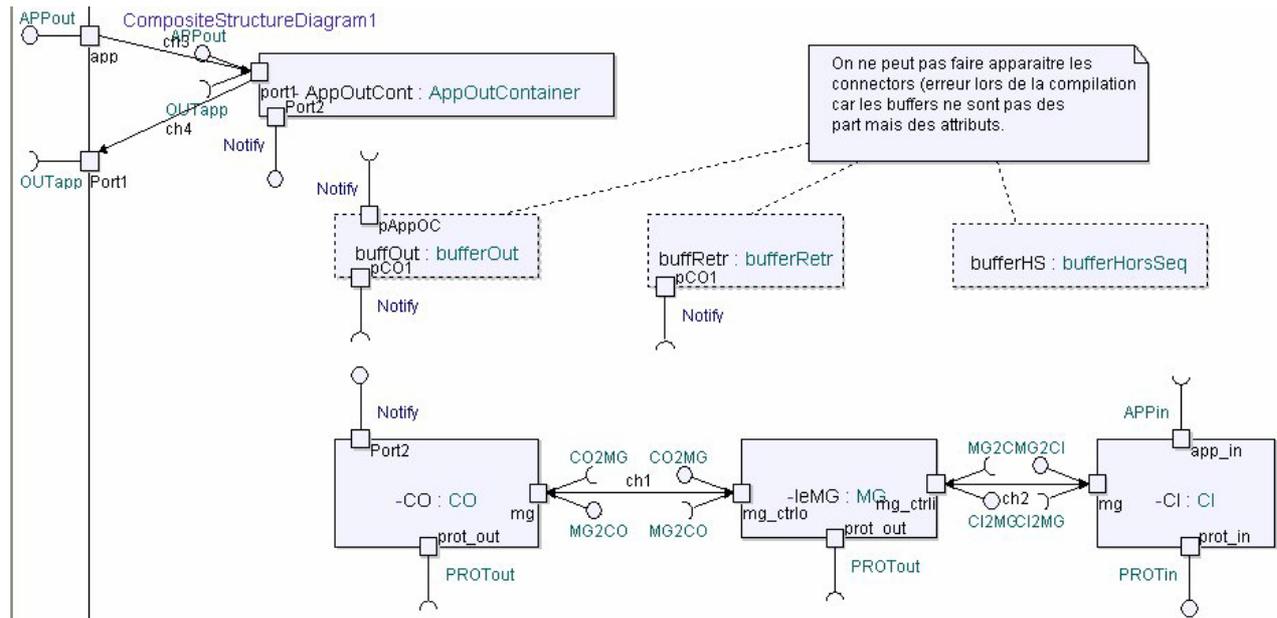


Figure 1 : Diagramme d'architecture de l'entité "Selective Repeat"

a) Le plan de contrôle d'émission (CO)

En émission, le plan de contrôle se situe entre les buffers de stockage servant de tampon avec l'application et l'entité protocolaire de niveau inférieur. Les interfaces qu'il implémente lui permettent une communication aussi bien avec cette dernière qu'avec le plan de management.

L'heuristique déployée au sein du CO se doit d'être la plus efficace possible afin de rester conforme à sa vocation qui est de maximiser le taux d'émission sans que d'autres traitements parallèles au mécanisme ne viennent diminuer les performances. C'est pourquoi, comme pour les mécanismes mis en œuvre dans [ARM05], la machine à états qui régit l'exécution de ce processus est restée simple.

A chaque fois que l'application émettrice place une donnée dans le buffer d'émission, cette donnée est convertie en PDU et un numéro de séquence lui est attribué. Une copie du PDU ainsi constitué est stockée dans le buffer de retransmission et le PDU est envoyé à l'entité protocolaire de niveau inférieur. Parallèlement à cela, le plan de management est informé de l'envoi du PDU afin de pouvoir effectuer les traitements adéquats.

Les mécanismes de retransmission n'étant pas du ressort du CO (le mécanisme de décision étant considéré comme un processus lent), c'est l'entité de management qui va se charger de la gestion des buffers afin de placer de façon prioritaire les PDUs à retransmettre dans le buffer d'émission.

La Figure 2 présente la machine à états du plan de contrôle d'émission.

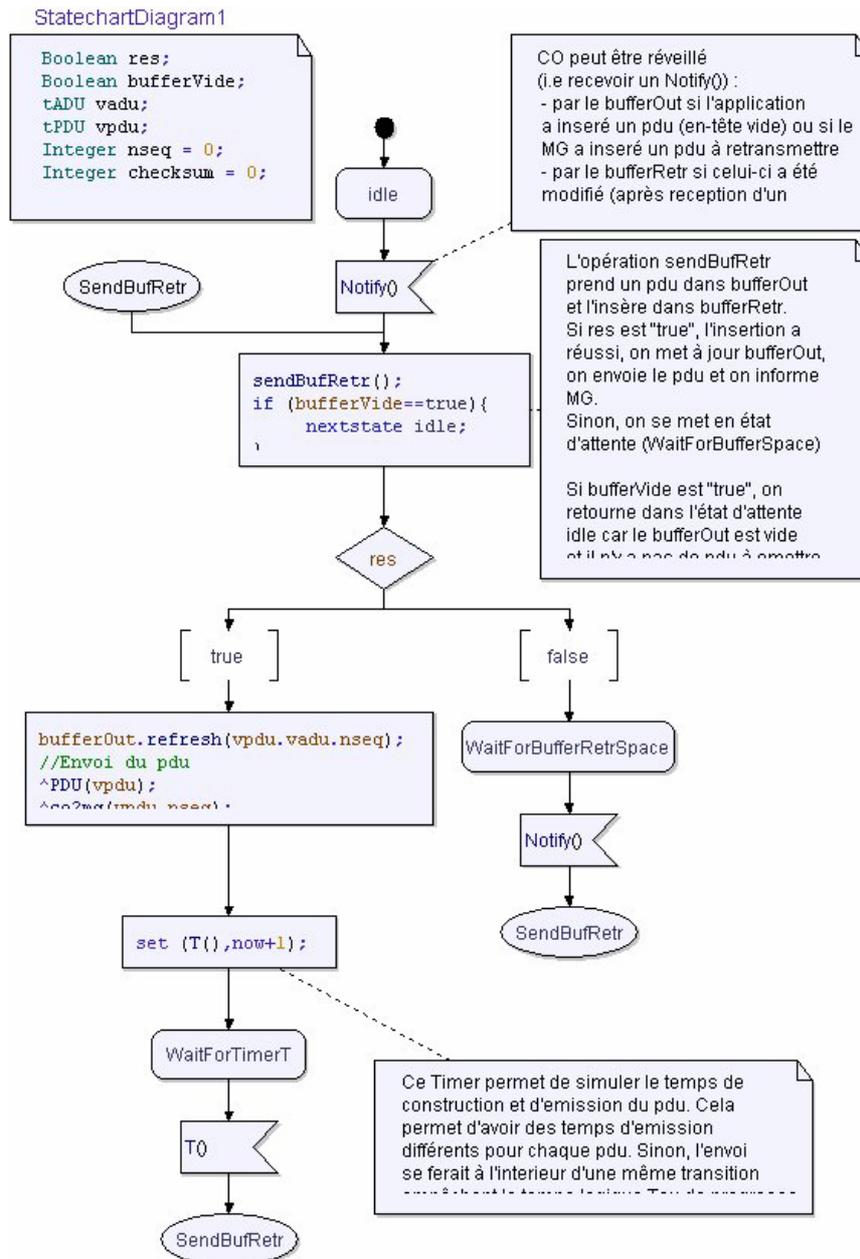


Figure 2 : Machine à état du plan d'émission (CO)

b) Le plan de contrôle de réception (CI)

Le comportement du plan de contrôle de réception est décomposé en deux interfaces, en effet, les messages qui régissent le fonctionnement de cette entité peuvent provenir d'une part du protocole de niveau inférieur et d'autre part du plan de management.

(i) Messages du protocole de niveau inférieur

Comme pour le CO, les mécanismes mis en œuvre au niveau du CI se veulent d'être optimaux et les actions à réaliser par ce plan doivent se limiter à des décisions simples et découlant immédiatement de l'évaluation de valeurs disponibles dans l'entête des PDUs manipulés.

Afin de garder la simplicité dans les mécanismes mis en œuvre, le comportement du CI est le même pour tous les PDUs de données. Sur réception d'un tel paquet, une validation rapide du checksum est effectuée afin de garantir l'intégrité des données. Si le test est positif, le paquet est alors stocké dans un buffer car le bon séquençement requis pour une remise directe à l'application ne peut pas être vérifié par le plan de réception qui ne connaît pas les euristiques décisionnelles mises en œuvre pour la prise de cette décision. Le plan de management est alors informé de la réception d'un paquet au checksum valide dont le numéro de séquence lui est communiqué.

Dans le cas des PDUs de contrôle, les champs important de leur entête est transmise au plan de management lors de leur réception.

Ces mécanismes sont décrits sur la machine à état présentée en Figure 4 page 10.

(ii) Commandes du plan de management

Une autre vocation du plan de réception étant la délivrance des ADUs à l'application réceptrice, un message issu du plan de management est prévu à cet effet. Sur réception de ce message, l'entité CI retire des buffers de réception les PDUs dont le numéro de série est indiqué par l'entité de management. Les ADUs contenus dans ces PDUs seront ensuite désencapsulés afin d'être remis à l'application. Ici encore, on retrouve le comportement propre au CI qui n'est que simplement exécutoire. Ce comportement est illustré sur la machine à états représentée Figure 3.

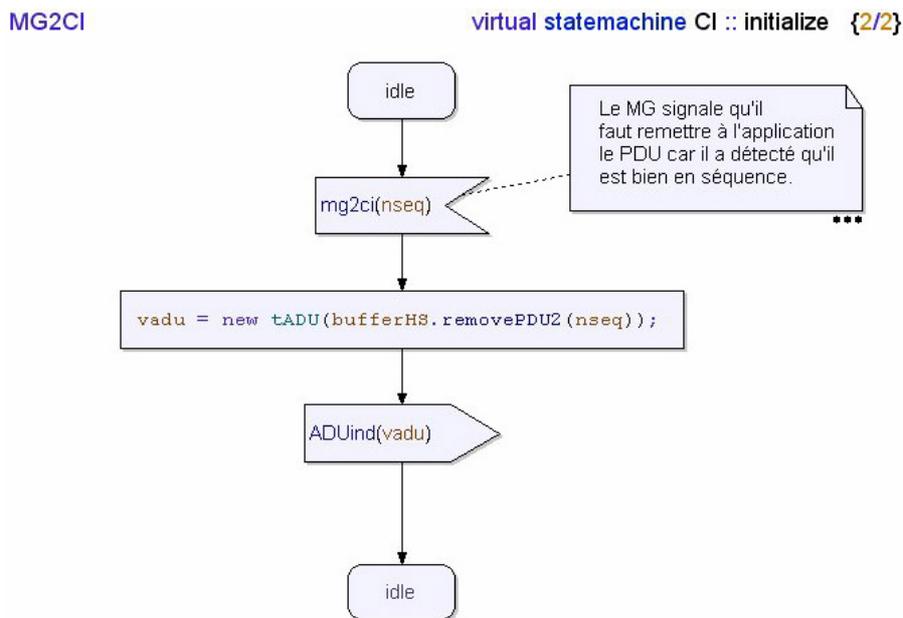


Figure 3 : Comportement sur commande du plan de gestion (MG)

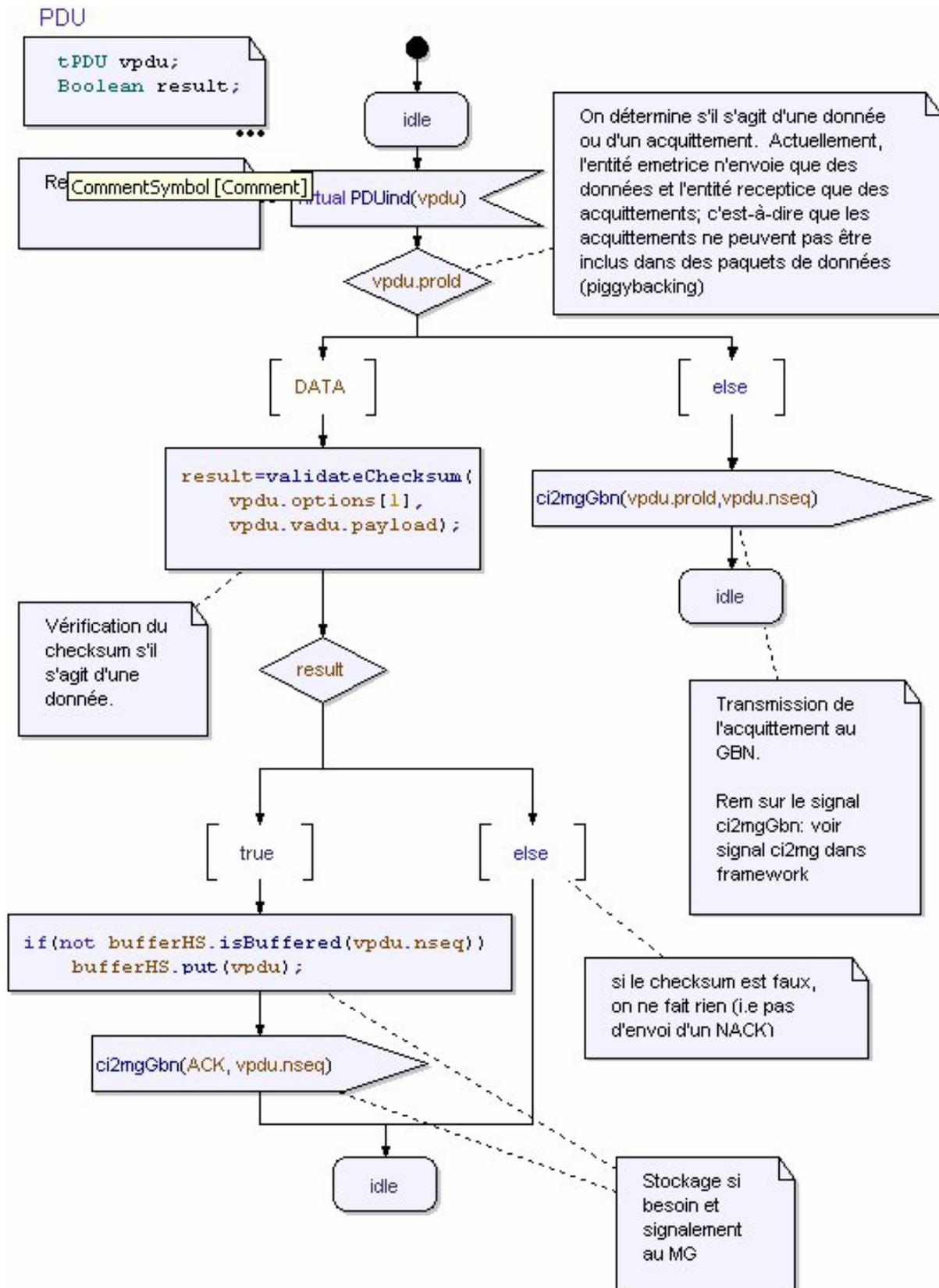


Figure 4 : Comportement du plan de contrôle de réception sur réception d'un PDU

c) Plan de gestion (MG)

Sans aucun doute, le plan de management (MG) met en place les heuristiques les plus complexes du mécanisme décisionnel du protocole. En effet, le « framework » ETP prévoit que les mécanismes « lourds » concernant la prise de décision dont le temps d'exécution est potentiellement plus long que ceux mis en place dans le plan de contrôle soient effectués en parallèle par le MG.

Le plan de gestion devient ainsi le centre intelligent du protocole. De par son architecture il ne peut recevoir aucun PDU directement, cette tâche étant réservée au plan de contrôle. Le MG est informé des événements extérieurs par les messages qui lui sont transmis par le CO et le CI et le comportement de l'entité sera différent selon qu'il s'agisse d'une notification de réception ou d'une notification d'émission. Ainsi, la machine à états qui décrit le fonctionnement de l'entité du plan de management est décomposée en trois sous machines chargées respectivement d'effectuer l'identification du mécanisme à mettre en œuvre, de la description des mécanismes mis en œuvre en réception et des mécanismes mis en œuvre en émission.

(i) Identification initiale

L'entité de gestion peut recevoir des signaux caractérisant soit des événements extérieurs qui lui sont signalés par le plan de contrôle, soit des événements internes liés aux mécanismes mis en œuvre (expiration d'un timer par exemple).

Les messages reçus provenant du plan de contrôle d'émission (CO) ont pour simple but de signaler l'émission d'un PDU, l'utilité de ces messages pour le plan de gestion sera explicitée lors de l'explication des mécanismes temporisés, dans la suite de ce chapitre (voir Fondement des mécanismes temporisés page 19). Il en va de même pour la réinitialisation des timers sur certaines des files d'exécution.

Le plan de contrôle de réception (CI) peut signaler au plan de gestion plusieurs événements, le but de l'identification initiale est de déterminer lequel des mécanismes mis en œuvre est à déclencher. Par exemple, lorsque le plan CI signale qu'il a reçu une demande de répétition d'un PDU, il faut diriger l'exécution de la machine à états vers le mécanisme propre à l'émetteur permettant la retransmission du PDU demandé.

Ce mécanisme de détermination de l'heuristique à mettre en œuvre est détaillé dans la machine à états présentée en Figure 5 page 12.

Une fois cette première identification effectuée, les actions à mener sont décrites sur différents diagrammes d'états qui seront présentés par la suite en distinguant les mécanismes prenant place du côté émetteur de ceux propres au récepteur.

IdentificationInitiale

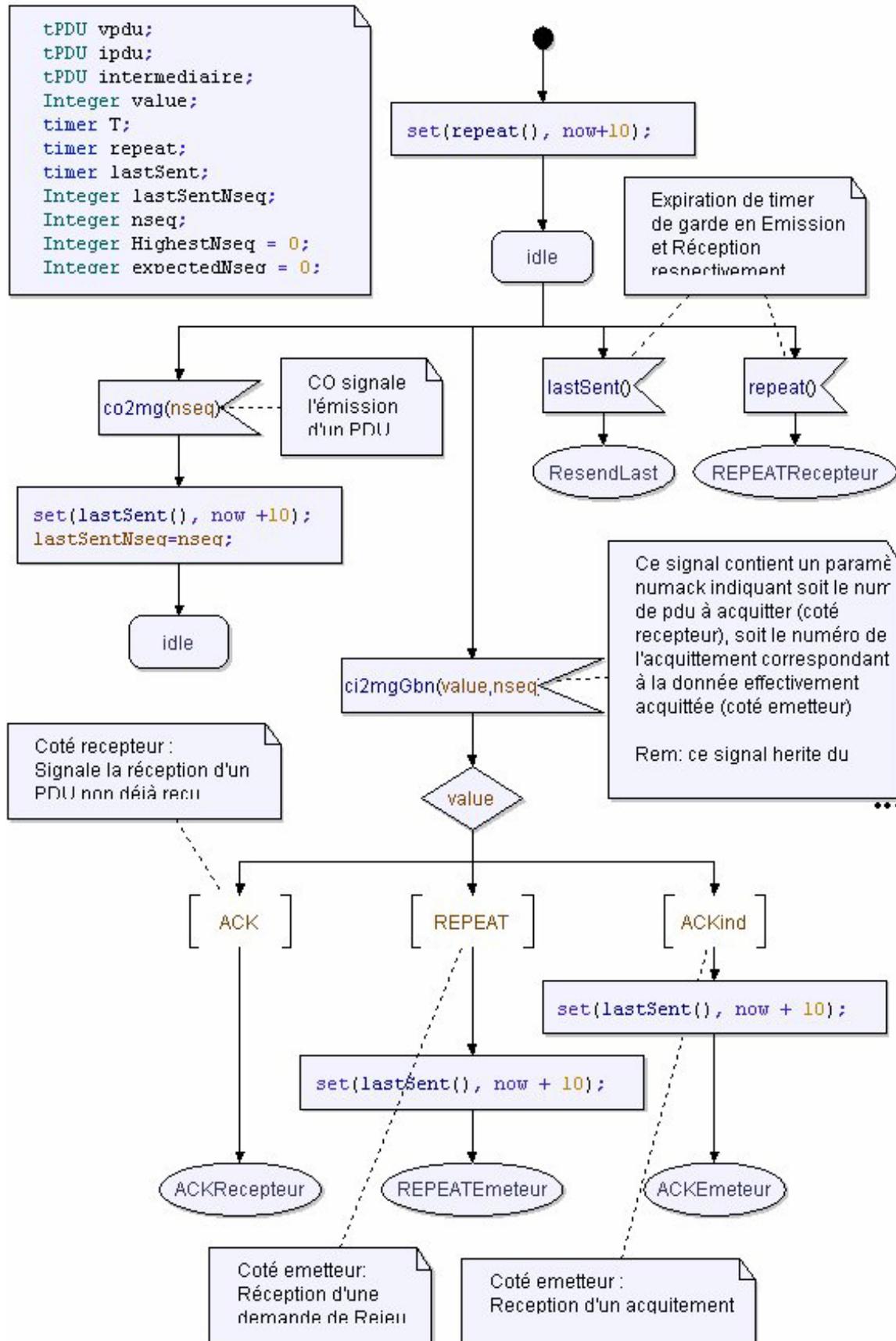


Figure 5 : Identification initiale et "routage" décisionnel pour le plan de Management (MG)

(ii) Mécanismes du récepteur

En fonction du résultat de l'identification initiale de la commande à exécuter en fonction des paramètres contenus dans le message reçu par l'entité de gestion, la file d'exécution du MG s'oriente vers l'un des mécanismes suivants :

- **ACKRecepteur** : Déclenché lorsqu'un PDU de données est signalé reçu.
 - **expectedReceived** : Déclenché par **ACKRecepteur** si le PDU reçu est en séquence.
- **REPEATRecepteur** : Déclenché sur expiration d'un timer de retransmission des demande de répétitions

Pour chacun de ces mécanismes, nous verrons leur fondement ainsi que le détail de leur exécution.

(a) ACKRecepteur

A chaque réception par le MG d'un évènement signalant la réception d'un PDU par le CI, le fil d'exécution est dirigé vers la branche **ACKRecepteur**. La tâche principale accomplie par cette branche porte sur le numéro de séquence du PDU qui vient d'être reçu. En effet, celui-ci est comparé avec le prochain numéro de séquence attendu par l'entité de réception.

Ceci permet de déterminer si le PDU reçu correspond :

- à un PDU dupliqué dont le contenu aurait déjà été remis à l'application (dans ce cas, le comportement se limite à supprimer le doublon et à acquitter à l'émetteur le dernier PDU reçu en séquence).
- au PDU attendu auquel cas, c'est le branchement **expectedReceived** qui est exécuté
- à un PDU hors séquence qui aura été placé dans les buffers de réception par le récepteur. Dans ce cas, tous les PDUs dont le numéro de séquence se trouve entre le prochain numéro de séquence attendu et le numéro de séquence du paquet en cours font l'objet d'un paquet de demande de répétition à l'émetteur.

Le diagramme d'états correspondant à l'exécution de cette branche est présenté sur la Figure 6 page 14.

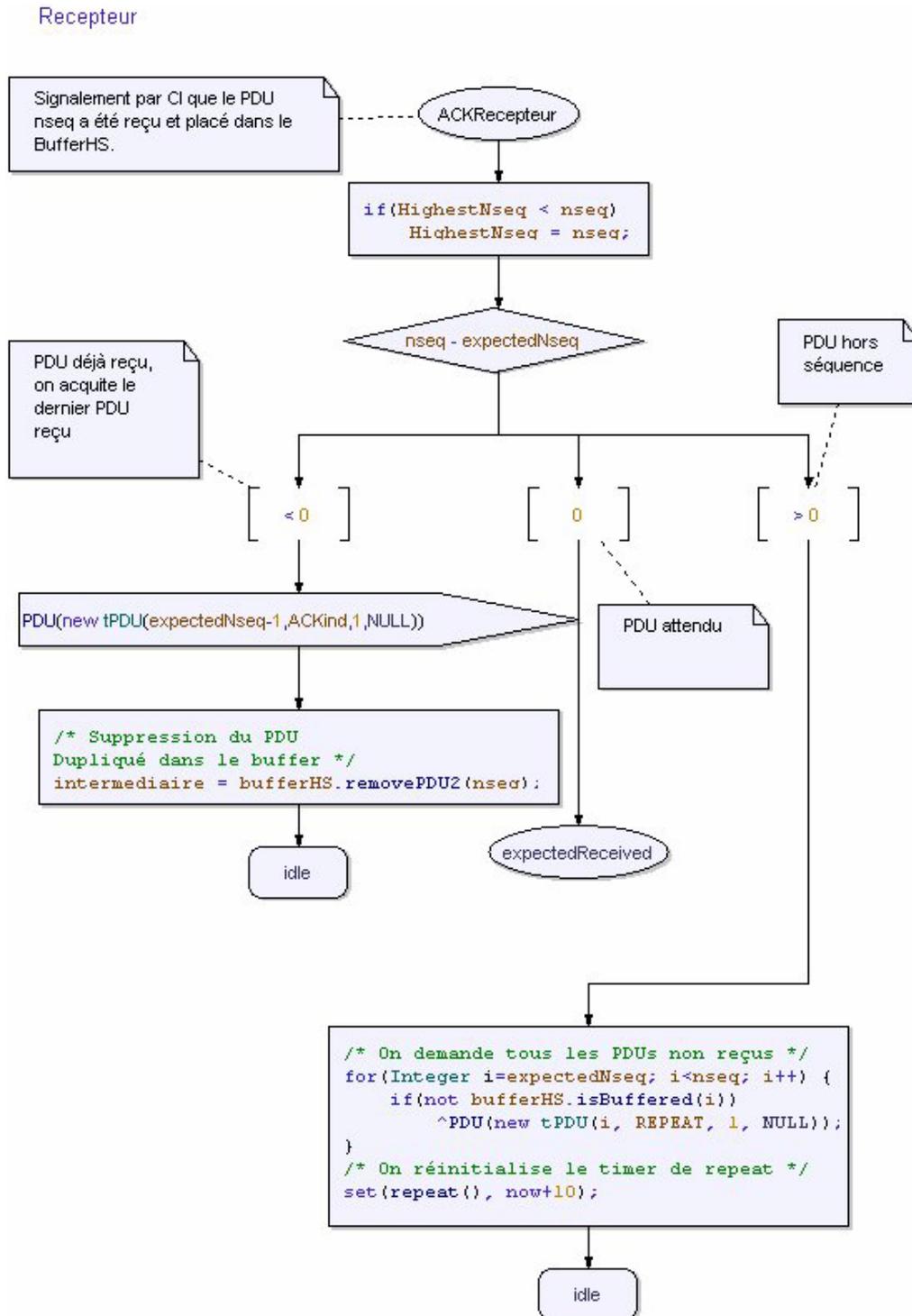


Figure 6 : ACKRecepteur

(b) *expectedReceived*

Lorsque le PDU reçu correspond au PDU attendu, l'exécution consiste à commander au CI de délivrer le PDU en question ainsi que tous les PDUs suivants présents dans les buffers. Lorsque la séquence est interrompue par un PDU manquant, un acquittement cumulatif est alors envoyé à l'émetteur afin qu'il puisse modifier ses

buffers en fonction. Ce mécanisme est illustré par le diagramme présenté Figure 7 page 15

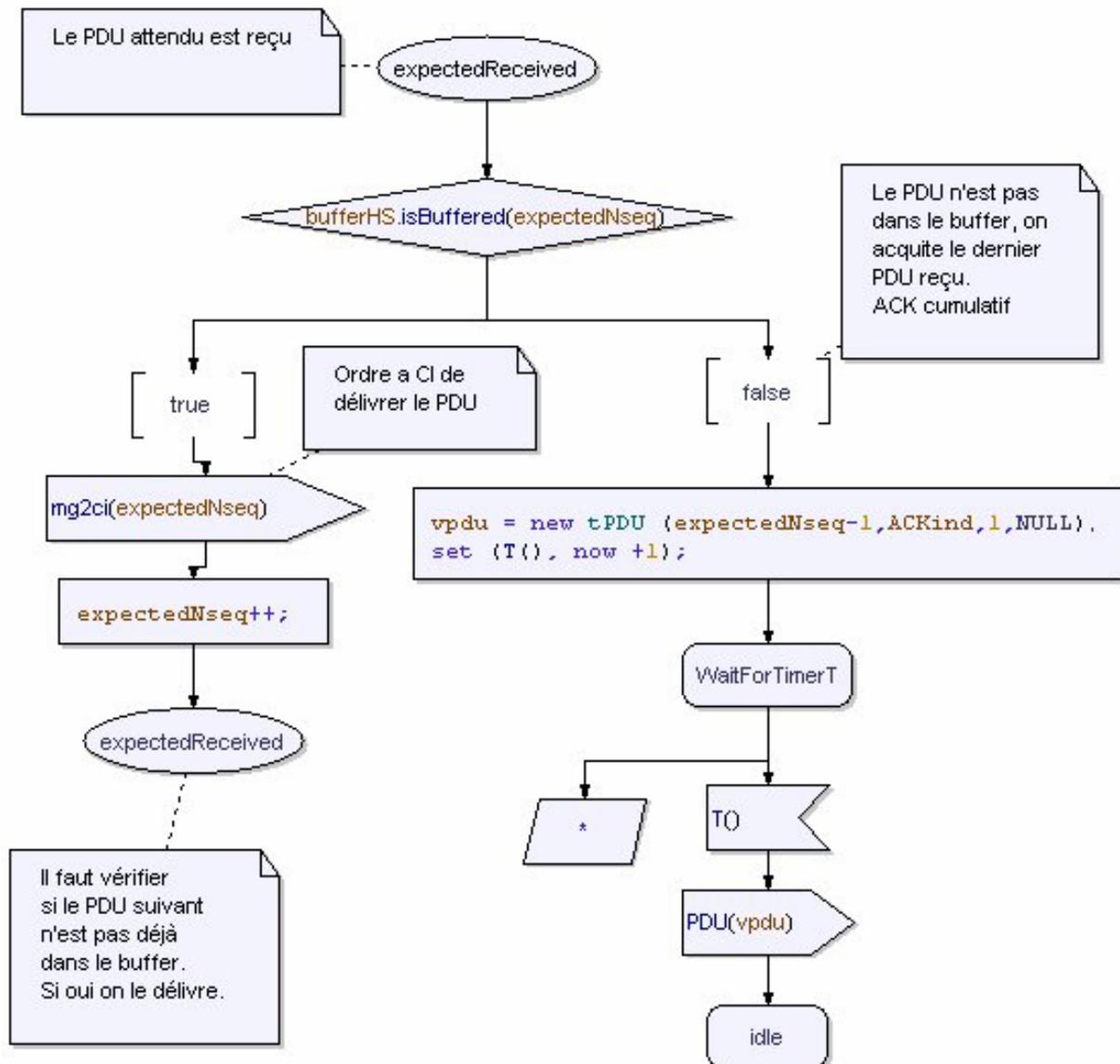


Figure 7 : expectedReceived

(c) REPEATRecepteur

Ce branchement conditionnel est exécuté lorsque le timer associé à la dernière demande de retransmission expire. Le besoin d'un tel processus sera explicité ultérieurement dans ce chapitre, son exécution consiste à vérifier dans un premier temps s'il y a déséquence dans la réception du flux et le cas échéant à redemander l'intégralité des PDU absents des buffers de réception par l'envoi d'un PDU de type REPEAT.

La machine à états décrivant ce procédé est présentée Figure 8 page 16.

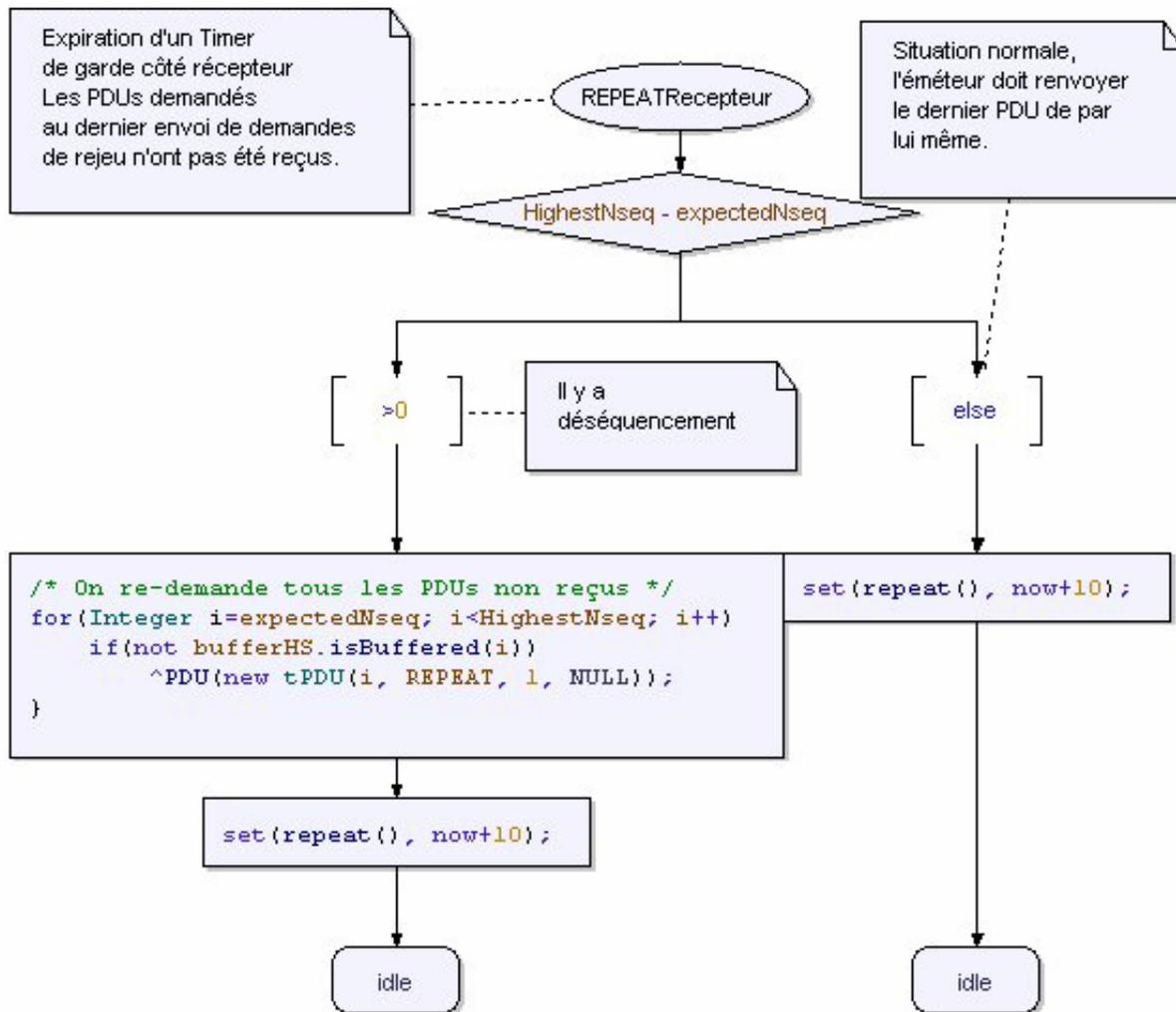


Figure 8 : REPEATRecepteur

(iii) Mécanismes de l'émetteur

Les messages susceptibles d'être reçu par l'entité émettrice conduiront l'exécution conditionnelle de l'entité de gestion vers l'un des mécanismes suivants :

- **ACKEmeteur** : Déclenché sur la réception d'un acquittement (coté émetteur)
- **REPEATEmeteur** : Déclenché sur la réception d'une demande de répétition.
- **ResendLast** : Déclenché sur expiration du timer de retransmission du dernier PDU transmit.

Pour chacun de ces mécanismes, nous verrons leur fondement ainsi que le détail de leur exécution.

(a) ACKEmeteur

Sur réception d'un acquittement, l'entité émettrice se doit de supprimer de ses buffers l'intégralité des PDUs dont le numéro de séquence est inférieur au numéro de séquence contenu dans l'acquittement. Il se peut également que l'acquittement concerne une PDU qui n'est plus dans les buffers auquel cas aucune action n'est nécessaire.

La machine à états correspondant à cette procédure est présentée sur la Figure 9 page 17

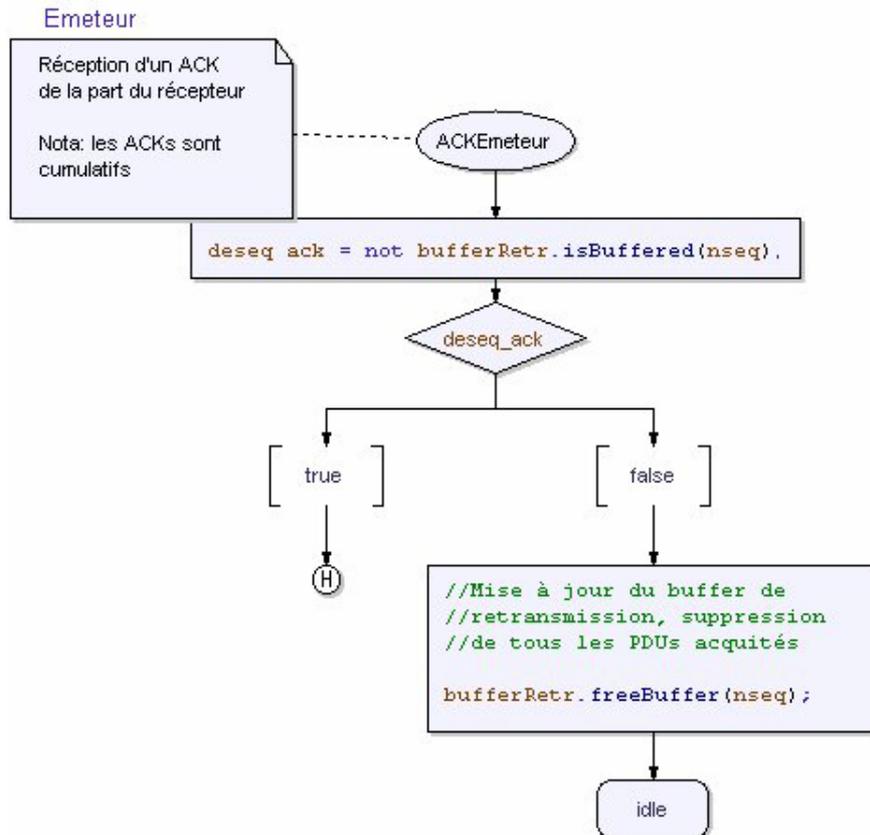


Figure 9 : ACKEmeteur

(b) REPEATEmeteur

Sur réception d'une demande de répétition, l'émetteur se doit de replacer le PDU à retransmettre dans le buffer de transmission, si toutefois, celui-ci n'a pas déjà été acquitté précédemment.

La machine à états correspondant à cette procédure est présentée sur la Figure 10 page 18

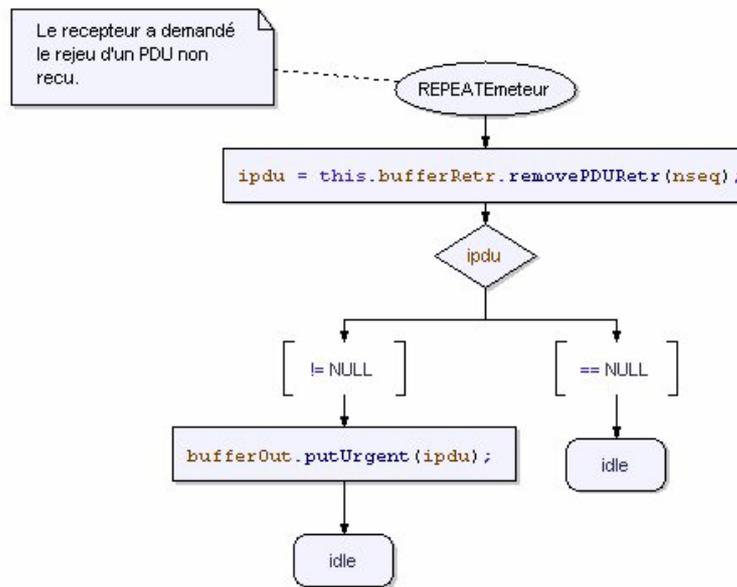


Figure 10 : REPEATEmeteur

(c) ResendLast

Lorsque le timer associé au dernier PDU transmis par l'entité émettrice expire, ce dernier doit être replacé dans les buffers d'émission. Les fondements de ce mécanisme seront explicités par la suite dans ce chapitre.

La machine à états correspondant à cette procédure est présentée sur la Figure 11.

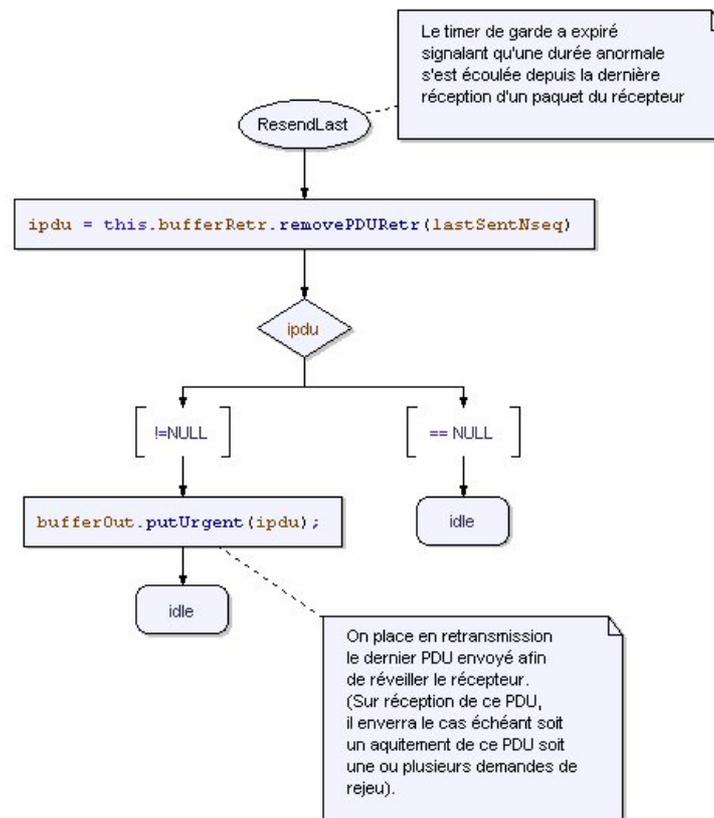


Figure 11 : ResendLast

(iv) Fondement des mécanismes temporisés

Le mécanisme du « selective repeat » présenté dans ce document comprend un certain nombre d'exécutions déclenchées par l'expiration de timers. La raison de la présence de ces timers n'est pas intuitive. En effet, dans un environnement purement théorique pour lequel il est possible de faire l'hypothèse que les PDUs ne peuvent pas être perdus ni déséquilibrés, le mécanisme peut se décrire de façon beaucoup plus simple.

Néanmoins, dans les environnements propices aux erreurs, il est nécessaire de mettre en place des mécanismes permettant aux entités concernées de protéger leurs envois.

(a) Perte des paquets de l'émetteur.

Lorsqu'un paquet émis est perdu, le mécanisme du « selective repeat » propose que celui-ci soit redemandé par l'entité réceptrice. Pour cela, il est nécessaire que cette dernière soit au courant de l'émission de ce paquet par la réception d'un paquet plus récent dans la séquence.

Néanmoins, il est possible que même ainsi, l'émetteur ne soit pas au courant de la perte du paquet dans le cas où la demande de répétition se perde.

Afin de pallier à cette difficulté sans pour autant perdre les avantages de performances qu'apporte le « selective repeat », un timer a été introduit. Ce timer est associé au dernier PDU transmis sur le réseau, à chaque émission il est réinitialisé et la réception de l'acquittement du dernier PDU émis le neutralise. De plus, tant que le récepteur envoie des demandes de répétition ou des acquittements, il n'est pas nécessaire de retransmettre le dernier PDU car il n'est pas possible d'affirmer qu'il a été perdu, le processus de réception n'ayant pas encore traité les PDUs précédents. C'est pour cela que le timer est là aussi réinitialisé.

(b) Perte des demandes de répétition.

Lorsqu'une demande de répétition se perd, il est possible de supposer que le prochain paquet reçu déclenchera le mécanisme qui consiste à redemander tous les paquets hors séquence. Malheureusement, cette hypothèse suppose que l'émetteur possède encore des paquets à transmettre, si ce n'est pas le cas le système risque le blocage.

Afin de pallier à cela, un timer de retransmission est placé en réception. Sur expiration de ce timer, tous les PDUs non bufferisés sont redemandés à l'entité émettrice. Néanmoins, afin de maintenir l'émission de paquets à son minimum, ce timer est réinitialisé par tout mécanisme indiquant une activité de la part de l'émetteur.

C. Simulation & Traces d'exécution

1. Mise en place de la simulation

Ce mécanisme étant modélisé en UML en utilisant le logiciel TAU développé par la société Télégologic, il est possible de procéder à une simulation du comportement du système pendant laquelle le programme trace le diagramme de séquence correspondant à l'exécution.

Afin de pouvoir procéder à ces tests, le « framework » ETP propose un nombre important de classes utilitaires tels des applications émettrices et réceptrices ainsi que différents types de médiums.

La composition des éléments ainsi déployée pour la simulation est décrite en UML dans un diagramme d'architecture qui permet de montrer les relations entre les classes ainsi que les différents canaux de communication mis en place.

Ce diagramme est donné Figure 12. On observe ici que le choix a été fait de modéliser l'application émettrice par une application supportant le contrôle de flux. De plus, les deux modules correspondants au mécanisme du « selective repeat » apparaissent bien à l'interface entre les applications et la couche protocolaire de niveau inférieur modélisée ici par le médium (chacun des médiums sont unidirectionnels mais leur union tel qu'elle est faite émule un médium bidirectionnel).

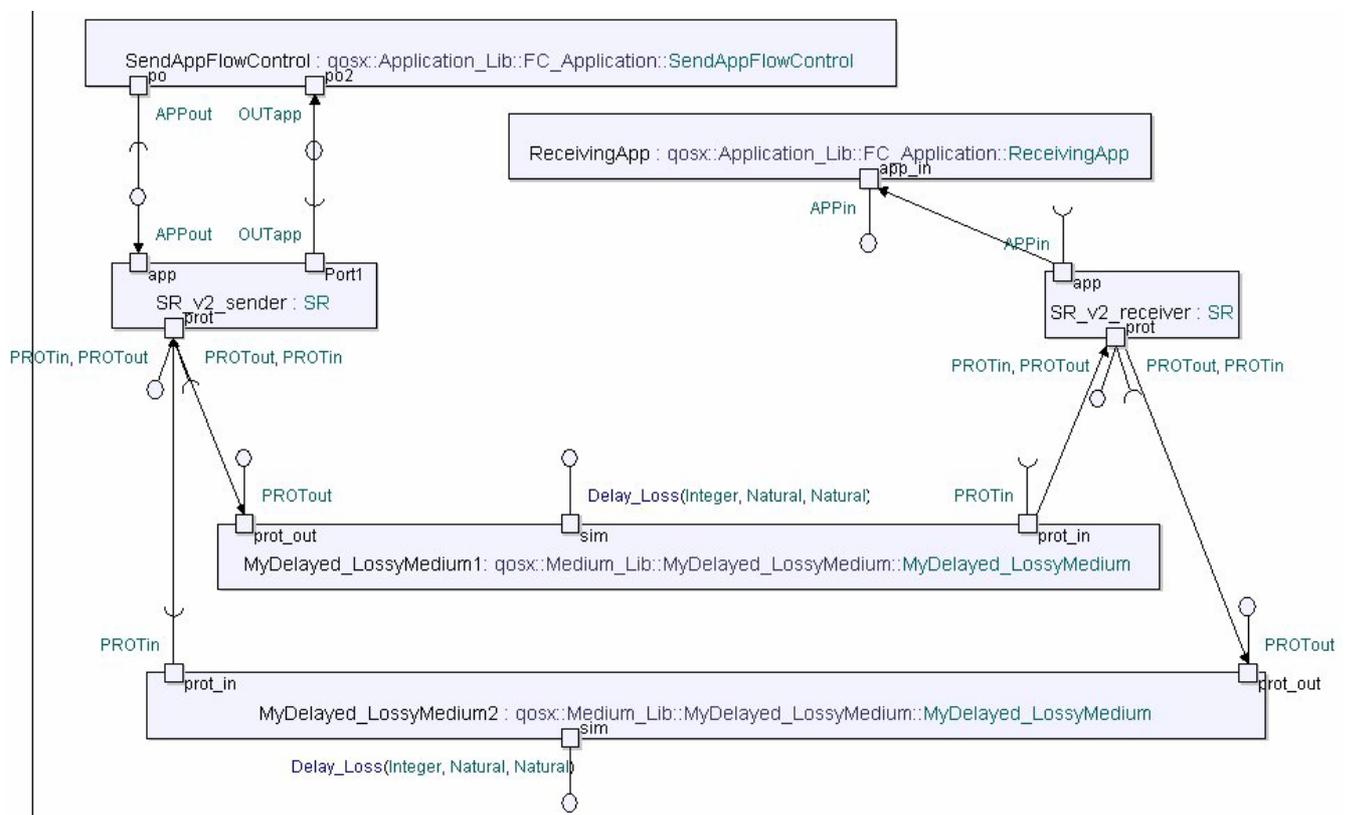


Figure 12 : Diagramme d'architecture de test

2. Traces d'exécution

Comme indiqué précédemment, le logiciel TAU génère, lors de la simulation du modèle UML réalisé, une trace sous la forme d'un diagramme de séquence. Bien que l'utilité de ces traces soit indiscutable dans le but de constater que les mécanismes mis en œuvre correspondent bien à ceux qui sont attendus, il ne faut pas les confondre avec une preuve formelle de conformité du protocole avec sa spécification.

Afin de ne pas trop alourdir la lecture de ce document, seule une retranscription des événements d'importance de quelques scénarios d'exécution choisis sera abordée. En effet, les traces générées sont exhaustives dans le sens où toutes les entités prenant part à l'exécution du protocole y possèdent une ligne de vie rendant ainsi très difficile, sans un peu d'habitude, la lecture du diagramme obtenu.

a) Pertes & déséquences ponctuels

L'intérêt du mécanisme mis en œuvre consiste en sa grande granularité qui permet d'accroître la réactivité de l'émetteur face aux éventuelles difficultés liées au réseau sous jacent. En effet, la sélectivité au paquet donne à la connexion une efficacité totale accrue de par la minimisation du nombre de paquets déjà reçus qui se verront retransmis.

Afin d'illustrer les mécanismes mis en œuvre dans la logique du « FrameWork » ETP, un diagramme de séquence UML décrivant le mécanisme mis en œuvre lors de la perte de deux paquets dans un flux composé de six paquets a été retranscrite Figure 13 page 22.

Le scénario consiste à l'émission successive par l'émetteur de six paquets de données. Pendant cette transmission, deux erreurs vont entraîner la perte des paquets 2 et 3. Nous illustrerons le mécanisme d'une part de détection de ces erreurs et d'autre part de détection de la correction de celles-ci.

1. La trace débute par l'émission par l'entité CO d'un PDU dont le numéro de séquence est zéro.
2. Cette trame est ensuite signalée émise au plan de gestion par l'intermédiaire du message local *co2mg(Integer)*.
3. Sur réception de ce PDU, l'entité CI du récepteur va placer le PDU dans des buffers de réception et va signaler au plan de gestion par l'intermédiaire du message *ci2mg(Integer, Status)* la réception d'un nouveau PDU.
4. Le plan de gestion va comparer le numéro de séquence du PDU qui lui a été signalé reçu avec le numéro de séquence attendu et va constater que le PDU reçu est en séquence. De plus, MG se doit également de vérifier la présence éventuelle dans les buffers de réception des PDU suivants de la séquence avant de commander la remise de ce/ces PDU(s) à l'application réceptrice par l'intermédiaire du message *mg2ci(Integer)*.

sd SR version 1

interaction 'REP-Systématique' {1/1}

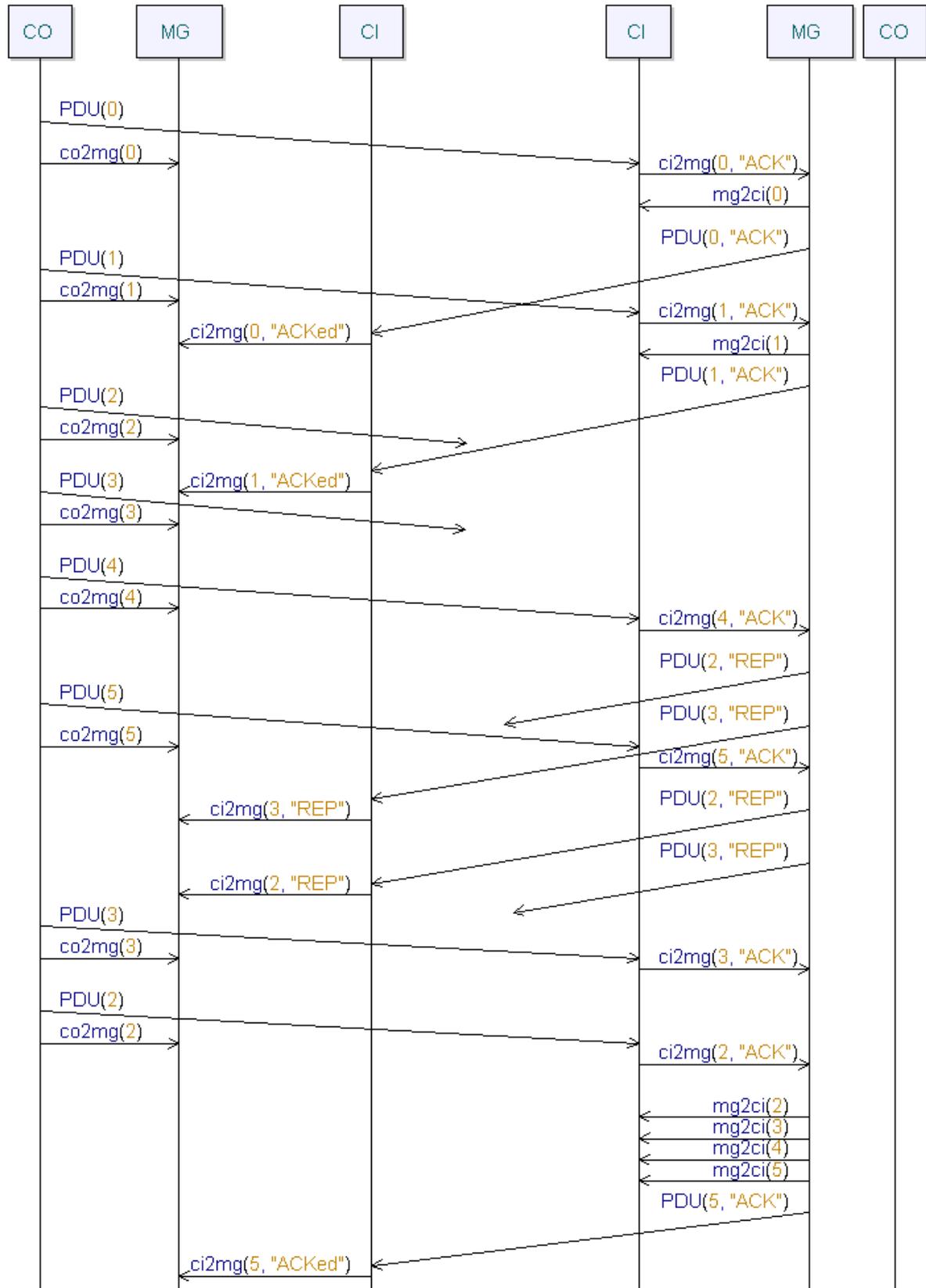


Figure 13 : Diagramme de séquence - Reprise sur pertes

5. Une fois cette euristique déployée, MG transmet un acquittement cumulatif à l'émetteur en acquittant ainsi tous les PDUs remis à l'application réceptrice.
6. Sur réception de cet acquittement, le CI de l'émetteur signale l'évènement au plan de gestion par l'intermédiaire du signal *ci2mg(Integer, Status)* qui supprimera le PDU signalé des buffers de retransmission.

Dans le cas où le numéro de séquence du PDU signalé comme reçu par l'entité CI du récepteur au MG à l'étape 3 ci-dessus est différent du prochain numéro de séquence attendu, le comportement du système varie comme suit (cas de la réception du PDU(4) sur la Figure 13 :

4. Le plan de gestion détecte un déséquencement, pour chaque PDU non reçu, une demande de répétition va être émise à destination de l'émetteur. Ceci se fait par la transmission d'un PDU dont le drapeau REP de répétition est activé.

Note : Afin de simplifier le diagramme ci-dessus, les PDUs 4 et 5 arrivant successivement, les demandes de répétition des PDUs 2 et 3 sont envoyés deux fois, néanmoins, afin d'éviter d'alourdir la figure par la représentation de la double retransmission engendrée par ce comportement, certains des PDUs de demande de répétition choisis spécifiquement sont considérés perdus.

5. Sur réception de la demande de retransmission, le CI de l'émetteur va informer le plan de gestion par le message *ci2mg(Integer, Status)*. Le plan de gestion placera alors les PDUs à retransmettre dans le buffer de transmission. Cette insertion est effectuée en tête afin de marquer le caractère urgent de la demande.
6. L'entité du plan de contrôle en émission (CO) va ensuite retransmettre ce PDU en le signalant au plan de gestion par le message *co2mg(Integer)*.
7. Sur réception de la demande de retransmission, l'entité CI du récepteur signale la réception d'un PDU au MG par le signal *ci2mg(Integer, Status)* et place le paquet dans les buffers de réception.
8. Le plan de gestion remarque que le PDU reçu est en séquence et scrute alors les buffers de réception afin de rechercher la présence éventuelle des PDUs suivants. La remise de tous les PDUs en séquence est ordonnée individuellement à l'entité CI par le message *mg2ci(Integer)*.
9. Une fois le processus terminé, un acquittement cumulatif est adressé à l'émetteur.

b) Importance des mécanismes temporisés

Lors de la description précédente de la trace étudiée dans le cas de la résolution des problèmes pertes et déséquencements, les timers mis en jeu en émission et en réception ont été volontairement omis. Ceci d'une part afin de garder la clarté du diagramme et d'autre part de montrer que le système mis en place possède, dans certains cas de figure, une exécution exempte de blocage.

Néanmoins, comme montré précédemment (Fondement des mécanismes temporisés - II.B.2.c)(iv), certaines exécutions peuvent conduire au blocage du système. Pour pallier à cela, deux mécanismes indépendants ont été mis en place.

C'est dans le but d'illustrer leur fonctionnement respectif que deux diagrammes de séquence sont proposés à la suite.

(i) Temporisations de l'émetteur

Le premier mécanisme de temporisation mis en place dans le mécanisme consiste en une temporisation du côté de l'émetteur visant à détecter une anomalie dans le processus de transmission du flux résultant de la perte successive de plusieurs PDUs et de leur demande de répétitions respectives. Il s'agit d'un timer visant à détecter une période anormale d'inactivité du récepteur depuis la dernière émission effective.

La mise en œuvre de cette temporisation est présentée sur le diagramme de séquence UML présent en Figure 14 page 25. On remarque sur cette trace la présence d'un stéréotype UML nouveau qui n'avait pas été présenté auparavant. Il s'agit en effet de la représentation UML correspondante aux temporisations.

Le scénario observé correspond à l'émission de cinq paquets successifs avec la perte du second paquet transmis (portant le numéro de séquence 1). De plus, on suppose que tous les paquets de demande de répétition provenant du récepteur sont perdus. Ainsi, seul le PDU zéro se voit acquitté et les buffers de retransmission étant de taille finie s'en retrouvent congestionnés du fait que les PDUs suivants, bien que reçus, ne sont pas acquittés (acquiescement cumulatif) du fait du déséquencement.

Dans cet état du système, il est évident que la seule façon envisageable de débloquent le système consiste à ce que le PDU numéro un soit retransmis par l'émetteur. Afin d'arriver à ce but, deux méthodes sont envisageables, en effet, le récepteur pourrait sans cesse demander la retransmission du prochain PDU attendu en séquence mais cette solution ne permettrait pas de débloquent un système pour lequel le récepteur n'a jamais reçu de données.

Lors de l'expiration du timer d'émission, le dernier PDU transmis est réémis, cela permet d'obtenir en retour de la part du récepteur l'image de ses buffers de réception qui est communiquée par la succession de demandes de répétitions qui en résulte. Les PDUs demandés peuvent ainsi être retransmis au plus vite.

Le bon fonctionnement de ce scénario réside sur l'hypothèse que le lien entre le récepteur et l'émetteur puisse à un moment être opérationnel, dans le cas contraire la communication ne reprendrait pas son cours normal. Le mécanisme ici repose sur la supposition que le temps d'expiration du timer de retransmission sera suffisamment long pour que les perturbations qui entraînaient l'impossibilité de faire circuler des flux entre le récepteur et l'émetteur se soient estompées.

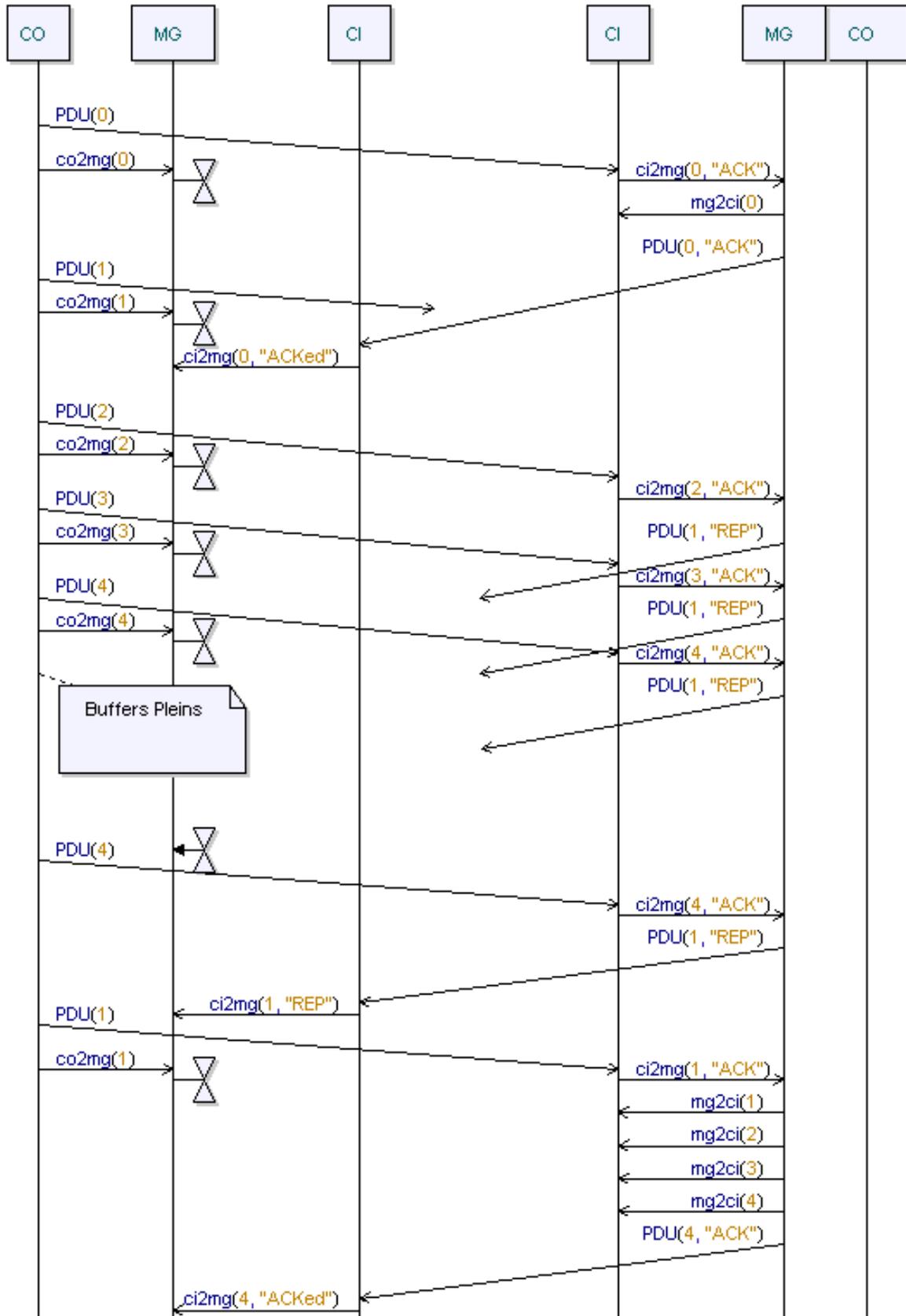


Figure 14 : Diagramme de séquence - Temporisation de l'émetteur

(ii) Temporisations du récepteur

Le mécanisme présenté ci-dessus présente l'avantage de fonctionner quelque soit l'état dans lequel se trouve le système. En effet, le problème lié à la perte successive des premiers paquets de la transmission n'est pas présent. Néanmoins, ce mécanisme ne semble pas très homogène avec la philosophie développée dans le cadre du mécanisme du « selective repeat ». En effet, il est plus naturel de penser que le seul décideur dans la demande de retransmission soit l'hôte récepteur car c'est le seul à connaître directement l'état de ses buffers de réception.

C'est dans cette optique que les mécanismes temporisés du côté du récepteur ont été introduits. En effet, en jouant sur les temps de temporisation des deux mécanismes respectivement en émission et en réception, il est possible de privilégier l'exécution de l'un ou l'autre des exécutions. De plus, le problème lié à la défaillance du lien pour l'un des sens de la communication est ainsi maîtrisé.

Le fonctionnement de ces temporisations est identique au fonctionnement présenté pour les timers d'émission. En effet, les messages à protéger sont maintenant les demandes de retransmission et non les PDUs, le mécanisme est donc transposé à l'identique du côté du récepteur. Afin de garder le mécanisme simple, un seul timer est utilisé. Il est réinitialisé à chaque émission d'une nouvelle demande de répétition et désactivé lorsqu'il n'y a plus de déséquencements.

Sur expiration de ce timer, l'état des buffers est scruté et tous les PDUs non bufferisés font l'objet d'une demande de retransmission spécifique de la part du récepteur.

Le diagramme de séquence UML présenté en Figure 15 page 27 fait état de ce mécanisme en l'illustrant sur un exemple simple. Lors d'une séquence d'émission de quatre paquets, le second est considéré comme perdu. De même, les demandes de retransmission émises sur la réception des deux paquets suivants sont elles aussi considérées perdues.

Pour chacune des deux demandes de retransmission, le timer de retransmission en réception est réinitialisé. Sur expiration de ce timer, l'état des buffers de réception est scruté et les demandes de répétitions adéquates sont effectuées. Sur réception du PDU manquant, l'entité de réception (MG) commande au plan de contrôle (CI) la remise des PDUs en séquence.

Ce mécanisme, bien que similaire au mécanisme présenté en émission ne lui est autre que complémentaire. En effet, le premier vise à palier aux erreurs qui sont indiscernables par le récepteur tandis que le second cherche à accroître l'efficacité globale du mécanisme en garantissant l'acheminement des demandes de retransmission en évitant ainsi au maximum l'émission par l'émetteur de PDUs déjà bufferisés.

III. Introduction du vecteur d'acquittement (SACK)

Dans une première approche, le mécanisme du « selective repeat » a été modélisé de façon simple, en considérant que chaque paquet non bufferisé par l'hôte récepteur faisait l'objet d'une demande de retransmission unique. Cette première modélisation a présenté l'intérêt de pouvoir s'adapter au « framework » ETP sans aucune modification de celui-ci. Néanmoins, afin d'accroître l'efficacité de ce mécanisme, une solution dans laquelle une émission périodique de vecteurs d'acquittement permettrait à l'émetteur d'obtenir une copie locale de l'état de buffers de réception a été envisagée. Cette optimisation du mécanisme se rapproche plus de celle présentée dans [EXP03] concernant le protocole FPTP.

Les avantages de cette technique sont nombreux. Tout d'abord, l'utilisation des ressources réseau est réduite de par la réduction du nombre de PDUs circulant sur le réseau. La possibilité de procéder au « piggybacking » des vecteurs d'acquittement est envisageable afin d'accroître l'optimisation. Malheureusement, sa mise en place dans le « framework » ETP demande une légère modification de celui-ci pour deux raisons :

- Lors de la génération de code pour la vérification de modèle, le logiciel TAU ne permet pas l'utilisation des mécanismes du polymorphisme.

Ceci pose problème au niveau du médium. En effet, celui-ci est modélisé de telle façon qu'il véhicule un objet de type tPDU. Si l'on souhaitait utiliser l'héritage afin de permettre au médium de véhiculer un PDU possédant un vecteur d'acquittement, les mécanismes du polymorphisme permettraient d'utiliser le même médium afin de les véhiculer.

Néanmoins, le logiciel ne permet pas l'utilisation de ce mécanisme refusant catégoriquement de véhiculer une classe fille dans une variable du type père.

- L'ajout de signaux à une interface ne peut pas se faire dans une classe dérivée.

Cette limitation entraîne une modification inévitable de l'interface de base du « framework ». En effet, du fait qu'il n'est pas possible d'ajouter des signaux aux interfaces réalisées et requises par les classes héritées, il n'est pas possible d'étendre la gamme de message que peuvent s'échanger le CI et le MG par exemple ce qui pose problème afin que le premier puisse informer le second de l'état des buffers de réception sur arrivage d'un SACK, par exemple.

Malgré ces difficultés, une méthode simple bien que difficilement conforme avec l'optique du « framework » en lui-même qui est de permettre l'extension du cadre de base par l'ajout de nouveaux messages et de nouvelles options dans l'entête des PDUs consiste à modifier directement les classes dans le « framework » de base. C'est cette technique qui a été utilisée ici bien qu'à contre cœur.

A. Modifications d'architecture

1. Ajout du SACK aux PDUs

Afin de permettre l'ajout d'un champ d'entête aux PDUs tout en respectant les mécanismes objets qui sont la base de la modélisation UML, une nouvelle classe a été introduite au « framework ». Cette classe porte le nom de tSACK et représente un vecteur d'acquiescement.

La classe tPDU du « framework » a elle aussi été modifiée pour y ajouter une référence à un vecteur d'acquiescement permettant son utilisation le cas échéant. L'architecture qu'il en résulte est décrite sur la Figure 16 ci-dessous. On observe que les différentes méthodes d'accès aux éléments du SACK sont prévues tout en conservant une classe tPDU relativement proche de l'originale.

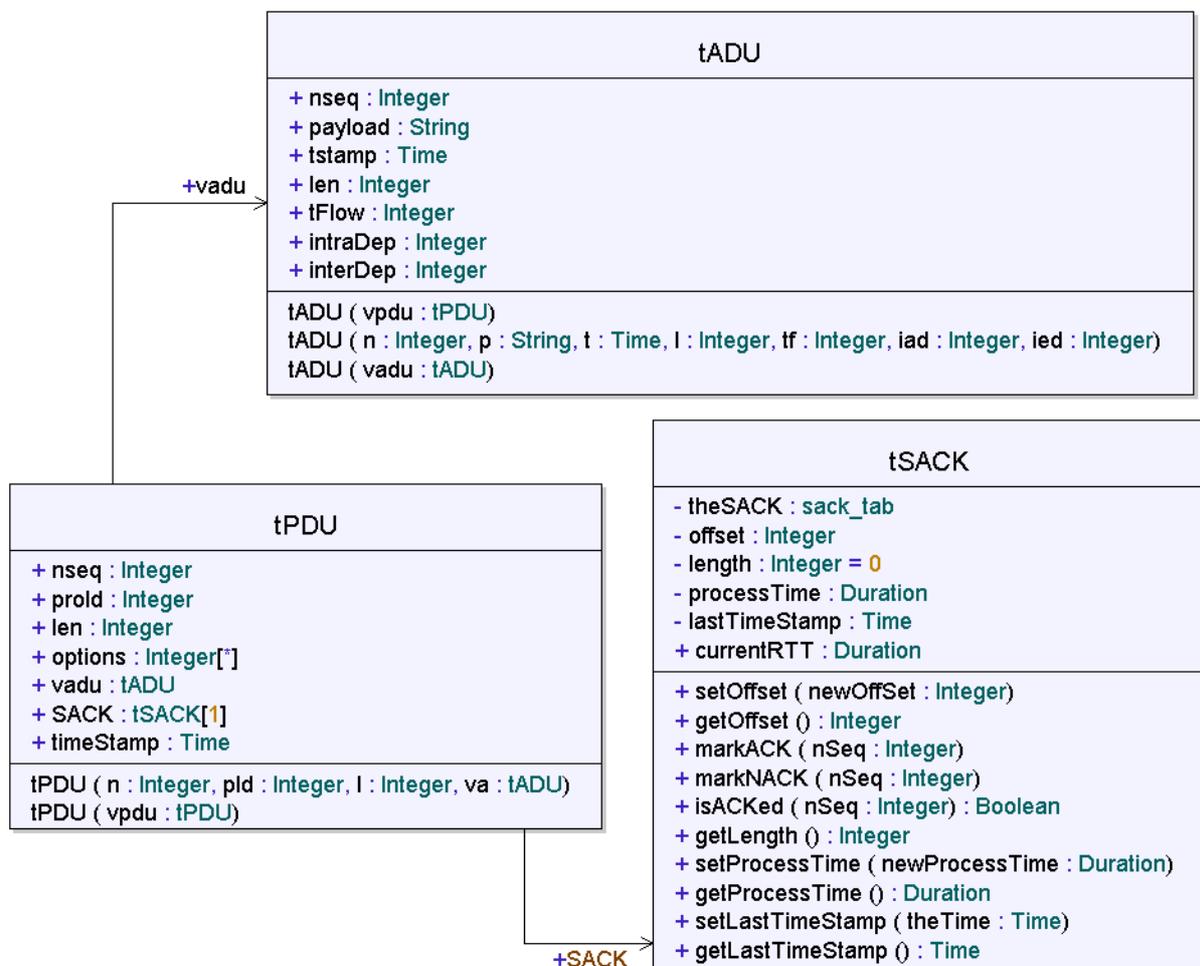


Figure 16 : Architecture de la classe tPDU avec SACK

2. Définition des messages de contrôle locaux

Afin de permettre au plan de gestion d'effectuer les traitements nécessaires sur réception des acquittements sélectifs, il est nécessaire que le plan de contrôle en réception puisse signaler au plan de gestion la réception ces paquets de contrôle. De plus, certaines des données contenues dans l'entête du paquet en question telles le vecteur d'acquiescement lui-même sont indispensables à la prise de décision et doivent elles aussi être communiquées au plan de gestion.

Afin de répondre à ce besoin, un nouveau message a été introduit dans l'interface CI2MG qui définit les messages potentiellement envoyés du plan de contrôle en réception au plan de gestion. Il s'agit du message mg2ciSACK(Integer, tSACK). Il contient comme paramètres le numéro de séquence du PDU reçu ainsi que le vecteur d'acquiescement correspondant.

L'entité de gestion a donc connaissance de l'état des buffers de réception et peut prendre les décisions nécessaires à la correction d'éventuelles pertes le cas échéant.

B. Modifications comportementales

Dans une première approche, un système permettant de conserver le fonctionnement simple décrit auparavant a été envisagé et mis en place. Néanmoins, l'ajout de fonctionnalités à un tel système s'est avéré très complexe du fait de la dualité et l'encombrement croissant des diagrammes par l'ajout d'embranchements conditionnels.

La solution retenue et présentée sur les diagrammes repris dans cette section correspond à la modélisation d'un système fonctionnant sur base du vecteur d'acquiescement sélectif uniquement.

1. Modifications apportés au plan de contrôle.

En ce qui concerne les mécanismes déployés au sein du plan de contrôle, le comportement du plan de contrôle en émission est resté inchangé. Les seules modifications ont été mises en place sur le plan de contrôle en réception (CI).

Le plan de contrôle en réception, tel que présenté en II.B.2.b) page 8, présente des interfaces avec d'une part le plan de contrôle et d'autre part l'entité protocolaire de niveau inférieur. Les modifications apportées se concentrent dans la communication qui prend place entre ce plan et le plan de gestion. En effet, le plan de gestion devra pouvoir être informé de la réception d'un vecteur d'acquiescement ou inversement, le plan de contrôle (CI) devra, sur réception d'un acquiescement sélectif, informer le plan de gestion de cet événement.

Afin de permettre ceci, une exécution conditionnelle au diagramme d'états correspondant aux traitements effectués sur réception d'un nouveau PDU présenté

Figure 17 ci-dessous. En effet, les acquittements sélectifs remplacent l'acquiescement classique, il n'y a donc plus que deux possibilités pour les messages reçus : soit le PDU contient des données applicatives, soit il s'agit d'un PDU de contrôle contenant un acquiescement sélectif. Dans chacun des cas, le signalement adéquat est fait au plan de gestion.

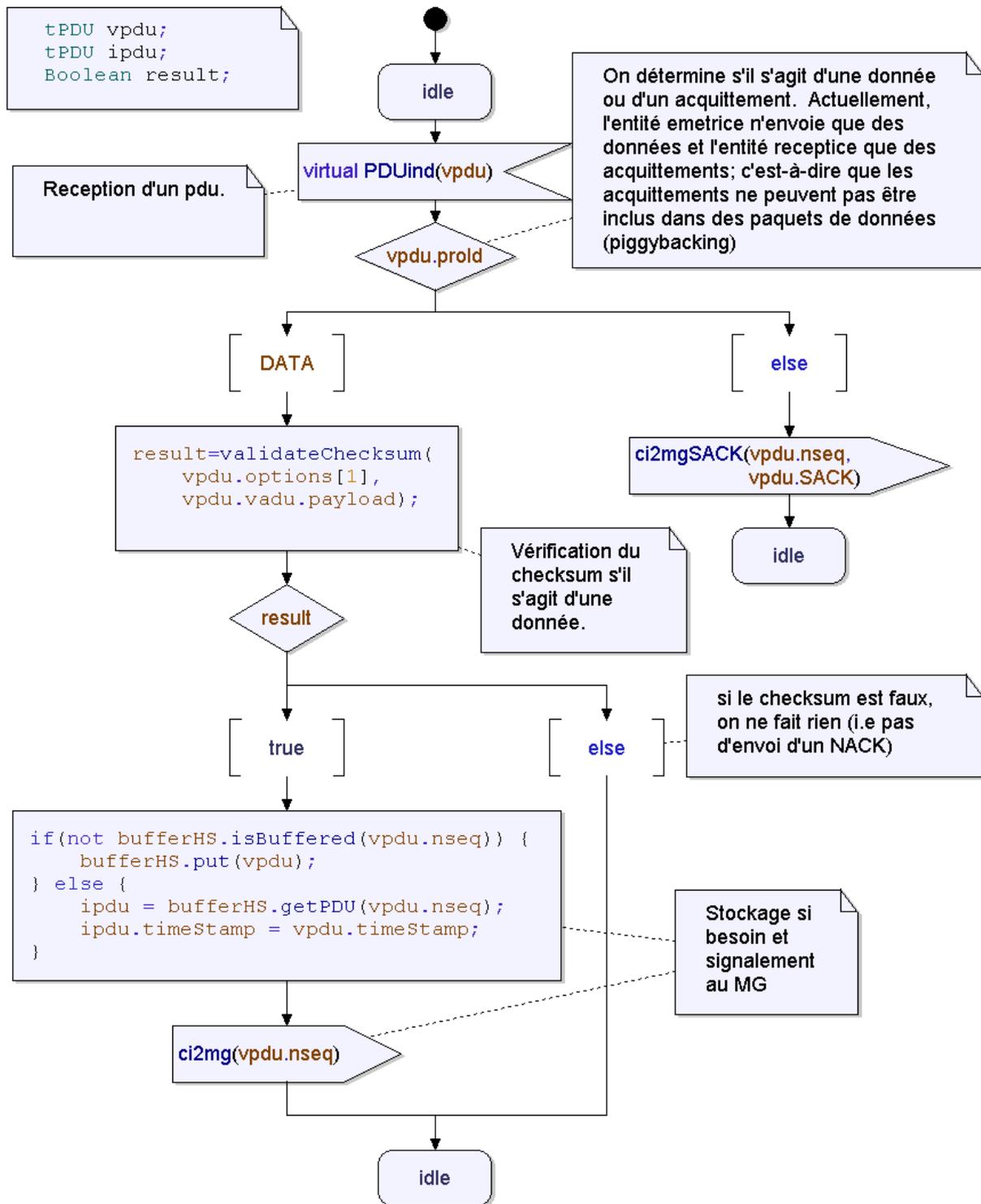


Figure 17 : Réception d'un PDU par le plan de contrôle (CI)

2. Modifications apportés au plan de gestion.

a) Identification Initiale

De par la simplification amenée par la réduction du nombre de types de messages que peuvent s'échanger les différentes entités protocolaires, le comportement initial du plan de gestion tel que présenté en Identification initiale I.B.2.c)(i) a lui aussi été modifié.

On remarque en effet sur la Figure 18 que le message ci2mgGbn(value, nseq) a été remplacé par le message ci2mg(nseq) d'une part, et que le message ci2mgSACK(nseq, ISACK) a été rajouté d'autre part. Ceci correspond au fait que les seuls PDUs de contrôles à prendre en compte sont les PDUs d'acquiescement sélectif.

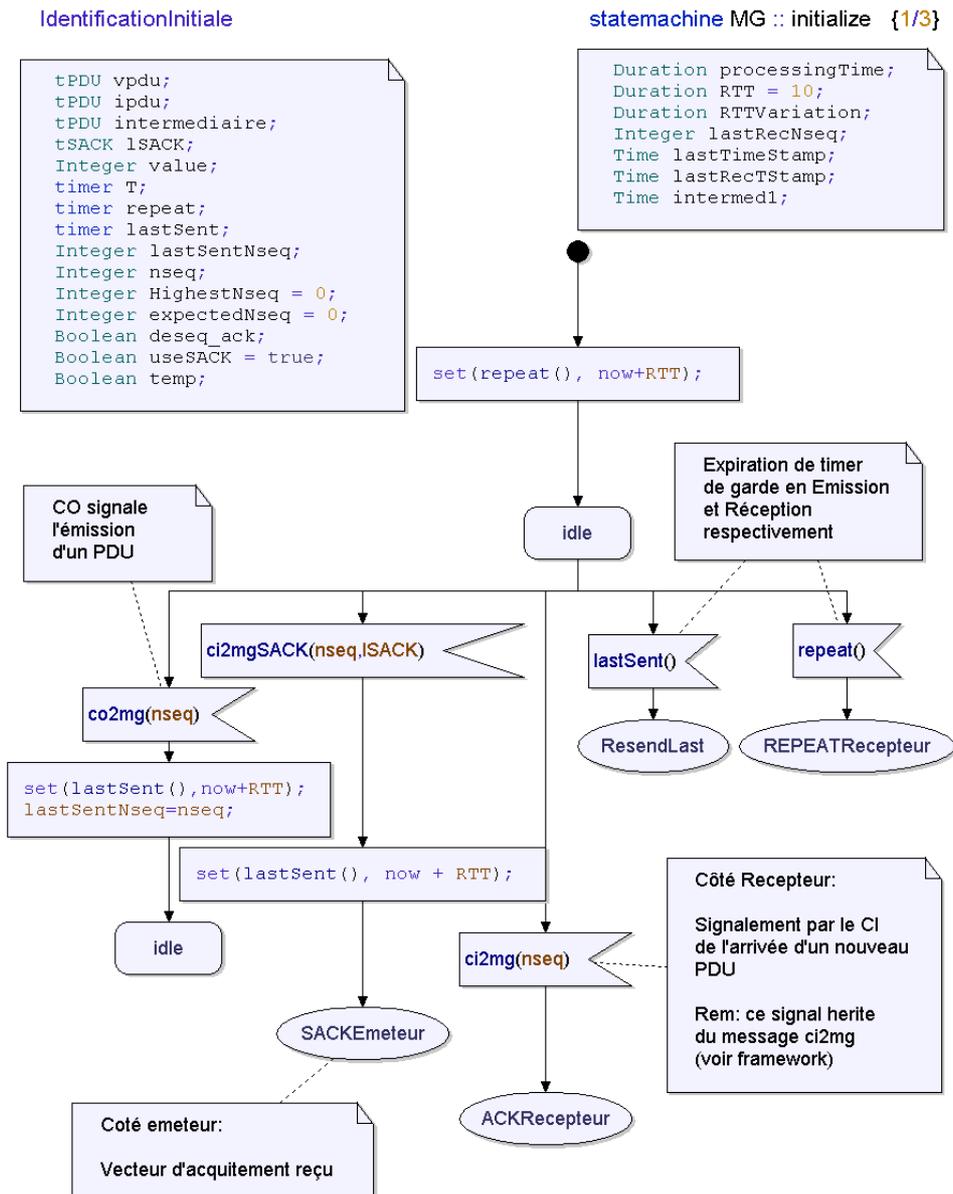


Figure 18 : Identification initiale du plan de gestion (MG)

b) Emission du vecteur d'acquiescement sélectif

Les mécanismes, une fois l'identification initiale terminée, n'ont pas changé dans leur fondement. Seul quelques modifications mineures ont été effectuées dans le but d'arriver à obtenir une meilleure correspondance avec la description faite dans [EXP03] concernant leur mise en œuvre pour FTP.

Tous les mécanismes d'émission d'acquiescement cumulatif ainsi que de demande de répétition sont remplacés par le mécanisme d'émission d'acquiescement sélectif présenté sur la Figure 19 ci-dessous.

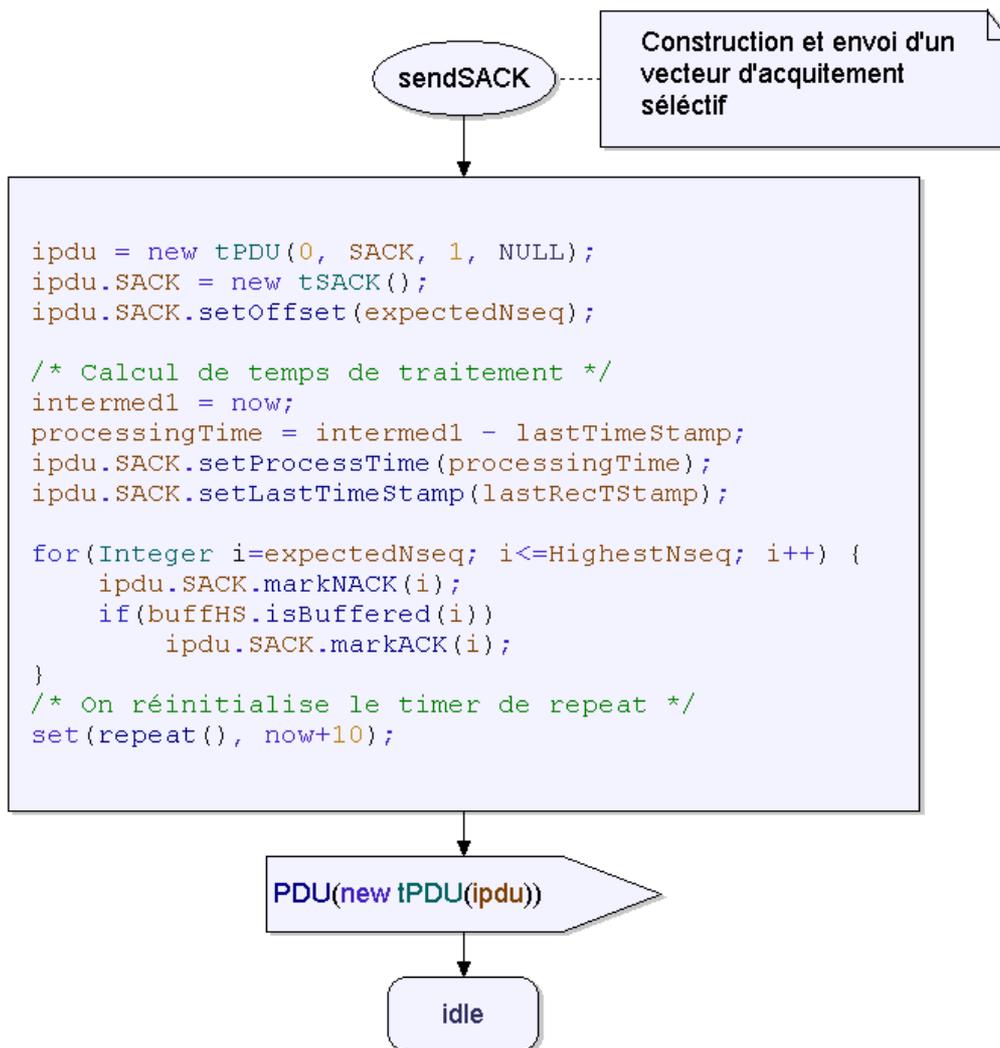


Figure 19 : Construction et émission du SACK

On notera que l'exécution consiste à construire un PDU en lui rajoutant un vecteur d'acquiescement sélectif (SACK) dont l'offset (qui sert d'acquiescement cumulatif pour le mécanisme de réception) est placé au numéro de séquence du prochain PDU attendu. Vient ensuite une partie concernant le calcul du temps de traitement qui entre en jeu dans le calcul du RTT (voir III.B.3 page 35) puis le vecteur d'acquiescement est rempli en fonction des PDUs présents dans les buffers. Le mécanisme s'achève avec la remise de ce PDU à l'entité de niveau inférieur.

c) Réception d'un acquittement sélectif

Sur réception d'un acquittement sélectif, l'hôte émetteur doit effectuer plusieurs traitements afin de supprimer de ses buffers de retransmission tous les PDUs reçus par l'hôte récepteur puis procéder à la retransmission éventuelle des PDUs marqués comme non reçus.

Ce mécanisme complexe est présenté Figure 20 ci-dessous. On notera sur la figure la présence d'instructions propres à la mise à jour du RTT, leur bien fondé sera présenté plus en détail de par la suite (III.B.3).



Figure 20 : Mécanisme déclenché à la réception d'un SACK (MG)

La première action effectuée correspond au traitement de l'acquittement cumulatif présent dans l'acquittement sélectif. En effet, l'hôte émetteur peut supprimer de ses buffers tous les PDUs dont le numéro de séquence est inférieur strictement à cet offset.

Ensuite, pour chacun des PDUs marqués dans le vecteur d'acquittement, le PDU est soit supprimé des buffers de retransmission, soit remis dans les buffers d'émission avec une priorité urgente si le temps écoulé depuis la dernière transmission de ce PDU est suffisamment long.

3. Nouveaux mécanismes ajoutés

Afin de permettre une mise en place du mécanisme qui soit la plus proche de celle décrite dans [EXP03] concernant l'implantation faite du protocole FPTP, plusieurs mécanismes ont dû être développés. Il était effectivement nécessaire pour pouvoir optimiser les retransmissions de PDUs supposés perdus de connaître la valeur moyenne du RTT, temps de parcours aller-retour.

Aussi, afin de servir de base à une éventuelle mesure de performances, certains comportements sont restés optionnels dans la mesure ou ils peuvent être « configurés » par l'intermédiaire d'une variable booléenne.

a) Evaluation du RTT

L'évaluation du RTT pour un système distribué n'est pas triviale. D'une part il est nécessaire que les différents messages échangés présentent un champ de contrôle dans lequel la notion du temps est véhiculée et d'autre part, afin d'éviter l'utilisation d'une horloge globale, seul l'un des deux hôtes se doit d'utiliser le temps absolu.

Le calcul du temps aller retour s'effectue selon l'algorithme suivant :

- Lors de l'émission de chaque PDU, l'hôte émetteur les marque d'une estampille temporelle.
- A chaque réception d'un PDU, le récepteur garde un mémoire du numéro de séquence de ce PDU ainsi que du temps absolu d'arrivée de ce dernier.
- Lors de l'émission d'un SACK, le récepteur transmet à l'émetteur les valeurs de l'estampille temporelle contenue dans le dernier PDU reçu (cette estampille ayant été marquée par l'émetteur lui-même) ainsi que le temps de traitement correspondant à la différence du temps entre le temps actuel et l'heure relevée lors de l'arrivée du dernier PDU reçu.
- Sur réception de ce SACK, l'émetteur peut retrouver le temps de transit aller retour en effectuant une simple soustraction des données fournies dans l'entête du PDU et de la date courante.

Le RTT ainsi calculé est utilisé par l'émetteur pour mettre en place le déclenchement du mécanisme NoFeedback (voir [EXP03]). En effet, c'est lorsque aucun message provenant du récepteur n'a été reçu pendant une période supérieure ou égale au RTT depuis l'envoi du dernier message par l'émetteur que le mécanisme de retransmission spontanée est déclenché.

Aussi, le RTT est utilisé par l'émetteur afin de spécifier l'algorithme de décision pour la retransmission des PDUs dont l'absence des buffers de réception est signalée dans le SACK. Ainsi, seuls les PDUs dont la date de dernière émission est éloignée de plus d'un RTT de l'instant présent sont retransmis. Ceci permet d'éviter une retransmission inutile des PDUs telle par exemple qu'elle est présentée dans la première itération de modélisation de ce document.

b) Contrôle d'erreur agressif

Tel qu'il est spécifié dans le protocole FTP, l'émission d'un vecteur d'acquiescement sélectif se fait à chaque réception d'un PDU détecté comme étant hors séquence. Bien que l'efficacité d'un tel mécanisme dans les environnements très bruités soit certainement très bonne, il en résulte à première vue une mauvaise utilisation des ressources dans le cas contraire.

Afin de permettre une éventuelle comparaison d'efficacité, un autre mécanisme, dans lequel la transmission d'un vecteur d'acquiescement sélectif se fait à intervalle constants (éventuellement liés au RTT) a été mise en place.

Afin de choisir parmi les deux modes, un booléen est présent en tant qu'attribut de la machine à états du plan de gestion. Ce booléen nommé `agressiveEC` déclenche le mécanisme propre à FTP ou le mécanisme temporisé selon qu'il prenne la valeur respectivement de « true » ou « false ».

Bibliographie

- [EXP03] Ernesto J. Exposito G. – Spécification et mise en oeuvre d’un protocole de transport orienté Qualité de Service pour les applications multimédias. – Rapport LAAS N° 03577
- [EXP03a] E. Exposito, M. Gineste, R. Peyrichou, P. Sénac, Michel Diaz, “XQOS: XML-based QoS specification language”, MMM’03 The 9th International Conference on Multi-Media Modeling, January 7-10, 2003, Taiwan.
- [ARM05] F. Armando – Projet de fin d’études – Nouveaux protocoles de transport pour l’adaptabilité au contexte. – INSA-LAAS – Juin 2005
- [Telelogic] Telelogic TAU Generation 2 – <http://www.telelogic.com>

Table des Figures

FIGURE 1 : DIAGRAMME D'ARCHITECTURE DE L'ENTITE "SELECTIVE REPEAT"	7
FIGURE 2 : MACHINE A ETAT DU PLAN D'EMISSION (CO)	8
FIGURE 3 : COMPORTEMENT SUR COMMANDE DU PLAN DE GESTION (MG)	9
FIGURE 4 : COMPORTEMENT DU PLAN DE CONTROLE DE RECEPTION SUR RECEPTION D'UN PDU	10
FIGURE 5 : IDENTIFICATION INITIALE ET "ROUTAGE" DECISIONNEL POUR LE PLAN DE MANAGEMENT (MG)	12
FIGURE 6 : ACKRECEPTEUR	14
FIGURE 7 : EXPECTEDRECEIVED	15
FIGURE 8 : REPEATRECEPTEUR	16
FIGURE 9 : ACKEMETEUR	17
FIGURE 10 : REPEATEMETEUR	18
FIGURE 11 : RESENDLAST	18
FIGURE 12 : DIAGRAMME D'ARCHITECTURE DE TEST	20
FIGURE 13 : DIAGRAMME DE SEQUENCE - REPRISE SUR PERTES	22
FIGURE 14 : DIAGRAMME DE SEQUENCE - TEMPORISATION DE L'EMETTEUR	25
FIGURE 15 : DIAGRAMME DE SEQUENCE - TEMPORISATIONS DU RECEPTEUR	27
FIGURE 16 : ARCHITECTURE DE LA CLASSE TPDU AVEC SACK	29
FIGURE 17 : RECEPTION D'UN PDU PAR LE PLAN DE CONTROLE (CI)	31
FIGURE 18 : IDENTIFICATION INITIALE DU PLAN DE GESTION (MG)	32
FIGURE 19 : CONSTRUCTION ET EMISSION DU SACK	33
FIGURE 20 : MECANISME DECLENCHE A LA RECEPTION D'UN SACK (MG)	34