

①9 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

①1 N° de publication :

2 776 155

(à n'utiliser que pour les  
commandes de reproduction)

②1 N° d'enregistrement national :

98 03274

⑤1 Int Cl<sup>6</sup> : H 04 L 12/56, H 04 L 1/00

①2

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 12.03.98.

③0 Priorité :

④3 Date de mise à la disposition du public de la  
demande : 17.09.99 Bulletin 99/37.

⑤6 Liste des documents cités dans le rapport de  
recherche préliminaire : *Se reporter à la fin du  
présent fascicule*

⑥0 Références à d'autres documents nationaux  
apparentés :

⑦1 Demandeur(s) : FRANCE TELECOM Société ano-  
nyme — FR.

⑦2 Inventeur(s) : KLAY FRANCIS et RABADAN CHRIS-  
TOPHE.

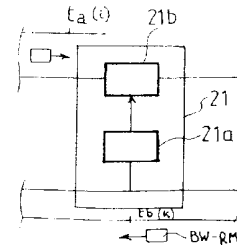
⑦3 Titulaire(s) :

⑦4 Mandataire(s) : CABINET LE GUEN ET MAILLET.

⑤4 PROCÉDE DE CONTROLE DE LA CONFORMITE DE DEBIT DES CELLULES DE DONNEES EMISES PAR UN  
TERMINAL SOURCE EN COMMUNICATION AVEC UN TERMINAL DESTINATAIRE.

⑤7 La présente invention concerne un procédé de contrô-  
le de la conformité du débit des cellules de données émises  
par un terminal source en communication avec un terminal  
destinataire.

Il consiste essentiellement, à l'arrivée d'une cellule de  
gestion de ressources retour (BW-RM) d'ordre  $k$  à ladite unité  
d'interface (21), à déterminer la date de programmation  
( $t_p(k)$ ) de la prise en compte par ledit processus de contrôle  
(21b) de la demande de débit (ER ( $k$ )) contenue dans le  
champ de ladite cellule (BW-RM).



FR 2 776 155 - A1



La présente invention concerne un procédé de contrôle de la conformité d'un flux de cellules émis par une application source d'un réseau à haut débit, par exemple d'un réseau utilisant le mode de transfert dit ATM (Asynchronous Transfer Mode). Plus précisément, l'invention est appliquée aux sources dont le mode de transfert est un mode à partage de ressources disponibles, encore appelé dans le domaine de la technique mode ABR (Available Bits Rate).

On rappelle à toutes fins utiles que le mode de transfert ATM consiste à transporter dans un réseau de données des paquets d'informations de taille fixe appelés cellules. Un terminal source envoie ainsi des cellules de données dans le réseau qui les transporte jusqu'à un terminal destinataire suivant un protocole de communication préétabli. Ces cellules sont ainsi dirigées et orientées au travers d'un canal préétabli entre les deux terminaux. Ce canal, qui est aussi appelé une connexion, est notamment caractérisé par son débit qui doit être suffisant pour transporter toute l'information émise par la source. Pour ce faire, à chaque élément traversé par ledit canal, un débit est réservé à ce canal.

L'intérêt majeur des réseaux à haut débit basés sur le mode de transfert asynchrone ATM, outre l'accès à des capacités de transmission et de commutation pouvant supporter des hauts débits, est de pouvoir intégrer sur un même support de transmission des applications dont les besoins en terme de qualité de service et de ressources sont différents. Ces réseaux ATM permettent, de plus, l'utilisation d'applications qui émettent des cellules avec des débits variables dans le temps, comme des applications de type vidéo ou de type Internet, ce qui entraîne une variation dans le temps de la consommation des ressources du réseau utilisées par ces applications. Pour intégrer toutes ces applications, la technique ATM doit satisfaire aux trois critères suivants : i) offrir le débit nécessaire à toute application connectée, ii) garantir le niveau de qualité de service requis par toute application connectée, iii) optimiser l'utilisation des ressources disponibles dans le réseau.

Les deux premiers points permettent d'établir les besoins d'une application donnée, qui après avoir été négociés et acceptés par le réseau lors de la procédure d'appel, constituent ce qui est appelé le contrat de trafic entre l'utilisateur de l'application envisagée, et le réseau. Ce contrat qui est défini par les deux interfaces appelées dans le domaine de la technique UNI (User Network Interface), entre le réseau et chacune des deux applications définit les caractéristiques de trafic de l'application source et la qualité de service QoS requise pour son bon fonctionnement.

Ce contrat garantit à l'application source la qualité de service QoS demandée si celle-ci respecte les caractéristiques de trafic qu'elle a négociées au moment de l'appel.

La présente invention s'applique pour l'essentiel au mode de transfert dit à partage des ressources disponibles ou ABR (Available Bit Rate). Ce mode de transfert  
5 ATM est généralement destiné aux applications capables d'adapter leur débit d'émission de cellules en cours de connexion. Parmi ces applications, on va par exemple trouver des applications de transfert de données : interconnexion ou émulation de réseaux locaux du type dit LAN (Local Area Network), ou encore l'accès à des données  
distantes.

10 Ce mode de transfert dont les mécanismes de contrôle sont mis en œuvre au niveau de la couche ATM du réseau a notamment été développé dans l'objectif d'utiliser plus efficacement la bande passante disponible dans un réseau ATM tout en garantissant un débit et un niveau de qualité de service QoS suffisants au bon  
fonctionnement des applications connectées.

15 Ce mode ABR est d'abord un protocole de communication qui, pour une connexion établie entre un terminal source et un terminal destinataire, permet de partager avec d'autres connexions les ressources disponibles dans chacun des éléments  
du réseau traversé par ladite connexion. Ces ressources disponibles correspondent à la somme des débits qui n'est utilisée par aucune des connexions sur l'élément considéré.  
20 Pour effectuer ce partage, le débit de chacune des sources qui fonctionnent également dans ce mode ABR est régulé en fonction, d'une part, de la quantité de ressources disponibles dans chaque élément de réseau et, d'autre part, du partage qui en est fait  
entre les connexions fonctionnant également selon le mode ABR. Le mode ABR fait donc intervenir un mécanisme de contrôle du débit des seules sources qui fonctionnent  
25 en mode ABR en fonction du débit alloué par les éléments de réseau pour chaque connexion fonctionnant dans ce mode. Ce mécanisme de contrôle permet d'ajuster dynamiquement le débit des sources en fonction des ressources qui sont disponibles  
dans le réseau.

30 Ce mécanisme est le suivant. Chaque source émet périodiquement une cellule de gestion de ressources aller encore appelée dans le domaine de la technique FW-RM comme "Forward-Ressource Management" et est retournée par l'application  
destinataire sous la forme d'une cellule de gestion de ressources retour encore appelée BW-RM comme "Backward-RM". Chaque élément de réseau traversé par l'une ou  
l'autre cellule a la possibilité de lui indiquer explicitement en chargeant un mot

approprié dans un de ses champs, son état de congestion. Cette état se présente sous la forme d'un débit noté par la suite ER, comme Explicit Rate. Après cet aller et retour, la cellule B-RM porte des directives de congestion qui permettent à la source d'adapter la valeur de son débit, par exemple entre une valeur minimum MCR et une valeur maximum PCR (respectivement Minimum Cell Rate et Peak Cell Rate) qui ont été  
5 négociées dans le contrat de trafic.

Ainsi, le débit réel d'émission de chaque source est contrôlé afin de vérifier qu'elles ne dépassent le débit qui lui a été réservé dans le réseau. En effet, si une source malveillante émet à un débit supérieur à celui qui lui a été alloué ou autorisé, elle peut  
10 venir saturer les capacités des éléments de réseau. Elle met ainsi en péril le transfert d'informations et donc la qualité de service de toutes les autres connexions qui traversent les mêmes éléments de réseau.

Ce contrôle est appelé contrôle de conformité et est mis en œuvre à l'interface entre le réseau et une source, soit à l'interface UNI (User-Network Interface) au moyen  
15 d'un opérateur dit UPC (Usage Parameter Control), soit à l'interface NNI (Network-Network Interface) au moyen d'un opérateur dit NPC (Network Parameter Network). Il est mis en œuvre au moyen d'un algorithme de contrôle de conformité qui est appelé, de manière générale GCRA comme Generic Cell Rate Algorithm. Appliqué au mode de transfert ABR pour lequel, comme on vient de le voir, le débit réservé change tout au  
20 long de la connexion en fonction des ressources disponibles dans les éléments de réseau, il est appelé algorithme DGCRA comme Dynamic Generic Cell Rate Algorithm.

A la Fig. 1, on peut voir un premier terminal 10, dit terminal source, connecté à un réseau 20. Un second terminal 30, dit terminal destinataire est connecté également  
25 au réseau 20. Dans la suite de cette description, on supposera que les terminaux 10 et 30 sont en communication l'un avec l'autre. De plus, ils sont l'un et l'autre du type fonctionnant en mode ABR à partage de ressources. Un canal virtuel est donc établi entre eux pour permettre la transmission de cellules de données émises par le terminal source 10. Des unités d'interface usager-réseau 21 (dit UNI dans le domaine de la  
30 technique) sont prévues entre le réseau 20 et les terminaux 10, 30. C'est dans ces unités 21 que sont mises en œuvre les fonctions des opérateurs UPC ainsi que l'algorithme DGCRA qui est maintenant décrit en relation avec la Fig 1.

On notera que le réseau 20 peut être constitué de plusieurs réseaux interconnectés par l'intermédiaire d'unités d'interface réseau-réseau (non représentées)

qui sont dits, dans le domaine de la technique, NNI qui mettent en œuvre les fonctions des opérateurs NPC ainsi que l'algorithme DGCRA.

Bien que chaque terminal usager peut être à la fois émetteur et récepteur de cellules de données, on considérera, par simplification un terminal source 10 qui émet  
5 des cellules de données vers un terminal destination 30. Les cellules de données, émises du terminal source 10, étant contrôlées à l'interface usager - réseau (UNI) 21 entre la source 10 et le réseau 20 ou encore à l'interface réseau - réseau (NNI) entre deux domaines différents, si c'est le cas.

Ainsi, pour la mise en œuvre de cet algorithme, le terminal source 10 émet  
10 périodiquement une cellule de gestion de ressources allée FW-RM à destination du terminal 30, lequel renvoie une cellule de gestion de ressources retour BW-RM à destination du terminal 10. Il émet par exemple une cellule de gestion de ressources toutes les 32 cellules de données. Les indications de congestion concernant les éléments traversés par la connexion sont remontées par les cellules BW-RM qui  
15 informent ainsi l'unité 21 côté terminal source du nouveau débit maximum alloué par les éléments du réseau. Le terminal source 10 est, à la réception de la cellule de gestion de ressources B-RM, informé de ce nouveau débit. Généralement, il applique le nouveau débit qu'à partir des prochaines cellules générées qui n'atteignent alors l'unité d'interface 21 qu'après un délai correspondant au temps de propagation des cellules  
20 entre le terminal source et l'unité d'interface 21.

On comprendra qu'il n'est pas possible d'appliquer, dans la fonction de contrôle du flux de cellules de données émis par le terminal 10, un changement de débit dès la réémission de la cellule de gestion de ressources retour BW-RM par l'unité d'interface 21. Il faut, en effet, attendre, d'une part, le temps pour que le terminal source 10 prenne  
25 connaissance du nouveau débit, temps correspondant au temps de propagation entre l'unité d'interface 21 et le terminal source 10 et, d'autre part, le temps pour que les cellules de données émises par le terminal 10 en respectant ce nouveau débit arrivent à l'unité d'interface 21, temps qui correspond alors au temps  $t_1$  de propagation entre le terminal source 10 et l'unité d'interface 21. Le temps total d'attente correspond donc à  
30 un temps de aller et retour  $t_2$ , dit également Round Trip Time (RTT), de propagation entre l'unité d'interface et le terminal source 10.

Ces temps de propagation dans le réseau en amont de l'interface varient en fonction de la charge du terminal source 10 et du réseau d'accès. Néanmoins, la variation du délai de propagation  $t_1$  entre le terminal source 10 et l'unité d'interface 21

peut être borné supérieurement par la variation du délai de cellule CDV (Cell Delay Variation) (notée également  $\tau_1$ ) tolérée à l'unité d'interface 21 qui est fixe pour la plage de débits compris entre le débit minimum MCR et le débit maximum PCR.

Quant au temps de propagation aller et retour  $t_2$ , il est compris entre les délais aller et retour maximum  $\tau_2$  et minimum  $\tau_3$ . Ainsi, un changement de débit va être effectif à l'unité d'interface 21 dans un intervalle de temps compris entre  $[\tau_3, \tau_2]$  après le départ de la cellule de gestion de ressources BW-RM correspondante de l'unité d'interface 21.

Différents algorithmes de conformité DGCRA ont été proposés. Le premier fut l'algorithme DGCRA dit A décrit au forum ATM par Berger & al.

A l'arrivée au temps noté  $t_a(i)$  d'une cellule de données d'ordre  $i$  émise par le terminal source 10, l'algorithme DGCRA A va déterminer, selon un premier processus de programmation représenté par une boîte 21a à la Fig. 1a, le débit de contrôle qu'il va appliquer dans un second processus de contrôle représenté par une boîte 21b à la Fig. 1a.

Pour ce processus de programmation, l'algorithme de conformité DGCRA A, pour chaque cellule de données d'ordre  $i$  arrivant à l'unité d'interface 21, au temps  $t_a(i)$ , met en œuvre une fonction de recherche du débit de valeur maximum dans une liste de demandes de débit. Chaque demande de débit de cette liste correspond à une demande portée par une cellule de gestion retour BW-RM arrivée précédemment à l'unité d'interface 21. Cette cellule BW-RM est partie de l'unité d'interface 21 à un temps précédent qui se trouve dans un intervalle de temps compris entre le temps actuel diminué du délai maximum aller-retour  $\tau_2$  et le temps actuel diminué du délai minimum aller-retour  $\tau_3$ .

Quant au processus de contrôle 21b, il applique le débit de contrôle  $ACR(t_a(i))$  à la cellule  $i$  arrivée à l'unité d'interface 21 au temps  $t_a(i)$  qui est défini par:

$$ACR(t_a(i)) = \text{MAX}(ER(i_{\text{max}}), ER(j))$$

où  $ER(i_{\text{max}})$  correspond au débit porté par la cellule BW-RM d'ordre  $i_{\text{max}}$  partie de l'unité d'interface 21 au temps  $t_b(i_{\text{max}})$  précédent le temps  $(t_a(i) - \tau_2)$  et  $ER(j)$  représente l'ensemble des valeurs de débit portées par les cellules B-RM parties de l'interface aux temps  $t_b(j)$  tel que  $t_a(i) - \tau_2 < t_b(j) < t_a(i) - \tau_3$ .

Alors qu'il peut n'y avoir qu'un seul changement de débit pour chaque cellule BW-RM et qu'une cellule de gestion de ressources RM n'est émise qu'une toutes les 32 cellules de données, l'algorithme DGCRA décrit ci-dessus recalcule à chaque arrivée d'une cellule de données le débit à appliquer au processus de contrôle de conformité.

5 Il détermine le débit à contrôler à un instant donné en recherchant dans une liste les demandes de changements de débit survenues dans le passé. Comme il ne programme pas dans le futur les dates de changements de débit, il ne sait pas *a priori* quand commencera et finira l'application d'un débit particulier au contrôle de conformité du flux de cellules. De ce fait, l'algorithme A recalcule un débit à contrôler  
10 à chaque nouvelle arrivée d'une cellule de données à l'unité d'interface aux instants  $t_a(k)$ . Cependant, un changement de débit est conditionné uniquement par le départ de l'unité d'interface d'une cellule de gestion de ressources retour B-RM, dont la fréquence est beaucoup plus faible que celle des cellules de données (par défaut: une cellule RM pour 32 cellules de données). La détermination de la valeur du débit à contrôler pour  
15 chaque arrivée d'une cellule de données complexifie fortement la réalisation d'un tel processus.

De plus, l'algorithme A utilise une liste non bornée d'événements qui correspondent aux indications de changements de débit survenues pendant le délai maximum de propagation RTT  $\tau_2$ . Lors d'une implémentation de cet algorithme, il sera  
20 nécessaire de limiter physiquement la taille de la mémoire nécessaire au stockage des informations de demande de débit  $(ER(j), t_b(j))$ . Comme cet algorithme ne définit pas de règles de réduction du nombre d'informations pertinentes stockées, il n'est pas robuste contre d'éventuels débordements de ladite mémoire. Lorsque le nombre de changements de débit survenus sur une durée  $\tau_2$ , dépasse la capacité de stockage de la  
25 mémoire, certains changements de débit peuvent être perdus et provoquer ainsi un contrôle de conformité défaillant. Par exemple, si l'on perd une demande de débit correspondant à une augmentation de débit, le contrôle de conformité peut alors s'effectuer à un niveau bas de débit provoquant inopportunément la destruction des cellules de données transmises à débit supérieur à ce niveau mais conforme à  
30 l'indication d'augmentation de débit.

Pour résoudre ce problème, on a prévu des algorithmes à deux mémoires. Ces algorithmes gèrent une liste des débits à programmer dans le futur. Cette liste est limitée à deux demandes de changement de débit.

L'inconvénient majeur de ces algorithmes concerne leur faible capacité à enregistrer les demandes de changement de débit survenues dans un intervalle de temps donné. Lorsque plus de deux demandes de changements interviennent dans un intervalle de temps  $\tau_2$ , les algorithmes à deux mémoires effectuent une sélection des débits à programmer. Cette sélection produit une élimination de certains débits qui provoque une surévaluation du débit contrôlé. Cette surévaluation peut être fréquente lorsque les délais de propagation entre la source et l'unité d'interface sont grands par rapport au temps entre deux cellules de gestion de ressources RM. Ceci est le cas lorsque la fonction de contrôle est loin de la source, ce qui est le cas des fonctions NPC (Network Parameter Control). De plus, la surévaluation peut être importante lorsque les ressources disponibles dans le réseau varient beaucoup d'une cellule de gestion de ressources RM à la suivante. En conséquence, le contrôle de conformité peut être réalisé à un niveau de débit supérieur à celui réellement autorisé par le réseau.

Le but de l'invention est de proposer un procédé de contrôle de la conformité d'un flux de cellules de données émis par une source fonctionnant dans un mode à partage de ressources disponibles ABR qui soit implémentable de manière plus simple que pour l'algorithme A et dont le comportement soit le plus proche du comportement de référence donné par cet algorithme A. Le but est donc de prévoir un tel procédé qui permette de contrôler beaucoup plus strictement et précisément le débit des sources ABR que les algorithmes de l'art antérieur, notamment des algorithmes à deux mémoires. Il est ainsi d'assurer une meilleure protection du réseau contre des sources malveillantes qui chercheraient à émettre à des débits supérieurs à celui autorisé.

Afin d'atteindre ces buts, un procédé de contrôle de la conformité du débit des cellules de données émises par un terminal source en communication avec un terminal destinataire selon l'invention est caractérisé en ce que ledit processus de programmation consiste, à l'arrivée d'une cellule de gestion de ressources retour à ladite unité d'interface, à déterminer la date, dite date de programmation ( $t_p(k)$ ), de la prise en compte par ledit processus de contrôle de la demande de débit contenue dans le champ de ladite cellule, ladite date de programmation étant égale à ladite date d'arrivée ( $t_b(k)$ ) retardée du délai minimum aller-retour ( $\tau_3$ ) entre l'interface et le terminal, dite date de programmation la plus proche, lorsque ladite prise en compte à cette date de programmation la plus proche aboutit à une augmentation de débit programmé et étant supérieur au temps de programmation le plus proche et inférieur à ladite date d'arrivée ( $t_b(k)$ ) retardée d'un délai maximum aller-retour ( $\tau_2$ ), ( $t_b(k) + \tau_2$ )

étant dite date de programmation la plus éloignée, si sa prise en compte audit temps de programmation le plus proche ( $t_b(k) + \tau_3$ ) avait abouti à une diminution de débit et, à transmettre, à l'échéance du temps de programmation ( $t_p(k)$ ), ladite valeur de débit à contrôler ( $ACR(t_p(k))$ ) audit processus de contrôle pour contrôle du flux émis par ledit terminal source.

Selon une autre caractéristique de l'invention, il consiste à tenir une liste d'événements de programmation, chaque événement étant défini par un couple de valeur dont la première représente une valeur de débit  $ACR(t_p(k))$  et dont la seconde représente la date de programmation  $t_p(k)$  de ladite valeur de débit  $ACR(t_p(k))$ , et ladite liste étant ordonnée suivant ladite date de programmation des événements.

Selon une autre caractéristique de l'invention, il consiste pour connaître le débit programmé ( $ACR(t_p^{\sim}(k))$ ) à la date d'arrivée  $t_b(k)$  retardée de la durée minimale d'aller retour  $\tau_3$  ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ), à définir une fonction  $ACR(l, t)$  qui, à partir des événements présents dans la liste  $l$ , détermine la valeur du débit programmé à un temps quelconque  $t$  de la manière suivante :

si, dans ladite liste  $l$ , il existe au moins un événement de programmation  $\{ACR(t_p), t_p\}$  prévu à un temps de programmation  $t_p$  compris entre le temps présent  $t_0$  et le temps  $t$ , ladite valeur est égale au débit de programmation de cet événement  $ACR(t_p)$  et, s'il y en a plusieurs, elle est égale au débit de programmation du dernier de ces événements, et

si, toujours dans cette liste, il n'existe pas d'événements programmés entre le temps présent  $t_0$  et  $t$ , ladite valeur est égale à la valeur du débit contrôlé actuellement dans le processus de contrôle.

Selon une autre caractéristique de l'invention, il consiste à :  
à l'arrivée à l'unité d'interface (21) auquel est connecté ledit terminal source, d'une cellule de gestion de ressources retour (BW-RM) de numéro de référence ( $k$ ), à comparer le débit ( $ER(k)$ ) que porte ladite cellule avec le débit programmé ( $ACR(l, t_p^{\sim}(k))$ ) à l'instant de programmation le plus proche, puis

si ladite comparaison effectuée à chaque arrivée d'une cellule de gestion de ressources (BW-RM) montre que le débit demandé  $ER(k)$  est supérieur au débit programmé au temps de programmation le plus proche ( $ACR(l, t_p^{\sim}(k))$ ), supprimer, de ladite liste  $l$ , tout événement de programmation de débit prévu à un temps supérieur ou égal au temps de programmation le plus proche ( $t_p^{\sim}(k)$ ) et rajouter, dans ladite liste  $l$ , un événement dont le débit contrôlé est égal à la valeur du débit  $ER(k)$  porté par ladite

cellule et dont le temps de programmation est le temps de programmation le plus proche ( $t_p^-(k) = t_b(k) + \tau_3$ ),

si ladite comparaison montre que le débit demandé  $ER(k)$  est égal au débit programmé  $ACR(l, t_p^-(k))$  au temps programmation le plus proche, supprimer, de la liste  $l$ , tout événement de programmation de débit dont le temps de programmation est supérieur au temps de programmation le plus proche ( $t_p^-(k) = t_b(k) + \tau_3$ ),

si ladite comparaison montre que le débit demandé  $ER(k)$  est inférieur au débit programmé  $ACR(l, t_p^-(k))$ , rechercher, dans ladite liste  $l$ , s'il existe un événement dont le temps de programmation ( $t'_p$ ) est le plus petit temps de programmation dans la liste supérieur au temps le plus proche et dont le débit est inférieur ou égal au débit demandé  $ER(k)$ ,

si cette recherche montre que cet événement n'existe pas, rajouter, dans la liste  $l$ , un événement dont le débit est égal à la valeur du débit demandé  $ER(k)$  et dont le temps de programmation est le temps de programmation le plus éloigné ( $t_p^+(k) = t_b(k) + \tau_2$ ) et,

si cette recherche montre que cet événement existe, supprimer, de la liste  $l$ , tout événement de programmation dont le temps de programmation est supérieur ou égal au temps de programmation ( $t'_p$ ) dudit événement ainsi trouvé et rajouter, dans la liste  $l$ , un événement dont le débit est égal à la valeur du débit demandé  $ER(k)$  et dont le temps de programmation est égal au temps de programmation dudit événement trouvé ( $t'_p$ ),

et, à changer le débit de contrôle du processus de contrôle (21b) avec le débit du premier événement  $\{ACR(1), t_p(1)\}$  programmé dans la liste d'événements de programmation lorsque le temps de programmation  $t_p(1)$  dudit premier événement arrive à échéance.

Selon une autre caractéristique de l'invention, il consiste à limiter le nombre d'événements stockés dans ladite liste  $l$  à un nombre fini  $N$  d'événements égal ou supérieur à deux, et, avant de rajouter un nouvel événement dans ladite liste, à vérifier si le nombre d'événements contenus dans ladite liste n'est pas supérieure audit nombre fini et, dans ce cas, à mettre alors en œuvre une fonction de réduction du nombre d'événements contenus dans la liste.

Selon une autre caractéristique de l'invention, ladite fonction de réduction consiste, lorsque le débit demandé ( $ER(k)$ ) que porte la cellule de gestion de ressources retour (BW-RM) d'ordre  $k$  arrivée à l'instant ( $t_b(k)$ ) est égal au débit programmé

( $ACR(l, t_p^{\sim}(k))$ ) à l'instant de programmation le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ), à supprimer, de la liste  $l$ , tout événement de programmation de débit dont le temps de programmation est supérieur ou égal au temps de programmation le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ) et à rajouter dans la liste  $l$  un événement dont le débit est égal à la valeur du débit ( $ER(k)$ ) porté par ladite cellule de gestion de ressources retour (BW-RM) et dont le temps de programmation est le temps de programmation le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ).

Selon une autre caractéristique de l'invention, il consiste, lors du rajout dans la liste d'un événement pour un temps correspondant au temps le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ), à vérifier, avant de rajouter ledit événement, si le nombre d'événements dans la liste  $l$  est égal audit nombre fini  $N$  et si le temps de programmation du dernier événement dans la liste ( $t_p(N)$ ) est inférieur ou égal au temps de programmation le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ), à définir le temps de programmation de l'événement à rajouter ( $t_p(k)$ ) comme étant égal audit temps de programmation du dernier événement de la liste, à supprimer le dernier événement de la liste, et à rajouter ledit événement à rajouter à la place du dernier événement dans la liste  $l$ , sinon, si le nombre d'événements dans la liste est inférieur audit nombre fini  $N$ , à définir le temps de programmation de l'événement à rajouter comme étant égal au temps de programmation le plus proche ( $t_p^{\sim}(k) = t_b(k) + \tau_3$ ), et à rajouter ledit événement dans la liste.

Selon une autre caractéristique de l'invention, il consiste, lors d'un rajout d'un événement dans la liste pour un temps correspondant au temps le plus éloigné, à vérifier, avant de rajouter ledit événement, si le nombre d'événements dans la liste  $l$  est égal audit nombre fini  $N$ , si c'est le cas il consiste, à remplacer le temps de programmation ( $t_p(k)$ ) de l'événement à rajouter par le temps de programmation du dernier événement de la liste ( $t_{pi}(N)$ ) si le débit dudit événement à rajouter ( $ER(k)$ ) est supérieur ou égal au dernier débit programmé dans la liste  $ER(N)$ , à programmer à la place de l'avant-dernier événement de la liste (d'ordre  $N-1$ ), un événement dont le temps de programmation est celui de l'avant dernier événement de la liste  $l$  ( $ACR(N-1), t_{pi}(N-1)$ ), et le débit est le plus grand débit entre les débits des deux derniers événements de la liste  $l$  ( $ACR(N-1) = \text{Max}(ACR(N-1), ACR(N))$ ), puis à rajouter l'événement à rajouter à la place du dernier événement dans la liste  $l$ .

Selon une autre caractéristique de l'invention, il consiste, lors du rajout d'un événement dans la liste  $l$ , à vérifier si le nombre d'événement contenu dans la liste  $l$  ne

dépasse pas ledit nombre fini, et si tel est le cas, à rechercher dans la liste / deux événements successifs pour lesquels le produit de l'augmentation ou de la diminution de débit par la mesure de l'intervalle de temps compris entre leurs deux temps de programmation soit minimum puis à leur substituer un événement dont le débit est le plus grand des deux débits et le temps de programmation est le premier des deux temps de programmation.

Les caractéristiques et avantages de l'invention mentionnées ci-dessus ainsi que d'autres apparaîtront plus clairement à la lecture de la description suivante d'exemples de réalisation, ladite description étant faite en relation avec les dessins joints parmi lesquels :

la Fig. 1 est un schéma d'un réseau auquel s'applique la présente invention,

la Fig. 1a est un schéma explicatif des processus mis en œuvre dans une unité d'interface utilisateur-réseau selon l'art antérieur,

la Fig. 1b est un schéma explicatif des processus mis en œuvre dans une unité d'interface utilisateur-réseau selon l'invention,

la Fig. 2 est un organigramme illustrant un premier mode de réalisation d'un procédé de programmation selon l'invention,

la Fig. 3 est un diagramme illustrant le déroulement d'un procédé selon le premier mode de réalisation de l'invention pour un exemple de six cellules arrivant à l'interface utilisateur-réseau,

la Fig. 4 est un organigramme illustrant un second mode de réalisation d'un procédé de programmation selon l'invention,

la Fig. 5 est un diagramme illustrant le déroulement d'un procédé selon le second mode de réalisation de l'invention pour un exemple de huit cellules arrivant à l'interface utilisateur-réseau,

la Fig. 6 est un organigramme illustrant un troisième mode de réalisation d'un procédé de programmation selon l'invention, et

la Fig. 7 est un diagramme illustrant le déroulement d'un procédé selon le troisième mode de réalisation de l'invention pour un exemple de huit cellules arrivant à l'interface utilisateur-réseau.

A l'instar des procédés selon l'état de la technique, le procédé de contrôle de la conformité selon l'invention consiste à mettre en œuvre, dans une unité 21 d'interface utilisateur-réseau, dite UNI dans le domaine de la technique (voir Fig. 1) ou, de même dans une unité d'interface réseau-réseau, dite NNI dans le domaine de la technique (non

représentée), deux processus 21a et 21b qui sont exécutés indépendamment l'un de l'autre. Ces deux processus 21a et 21b tels qu'ils sont arrangés dans le cadre de la présente invention sont maintenant décrits en relation avec la Fig. 1b.

5 Le premier processus est un processus de contrôle 21b proprement dit qui vérifie que la cellule de donnée d'ordre  $i$  est, à son arrivée à l'unité d'interface 21 au temps  $t_a(i)$ , conforme aux paramètres de trafic que l'on écrit  $(ACR(t_a(i)), \tau_1)$ , où  $ACR(t_a(i))$  est le débit à contrôler à l'instant de l'arrivée  $t_a(i)$  de la cellule de donnée d'ordre  $i$  et  $\tau_1$  la variation du délai cellule tolérée CDV associée. Ce premier processus ne fait pas à proprement partie de la présente invention et ne sera donc pas décrit plus en détail.

10 Le second processus est un processus de programmation représenté par une boîte 21a à la Fig. 1b qui, à l'arrivée au temps  $t_b(k)$  d'une cellule de gestion de ressources retour BW-RM d'ordre  $k$ , détermine, à partir d'une demande de débit  $ER(k)$  qu'un champ de ladite cellule BW-RM contient, la valeur du débit à contrôler  $ACR(t_p(k))$  à un instant futur  $t_p(k)$  que l'on nommera, dans la suite de la description, temps de programmation. Au temps  $t = t_p(k)$ , le processus de programmation transmet cette  
15 valeur  $ACR(t_p(k))$  au processus de contrôle 21b.

On rappelle que chaque cellule de gestion retour BW-RM comprend un champ spécifique qui porte une demande de débit que l'on notera, dans la suite de la description,  $ER(k)$ ,  $k$  étant l'ordre de ladite cellule.

20 Selon l'invention, une demande de débit qui correspond à une augmentation par rapport au débit programmé au temps  $t_b(k)$  retardé du délai minimum aller-retour  $\tau_3$ , sera prise en compte au temps de départ  $t_b(k)$  retardé du délai minimum aller-retour  $\tau_3$ . A l'inverse, une demande de débit qui correspond à une diminution par rapport au débit programmé au temps  $t_b(k)$  retardé du délai minimum aller-retour  $\tau_3$ , sera prise en  
25 compte à un temps supérieur au temps de départ  $t_b(k)$  retardé du délai minimum aller-retour  $\tau_3$ . Si en plus la demande de débit est inférieure à tout les débits programmés au delà du temps de départ  $t_b(k)$  retardé du délai minimum aller-retour  $\tau_3$ , elle sera programmée au temps de départ  $t_b(k)$  retardé du délai maximum aller-retour  $\tau_2$ .

Dans le premier cas, le changement effectué au contrôle du flux de données émis  
30 par le terminal source 10 (dans le sens aller) sera donc être appliqué le plus tôt possible, c'est-à-dire à l'instant de programmation  $t_p^-(k) = t_b(k) + \tau_3$ . Ce temps de programmation sera appelé dans la suite de la description, temps de programmation le plus proche et on le notera  $t_p^-(k)$ .

Par contre, dans le dernier cas, ce changement sera appliqué le plus tard possible à l'instant de programmation  $t_p^+(k) = t_b(k) + \tau_2$ . Dans la suite de la description, on appellera temps de programmation le plus éloigné ce temps de programmation et on le notera  $t_p^+(k)$ .

5           Ainsi, le processus de programmation programme donc les changements de débit de contrôle sur la base de situations "pire cas" favorable à l'utilisateur, en fonction du type de changement.

On notera que le processus de programmation 21a selon l'invention est mise en œuvre à chaque arrivée d'une cellule BW-RM à l'unité d'interface 21. La complexité est  
10           donc diminuée par rapport aux processus de programmation de l'algorithme A de référence de l'art antérieur qui effectuent une programmation pour chaque cellule de données arrivant à l'unité d'interface, les cellules de données étant en nombre supérieur au nombre de cellules de gestion de ressources.

Dans la suite de la description, on notera donc le débit  $ACR(t)$  le débit contrôlé  
15           par l'unité d'interface 21 à l'instant  $t$ . Si le temps  $t$  est dans le futur, on préférera appeler ce débit "débit programmé à l'instant  $t$ ".

Le procédé de l'invention est maintenant décrit en relation avec la Fig. 2. A l'arrivée à l'unité d'interface d'une cellule de gestion de ressources retour BW-RM de numéro de référence  $k$ , on compare, à une étape 100, le débit demandé  $ER(k)$  qu'elle  
20           porte avec le débit programmé  $ACR(I, t_p^-(k))$  à l'instant de programmation le plus proche correspondant donc au temps éloigné du temps de cette arrivée  $t_b(k)$  de la durée minimale d'aller retour  $\tau_3$  ( $t_p^-(k) = t_b(k) + \tau_3$ ).

Si le débit demandé  $ER(k)$  est supérieur au débit programmé au temps de programmation le plus proche  $ACR(I, t_p^-(k))$ , alors, à l'étape 210, on supprime toute  
25           programmation de débit prévue au delà de ce temps de programmation  $t_p^-(k)$  et, à l'étape 211, on programme le débit contrôlé à ce temps  $t_p^-(k) = t_b(k) + \tau_3$  et au delà de ce temps à la valeur du débit demandé  $ER(k)$ . On a donc  $ACR(t)$  pour  $t \geq t_p^-(k)$  qui est égal à  $ER(k)$ .

Si le débit demandé  $ER(k)$  est égal au débit programmé  $ACR(I, t_p^-(k))$  au temps  
30           de programmation le plus proche, alors on supprime, à l'étape 220, toute programmation de débit prévue au delà du temps de programmation  $t_p^-(k) = t_b(k) + \tau_3$ .

Si le débit demandé  $ER(k)$  est inférieur au débit programmé  $ACR(I, t_p^-(k))$ , on recherche, à l'étape 230, s'il existe un changement de débit de programmation  $ER'$  pour un temps de programmation  $t_p'$  qui soit, d'une part, à la fois supérieur et le plus proche

du temps de programmation le plus proche  $t_p^-(k) = t_b(k) + \tau_3$  et qui, d'autre part, corresponde à un changement vers un débit inférieur au débit demandé  $ER(k)$  ( $ER' \leq ER(k)$ ).

Si ce changement n'existe pas, on programme, à l'étape 231, le débit contrôlé au temps  $t_p^+(k) = (t_b(k) + \tau_2)$  et au delà de ce temps à la valeur du débit demandé  $ER(k)$ . On a donc  $ACR(t)$  pour  $t \geq t_p^+(k)$  qui est égal à  $ER(k)$ .

Par contre, s'il existe, on supprime, à l'étape 232, toutes programmations prévues au delà du temps  $t'_p$  de programmation de ce changement, c'est-à-dire pour lesquels  $t_p \geq t'_p$ , et on programme, à l'étape 233, pour ce temps  $t'_p$  et tout temps au delà, un débit égal au débit demandé  $ER(k)$ . On a donc  $ACR(t)$  pour  $t \geq t'_p$  qui est égal à  $ER(k)$ .

Pour illustrer ce procédé, on a construit le graphe de la Fig. 3 relative à un exemple de six arrivées de cellules de gestion de ressources BW-RM. Le délai minimum aller-retour  $\tau_3$  est égal à 5 unités de temps alors le délai maximum  $\tau_2$  est égale à 8 unitésde temps.

La première arrive au temps  $t_b(0) = 3$  et porte le débit  $ER(0) = 70$ . Au temps  $t = 3$ , le temps de programmation le plus proche est le temps  $t_b(0) + \tau_3 = 8$ . A ce temps, le débit programmé est 100. En conséquence, le débit porté par la cellule 0 est inférieur au débit programmé au temps de programmation le plus proche. Par ailleurs, il n'y a pas de programmation à un débit inférieur à  $ER(0)$  qui soit prévue au delà du temps de programmation le plus proche. Il est résulte que l'on prévoit une programmation au débit de 70 pour le temps de programmation le plus éloigné soit  $t_b(0) + \tau_2 = 11$ . On a marqué I l'échelon résultant.

Au temps,  $t_b(1) = 5$ , la seconde cellule arrive en portant le débit  $ER(1) = 50$ . Au temps  $t = 5$ , le temps de programmation le plus proche est le temps  $t_b(1) + \tau_3 = 10$ . A ce temps, le débit programmé est de 100. En conséquence, le débit porté par la cellule 1 est inférieur au débit programmé au temps de programmation le plus proche. Par ailleurs, il n'y a pas de programmation à un débit inférieur au débit  $ER(1)$  qui soit prévue au delà du temps de programmation le plus proche  $t_b(1) + \tau_3 = 10$ . Il en résulte que l'on prévoit une programmation au débit de 50 pour le temps de programmation le plus éloigné soit  $t_b(1) + \tau_2 = 13$ . On a marqué II l'échelon résultant.

Au temps,  $t_b(2) = 7$ , la troisième cellule arrive en portant le débit  $ER(2) = 60$ . Au temps  $t = 7$ , le temps de programmation le plus proche est le temps  $t_b(2) + \tau_3 = 12$ . A ce temps, le débit programmé est de 70. En conséquence, le débit porté par la cellule 2 est inférieur au débit programmé au temps de programmation le plus proche. Par

ailleurs, on constate que la première programmation au delà du temps de programmation le plus proche et pour laquelle le débit est inférieur au débit demandé se trouve au temps  $t_p' = 13$ . Il en résulte que l'on prévoit une programmation au débit de 60 pour le temps de programmation  $t_p = 13$ . On a marqué III l'échelon résultant.

5        Au temps,  $t_b(3) = 9$ , la quatrième cellule arrive en portant le débit  $ER(3) = 80$ . Au temps  $t = 9$ , le temps de programmation le plus proche est le temps  $t_b(3) + \tau_3 = 14$ . A ce temps, le débit programmé est de 60. En conséquence, le débit porté par la cellule 3 est supérieur au débit programmé au temps de programmation le plus proche. Il en résulte que l'on prévoit une programmation au débit de 80 pour le temps de programmation  $t = 14$ . On a marqué IV l'échelon résultant.

10        Au temps,  $t_b(4) = 11$ , la cinquième cellule arrive en portant le débit  $ER(4) = 60$ . Au temps  $t = 11$ , le temps de programmation le plus proche est le temps  $t_b(4) + \tau_3 = 16$ . A ce temps, le débit programmé est de 80. En conséquence, le débit porté par la cellule 4 est inférieur au débit programmé au temps de programmation le plus proche. Par ailleurs, il n'y a pas de programmation prévue au delà du temps de programmation le plus proche. Il est résulte que l'on prévoit une programmation au débit de 60 pour le temps de programmation le plus éloigné soit  $t_b(4) + \tau_2 = 19$ . On a marqué V l'échelon résultant.

20        On notera qu'à ce temps d'arrivée de cellule égal à 11, le débit de contrôle est passé de 100 à 70 comme cela avait été précédemment programmé.

      Au temps,  $t_b(5) = 13$ , la sixième cellule arrive en portant le débit  $ER(5) = 80$ . Au temps  $t = 13$ , le temps de programmation le plus proche est le temps  $t_b(3) + \tau_3 = 18$ . A ce temps, le débit programmé est déjà de 80. En conséquence, le débit porté par la cellule 5 est égal au débit programmé au temps de programmation le plus proche. Il en résulte que l'on supprime toute programmation au delà du temps de programmation le plus proche soit  $t = 18$  et, en particulier, la programmation, marquée V, qui était prévue au temps  $t = 19$ .

      On notera encore qu'à ce temps d'arrivée de cellule égal à 13, le débit de contrôle est passé de 70 à 60 comme cela avait été précédemment programmé.

30        Pour la mise en œuvre du procédé, on a prévu de tenir une liste / d'événements de programmation chacun définis pour une cellule BW-RM d'ordre  $k$ , d'une part, par la date de programmation  $t_p(k)$  et, d'autre part, par le débit à programmer à cette date  $ACR(t_p(k))$ . L'indice  $p$  indique qu'il s'agit d'un temps de programmation et la référence  $k$  indique que cette date a été déterminée à l'arrivée de la cellule d'ordre  $k$ . Ainsi, on

appelle un événement de programmation un couple de valeurs  $\{ACR(t), t\}$  où  $ACR(t)$  est le débit de contrôle qui est à programmer à un temps  $t$ .

Par ailleurs, on définit une fonction qui donne à un temps  $t$  quelconque, la valeur du débit de contrôle. Cette fonction est notée  $ACR(I, t)$  et est définie de la manière suivante :

si, dans la liste  $I$ , il existe au moins un événement de programmation  $\{ACR(t_p), t_p\}$  prévu à un temps  $t_p$  compris entre le temps présent  $t_0$  et le temps  $t$ , elle est égale au débit de programmation de cet événement  $ACR(t_p)$  et, s'il y en a plusieurs, elle est égale au débit de programmation du dernier de ces événements, et

si, toujours dans cette liste, il n'existe pas d'événements de programmés, la valeur de la fonction  $ACR(I, t)$  est égale à la valeur du débit de programmation actuel.

Par exemple, si l'on considère à un temps  $t_0 = 0$ , la liste suivante :

100	5
70	10
50	15

la fonction  $ACR(I, 3)$  renvoie à la valeur du débit de programmation actuel qui est par exemple de 110. La fonction  $ACR(I, t_1 = 7)$  est égale au débit de programmation de l'événement qui se trouve entre le temps  $t_0 = 0$  et le temps  $t_1 = 7$ , soit le temps  $t_p(k) = 5$ . Cette valeur est donc 100. La fonction  $ACR(I, t_2 = 17)$  est égale au débit de programmation de l'événement qui se trouve entre le temps  $t_0 = 0$  et le temps  $t_2 = 17$  et qui est le plus proche du temps  $t_2 = 17$ . Il s'agit donc du temps  $t_p(k) = 15$  pour lequel le débit est 50. On a donc  $ACR(I, t_2 = 17) = 50$ .

Dans la suite de la description, on appellera la valeur de cette fonction  $ACR(I, t)$ , "débit programmé au temps  $t$ ".

Le procédé de l'invention peut alors s'écrire, sous la forme de pseudocode, de la façon suivante :

Pour tout instant  $t_b(k)$  correspond à l'arrivée d'une cellule B-RM portant le débit  $ER(k)$  :

Si  $ER(k) > ACR(I, t_p(k) = t_b(k) + \tau_3)$  alors supprimer dans la liste ordonnée  $I$  tous les événements de programmation  $\{ACR(i), t_p(i)\}$  tel que  $t_p(i) \geq t_p(k)$  et

rajouter l'événement  $\{ACR(t_p(k)), t_p(k)\}$  avec  $ACR(t_p(k)) = ER(k)$  et  $t_p(k) = t_b(k) + \tau_3$ .

5 Si  $ER(k) = ACR(l, t_p^-(k))$  alors supprimer dans la liste ordonnée  $l$  tous les événements de programmation  $\{ACR(t_p(i)), t_p(i)\}$  tel que  $t_p(i) > t_p^-(k)$ .

Si  $ER(k) < ACR(l, t_p^-(k))$  alors rechercher dans la liste l'événement  $\{ACR', t'_p\}$  de plus petit  $t'_p$  tel que  $t'_p > t_p^-(k)$  et  $ACR' \leq ER(k)$ , puis

10 - si l'élément  $\{ACR', t'_p\}$  existe alors supprimer dans la liste ordonnée  $l$  tous les événements de programmation  $\{ACR(t_p(i)), t_p(i)\}$  tel que  $t_p(i) \geq t'_p$  et rajouter l'événement  $(ER(k), t'_p)$ .

- si l'élément  $\{ACR', t'_p\}$  n'existe pas alors rajouter l'élément  $\{ER(k), t_p^+(k)\}$ .

15

A l'expiration de la date de programmation ( $t = t_p(k)$ ), on procède aux opérations suivantes :

20 - égaliser le débit de contrôle  $ACR(t)$  au débit programmé  $ACR(t_p(k))$  pour cette date,

- Supprimer de la liste  $l$  l'événement correspondant  $\{ACR(t_p(k)), t_p(k)\}$ .

25 On va reprendre l'exemple ci-dessus pour expliciter ce premier mode de réalisation avec liste  $l$  des événements..

La première cellule BW-RM arrive au temps  $t_b(0) = 3$  et porte le débit  $ER(0) = 70$ . On détermine la valeur de la fonction  $ACR(l, t_p(k) + \tau_3)$  qui, puisqu'il n'y a pas d'événements de prévus dans la liste  $l$ , est égale à la valeur de débit de contrôle actuelle, soit, dans l'exemple donné 100.

30 Au temps  $t = 3$ , le débit porté par la cellule 0 est inférieur au débit programmé au temps de programmation le plus proche. Comme il n'y a pas de programmation prévue au delà du temps de programmation le plus proche, on place dans la liste  $l$  l'événement  $\{70, 11\}$ .

Au temps,  $t_b(1) = 5$ , la seconde cellule arrive en portant le débit  $ER(1) = 50$ . Le temps de programmation le plus proche est égal à 10. Il n'y a donc pas de programmations de prévues entre le temps actuel égal à 5 et ce temps de programmation le plus proche égal à 10. En conséquence, la valeur de la fonction  $ACR(l, t_p(k))$  est de nouveau égale à 100.

Ainsi, le débit porté par la cellule est encore inférieur au débit programmé au temps de programmation le plus proche. Comme, il n'y a pas de programmation à un débit inférieur à  $ER(1)$  qui soit prévue au delà du temps de programmation le plus proche, on ajoute dans la liste l'événement  $\{50, 13\}$ . La liste  $l$  est maintenant la suivante :

70	11
50	13

Au temps,  $t_b(2) = 7$ , la troisième cellule arrive en portant le débit  $ER(2) = 60$ . Le temps de programmation le plus proche est égal à  $t_b(2) + \tau_3 = 12$ . L'événement de programmation  $\{70, 11\}$  est prévu entre le temps actuel et le temps de programmation le plus proche. Donc la fonction  $ACR(l, t_p(k) + \tau_3)$  est égal à 70. Il en résulte que le débit porté par la cellule BW-RM est inférieur à la valeur de cette fonction. Par ailleurs, le premier événement prévu au delà du temps de programmation le plus proche et pour lequel le débit est inférieur au débit demandé se trouve au temps  $t_p' = 13$ . On supprime donc de la liste  $l$  l'événement  $\{50, 13\}$  et on y ajoute l'événement  $\{60, 13\}$ . La liste est maintenant la suivante :

70	11
60	13

Au temps,  $t_b(3) = 9$ , la quatrième cellule arrive en portant le débit  $ER(3) = 80$ . Au temps  $t = 9$ , le temps de programmation le plus proche est le temps  $t_b(3) + \tau_3 = 14$ . On comprendra que la fonction  $ACR$  est égal à 60. En conséquence, le débit porté par la cellule est supérieur à cette valeur. On ajoute donc à la liste  $l$  l'événement  $\{80, 14\}$ .

La liste  $l$  devient la suivante :

70	11
----	----

60	13
80	14

Au temps  $t = 11$ , le débit de contrôle  $ACR(t)$  devient égal à 70 et l'événement correspondant de la liste  $I$  y est retiré. La liste  $I$  devient :

60	13
80	14

5

Par ailleurs, à ce temps, la cinquième cellule arrive en portant le débit  $ER(4) = 60$ . Le temps de programmation le plus proche est le temps  $tb(3) + \tau_3 = 16$ . La fonction  $ACR$  est alors égale à 80. En conséquence, le débit porté par la cellule 4 est inférieur à cette valeur. Par ailleurs, il n'y a pas, dans la liste  $I$ , d'événement prévu pour un temps au delà du temps de programmation le plus proche. Il est résulte que l'on

10 ajoute à la liste  $I$  l'événement  $\{60, 19\}$ . La liste  $I$  devient :

60	13
80	14
60	19

Au temps  $t = 13$ , le débit de contrôle  $ACR(t)$  devient égal à 60 et le premier

15 événement de la liste  $I$  y est retiré. La liste  $I$  devient donc :

80	14
60	19

A ce temps, la quatrième cellule arrive en portant le débit  $ER(5) = 80$ . Le temps de programmation le plus proche est le temps  $tb(3) + \tau_3 = 18$ . Il existe un événement

20 entre le temps présent et le temps de programmation le plus proche qui est l'événement  $\{80, 14\}$ . La fonction  $ACR$  est donc égale à 80. Le débit porté par la cellule 5 est donc égal à cette valeur. Il en résulte que l'on supprime de la liste tout événement prévu à des temps au delà du temps de programmation le plus proche soit  $t = 18$  et, en particulier, l'événement  $\{60, 19\}$ . La liste devient donc :

25

80	14
----	----

On peut montrer mathématiquement que le procédé qui vient d'être décrit donne le même comportement que l'algorithme A de l'art antérieur.

Pour éviter des débordements des files de mémoire dans lesquelles la liste d'événements est enregistrée, on a cherché à améliorer le procédé ci-dessus décrit de manière à limiter le nombre d'éléments qu'elle contient. On a donc cherché à limiter ce nombre d'éléments à N événements de programmation  $\{ER(i), t_p(i)\}$  et pour ce faire, on va définir une règle de réduction qui sera mise en oeuvre par défaut lorsque N + 1 éléments se trouveront dans la liste / et qui aura pour objet d'éliminer un événement de la liste /.

La contrainte à respecter pour les règles de réduction à N programmations pour l'algorithme générique consiste à fournir une trace de débit à contrôler supérieure à celle qui serait fournie par l'algorithme de référence A ou l'algorithme générique sans limitation de mémoire. Ainsi, il est possible d'envisager toute type de règle de réduction à N programmations à condition que cette règle est respectée.

On décrit ci-après, en relation avec la Fig. 4, un premier mode de réalisation d'un procédé selon l'invention dont la liste est limitée à N éléments. Selon ce mode, comme dans le mode de réalisation précédent, on compare, à l'étape 100, le débit demandé  $ER(k)$  porté par la cellule de gestion de ressources BW-RM d'ordre k qui arrive à l'instant  $t_b(k)$  à l'unité d'interface 21 à la valeur prise par la fonction  $ACR(/, t_b(k) + \tau_3)$ . Cette fonction est identique à celle qui a été précédemment décrite.

Suite à cette comparaison, si le débit demandé  $ER(k)$  est supérieur ou égal au débit programmé au temps de programmation le plus proche  $ACR(/, t_b(k) + \tau_3)$ , alors, comme précédemment on supprime, à l'étape 310, de la liste /, tous les éléments de programmation prévus à un temps égal ou supérieur au temps de programmation le plus proche  $(t_b(k) + \tau_3)$ .

Si la liste / n'est pas pleine, c'est-à-dire si le nombre d'éléments de programmation  $N_1$  contenu dans la liste / est inférieur au nombre limite N, le temps de programmation  $t_p(k)$  est, comme précédemment, égal au temps de programmation le plus proche  $t_p(k) = t_b(k) + \tau_3$  (voir étape 311).

Par contre, si la liste est pleine (le nombre d'éléments  $N_1$  dans la liste est égal à N), alors si le temps de programmation  $t_{pi}(N)$  du dernier événement de la liste / (c'est-à-dire de l'événement dont le temps de programmation, noté  $t_{pi}(N)$ , est le plus éloigné)

est inférieur ou égal au temps de programmation le plus proche  $t_p^*(k)$ , le temps de programmation  $t_p(k)$  devient égal au temps de programmation  $t_{pl}(N)$  du dernier élément enregistré dans la liste (étape 312). De plus, on supprime de la liste, le dernier élément enregistré dans la liste  $\{ACR(N), t_{pl}(N)\}$  (étape 313).

- 5 Si le temps de programmation  $t_{pl}(N)$  du dernier élément de la liste est supérieur au temps de programmation le plus proche  $t_p^*(k)$ , le temps de programmation  $t_p(k)$  devient égal au temps de programmation le plus proche  $t_p(k) = t_p^*(k)$  (voir étape 311).

A la suite de ces opérations, l'élément  $\{ACR(t_p(k)), t_p(k)\}$  avec  $ACR(t_p(k)) = ER(k)$  est enregistré, à l'étape 314, dans la liste  $l$ .

- 10 La notation suivie ici correspond à l'utilisation d'une liste dont les événements sont ordonnés selon l'ordre croissant des temps de programmation que l'on note  $t_{pl}$ , avec  $p$  signifiant programmation et  $l$  signifiant présent dans liste. Ainsi, pour une liste  $l$  de dimension  $N$ , le dernier événement est celui d'ordre  $N$ . Le premier événement de liste  $l$  est celui d'ordre 1 et correspond à l'événement dont le temps de programmation
- 15  $t_{pl}$  est le plus petit de la liste  $l$ . De manière générale, l'événement d'ordre  $X$  est noté  $\{ACR(X), t_{pl}(X)\}$ .

- Si le débit demandé  $ER(k)$  est inférieur au débit programmé  $ACR(t_p^*(k), l)$ , on recherche dans la liste  $l$ , à l'étape 320, s'il existe, l'événement  $\{ACR(t'_p), t'_p\}$  programmé à un temps  $t'_p$  au delà du temps de programmation le plus proche  $t_p^*(k)$  pour un débit  $ACR(t'_p)$  inférieur ou égal au débit demandé  $ER(k)$ . Au cas où il existerait plusieurs événements  $\{ACR(t'_p), t'_p\}$ , on recherchera seulement celui qui a le plus petit des temps  $t'_p$ .
- 20

Si cet événement n'existe pas, le temps de programmation  $t_p(k)$  devient égal au temps de programmation le plus éloigné  $t_p^*(k) = t_b(k) + \tau_2$  (étape 321).

- 25 Par contre, si cet élément  $\{ACR(t'_p), t'_p\}$  existe, on supprime de la liste  $l$ , à l'étape 322, tout événement programmé à un temps égal et au delà du temps de programmation  $t'_p$  de cet événement. De plus, on égalise, à l'étape 323, le temps de programmation  $t_p(k)$  au temps de programmation  $t'_p$  de cet élément.

- 30 On vérifie, à l'étape 324, si le nombre d'éléments de programmation présents dans la liste  $l$  est inférieur ou s'il est égal à  $N$ .

Si la liste  $l$  est pleine ( $N$  éléments de programmation se trouvent dans la liste  $l$ ) (étape 325), alors si le débit demandé  $ER(k)$  est supérieur ou égal au débit de programmation  $ACR(N)$  du dernier élément enregistré dans la liste  $l$ , on modifie, à

l'étape 326, le temps de programmation  $t_p(k)$  pour qu'il devienne égal au temps de programmation  $t_{pi}(N)$  de cet élément.

L'avant-dernier élément de la liste est modifié à l'étape 327 de manière que son débit  $ACR(N-1)$  soit la plus grande des deux valeurs correspondant respectivement aux débits de programmation du dernier élément et de l'avant dernier élément :  
 5  $ACR(N-1) = \text{MAX}(ACR(N), ACR(N-1))$ . Puis on supprime le dernier élément  $\{ACR(N), t_{pi}(N)\}$ .

Dans tous les cas, on rajoute ensuite dans la liste  $I$  l'élément  $\{ACR(t_p(k)), t_p(k)\}$  avec  $ACR(t_p(k)) = ER(k)$  correspondant au débit demandé  $ER(k)$  pour le temps de programmation  $t_p(k)$  déterminé.  
 10

Pour illustrer ce mode de réalisation, on a construit le graphe de la Fig. 5 pour l'arrivée de 8 cellules au temps  $t_b = 5, 7, 9, 11, 16, 21, 25$  et  $35$ . Ces cellules portent respectivement les débits  $ER = 110, 130, 150, 100, 70, 50$  et  $60$ . Le délai le plus court  $\tau_3$  est égal à 10 alors que le délai le plus long  $\tau_2$  est égal à 20. Le nombre d'événements supporté dans la liste  $I$  est  $N = 2$ .  
 15

La première cellule arrive donc au temps  $t_b(0) = 5$  et porte le débit  $ER(0) = 110$ . Le temps de programmation le plus proche est égal à 15 pour lequel le débit programmé est 100. On comprendra que le débit programmé est en réalité le résultat de la fonction  $ACR(I, t_b(k) + \tau_3)$  telle qu'elle est décrite ci-dessus. Comme précédemment,  
 20 le premier élément dans la liste sera donc le suivant :  $\{110, 15\}$  correspondant au débit porté  $ER(0)$  et au temps de programmation le plus proche  $t_b(0) + \tau_3$ . On a marqué I l'échelon résultant correspondant.

Le processus sera semblable pour la seconde cellule 1 arrivée au temps  $t_b(1)$  et aboutit à l'enregistrement dans la liste  $I$  du second élément  $\{130, 17\}$ . On a marqué II  
 25 l'échelon correspondant.

La liste  $I$  se présente alors sous la forme suivante :

110	15
130	17

La troisième cellule arrive au temps  $t_b(2) = 9$  et le débit porté est  $ER(2) = 150$ .  
 30 Ce débit est supérieur à celui qui est prévu au temps de programmation le plus proche, à savoir  $t_b(2) + \tau_3 = 19$  qui est 130. Cependant la liste  $I$  est pleine (le nombre d'éléments dans la liste  $I$  est égal à  $N = 2$ ). Le dernier élément enregistré dans la liste a pour temps

de programmation  $t_{pi}(N)$  le temps 17 qui est inférieur au temps de programmation le plus proche. Il est résulte que le temps de programmation devient égal à ce temps de programmation  $t_p(k) = t_{pi}(N)$ . On supprime le dernier élément de la liste, à savoir l'élément  $\{130, 17\}$ . Puis on rajoute dans la liste l'élément  $\{ER(k), t_p(k)\}$  soit l'élément

5  $\{150, 17\}$ . On a marqué III l'échelon correspondant.

La liste  $l$  se présente alors sous la forme suivante :

110	15
150	17

10 La quatrième cellule arrive au temps  $t_b(3) = 11$ . Le débit porté est  $ER(3) = 100$  qui est inférieur au débit programmé au temps de programmation le plus proche qui est maintenant 150. On peut constater que le temps de programmation est maintenant égal à  $t_p(3) = t_b(3) + \tau_2$ , soit le temps  $t_p(3) = 31$ . Cependant la liste est pleine. Le temps de programmation n'est pas modifié puisque le débit demandé  $ER(3) = 100$  est inférieur au

15 débit du dernier élément enregistré dans la liste  $l$ , soit 150.

Le débit de l'avant-dernier élément  $ACR(N-1)$  est égalé au débit du dernier élément  $ACR(N) = 150$ . On a donc maintenant l'événement  $\{150, 15\}$ . Le dernier élément  $\{AC(N), t_{pi}(N)\} = \{150, 17\}$  est alors supprimé de la liste  $l$ .

La liste  $l$  se présente alors sous la forme suivante :

20

150	15

Enfin, l'élément  $\{ACR(k), t_p(k)\}$  est ajoutée à la liste  $l$  qui est maintenant la suivante :

25

150	15
100	31

On a marqué IV l'échelon correspondant à cette suite d'opérations.

Au temps  $t = 15$ , le débit contrôlé passe de 100 à 150 et l'événement correspondant  $\{150, 15\}$  est supprimé de la liste  $l$  qui ne comporte plus que l'élément  $\{100, 31\}$ . Elle se présente donc sous la forme :

100	31

5

La cinquième cellule arrive au temps  $t_b(4) = 16$  avec un débit porté qui est égal à  $ER(4) = 70$  inférieur au débit programmé au temps de programmation le plus proche. On peut alors constater que le temps de programmation est maintenant égal à  $t_p(4) = t_b(4) + \tau_2$ , soit le temps  $t_p(4) = 36$ .

10 L'élément  $\{ER(k), t_p(k)\}$  est ajoutée à la liste  $l$  qui est maintenant la suivante :

100	31
70	36

On a marqué V l'échelon correspondant.

15 La sixième cellule arrive au temps  $t_b(5) = 21$ . Le débit porté est  $ER(5) = 50$  qui est inférieur au débit programmé au temps de programmation le plus proche. On peut constater que le temps de programmation est maintenant égal à  $t_p(5) = t_b(5) + \tau_2$ , soit le temps  $t_p(5) = 41$ . Cependant la liste est pleine. Le temps de programmation n'est pas modifié puisque le débit demandé  $ER(k)$  est inférieur au débit du dernier élément enregistré dans la liste  $l$ .

20

Le débit de l'avant-dernier élément  $ACR(N-1)$  n'est pas modifié et le dernier élément  $\{ACR(N), t_{pl}(N)\} = \{70, 36\}$  est supprimé.

Enfin, l'élément  $\{ER(k), t_p(k)\}$  est ajoutée à la liste  $l$  qui est maintenant la suivante :

25

100	31
50	41

On a marqué VI l'échelon correspondant.

L'algorithme selon ce second mode de mise en œuvre peut s'écrire sous la forme dite de pseudo-code de la manière suivante :

A) Pour chaque instant  $t_b(k)$  correspondant à l'arrivée d'une cellule portant le débit demandé  $ER(k)$  :

5

1) Si  $ER(k) \geq ACR(l, t_p^-(k))$  alors

Supprimer dans la liste ordonnée  $l$  tous les éléments de programmation  $\{ACR(i), t_{pi}(i)\}$  tel que  $t_{pi}(i) \geq t_p^-(k)$ ,

10

Si la liste est pleine ( $N$  éléments dans la liste) alors

Si  $(t_{pi}(N) \leq t_p^-(k))$  alors  $t_p(k) = t_{pi}(N)$

Supprimer le dernier élément  $(ACR(N), t_{pi}(N))$

Sinon  $t_p(k) = t_p^-(k)$

15

2) Si  $ER(k) < ACR(l, t_p^-(k))$  alors

Rechercher dans la liste  $l$  l'élément  $\{ACR', t_p'\}$  de plus petit  $t_p'$  tel que  $ACR' \leq ACR(k)$  et  $t_p' > t_p^-(k)$

Si l'élément  $\{ACR', t_p'\}$  existe alors

20

Supprimer dans la liste ordonnée  $l$  tous les éléments de programmation  $\{ACR(i), t_{pi}(i)\}$  tel que  $t_{pi}(i) \geq t_p'$  et faire  $t_p(k) = t_p'$ ,

Si l'élément  $(ER', t_p')$  n'existe pas alors  $t_p(k) = t_p^+(k)$ .

Si la liste est pleine ( $N$  éléments dans la liste) alors

si  $ER(k) \geq ACR(N)$  alors  $t_p(k) = t_{pi}(N)$

$ACR(N-1) = \text{Max}(ACR(N-1), ACR(N))$

25

Supprimer le dernier élément  $\{ACR(N), t_{pi}(N)\}$

3) Rajouter dans la liste après la dernière programmation l'élément  $\{ACR(t), t\}$  avec  $ACR(t) = ER(k)$  et  $t = t_p(k)$ .

B) A l'expiration de la première date de programmation ( $t = t_p(1)$ ) :

$$ACR(t) = ACR(1);$$

supprimer le premier élément  $\{ACR(1), t_p(1)\}$  de la liste  $l$ .

5 On propose ci-après un autre mode de mise en oeuvre selon l'invention qui est décrit en relation avec la Fig. 6.

Comme dans les modes de réalisation précédemment décrits, on compare, à l'étape 400, le débit demandé  $ER(k)$  avec le débit programmé au temps de programmation le plus proche. On rappelle que la valeur de ce débit est déterminée par  
10 la fonction  $ACR(l, t)$  décrite ci-dessus.

Si le débit demandé  $ER(k)$  est supérieur au débit programmé au temps de programmation le plus proche  $ACR(t_b(k) + \tau_3)$ , alors, comme précédemment on supprime, à l'étape 410, de la liste  $l$ , tous les événements de programmation prévus à un temps supérieur ou égal au temps de programmation le plus proche ( $t_b(k) + \tau_3$ ).

15 Si la liste est pleine (le nombre d'événements stockés dans la liste  $l$  est égal à  $N$ )(étape 411), on met en oeuvre, à l'étape 412, une fonction  $F_R$  dont l'objet est de réduire le nombre d'éléments dans la liste  $l$ . Un exemple de mise en oeuvre de cette fonction est donnée ci-dessous. Par contre, si la liste n'est pas pleine, on passe directement à l'étape suivante. Celle-ci, étape 413, consiste à programmer l'événement  
20  $\{ACR(t_p(k)), t_p(k)\}$  avec  $ACR(t_p(k)) = ER(k)$  et  $t_p(k) = t_p^-(k)$ .

Si le débit demandé  $ER(k)$  est égal au débit programmé au temps de programmation le plus proche  $ACR(l, t_p^-(k))$ , alors on supprime, à l'étape 420, dans la liste  $l$ , tous les éléments de programmation prévus au delà de ce temps de programmation le plus proche  $t_p^-(k) = (t_b(k) + \tau_3)$ .

25 Si le débit demandé  $ER(k)$  est inférieur au débit programmé  $ACR(l, t_p^-(k))$ , on recherche dans la liste  $l$ , à l'étape 430, s'il existe, l'événement  $\{ACR(t'_p(k)), t'_p(k)\}$  programmé à un temps  $t'_p(k)$  au delà du temps de programmation le plus proche tel que le débit  $ACR(t'_p(k))$  est inférieur au débit demandé  $ER(k)$ . Si plusieurs événements  $\{ACR(t'_p(k)), t'_p(k)\}$  existent, on recherchera que le premier de ces événements pour  
30 lequel le temps  $t'_p(k)$  est le plus petit.

Si cet événement n'existe pas et que la liste  $l$  est pleine (étape 431), on met en oeuvre, à l'étape 432, la fonction de réduction  $F_R$  mentionnée ci-dessus. Puis à l'étape 433, on rajoute, dans la liste  $l$ , l'élément de programmation  $\{ACR(t_p(k)), t_p(k)\}$  dont le

débit  $ACR(t_p(k))$  correspond au débit demandé  $ER(k)$  et dont le temps de programmation est le temps de programmation le plus éloigné  $t_p(k) = t_b(k) + \tau_2$ .

Si l'événement recherché à l'étape 430 n'existe pas et qu'en plus la liste  $I$  n'est pas pleine (étape 431), on ajoute directement l'élément de programmation  
 5  $\{ACR(t_p(k)), t_p(k)\}$  dont le débit  $ACR(t_p(k))$  correspond au débit demandé  $ER(k)$  et dont le temps de programmation est le temps de programmation le plus éloigné  $t_p^+(k) = t_b(k) + \tau_2$  (étape 433).

Par contre, si l'élément recherché  $(ACR', t_p')$  existe, on supprime, à l'étape 434, toute programmation prévue à un temps supérieur ou égal au temps de programmation  
 10  $t_p'$ .

Si la liste  $I$  est pleine ( $N$  éléments de programmation se trouvent dans la liste  $I$ ) (étape 435), on met en œuvre, à l'étape 436, la fonction de réduction  $F_R$  mentionnée ci-dessus, puis on ajoute, à l'étape 437, à la liste  $I$ , l'élément  $\{ACR(t_p(k)), t_p(k)\}$  dont le  
 15 débit  $ACR(t_p(k))$  correspond au débit demandé  $ER(k)$  et dont le temps de programmation  $t_p(k)$  devient égal au temps de programmation  $t_p'$  de l'élément de programmation recherché.

Par contre, si la liste  $I$  n'est pas pleine, on ajoute directement, également à l'étape 437, l'élément  $\{ACR(t_p(k)), t_p(k)\}$  dont le débit  $ACR(t_p(k))$  correspond au débit  
 20 demandé  $ER(k)$  et dont le temps de programmation  $t_p(k)$  devient égal au temps de programmation  $t_p'$  de l'élément de programmation recherché.

La fonction  $F_R$  va d'abord déterminer deux événements de programmation consécutifs de la liste  $I$  et va donner finalement un seul événement qui sera tel qu'il en  
 25 résultera une surévaluation du débit à contrôler. Le choix des deux événements considérés est avantageusement fait de manière à limiter la surévaluation de débit de contrôle qui résulte de l'application de la fonction  $F_R$ .

Selon un mode avantageux de réalisation de la fonction de réduction  $F_R$ , celle-ci consiste à rechercher d'abord dans la liste  $I$ , qui est ordonnée selon l'ordre croissant des  
 30 temps de programmation  $t_{pi}$ , les deux éléments successifs pour lesquels le produit de l'augmentation ou de la diminution de débit par la mesure de l'intervalle de temps compris entre leurs deux temps de programmation soit minimum. Ainsi, sont recherchés les deux éléments  $\{ACR(X), t_{pi}(X)\}$  et  $\{ACR(X-1), t_{pi}(X-1)\}$  tels que la valeur de l'expression  $|(ACR(X)-ACR(X-1)) \cdot (t_{pi}(X)-t_{pi}(X-1))|$  soit minimale,  $X$  étant l'ordre dans la liste  $I$  du deuxième de ces deux éléments successifs.

Si le débit de programmation  $ACR(X)$  du second élément est supérieur au débit de programmation  $ACR(X-1)$  du premier élément, on élimine le second élément et on affecte au premier le débit du second. Ainsi, des deux éléments  $\{ACR(X), t_{pi}(X)\}$  et  $\{ACR(X-1), t_{pi}(X-1)\}$ , il ne restera qu'un élément  $\{ACR(X), t_{pi}(X-1)\}$  dont le débit correspond au débit de programmation du second élément et dont le temps de programmation est celui de premier.

Par contre, si le débit de programmation  $ACR(X)$  du second élément n'est pas supérieur au débit de programmation  $ACR(X-1)$  du premier élément, on élimine le second élément.

L'ordre des éléments d'ordre supérieur ou égal à  $X$  est modifié de manière que l'ordre de la case libre de la liste  $l$  soit celle d'ordre  $N$ . Ceci se traduit par la mise en oeuvre de l'instruction suivante :

Pour  $i = X$  à  $N-1$ ,  $ACR(i) = ACR(i+1)$  et  $t_{pi}(i) = t_{pi}(i+1)$ .

L'ensemble de ces opérations peut s'exprimer par le pseudocode suivant :

Si  $ACR(X) > ACR(X - 1)$  alors on programme le plus grand des deux derniers débits.

$ACR(X-1) = ACR(X)$

Supprimer l'élément d'ordre  $X$ .

Pour  $i = X$  jusqu'à  $N-1$  alors on décale les programmations d'une case mémoire.

$ACR(i) = ACR(i + 1)$  et  $t_{pi}(i) = t_{pi}(i + 1)$

Fin pour

Insérer à l'ordre  $N$  le nouvel événement  $\{ACR(N), t_p(N)\} = \{ER(k), t_p(k)\}$ .

On va donner ci-dessous, en relation avec la Fig. 7, un exemple de mise en oeuvre du procédé selon ce mode de réalisation pour l'arrivée de 9 cellules aux temps  $t_b = 5, 7, 9, 11, 14, 16, 21, 25$  et  $35$ . Ces cellules portent respectivement les débits  $ER = 110, 150, 160, 130, 120, 70, 50, 60$  et  $90$ . Le délai le plus court  $\tau_3$  est égal à  $10$  alors que le délai le plus long  $\tau_2$  est égal à  $20$ . Le nombre d'éléments supporté dans la liste  $l$  est  $N = 4$ .

La première cellule arrive donc au temps  $t_b(0) = 5$  et porte le débit  $ER(0) = 110$ . Le temps de programmation le plus proche est égal à  $15$  pour lequel le débit programmé est  $100$ . Comme précédemment, le second élément dans la liste sera donc

le suivant :  $\{110, 15\}$  correspondant au débit porté  $ER(0)$  et au temps de programmation le plus proche  $t_b(0) + \tau_3$ . On a marqué I l'échelon de débit correspondant.

Le processus sera semblable pour les seconde et troisième cellules et aboutit à l'enregistrement dans la liste  $I$  du second élément  $\{150, 17\}$  et du troisième élément  $\{160, 19\}$ . On a marqué II et III les échelons respectifs correspondants.

La quatrième cellule arrive au temps  $t_b(3) = 11$  et le débit porté est  $ER(3) = 130$ . Ce débit est inférieur à celui qui est prévu au temps de programmation le plus proche, à savoir  $t_b(3) + \tau_3 = 21$  qui est 160. L'étape de recherche n'aboutit pas. La liste  $I$  n'est pas pleine et l'élément  $\{130, 31\}$  dont le débit est égal au débit demandé  $ER(3)$  et dont le temps de programmation est le temps de programmation le plus éloigné  $t_p(3) = t_b(3) + \tau_2$  est ajouté à la liste  $I$ .

La liste  $I$  se présente donc sous la forme suivante :

110	15
150	17
160	19
130	31

15

On a marqué IV l'échelon correspondant à cette dernière opération.

La cinquième cellule arrive au temps  $t_b(4) = 14$  et le débit porté est  $ER(4) = 120$  qui est inférieur au débit programmé au temps de programmation le plus proche. La recherche n'aboutit pas. Cependant la liste  $I$  est pleine. On met donc en œuvre la fonction de réduction qui, comme on peut aisément le vérifier conduit à la suppression de l'élément de programmation  $\{150, 17\}$  et à l'ajout de l'événement  $\{160, 17\}$ . En effet, le produit  $|(ACR(X) - ACR(X-1)) \cdot (t_{pl}(X) - t_{pl}(X-1))| = |160 - 150| \cdot (19 - 17)$  est minimum. Puis, on ajoute, dans la liste  $I$ , l'élément dont le débit est égal au débit demandé  $ER(4)$  et donc le temps de programmation est le temps de programmation le plus éloigné  $t_p(4) = t_b(4) + \tau_2$  est ajouté à la liste  $I$ .

25

La liste  $I$  devient donc la suivante :

110	15
160	17
130	31

120	34
-----	----

On a marqué V l'échelon correspondant résultant à ces opérations.

Le contrôle correspondant à l'élément  $\{110, 15\}$  est mis en œuvre et cet élément est supprimé de la liste.

- 5 La sixième cellule arrive au temps  $t_b(5) = 16$  avec un débit demandé égal à 70. Ce débit est inférieur à toute programmation et l'élément  $\{70, 36\}$  est ajouté à la liste  $l$  qui devient :

160	19
130	31
120	34
70	36

- 10 On a marqué VI l'échelon correspondant résultant à ces opérations.

Le contrôle correspondant à l'élément  $\{160, 19\}$  est mis en œuvre et cet élément est supprimé de la liste.

Au temps  $t_b(6)$  égal à 21 arrive la septième cellule. Elle porte le débit demandé  $ER(6) = 50$  qui est inférieur à toute programmation. Il en résulte que l'élément  $\{50, 41\}$

- 15 est ajouté à la liste qui devient :

130	31
120	34
70	36
50	41

On a marqué VII l'échelon correspondant résultant à ces opérations.

Au temps  $t_b(7)$  égal à 25 arrive la huitième cellule laquelle porte le débit demandé

- 20 60. La recherche aboutit à l'élément  $\{50, 41\}$  qui est supprimé de la liste  $l$ . La liste n'est pas pleine et l'élément  $\{60, 41\}$  est ajouté à la liste  $l$  qui devient :

130	31
120	34
70	36

60	41
----	----

On a marqué VIII l'échelon correspondant résultant à ces opérations.

Les contrôles correspondant aux éléments {130,31} et {120,34} sont mis en œuvre et ces éléments sont supprimés de la liste *l*.

- 5 Au temps  $t_b(8)$  égal à 35 arrive la neuvième cellule laquelle porte le débit demandé 90 supérieur au débit programmé pour le temps de programmation le plus proche. La liste *l* n'est pas pleine et l'élément de programmation {90,45} est ainsi ajouté à la liste qui devient :

70	36
60	41
90	45

10

On a marqué IX l'échelon correspondant résultant à ces opérations.

## REVENDEICATIONS

1) Procédé de contrôle de la conformité du débit des cellules de données émises par un terminal source en communication avec un terminal destinataire, via un réseau à haut débit, par exemple un réseau utilisant le mode de transfert dit ATM, ledit réseau comportant des unités d'interface utilisateur-réseau auxquels sont connectés lesdits terminaux, et, éventuellement, des unités d'interface réseau-réseau, ledit terminal source émettant périodiquement une cellule dite de gestion de ressources qui est retournée par le terminal destinataire audit terminal source sous la forme d'une cellule dite de gestion de ressources retour, ladite cellule de gestion de ressources comprenant un champ dans lequel est stockée la valeur d'un débit demandé, ledit procédé comprenant un processus de contrôle (21b) consistant à vérifier la conformité du débit des cellules émises par ledit terminal source avec un débit programmé par un processus dit processus de programmation, caractérisé en ce que ledit processus de programmation consiste, à l'arrivée d'une cellule de gestion de ressources retour (BW-RM) d'ordre  $k$  à ladite unité d'interface (21), à déterminer la date, dite date de programmation ( $t_p(k)$ ), de la prise en compte par ledit processus de contrôle (21b) de la demande de débit ( $ER(k)$ ) contenue dans le champ de ladite cellule (BW-RM), ladite date de programmation étant égale à ladite date d'arrivée ( $t_b(k)$ ) retardée du délai minimum aller-retour ( $\tau_3$ ) entre l'interface et le terminal, dite date de programmation la plus proche, lorsque ladite prise en compte à cette date de programmation la plus proche aboutit à une augmentation de débit programmé et étant supérieur au temps de programmation le plus proche et inférieur à ladite date d'arrivée ( $t_b(k)$ ) retardée d'un délai maximum aller-retour ( $\tau_2$ ), ( $t_b(k) + \tau_2$ ) étant dite date de programmation la plus éloignée, si sa prise en compte audit temps de programmation le plus proche ( $t_b(k) + \tau_3$ ) avait abouti à une diminution de débit et, à transmettre, à l'échéance du temps de programmation ( $t_p(k)$ ), ladite valeur de débit à contrôler ( $ACR(t_p(k))$ ) audit processus de contrôle (21b) pour contrôle du flux émis par ledit terminal source.

2) Procédé selon la revendication 1, caractérisé en ce qu'il consiste à tenir une liste (l) d'événements de programmation, chaque événement étant défini par un couple de valeur dont la première représente une valeur de débit  $ACR(t_p(k))$  et dont la seconde représente la date de programmation  $t_p(k)$  de ladite valeur de débit  $ACR(t_p(k))$ , et ladite liste étant ordonnée suivant ladite date de programmation des événements.

3) Procédé selon la revendication 2, caractérisé en ce qu'il consiste pour connaître le débit programmé ( $ACR(t_p^*(k))$ ) à la date d'arrivée  $t_b(k)$  retardée de la durée minimale d'aller retour  $\tau_3$  ( $t_p^*(k) = t_b(k) + \tau_3$ ), à définir une fonction  $ACR(I, t)$  qui, à partir des événements présents dans la liste  $I$ , détermine la valeur du débit programmé à un temps quelconque  $t$  de la manière suivante :

5 si, dans ladite liste  $I$ , il existe au moins un événement de programmation  $\{ACR(t_p), t_p\}$  prévu à un temps de programmation  $t_p$  compris entre le temps présent  $t_0$  et le temps  $t$ , ladite valeur est égale au débit de programmation de cet événement  $ACR(t_p)$  et, s'il y en a plusieurs, elle est égale au débit de programmation du dernier de ces événements, et

10 si, toujours dans cette liste, il n'existe pas d'événements programmés entre le temps présent  $t_0$  et  $t$ , ladite valeur est égale à la valeur du débit actuellement contrôlé dans le processus de contrôle.

4) Procédé selon la revendication 2 ou 3, caractérisé en ce qu'il consiste à :

15 à l'arrivée à l'unité d'interface (21) auquel est connecté ledit terminal source, d'une cellule de gestion de ressources retour (BW-RM) de numéro de référence ( $k$ ), à comparer le débit ( $ER(k)$ ) que porte ladite cellule avec le débit programmé ( $ACR(I, t_p^*(k))$ ) à l'instant de programmation le plus proche, puis

20 si ladite comparaison effectuée à chaque arrivée d'une cellule de gestion de ressources (BW-RM) montre que le débit demandé  $ER(k)$  est supérieur au débit programmé au temps de programmation le plus proche ( $ACR(I, t_p^*(k))$ ), supprimer, de ladite liste  $I$ , tout événement de programmation de débit prévu à un temps supérieur ou égal au temps de programmation le plus proche ( $t_p^*(k)$ ) et rajouter, dans ladite liste  $I$ , un événement dont le débit contrôlé est égal à la valeur du débit  $ER(k)$  porté par ladite

25 cellule et dont le temps de programmation est le temps de programmation le plus proche ( $t_p^*(k) = t_b(k) + \tau_3$ ),

si ladite comparaison montre que le débit demandé  $ER(k)$  est égal au débit programmé  $ACR(I, t_p^*(k))$  au temps programmation le plus proche, supprimer, de la liste  $I$ , tout événement de programmation de débit dont le temps de programmation est

30 supérieur au temps de programmation le plus proche ( $t_p^*(k) = t_b(k) + \tau_3$ ),

si ladite comparaison montre que le débit demandé  $ER(k)$  est inférieur au débit programmé  $ACR(I, t_p^*(k))$ , rechercher, dans ladite liste  $I$ , s'il existe un événement dont le temps de programmation ( $t'_p$ ) est le plus petit temps de programmation dans la liste

supérieur au temps le plus proche et dont le débit est inférieur ou égal au débit demandé  $ER(k)$ ,

si cette recherche montre que cet événement n'existe pas, rajouter, dans la liste  $l$ , un événement dont le débit est égal à la valeur du débit demandé  $ER(k)$  et dont le temps de programmation est le temps de programmation le plus éloigné  
5  $(t_p^+(k) = t_b(k) + \tau_2)$  et,

si cette recherche montre que cet événement existe, supprimer, de la liste  $l$ , tout événement de programmation dont le temps de programmation est supérieur ou égal au temps de programmation  $(t_p^-)$  dudit événement ainsi trouvé et rajouter, dans la liste  $l$ ,  
10 un événement dont le débit est égal à la valeur du débit demandé  $ER(k)$  et dont le temps de programmation est égal au temps de programmation dudit événement trouvé  $(t_p^-)$ ,

et, à changer le débit de contrôle du processus de contrôle (21b) avec le débit du premier événement  $\{ACR(1), t_p(1)\}$  programmé dans la liste d'événements de programmation lorsque le temps de programmation  $t_p(1)$  dudit premier événement arrive à échéance.  
15

5) Procédé selon une des revendications 2 à 4, caractérisé en ce qu'il consiste à limiter le nombre d'événements stockés dans ladite liste  $l$  à un nombre fini  $N$  d'événements égal ou supérieur à deux, et, avant de rajouter un nouvel événement dans  
20 ladite liste, à vérifier si le nombre d'événements contenus dans ladite liste n'est pas supérieure audit nombre fini et, dans ce cas, à mettre alors en œuvre une fonction de réduction du nombre d'événements contenus dans la liste.

6) Procédé selon la revendication 4 ou 5, caractérisé en ce que ladite fonction de réduction consiste, lorsque le débit demandé ( $ER(k)$ ) que porte la cellule de gestion de ressources retour (BW-RM) d'ordre  $k$  arrivée à l'instant ( $t_b(k)$ ) est égal au débit  
25 programmé ( $ACR(l, t_p^-(k))$ ) à l'instant de programmation le plus proche ( $t_p^-(k) = t_b(k) + \tau_3$ ), à supprimer, de la liste  $l$ , tout événement de programmation de débit dont le temps de programmation est supérieur ou égal au temps de programmation le plus proche ( $t_p^-(k) = t_b(k) + \tau_3$ ) et à rajouter dans la liste  $l$  un événement dont le débit est  
30 égal à la valeur du débit ( $ER(k)$ ) porté par ladite cellule de gestion de ressources retour (BW-RM) et dont le temps de programmation est le temps de programmation le plus proche ( $t_p^-(k) = t_b(k) + \tau_3$ ).

7) Procédé selon l'une des revendications 4 à 6, caractérisé en ce qu'il consiste, lors du rajout dans la liste d'un événement pour un temps correspondant au temps le

plus proche ( $t_p(k)=t_b(k)+\tau_3$ ), à vérifier, avant de rajouter ledit événement, si le nombre d'événements dans la liste l est égal audit nombre fini N et si le temps de programmation du dernier événement dans la liste l ( $t_p(N)$ ) est inférieur ou égal au temps de programmation le plus proche ( $t_p(k)=t_b(k)+\tau_3$ ), à définir le temps de programmation de l'événement à rajouter ( $t_p(k)$ ) comme étant égal audit temps de programmation du dernier événement de la liste, à supprimer le dernier événement de la liste, et à rajouter ledit événement à rajouter à la place du dernier événement dans la liste l, sinon, si le nombre d'événements dans la liste est inférieur audit nombre fini N, à définir le temps de programmation de l'événement à rajouter comme étant égal au temps de programmation le plus proche ( $t_p(k)=t_b(k)+\tau_3$ ), et à rajouter ledit événement dans la liste.

8) Procédé selon une des revendications 4 à 7, caractérisé en ce qu'il consiste, lors d'un rajout d'un événement dans la liste pour un temps correspondant au temps le plus éloigné, à vérifier, avant de rajouter ledit événement, si le nombre d'événements dans la liste l est égal audit nombre fini N, et, si c'est le cas, à remplacer le temps de programmation ( $t_p(k)$ ) de l'événement à rajouter par le temps de programmation du dernier événement de la liste ( $t_p(N)$ ) si le débit dudit événement à rajouter ( $ER(k)$ ) est supérieur ou égal au dernier débit programmé dans la liste  $ER(N)$ , puis à programmer à la place de l'avant-dernier événement de la liste (d'ordre N-1), un événement dont le temps de programmation est celui de l'avant dernier événement de la liste l ( $ACR(N-1), t_{pi}(N-1)$ ), et le débit est le plus grand débit entre les débits des deux derniers événements de la liste l ( $ACR(N-1) = \text{Max}(ACR(N-1), ACR(N))$ ), puis à rajouter l'événement à rajouter à la place du dernier événement dans la liste l.

9) Procédé selon la revendication 5, caractérisé en ce qu'il consiste, lors du rajout d'un événement dans la liste l, à vérifier si le nombre d'événement contenu dans la liste l ne dépasse pas ledit nombre fini, et si tel est le cas, à rechercher dans la liste l deux événements successifs pour lesquels le produit de l'augmentation ou de la diminution de débit par la mesure de l'intervalle de temps compris entre leurs deux temps de programmation soit minimum puis à leur substituer un événement dont le débit est le plus grand des deux débits et le temps de programmation est le premier des deux temps de programmation.

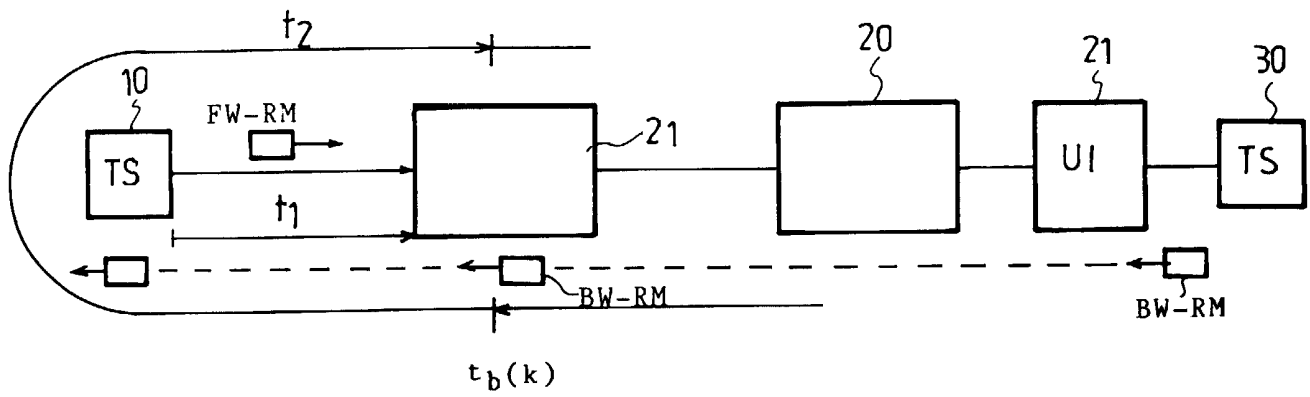


FIG.1

FIG.1a

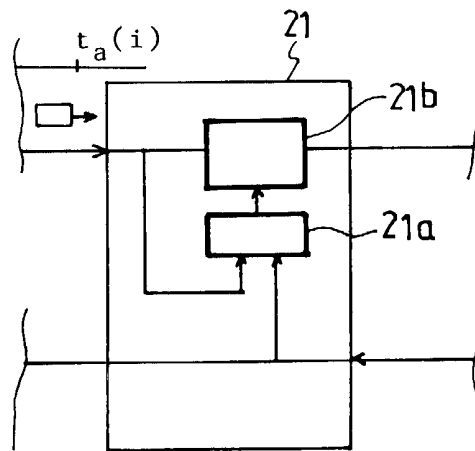
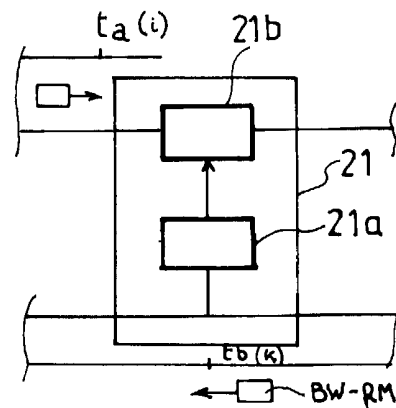


FIG.1b



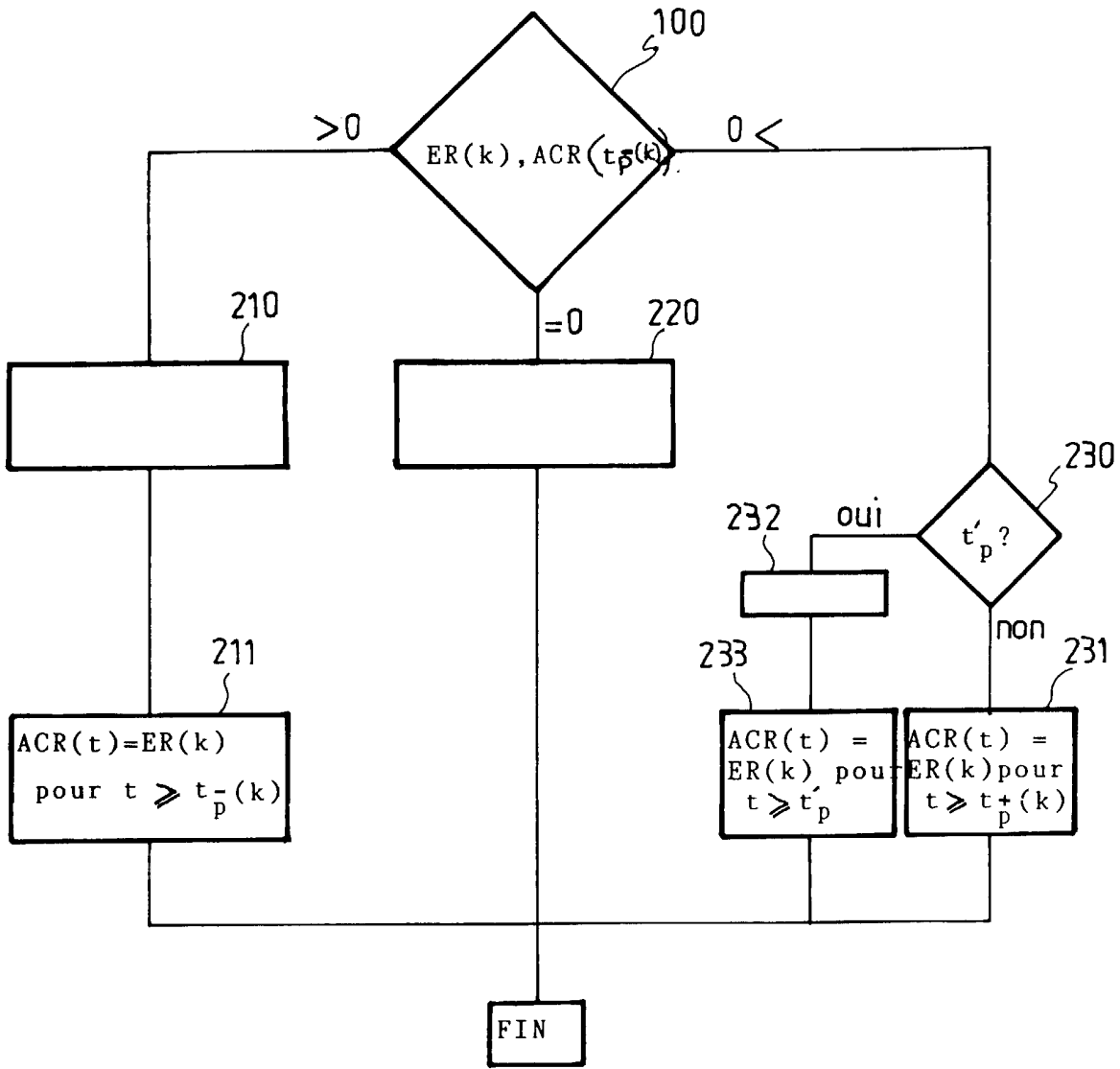
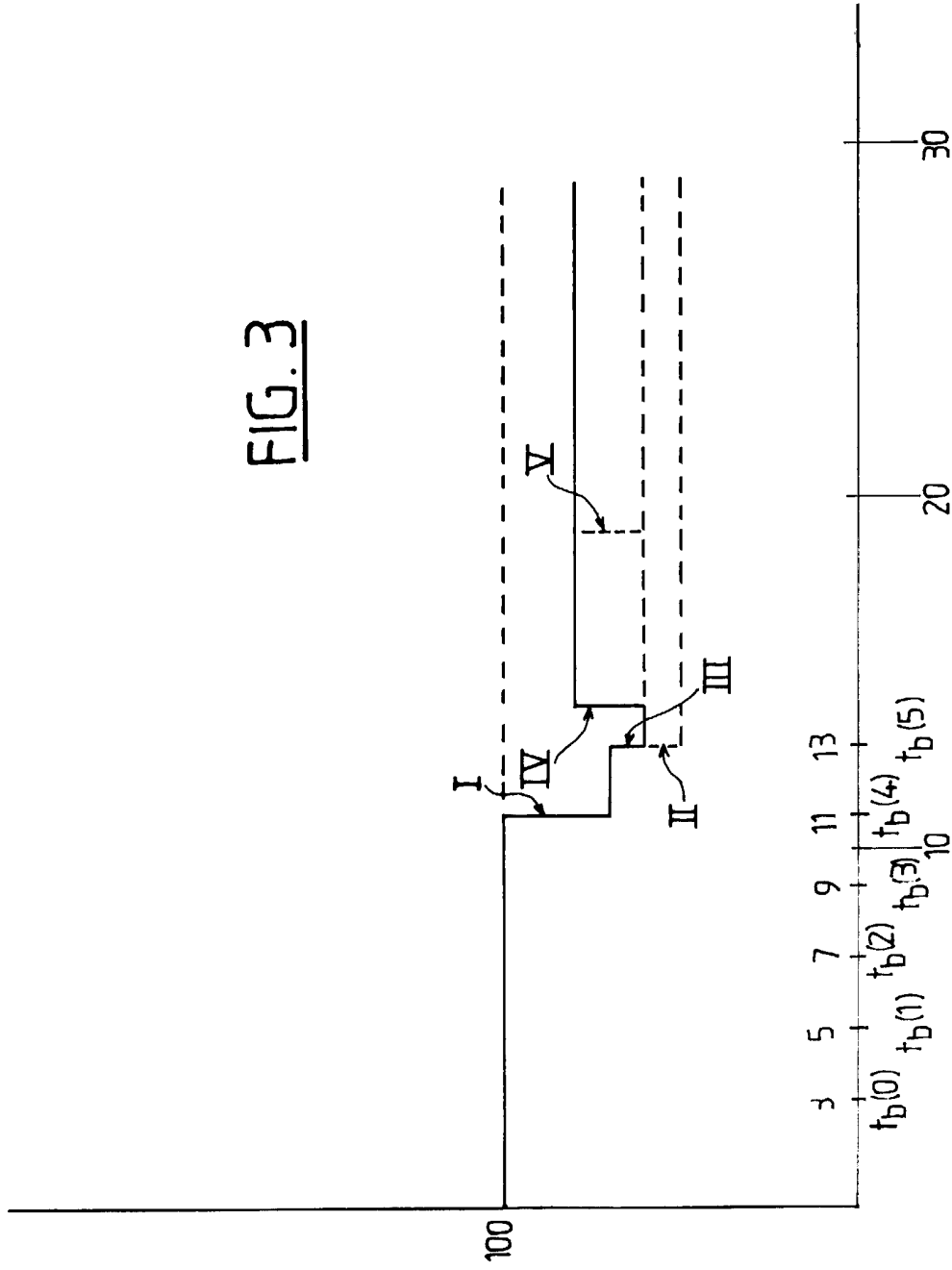


FIG. 2

$\tau_3 = 5$   
 $\tau_2 = 8$

FIG. 3



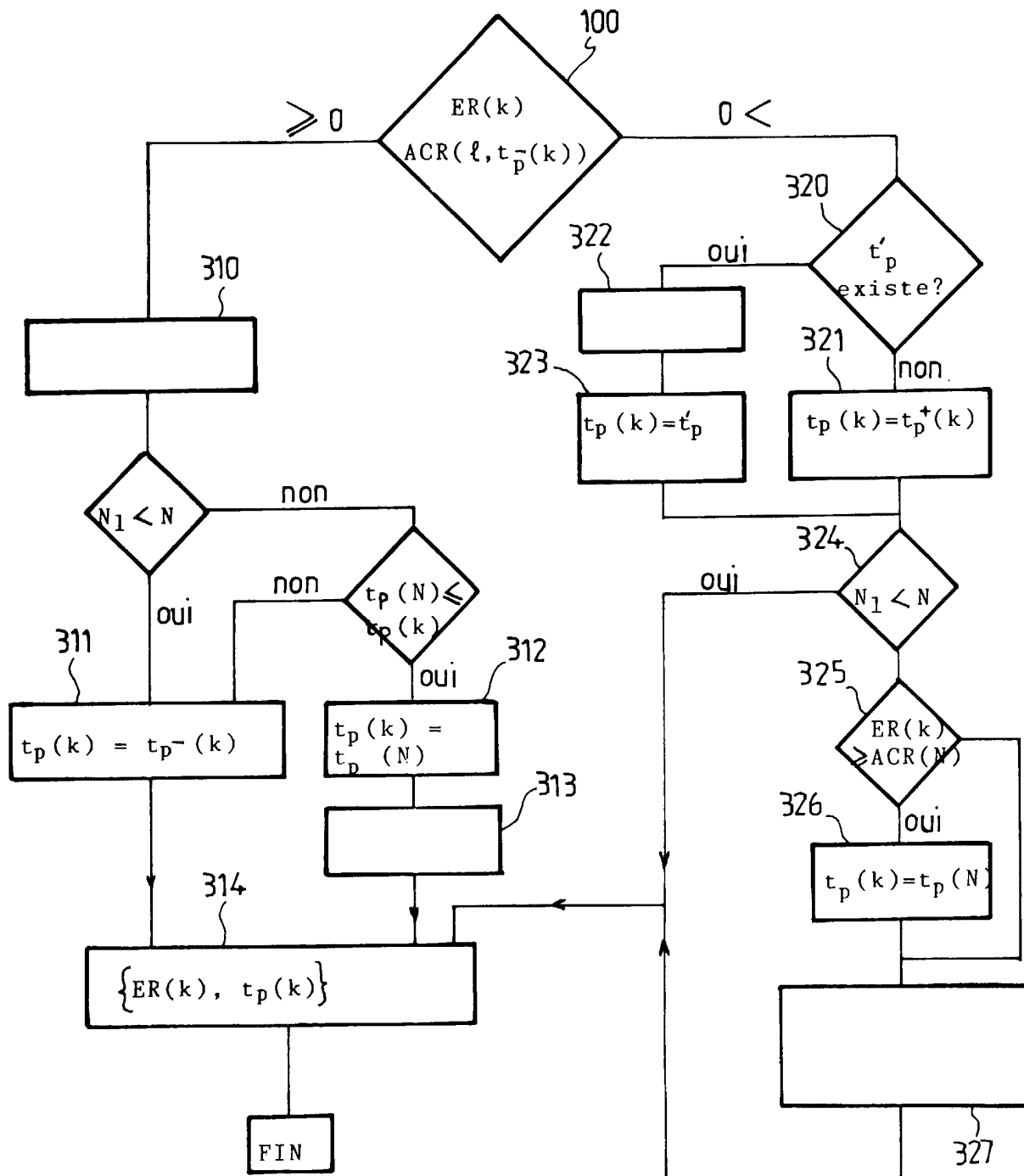


FIG. 4

$$\zeta_3 = 10$$

$$\zeta_2 = 20$$

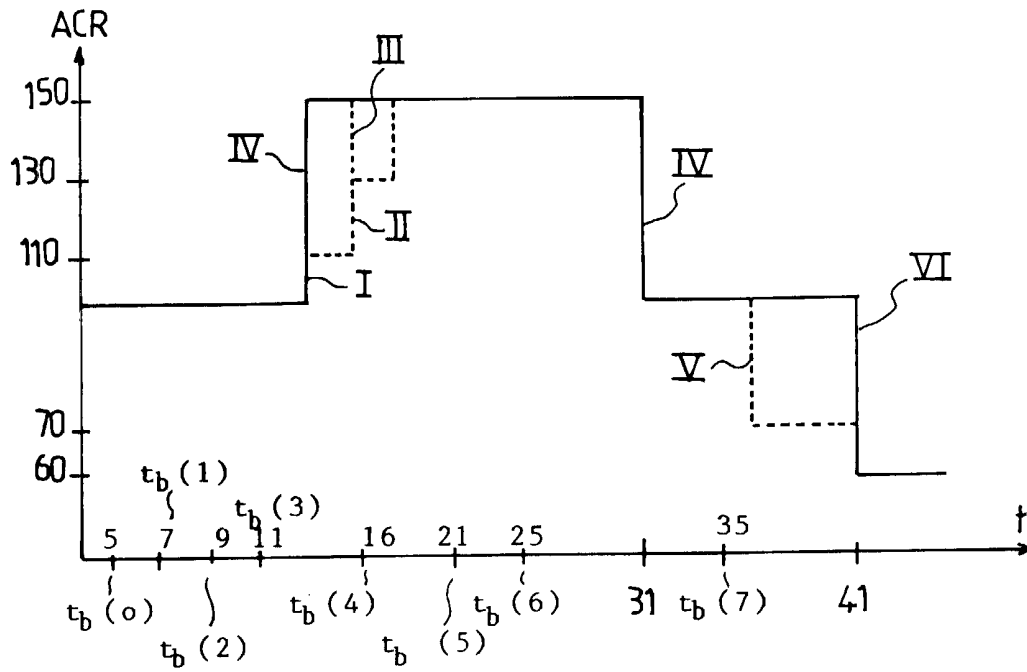


FIG.5

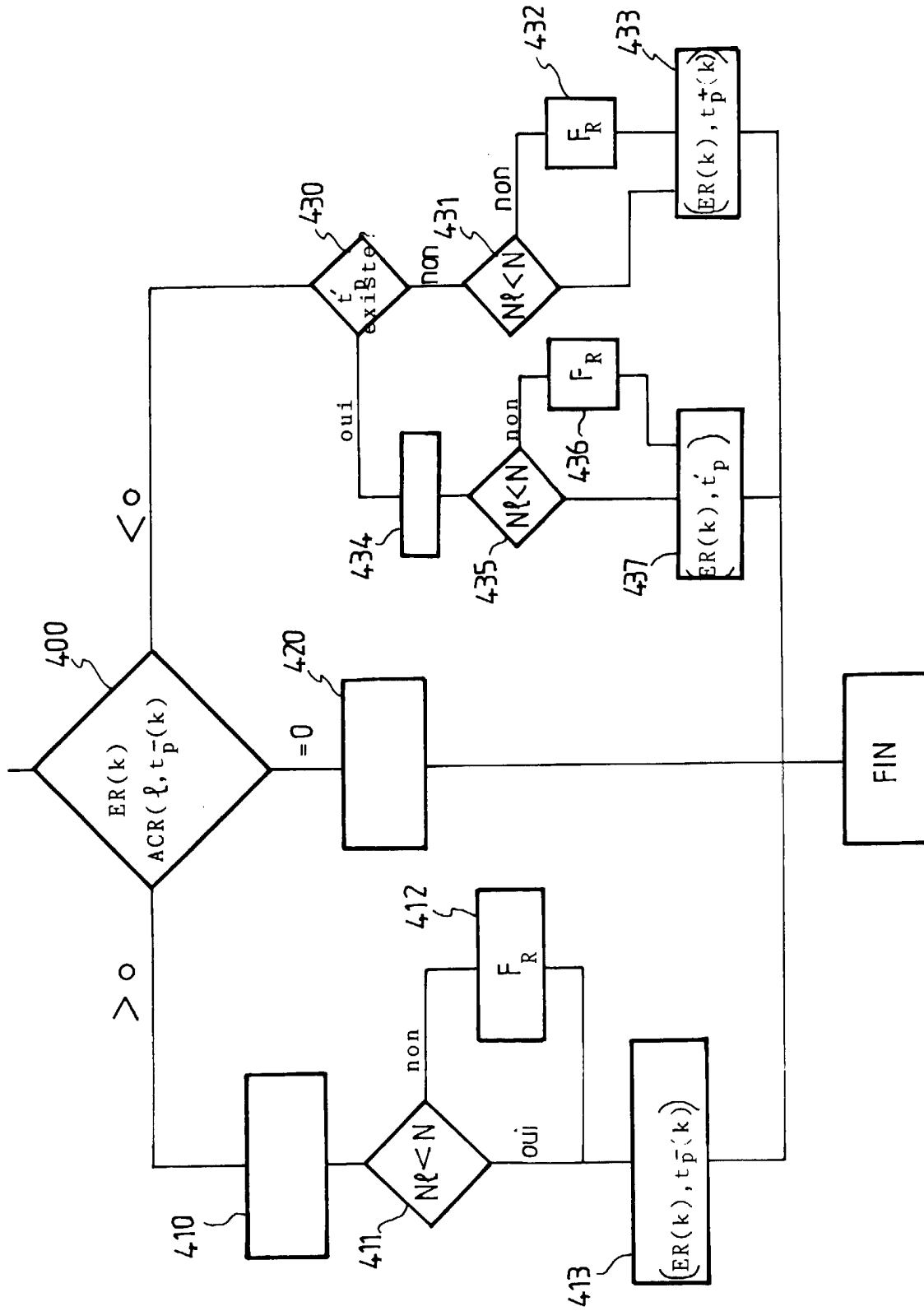


FIG. 6

$$\zeta_3 = 10$$

$$\zeta_2 = 20$$

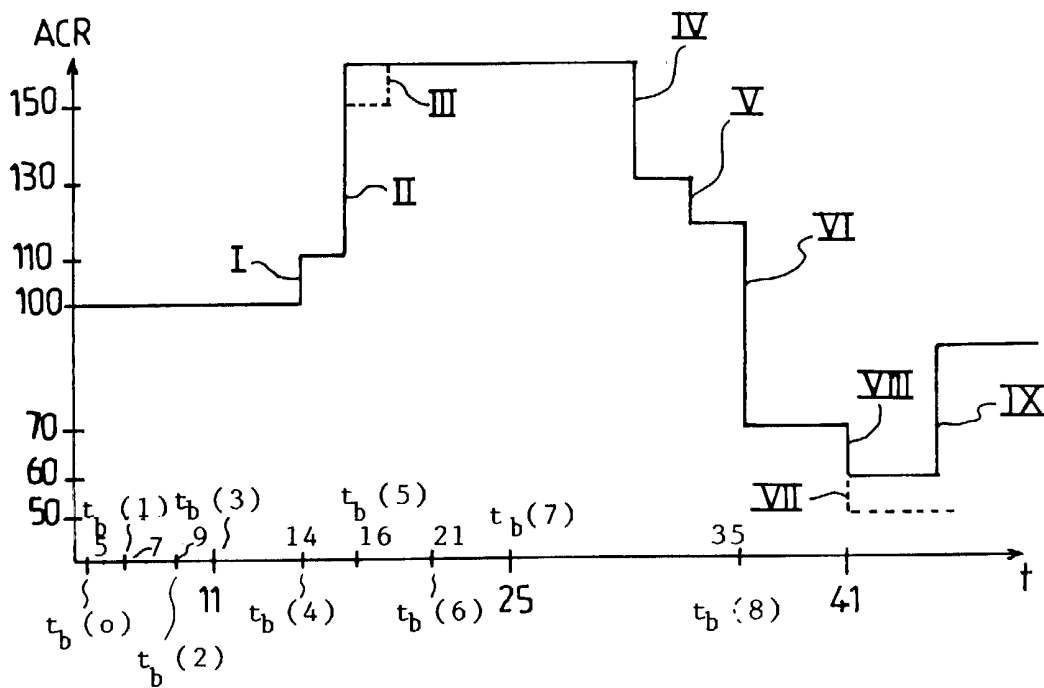


FIG. 7

RAPPORT DE RECHERCHE  
PRELIMINAIRE

établi sur la base des dernières revendications  
déposées avant le commencement de la recherche

N° d'enregistrement  
national

FA 558078  
FR 9803274

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
A	GB 2 301 977 A (FUJITSU LTD) 18 décembre 1996 * page 15, ligne 7 - page 18, ligne 28 *	1-9
A	LIU K ET AL: "DESIGN AND ANALYSIS OF A BANDWIDTH MANAGEMENT FRAMEWORK FOR ATM-BASED BROADBAND ISDN" IEEE COMMUNICATIONS MAGAZINE, vol. 35, no. 5, mai 1997, pages 138-145, XP000657119 * page 141, colonne de gauche, alinéa 6 - colonne de droite, alinéa 2 *	1-9
		DOMAINES TECHNIQUES RECHERCHES (Int.CL.6)
		H04L H04Q
Date d'achèvement de la recherche		Examineur
15 septembre 1998		Lindner, A
<p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons &amp; : membre de la même famille, document correspondant</p>		

1