



HAL
open science

Transport en circuits virtuels internes sur reseau local et connexion Transpac

Michel Tournois, Raymond Trepos

► **To cite this version:**

Michel Tournois, Raymond Trepos. Transport en circuits virtuels internes sur reseau local et connexion Transpac. [Rapport de recherche] RT-0016, INRIA. 1982. inria-00071341

HAL Id: inria-00071341

<https://inria.hal.science/inria-00071341>

Submitted on 23 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IRIA

CENTRE DE RENNES

IRISA

Rapports Techniques

N° 16

TRANSPORT EN CIRCUITS VIRTUELS INTERNES SUR RÉSEAU LOCAL ET CONNEXION TRANSPAC

Ho Revu le 17.07.82
Revue 316

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tel 954 90 20

Michel TOURNOIS
Raymond TREPOS

Juillet 1982

IRISA

PUBLICATIONS INTERNES

**TRANSPORT EN CIRCUITS
VIRTUELS INTERNES
SUR RESEAU LOCAL
ET CONNEXION TRANSPAC**

**Michel TOURNOIS
Raymond TREPOS**

Publication Interne n°167 - Mai 1982



PAPIER RECUPERÉ ET RECYCLÉ

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
TÉL. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

TRANSPORT EN CIRCUITS VIRTUELS INTERNES SUR RÉSEAU LOCAL ET CONNEXION TRANSPAC

Michel *TOURNOIS* *

Raymond *TREPOS* *

Publication Interne n° 167 - Mai 1982 - 90 pages

RESUME : Ce document présente la réalisation d'un logiciel de transport type circuits Virtuels X25 sur le réseau local ARC (DATAPOINT).

Le choix des circuits Virtuels X25 permet :

- d'offrir un interface application unique indépendant de la localisation de l'utilisateur (réseau local ou réseau public X25),
- de faciliter l'interconnexion du réseau local aux réseaux publics X25.

La réalisation et l'intégration sous système constructeur de ce logiciel ainsi que la passerelle Transpac y sont détaillées.

ABSTRACT : This paper presents the implementation of a Transport software (based on X25 Virtual Circuits) on the ARC local data network (DATAPOINT).

The choice of X25 Virtual Circuits allows :

- a single application interface independent of the localisation of the user (either on the local network or on X25 public network),
- an easy interconnection of the local network to X25 public networks.

Implementation and integration of this software in the constructor's system are detailed in this paper. A chapter describes the Transpac gateway.

* I.R.I.S.A. - Campus de Beaulieu - 35042 RENNES CEDEX

Secrétariat de rédaction : Melle F. MOINET - IRISA - Lab. Informatique
U.E.R. Mathématiques et Informatique - Campus de Beaulieu
35042 RENNES CEDEX

PLAN

pages

I - <u>INTRODUCTION</u>	1
I.1. But du projet.	
I.2. Matériel utilisé.	
I.3. Logiciels DATAPOINT utilisés.	
I.4. Logiciel de Transport.	
II - <u>PROTOCOLE DE TRANSPORT INTERNE ET X25</u>	5
II.1. Choix d'un protocole de Transport interne.	
II.2. Services offerts par le réseau local.	
II.3. Problèmes posés par l'implantation de X25 niveau paquet sur le réseau local.	
II.4. Découpage modulaire des différents logiciels.	
III - <u>STRUCTURATION DES LOGICIELS</u>	14
III.1. Contraintes.	
III.2. Choix effectués.	
III.3. Structuration en processus.	
III.4. Scheduling, partage de temps.	
IV - <u>NIVEAU VOIE</u>	24
IV.1. Rôle.	
IV.2. Structuration.	
IV.3. Interfaces.	
IV.4. Automates.	
IV.5. Caractéristiques de la réalisation effectuée.	
V - <u>DRIVER RIM</u>	38
V.1. Rôle.	
V.2. Structuration.	
V.3. Primitives "utilisateur".	
V.4. Gestion des buffers.	
V.5. Le gérant RIM.	
V.6. Horloge de garde.	

- V.7. Test-isolement.
- V.8. Gestion efficace des échanges.

VI - NIVEAU LIEN 45

- VI.1. Rôle.
- VI.2. "Accès Réseau".
- VI.3. Protocole LIEN.
- VI.4. Structuration et interfaces.
- VI.5. Format des paquets lien.

VII - LA PASSERELLE 54

- VII.1. Intérêt de la passerelle.
- VII.2. Rôle de la passerelle.
- VII.3. Réalisation de la passerelle.
- VII.4. Choix effectués.
- VII.5. Traitement des appels.
- VII.6. Traitement des autres paquets de signalisation.
- VII.7. Traitement des paquets de données.

VIII - CONCLUSION 67

Annexe 1 : Interface utilisateur - niveau voie.

Annexe 2 : Interface niveau voie - niveau inférieur.

Annexe 3 : Format des paquets.

Annexe 4 : Interface utilisateur - driver RIM.

1 - INTRODUCTION

1.1. But du projet

Dans ce projet, nous avons défini et réalisé un logiciel de transport permettant indifféremment les communications entre des calculateurs connectés :

- au même ^{niveau} niveau local
- à des réseaux locaux interconnectés par TRANSPAC
- directement à TRANSPAC.

Ce logiciel de transport se caractérise par :

- 1) son aspect modulaire lui permettant de s'intégrer dans les différentes configurations (connexion au réseau local, connexion directe à Transpac, passerelle) et sous différents systèmes d'exploitation.
- 2) la définition d'un interface unique : le fonctionnement et la programmation des applications utilisant ce logiciel de transport sont identiques dans les différents cas de connexion.

Le réseau local utilisé supporte le système ARC [4] de MATRA-DATAPOINT et les calculateurs hôtes du réseau local sont homogènes : ils appartiennent à la gamme du constructeur DATAPOINT. Toutefois, une étude complémentaire a montré qu'il était peu coûteux de connecter d'autres matériels au réseau local [2].

1.2. Le matériel utilisé

- Le réseau local [1] est un réseau à diffusion (débit 2,5 Mbits/s) exploité en commutation de paquets (taille 256 octets). Le réseau local comprend :

- . un câble coaxial (longueur maximum 6 km)
- . des communicateurs appelés RIM (255 maximum) connectés aux calculateurs comme des coupleurs banalisés
- . les calculateurs de la gamme MATRA-DATAPOINT.

- les microprocesseurs 8080A et 8085A articulés autour du MULTIBUS INTEL (famille iSBC), matériel paraissant mieux adapté (performance, coûts) qu'un ordinateur DATAPOINT pour supporter la passerelle.

1.3. Les logiciels DATAPOINT utilisés

- Le système de base DOS comprenant essentiellement :

- . la gestion de l'écran-clavier
- . les accès disques.

- Le DATASHARE est un système multi-tâches sous DOS. Il permet la gestion simultanée de 1 à 24 terminaux conversationnels. De plus, sous DATASHARE, un interface de communication "universel" permet d'introduire un logiciel gérant une (ou plusieurs) liaison conversationnelle entre les terminaux et un (ou plusieurs) ordinateur central éloigné. Cet interface s'appelle MULTILINK.

N.B. Les logiciels développés doivent fonctionner sous les systèmes DOS et DATASHARE.

1.4. Logiciel de transport

Les applications utilisant le logiciel de transport peuvent :

1) entretenir des conversations internes, appelées circuits virtuels internes, au réseau local (cf. figure I.1.).

2) entretenir des conversations externes, via TRANSPAC (cf. figure I.2.)

- soit directement : cas des applications situées sur un processeur DATAPOINT isolé connecté directement à TRANSPAC.

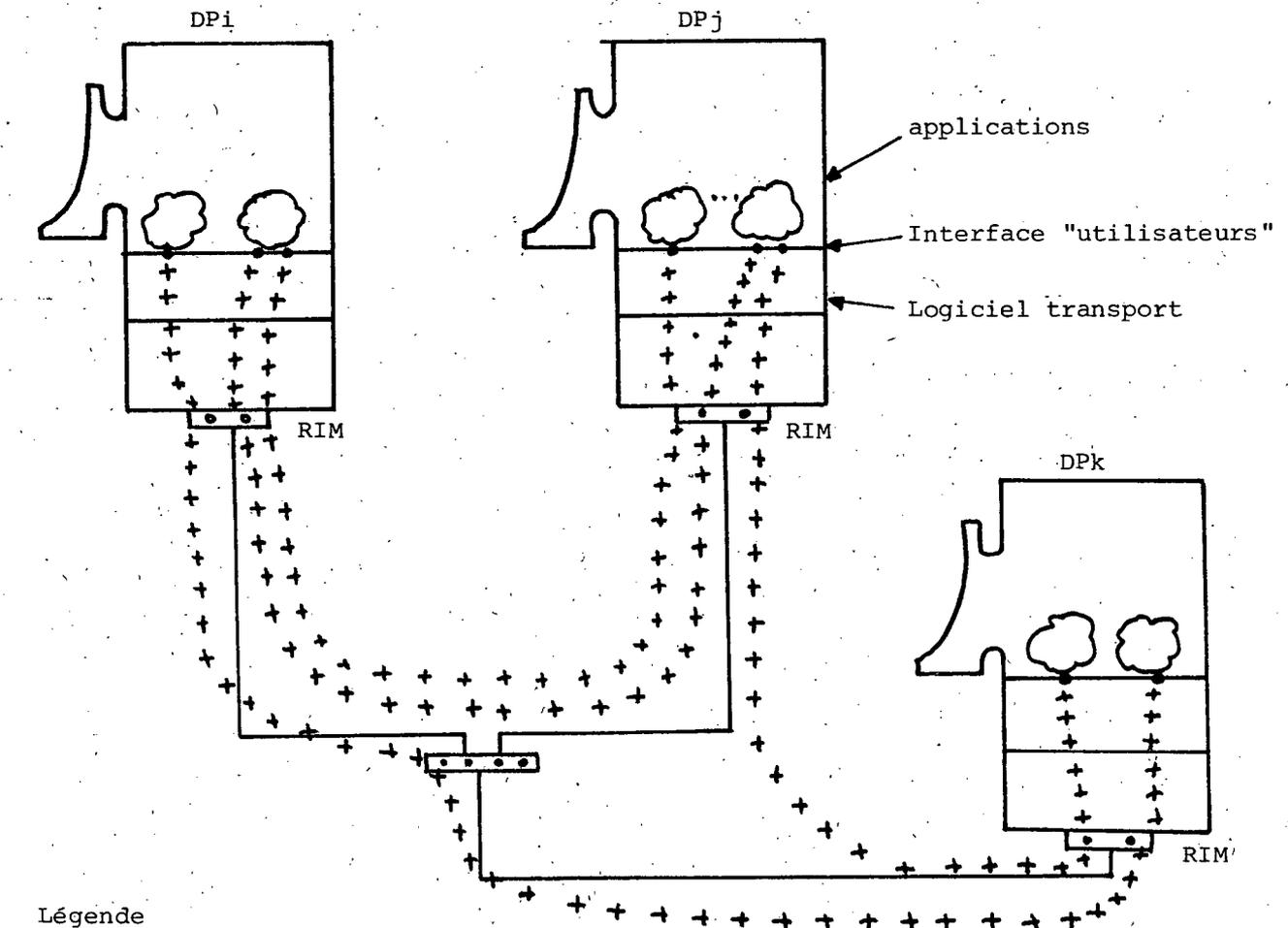
- soit par l'intermédiaire d'une passerelle (fonction GATEWAY) : cas des applications situées sur les processeurs connectés au réseau local.

L'ensemble du réseau local étant considéré par TRANSPAC comme un seul abonné TRANSPAC (ETTD), les différentes applications du réseau local sont identifiées par :

<numéro ETTD du > , <numéro complémentaire propre >
réseau local à chaque application

Le logiciel de transport offrant un interface utilisateur unique, l'écriture d'une application est la même quelque soit sa situation et celle des applications dialoguant avec elles. La seule différence provient évidemment des adresses indiquées à l'établissement des conversations dans les paquets d'appels.

Figure I.1. - Conversations internes au réseau local.



Légende



Applications = différents logiciels utilisant le service de transport

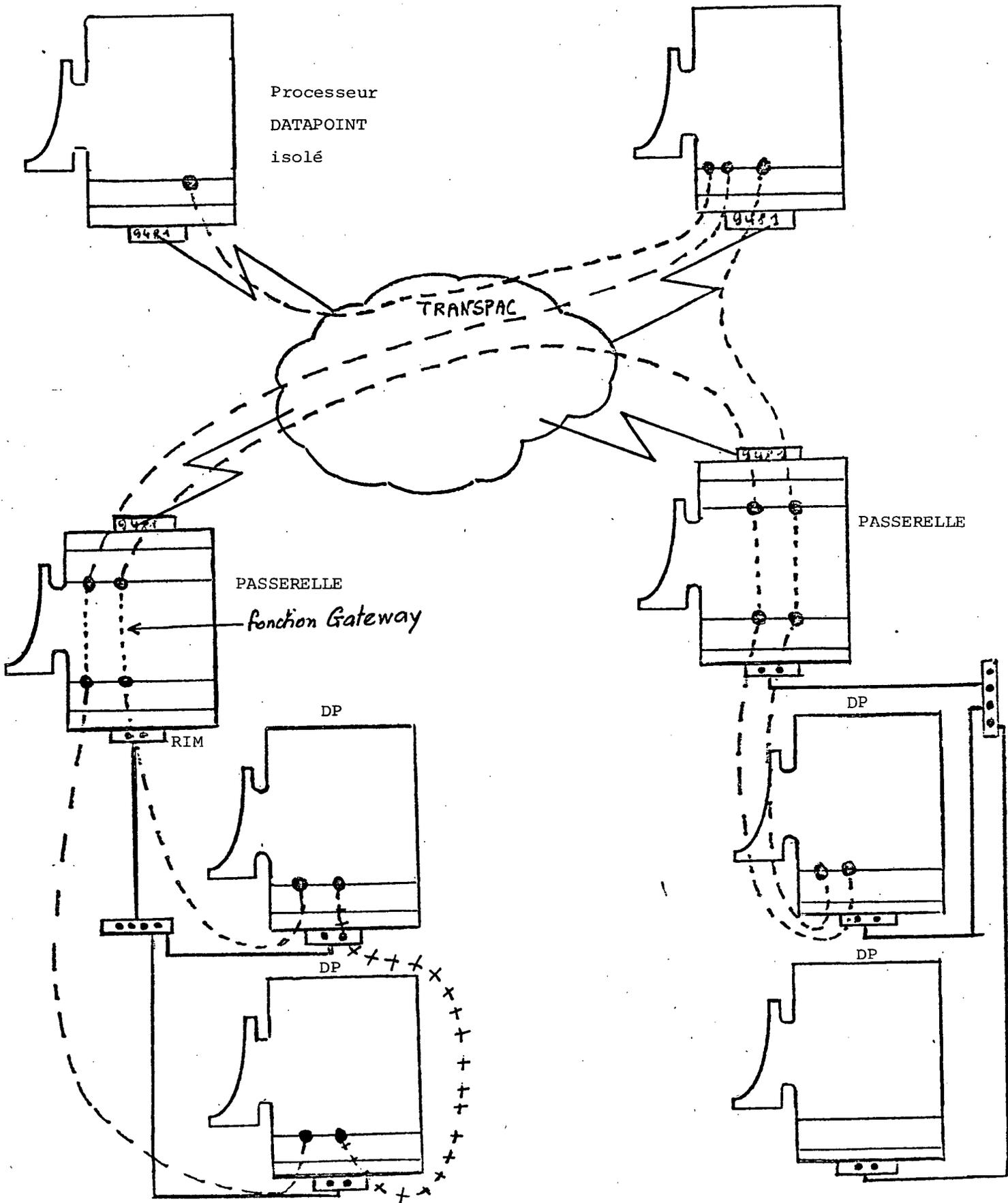
D P

calculateur DATAPOINT

++++

conversation interne au réseau local (désigné ensuite sous le nom de circuit virtuel interne ou CVI)

Figure I.2.



II - PROTOCOLE DE TRANSPORT INTERNE ET X 25

II.1. Choix d'un protocole de transport interne

Dans le chapitre précédent, nous venons de voir que les applications devaient communiquer suivant le même protocole qu'elles soient situées sur :

- un processeur local
- un processeur éloigné.

Cet objectif nous a conduit à choisir X25 niveau paquet comme protocole de transport pour les conversations internes au réseau local. Notons qu'en alignant le protocole de transport interne sur X25 niveau paquet la réalisation de la passerelle est plus aisée : le problème est de raccorder une conversation interne à une conversation externe, toutes deux régies par le même protocole. Le protocole de transport interne choisi, le problème est donc d'implanter X25 sur le réseau local DATAPOINT et ses calculateurs locaux. Par analogie à TRANSPAC, il s'agit de faire communiquer entre eux plusieurs ETTD non pas via TRANSPAC mais via le réseau local ; les ETTD étant les calculateurs connectés au réseau local.

II.2. Services offerts par le réseau local

N.B. L'utilisation détaillée du réseau local est décrite dans le document [1].

Le réseau local proprement dit comprend un câble coaxial sur lequel sont branchés des communicateurs (255 au maximum) appelés RIM ; chaque calculateur local étant connecté à une RIM.

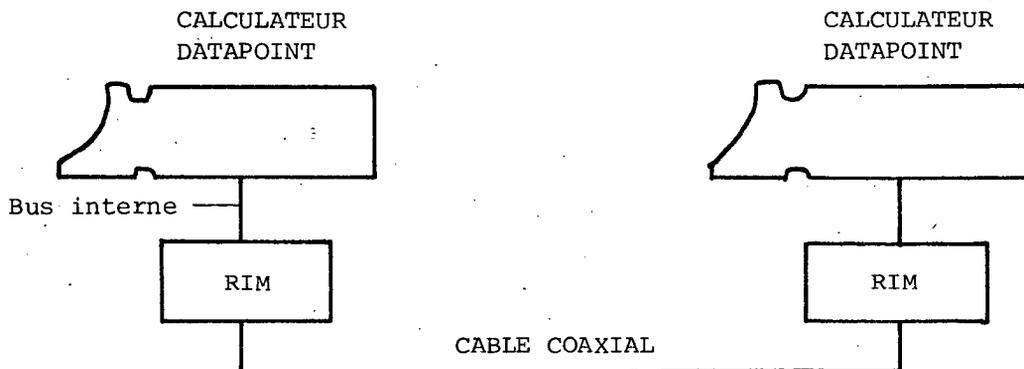


Figure II.3.

Vu du calculateur, la RIM comprend :

- 4 buffers de 256 octets
- 1 mot d'état
- 1'émetteur (TRANSMITTER)
- le récepteur (RECEIVER)
- 1'interface avec le câble coaxial (INTERFACE LINE).

La RIM comme les autres coupleurs ne génère aucun signal, c'est au calculateur d'aller consulter le mot d'état de la RIM pour connaître les événements RIM. Le mot d'état indique notamment :

- le bon fonctionnement de la RIM
- les erreurs de parité
- la disponibilité de l'émetteur (resp la fermeture du récepteur) indiquant de manière indirecte si un paquet a été émis (resp reçu)
- l'acquittement d'émission d'un paquet.

Dans tout ce qui va suivre, nous nous intéressons uniquement au mode adressé : un paquet est émis vers une seule RIM. Le mode diffusion qui permet d'émettre vers toutes les RIM ne sera pas abordé.

11.2.a) Format des paquets émis ou reçus

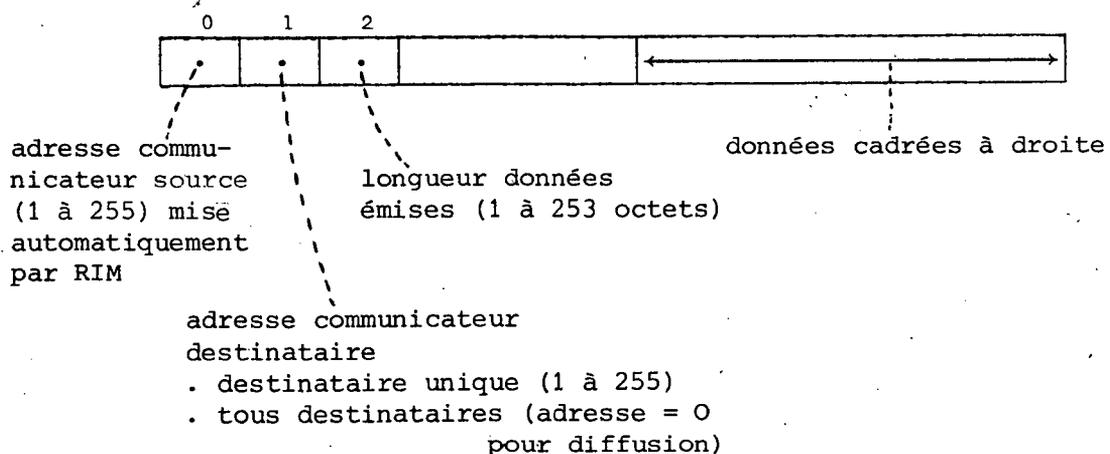


Figure II.4.

11.2.b) Pour émettre un paquet, le calculateur doit :

- 1) "Sélectionner" un buffer disponible.
- 2) Ecrire octet par octet le paquet à émettre dans le buffer choisi.
- 3) Indiquer à l'émetteur de la RIM le numéro de buffer contenant le paquet à émettre. Cet ordre ne sera accepté par la RIM que si l'émetteur est disponible (aucune émission en cours). L'émission est alors sous le contrôle de la RIM. Laquelle RIM, lorsqu'elle aura le droit de parole, transmettra le paquet après accord de la RIM réceptrice.
- 4) Examiner le mot d'état de la RIM pour savoir :
 - si l'émission a eu lieu (accord de la RIM réceptrice nécessaire pour l'émission du paquet)
 - si l'émission s'est bien déroulée (envoi d'un CRC) et si la RIM réceptrice a acquitté le paquet.

N.B. L'accord avant l'émission du paquet et l'accusé de réception de la RIM réceptrice montre qu'il existe un contrôle d'erreur et de flux sur le réseau local. Ces messages de contrôle (accord pour recevoir et accusé de réception) échappent aux calculateurs, seul le résultat (message émis, accusé de réception reçu) se reflète dans le mot d'état de la RIM.

11.2.c) Pour la réception des paquets, le calculateur doit :

- 1) ouvrir le récepteur en lui affectant un buffer; cette opération n'est valide que si le récepteur est fermé.
- 2) examiner périodiquement le mot d'état pour savoir si un paquet est arrivé (récepteur redevient fermé).
- 3) le paquet arrivé, "sélectionner" le buffer contenant le paquet arrivé.
- 4) lire octet par octet tout ou partie du paquet arrivé.

N.B. Le communicateur ne pourra rien recevoir tant que le récepteur n'aura pas été réouvert.

11.2.d) Parallélisme de fonctionnement

Il est important de souligner le parallélisme de fonctionnement entre la RIM et le calculateur hôte. Le calculateur lit (resp émet) les paquets reçus

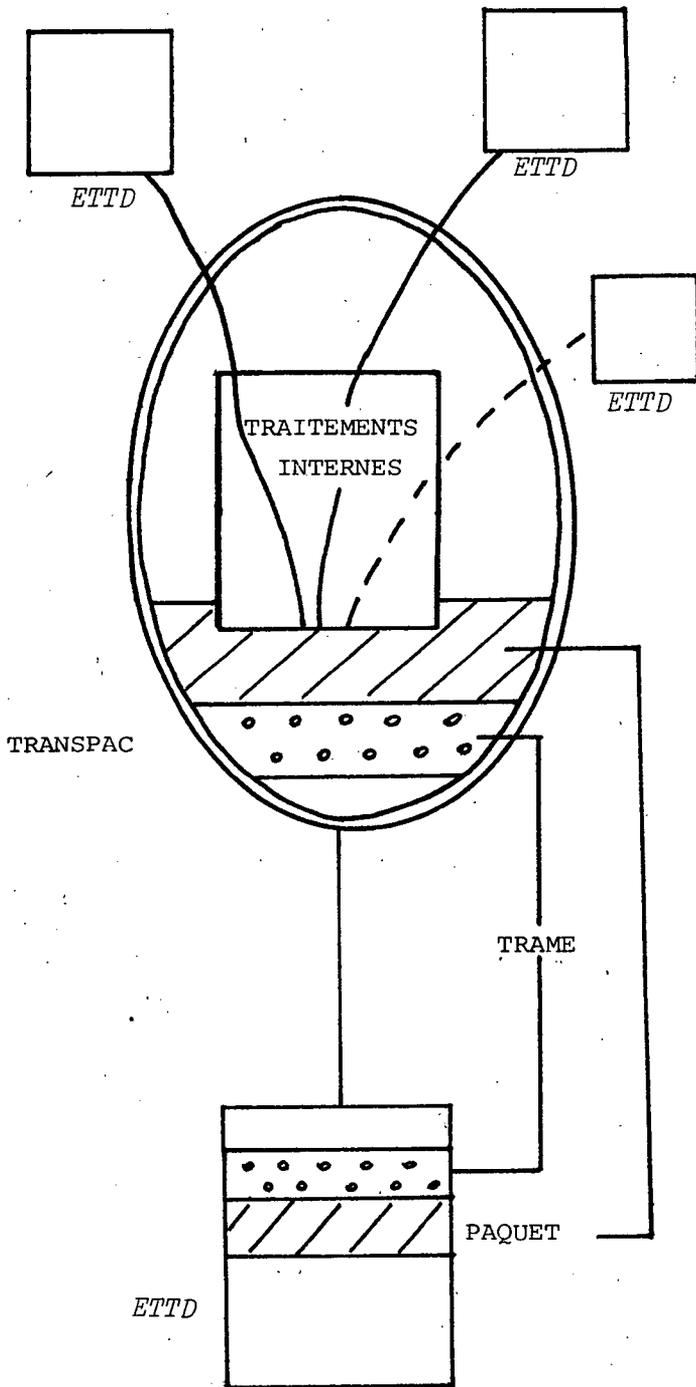


Figure II.5. - Différents niveaux des circuits virtuels.

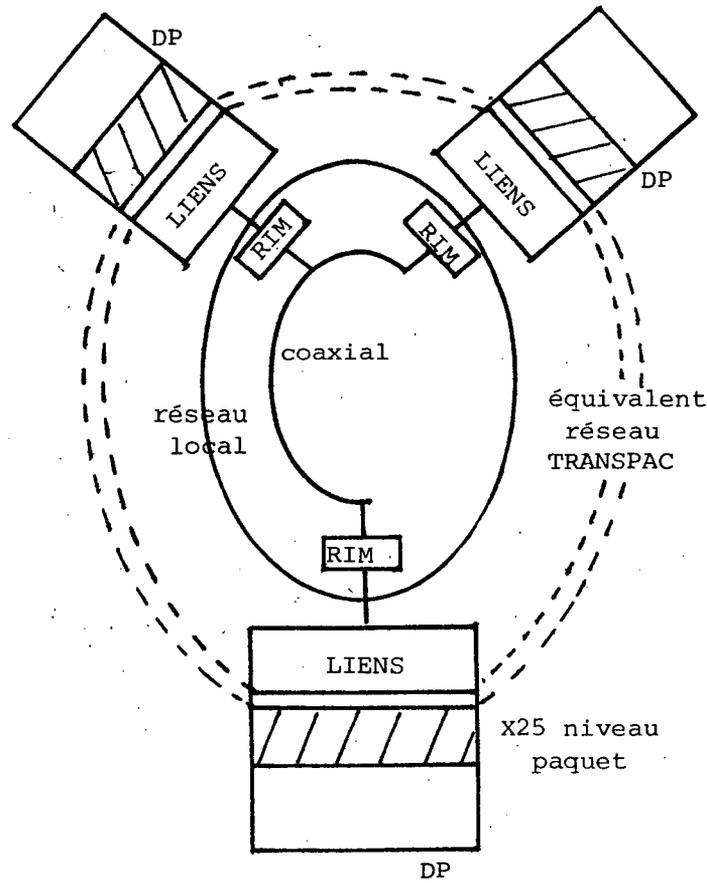


Figure II.6. - Différents niveaux des circuits virtuels internes.

DP = calculateur DATAPOINT connecté au réseau local.

(resp à émettre) dans la RIM et donne des directives à la RIM mais le transfert des paquets entre RIM est entièrement assuré par les RIM sans intervention des calculateurs.

11.3. Problèmes posés par l'implantation de X25 niveau paquet sur le réseau local.

Reprenons l'analogie avec TRANSPAC. Plus précisément, examinons les fonctions indispensables à rajouter au réseau local pour qu'il joue le même rôle vis à vis des calculateurs locaux que TRANSPAC vis à vis des ETTD.

11.3.a) Fonctions indispensables à rajouter au réseau local

- Sur TRANSPAC, le numéro de voie logique suffit à identifier un circuit virtuel entre l'ETTD et TRANSPAC.

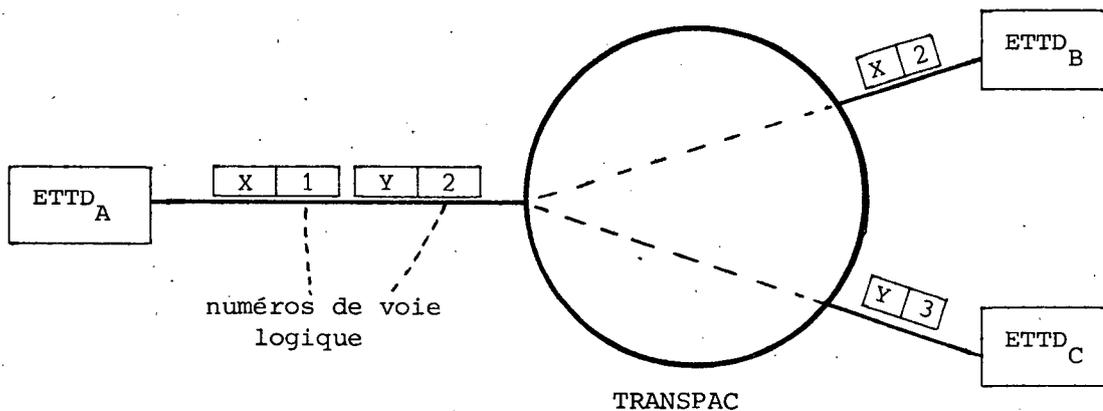


Figure II.7.

Le numéro de voie logique est purement local. Sur la figure II.7. par exemple pour le circuit virtuel X reliant l'ETTD A à l'ETTD B, les paquets sont émis ou reçus avec le n° de voie 1 pour l'ETTD A, et le n° de voie 2 pour l'ETTD B.

A l'établissement de chaque circuit virtuel, TRANSPAC associe deux numéros de voie logique : le n° de voie logique fourni par l'ETTD émetteur du

paquet et le plus petit numéro de voie logique disponible chez l'ETTD récepteur. Une fois le cv établi, TRANSPAC convertit pour chaque paquet transmis le n° de voie fourni par l'ETTD émetteur en le n° de voie associé de l'ETTD récepteur. Donc cette gestion des voies logiques des abonnés ETTD est indispensable à rajouter au réseau local.

- D'autre part, TRANSPAC connaît en permanence l'état des ETTD : (absent, saturé...), ce qui permet à TRANSPAC de refuser un appel tout en indiquant le motif du refus.

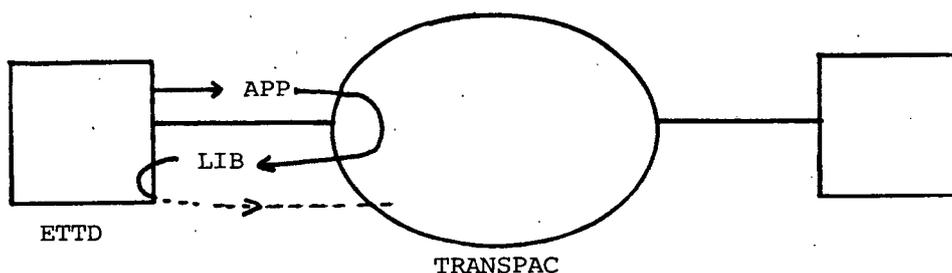


Figure II.8. - Cas d'un appel refusé.

Cet exemple montre que l'ETTD dialogue avec le réseau TRANSPAC ; dans le cas de l'appel refusé par TRANSPAC, l'ETTD visé ne joue aucun rôle. La connaissance de l'état des abonnés par le réseau, diffusé aux applications par les libérations est nécessaire à rajouter au réseau local.

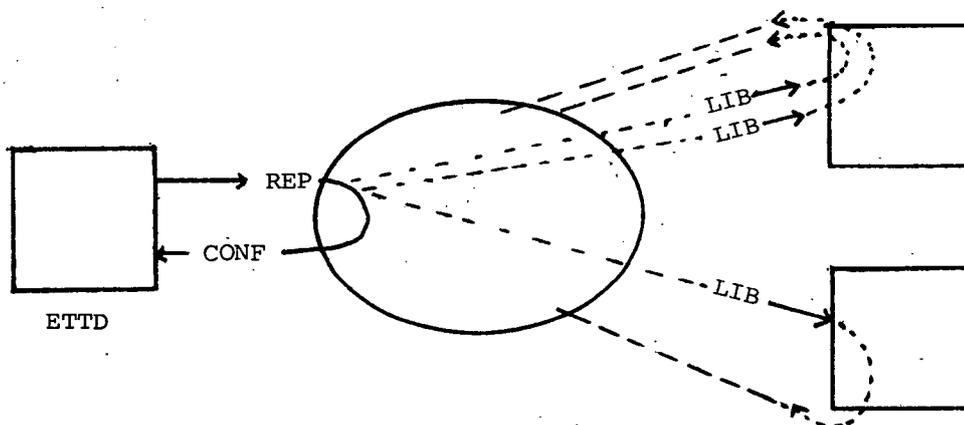


Figure II.9. - Reprise.

Il s'agit comme pour TRANSPAC (Fig. II.9.) de transformer le paquet de reprise émis par un calculateur en paquets de libération sur tous ses circuits virtuels ouverts. Notons aussi que le paquet REPRISE pour TRANSPAC peut être émis par le réseau; il sert donc aussi à signaler aux ETTD l'état opérationnel ou non opérationnel du réseau.

11.3.b) Le niveau LIENS

En résumé, les fonctions à rajouter au réseau local sont :

- gestion des voies logiques et routage des paquets
- surveillance de l'état opérationnel des abonnés par le réseau
- signalisation de l'état opérationnel ou non opérationnel du réseau
- traitement des paquets de reprise.

Les communications du réseau local n'étant pas prévues pour être facilement modifiées, l'ensemble de ces fonctions est assuré par un logiciel réparti sur les calculateurs connectés au réseau local et appelé LIENS.

Remarque : Les calculateurs locaux possèdent donc la version X25 niveau paquet plus le niveau LIENS. Le réseau local jouant un rôle uniquement de transport ; les paquets sont directement échangés entre les calculateurs locaux. Ce dialogue direct n'existe pas pour les abonnés TRANSPAC, le réseau jouant là un rôle actif dans l'échange des paquets.

11.3.c) Accès au réseau

L'accès au réseau TRANSPAC pour les ETTD en mode paquet se fait sur une (ou plusieurs) liaisons gérée par la procédure TRAME. Pour les CVI, l'équivalent du réseau TRANSPAC étant réparti sur chaque calculateur local, l'équivalent du niveau TRAME disparaît : le niveau paquet CVI et le niveau LIENS communiquent par mémoire commune.

11.4. Découpage modulaire des différents logiciels

11.4.a) Logiciel de transport des circuits virtuels internes

(cf. fig. II.10.a)

Le logiciel de transport des calculateurs locaux connectés au réseau

local comprendra 3 niveaux :

- le niveau voie (ou niveau paquet) correspondant au niveau 3 de X25
- le niveau LIENS
- le DRIVER RIM assurant la gestion du coupleur RIM.

11.4.b) Logiciel de circuit virtuel (cf. fig. 11.10.b)

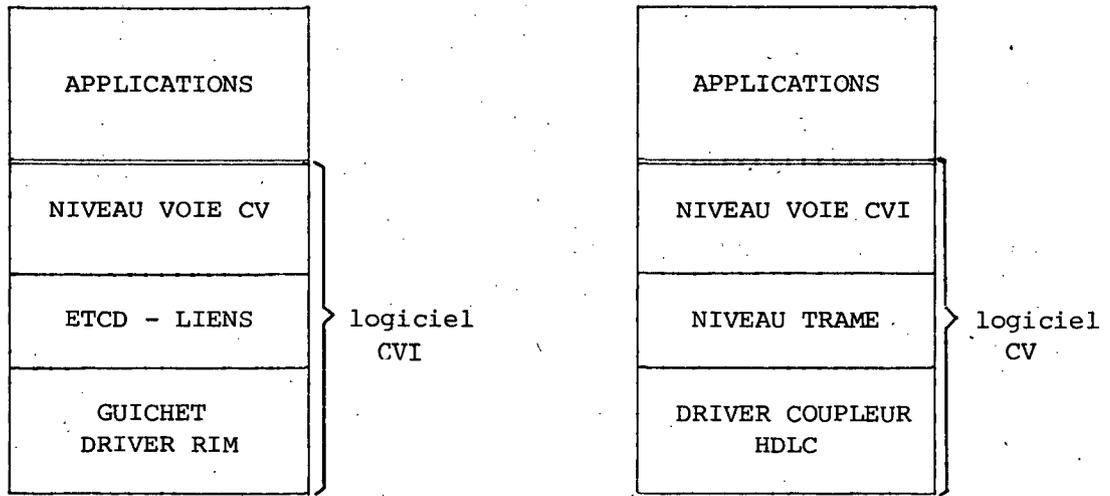
Les processeurs DATAPOINT éloignés isolés possèdent :

- le niveau voie identique à celui des CVI
- le niveau TRAME
- la gestion du coupleur HOLC ou 9481.

11.4.c) Logiciel de la passerelle (voir fig. 11.10.c)

La passerelle gérant l'accès à TRANSPAC et étant un processeur local banalisé comporte les 2 versions de logiciel : circuit virtuel CV et circuit virtuel interne CVI plus un module JONCTION faisant le raccord entre les CV et CVI.

Figure II.10. - Logiciels



II.10.a - processeur connecté au réseau local (autre que la passerelle)

II.10.b - processeur isolé connecté à TRANSPAC

JONCTION	
NIVEAU VOIE CVI	NIVEAU VOIE CV
ETCD-LIENS	NIVEAU TRAME
DRIVER RIM	DRIVER COUPLEUR HDLC

II.10.c - Passerelle

III - STRUCTURATION DES LOGICIELS

III.1. Contraintes

III.1.a) Servir le coupleur HDLC (9481)

Le coupleur 9481 dispose de 2 buffers de 63 caractères : un affecté pour l'émission et le second pour la réception.

Pour piloter le 9481 à la bonne cadence, il faudra :

- en réception aller lire les caractères suffisamment vite pour que le buffer réception ne soit pas plein à l'arrivée d'un caractère
- pour une trame en cours d'émission, éviter que le buffer émission soit vide; le coupleur 9481 répète le dernier caractère émis lorsque le buffer émission est vide.

Donc pour gérer efficacement le 9481, il faudra disposer d'une horloge précise pour suivre le rythme des émissions et des réceptions sans aller examiner le coupleur trop fréquemment, (le coupleur 9481 ne générant aucun signal).

III.1.b) Fonctionner sous DOS et MULTILINK de DATASHARE

Sous ces systèmes, on distingue 2 niveaux d'exécution de logiciel :

- le "foreground" activé toutes les millisecondes
- le "background" interrompu toutes les ms par le foreground et ayant le contrôle lorsque le foreground ne l'a pas.

L'activation du foreground se fait par une horloge hardware qui, si elle n'est pas masquée, interrompt le background. Le masquage de l'interruption horloge est effectif lorsque le foreground garde le contrôle plus d'une ms ou lorsque le background masque l'interruption.

Sous le MULTILINK de DATASHARE, le logiciel de transmission s'exécute entièrement en foreground. Si on ne veut pas perdre d'activations foreground, les traitements ont une durée maximum inférieure à 1 ms. Si l'on tient compte que sous le niveau foreground peuvent s'exécuter des traitements des processus foreground propres au DATASHARE ; la durée maximum d'une activation sous

Multilink est de l'ordre de $QT \approx 600 \mu s$; durée au bout de laquelle le processus activé sous MULTILINK doit rendre le contrôle au DATASHARE. D'autre part, la place mémoire d'un logiciel ne doit pas dépasser 8 k octets.

III.2. Choix effectués

III.2.a) Non recopie des données

- . Les buffers émission et réception sont :
 - fournis par les applications utilisant le niveau paquet
 - attachés à une voie donnée.

L'objectif est d'éviter ou de minimiser la recopie de données entre les niveaux paquet et le niveau inférieur. Les données reçues du coupleur sont directement écrites dans le buffer réception de l'application, le traitement étant symétrique pour l'émission.

Sur TRANSPAC par exemple, le fait d'écrire directement du coupleur dans le buffer associé à la voie va nous obliger à un échange d'informations entre le niveau TRAME et le niveau voie durant la réception d'une trame contenant un paquet donné. En effet il faudra :

- lire l'entête du paquet de données
- analyser cet entête au niveau voie pour qu'il retourne au niveau trame le buffer ad hoc (si le paquet de données est attendu).
- le buffer déterminé, écrire directement dans le buffer et à la fin de la trame, prévenir le niveau voie de la validité ou non de la trame reçue.

Remarque : Le choix effectué d'associer un buffer à une voie nous a conduit à écarter des solutions de gestion dynamique de mémoire comme l'échange d'un buffer plein contre un buffer vide entre le niveau voie et le niveau TRAME. Ce choix est particulièrement intéressant pour recevoir plusieurs paquets successifs dans une zone mémoire contiguë ; exemple : 2 paquets de 128 octets pour 1 buffer disque de 256 octets.

III.2.b) Répercussion directe des événements

Nous venons de voir que la réception d'une trame contenant un paquet de données imbriquait des traitements de différents niveaux (coupleur, trame, voie). Cette imbrication n'est possible que si les traitements sont séquentiels. Nous avons gardé cette philosophie pour tous les événements :

- acquittement d'émission
- réception de paquet de signalisation
- progression de la fenêtre, etc...

Donc la détection d'un événement est immédiatement répercutée sur les niveaux adjacents ; cela se traduit par une cascade d'appels de procédures des différents niveaux jusqu'au niveau utilisateur. Ces procédures sont appelées procédures-événements. Une procédure-événement est appelée par le niveau inférieur qui lui signale l'événement dès qu'il se produit et généralement toutes les informations concernant cet événement. On évite ainsi au maximum les files d'attente intermédiaires pour mémoriser les événements ; ceci bien entendu dans les limites permises par les contraintes précédentes (servir le coupleur et activation $\sim 600 \mu\text{s}$).

III.2.c) Minimiser le nombre de processus

Nous avons vu que sous MULTILINK, les contraintes de fonctionnement étaient fortes (servir efficacement le coupleur, découper en traitements inférieurs à $600 \mu\text{s}$) dans le cas des circuits virtuels. Comme d'autre part, les outils systèmes fournis étant très limités, nous nous sommes efforcés de structurer les logiciels de transport avec un nombre minimum de processus. Ainsi, il est plus facile de trouver un pilotage des processus répondant aux contraintes (cf III.4.).

III.2.d) Niveau voie unique

Le niveau voie étant le même pour les CV et les CVI, l'interface niveau voie - niveau inférieur est aussi identique (cf chapitre IV).

III.3. Structuration en processus

III.3.a) Structuration logiciel CV

Les processus composant le logiciel CV comprennent (cf. fig. III.11) :

- le processus RT (Réception Trame) gérant la réception du coupleur, le traitement des trames reçues au niveau trame et au niveau paquet. Ce processus sera activé à des périodes inférieures au temps de remplissage du buffer de réception du coupleur.

- le processus ALT (Alimentation Trame) activé lorsqu'il y a un paquet à émettre au niveau voie.

- le processus ET (Emission Trame) activé par :

- . ALT pour les trames d'information à émettre

- . RT pour les trames de supervision

- . le processus Horloge pour les sonneries des réveils trame impliquant une réémission.

- le processus H (Horloge) gérant les réveils trame et voie.

III.3.b) Structure du logiciel CVI (cf. fig. III.12).

Les processus constituant le logiciel CVI comprennent :

- le gérant RIM (GR) qui va examiner fréquemment l'état de la RIM pour détecter l'arrivée d'un message, l'émission d'un message, les anomalies. Ces événements sont communiqués au processus RL. La RIM comme les autres coupleurs DATAPOINT ne génère aucun signal, le processus GR devra être activé périodiquement pour gérer efficacement les réceptions et les émissions ; notamment, pour relancer l'émission d'un buffer contenant un message prêt à émettre ou réouvrir le récepteur sur un buffer vide (voir chapitre DRIVER RIM).

- le processus RL (Réception Lien) chargé comme le processus Réception Trame de lire les messages reçus et de répercuter l'émission correcte des paquets au niveau voie. Il est activé par le gérant RIM.

- le processus EL (Emission Lien) chargé d'écrire dans la RIM les paquets à émettre pour le compte du niveau voie (activé par ALT) et pour le compte du niveau liens (activé par RL).

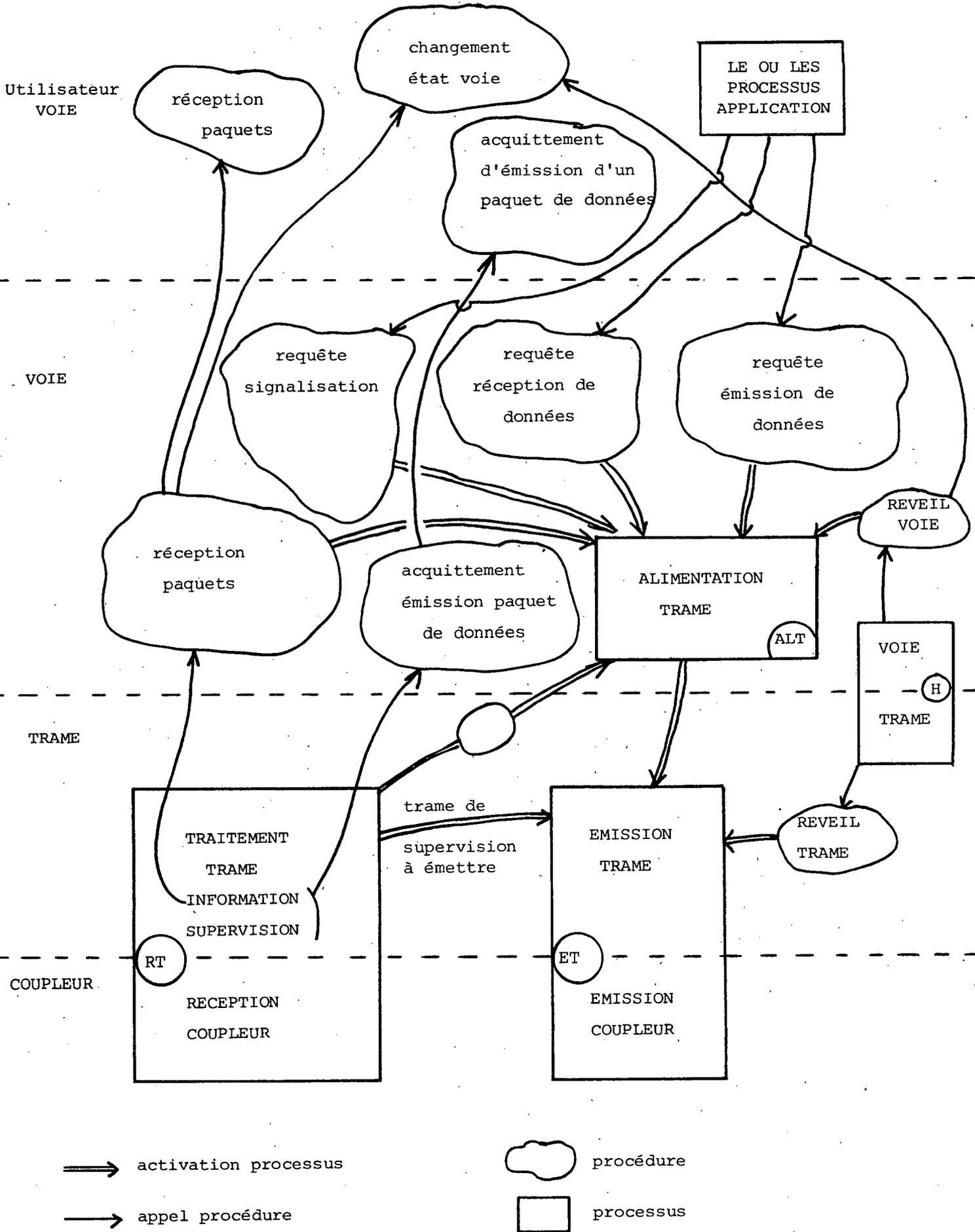


Figure III.11. - Logiciel CV

- le processus ALT (Alimentation trame)
- le processus Horloge H chargé de gérer les réveils au niveau voie, liens et RIM.

III.4. Scheduling, partage de temps

III.4.1. Scheduling

Il a été défini un scheduling spécifique pour satisfaire aux contraintes et aux choix indiqués ci-dessus.

Les caractéristiques principales de ce scheduling sont :

- une file des processus activables ; la priorité interprocessus est définie une fois pour toutes.
- un niveau de primitives très simple qui permet à un processus de :
 - . rendre activable un processus (ACTIVE)
 - . se désactiver (DACTIVE) c'est-à-dire sortir de la file des activables
 - . changer le point d'entrée d'un processus (NEWPOINT)
- une gestion des horloges du système faite dans le noyau du scheduler.

Les contraintes dues au système DATASHARE-MULTILINK (tout en foreground, traitement inférieur à un quantum de temps donné QT (une fraction de ms)) ont conduit à prendre le type de fonctionnement suivant :

- . le scheduler ne donne le contrôle qu'à un seul processus par ms : le plus prioritaire, c'est-à-dire celui qui est en tête de la file des processus activables.

- . chaque processus doit limiter son temps de traitement à un maximum QT. Cela a nécessité d'évaluer à priori le temps d'exécution de chaque traitement et de fractionner certains traitements. Le processus rend alors le contrôle (en changeant éventuellement son point d'entrée) tout en restant dans la file des activables. Cela impose aussi à l'"application" de limiter le temps de traitement dans les "procédure s-événements".

Les contraintes matérielles (coupleur HDLC ou communicateur RIM) ont conduit à distinguer deux sortes de processus :

- les processus qui nécessitent d'être activés périodiquement par exemple Réception Trame (RT) ou gérant RIM (GR). Ces processus sont introduits dans la file à leur niveau de priorité sur sonnerie d'horloge.

- les processus qui sont rendus activables par les autres processus, par exemple l'alimentation trame (ALT) ou les processus "application".

Il y a toujours au moins un processus activable : c'est le processus "application" de plus basse priorité.

III.4.2. Partage de temps TRANSPAC

Les contraintes de service du coupleur HDLC peuvent se résumer ainsi : garantir une bonne cadence en réception et émission et éviter au maximum les accès inutiles au coupleur (lorsque celui-ci est vide) pour ne pas monopoliser pour rien le quantum de temps Foreground.

On a donc deux processus accédant directement au coupleur (cf III.3) lancés périodiquement, dont le fonctionnement est décrit ci-dessous :

le processus RT est le processus de plus haute priorité. Il prend le contrôle à intervalles de temps T ms ; T est le temps que met à se remplir le buffer réception du coupleur ; par exemple, pour le coupleur 9481 à 9800 bauds disposant d'un buffer de 63 caractères, T est égal à $\frac{63 \times 8}{9800} \approx 52$ ms et avec une marge de sécurité T = 50 ms. RT reste activable tant que le coupleur n'est pas vide : le traitement réception est donc bloqué en début de cycle de T ms. L'estimation à priori des durées des traitements réception montre que RT se termine avant T/2 quel que soit l'état de remplissage du coupleur. Pour vérifier que cette limite était bien respectée dans tous les cas de figure on a examiné les cas les plus défavorables : par exemple, la présence en début de cycle dans le coupleur de plusieurs trames de service ou contenant des paquets courts nécessitant un traitement important soit au niveau trame, soit au niveau voie, soit les deux. Cette analyse est détaillée dans "spécification de réalisation Trame". Le lancement du processus ET a donc été cadré à T/2. Celui-ci se termine avant T soit parce que le coupleur est plein (il reste alors activable au cycle suivant) soit parce que l'émission est terminée.

Le partage de temps du cycle T peut donc se schématiser ainsi :

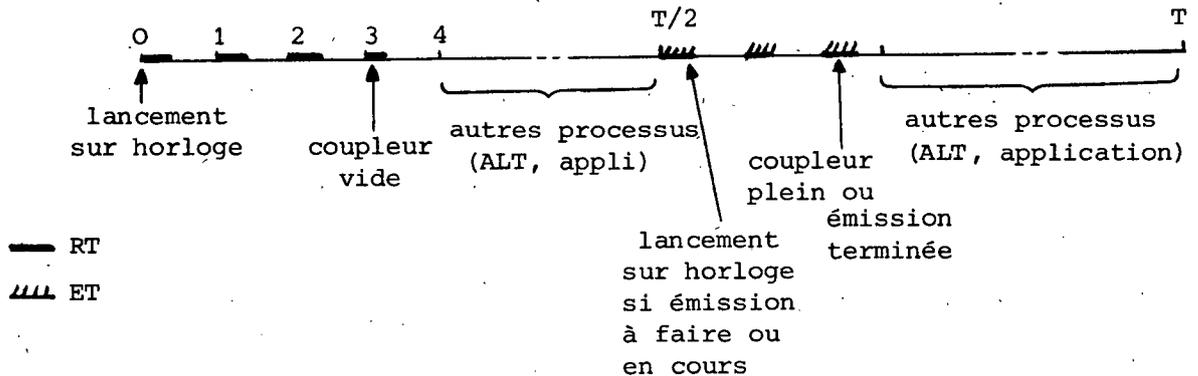


Figure III.13

Les processus gérés par le scheduler sont par priorité décroissante : RT, ET, ALT et les processus "application".

III.4.3. Partage de temps CVI

Il n'y a plus de contraintes liées au bon fonctionnement du coupleur : on a en principe tout son temps pour aller prélever le mot d'état du communicateur. Néanmoins le processus de plus bas niveau le Gérant RIM (GR) est toujours lancé sur horloge.

La période d'activation est un compromis entre deux types de contraintes :

- favoriser le parallélisme du réseau local par rapport au calculateur c'est-à-dire lancer les émissions RIM et permettre les réceptions RIM le plus rapidement possible.
- éviter les accès inutiles au mot d'état du communicateur (lorsqu'il n'y a rien à faire) compte tenu qu'un accès au mot d'état constitue une activation de processus, donc occupe un quantum de temps.

Cette période d'activation doit être ajustée en fonction des applications et des configurations. Des essais avec mesures de débit ont conduit avec

des applications - tests et une configuration réduite (2 ou 3 processeurs) à une période d'activation de GR de l'ordre de 3 à 8 ms.

En CVI s'introduit un niveau de protocole supplémentaire (cf structuration CVI et LIENS) ce qui ajoute les processus intermédiaires Emission Lien (EL) et Réception Lien (RL). Ces processus sont rendus activables soit par le Gérant RIM soit par le processus ALT.

Les processus gérés par le scheduler sont par priorité décroissante : GR, RL, EL, ALT et les processus "application".

IV - NIVEAU VOIE

IV.1. Rôle

Il remplit les fonctions du niveau paquet de X25 c'est-à-dire la gestion des circuits virtuels (CV et le multiplexage sur le niveau inférieur (TRAME dans le cas TRANSPAC, ETCD-LIENS dans le cas CVI) :

- vérification du respect du protocole X25 niveau paquet
- gestion des horloges de garde
- attribution des numéros de voie pour les appels sortants
- contrôle de flux paquet
- formatage des paquets et alimentation du niveau inférieur.

X25 TRANSPAC est décrit dans le manuel TRANSPAC :

'Spécifications Techniques d'Utilisation du Réseau' [3].

Le niveau voie offre à l'application un interface d'exploitation des CV basé sur des "requêtes" et des "événements". Il a été conçu de telle façon qu'il soit possible à une application de passer de CVI à TRANSPAC et inversement sans différence de comportement et sans changer d'interface (cf chapitres I à III) mis à part l'adressage dans les données d'appel (cf annexe 3).

IV.2. Structuration

La structuration a été décrite au chapitre III.

On peut noter :

Les traitements de niveau voie sont effectués sous le contrôle

- des processus de niveau inférieur en ce qui concerne les événements (réception, émission trame dans le cas TRANSPAC, émission, réception lien dans le cas CVI)

- des processus application en ce qui concerne les requêtes.

Il y a un seul processus spécifique au niveau voie alimentation trame (ALT).

Le rôle de ce processus est de formater les entêtes paquets à émettre en format TRANSPAC ou CVI suivant le cas. Ce formattage a été fait dans un processus particulier pour décharger les autres processus (de niveau inférieur ou application) et parce qu'il n'y a pas de contraintes de temps liées à cette tâche. Ce processus peut donc s'exécuter de façon non synchronisée par rapport aux processus périodiques. Il est toutefois prioritaire par rapport aux processus application.

Il peut être rendu activable par :

- les processus application suite à une requête entraînant une émission paquet
- les processus de niveau inférieur :
 - sur réception d'autorisation à émettre
 - ou sur progression de la fenêtre Trame (cf plus loin)
- l'horloge de garde.

IV.3. Interfaces

Les deux interfaces (avec l'application et le niveau inférieur) ont été définis en tenant compte :

- de la structuration en processus induite des contraintes et des choix décrits dans le chapitre III : ce qui implique en particulier la répercussion directe des événements par les processus de niveau inférieur (y compris les acquittements émission)
- du fait que c'est à l'application de fournir les buffers autant en émission qu'en réception et qu'on a voulu minimiser les recopies d'informations entre niveaux : les adresses buffer doivent donc être rendues accessibles à tous les niveaux.

IV.3.1. Interface niveau voie-application

La liste complète des requêtes et événements est donnée en annexe 1. On y trouvera :

- les noms des requêtes et événements
- les paramètres (à fournir par l'application pour une requête, fournis à l'application pour un événement)

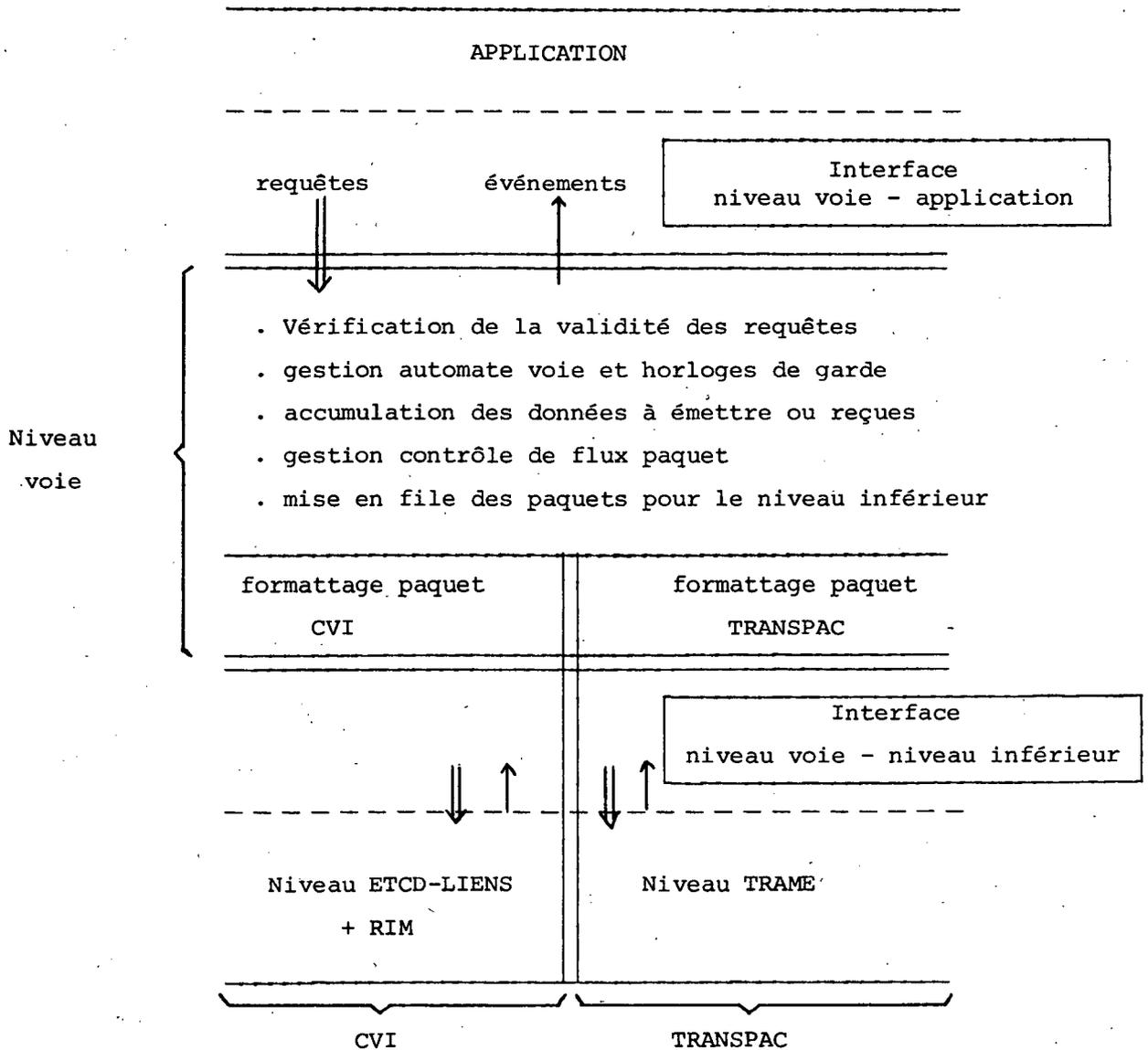


Figure IV.14.

Position des deux interfaces dans la structure.

Ce schéma ne fait pas apparaître la structuration en processus (cf chapitre III).

- des indications sur leur utilisation (requête) ou conséquences (événements).

On peut décomposer le fonctionnement paquet en 3 phases :

- la phase globale relative à l'initialisation/réinitialisation du niveau voie concerne toutes les voies logiques
- la phase signalisation, relative, à une voie, qui regroupe l'établissement, la libération, la réinitialisation d'un CV
- la phase données pendant laquelle les émissions/réceptions sont autorisées sur un CV.

On distingue alors 3 types principaux de requêtes-événements :

1) requêtes et événements globaux :

- événements connexion, déconnexion du niveau inférieur :
ETCNX, ETDISC, ETHS
- requêtes et événements relatifs à la reprise paquet :
REP, OKREP, EREPRIS, EFINREP, EVOIHS.

L'enchaînement des requêtes-événements est décrit par l'automate global.

2) les requêtes et événements relatifs à la phase signalisation :

- requêtes et événements d'établissement, libération, réinitialisation de CV : APPEL, OKAPPEL, EAPPEL, ECVIDON, LIB, OKLIB, ELIB1 à ELIB5, EPRET, REINI, OKREINI, EREINI.
- requêtes d'initialisation d'adresses buffers (INIBUFR) (cf note 1).

L'enchaînement de ces requêtes événements est décrit par l'automate de signalisation.

Une requête peut échouer si elle n'est pas autorisée compte tenu de l'état automate : aucune action n'est faite alors au niveau voie et un compte rendu d'échec est donné à l'application.

3) les requêtes et événements relatifs à la phase "données"

- requêtes demande d'émission (BUFEM), fourniture de buffer réception (BUFREC), acquisition des paramètres de paquet reçu (GETADR, CONSOM)
- événements acquittement d'émission de buffer (EFINBUF), réception de paquet (ERECDON).

Les demandes d'émission et les fournitures de buffers réception doivent respecter le contrôle de flux (fenêtre). Le non respect de la fenêtre se traduit par un échec de la requête et aucune action n'est faite au niveau voie. Il y a aussi échec d'une telle requête lorsque la voie n'est plus en phase "données". Notons que dans ce cas l'application a été prévenue par un événement "signalisation" ou global.

L'application ne peut faire de demandes d'émission qu'après l'événement "passage en phase données". (cf automate signalisation et annexe). Une demande d'émission se traduira par une émission réelle si la fenêtre est ouverte (c'est toujours le cas à l'établissement ou l'initialisation d'un CV) sinon elle sera enregistrée au niveau voie sous réserve que le nombre de demande émission enregistrées ne dépasse pas la valeur de la fenêtre paquet (W). Du fait de la non recopie de buffer aux niveaux inférieurs l'application devra attendre l'acquiescement d'émission d'un buffer pour le réutiliser (EFINBUF).

Une fourniture de buffer réception (BUFREC) se traduit par une progression de la fenêtre réception ; on ne peut pas anticiper de plus de W fournitures de buffers. Remarquons que les paramètres fournis dans les événements EFINBUF (fin d'émission) et ERECDON (réception) peuvent décharger l'application de la mémorisation des adresses buffer utilisées puisque ces adresses sont fournies explicitement.

Note 1

A l'initialisation d'un CV (après établissement ou réinitialisation) la fenêtre paquet est entièrement ouverte : en réception on peut donc recevoir W paquets d'où la nécessité de disposer de buffer et donc de pouvoir fournir leurs adresses au niveau voie avant l'initialisation du CV. Cette requête INIBUFR ne figure pas dans l'automate.

IV.2.2. Interface niveau voie - niveau inférieur

Ce paragraphe décrit les interactions entre le niveau voie et le niveau inférieur. Sur le principe il est identique pour CVI et TRANSPAC. L'interface est détaillé en annexe 2.

1) Emission

Le niveau voie dispose d'une file des voies pour lesquelles il y a un ou plusieurs paquets à émettre (file des voies en émission). Le processus "alimentation" prend cette file en entrée et "formate" les entêtes paquets dans une zone gérée par le niveau inférieur (zone entêtes paquets).

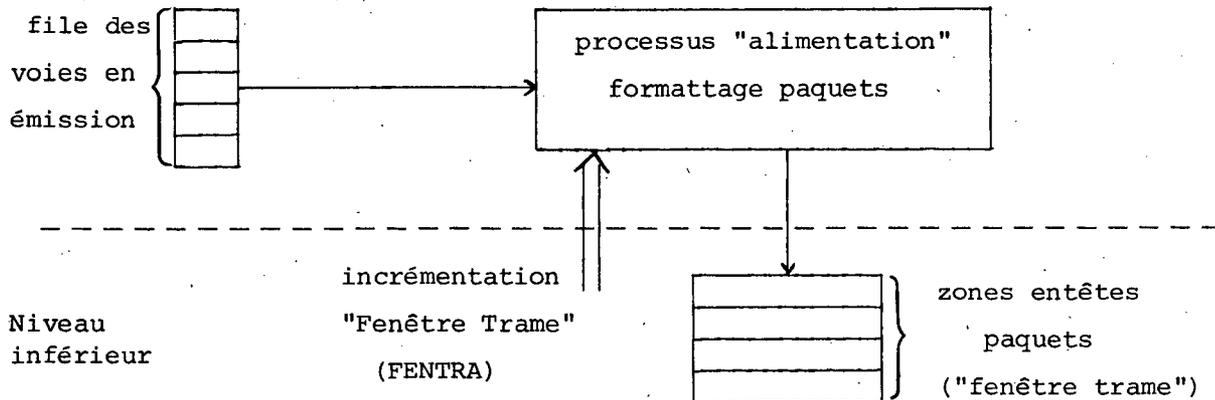


Figure IV.15.

Les zones entêtes constituent la fenêtre trame émission dans le cas TRANSPAC. Dans le cas CVI elles sont une zone intermédiaire qu'on appellera "pseudo fenêtre trame".

La structure d'une zone entête paquet est décrite dans l'annexe 2. Elle contient en particulier l'adresse du buffer émission. La taille de ces zones entêtes est prévue pour pouvoir contenir des paquets de signalisation en entier.

Les interactions entre le niveau inférieur et le processus alimentation se font par :

- l'appel par le niveau inférieur d'un sous programme (FENTRA) du niveau voie qui rend activable le processus alimentation si la file des voies en émission n'est pas vide. Cet appel se fait lorsque une (ou plusieurs) zone(s) entête est libérée(s) suite à la terminaison correcte d'émission (notée incrémentation "fenêtre trame" dans le schéma).

- l'appel par le processus alimentation d'un sous-programme (PTLIBR) du niveau inférieur qui fournit l'adresse de la zone entête à utiliser pour l'alimentation courante (cf annexe 2).

Tout paquet alimenté dans cette zone est irrattrapable par le niveau voie. Il est dit alors "engagé" au niveau inférieur.

L'acquittement d'émission paquet est répercuté par le niveau inférieur sur le niveau voie en précisant le paquet considéré. Cet acquittement peut faire évoluer l'automate voie et se traduire par un événement "signalisation" (voir automate) ou un acquittement d'émission de buffer (EFINBUF), ou niveau application.

2) Réception

Le niveau inférieur dispose d'une "zone entête" (structurée comme à l'émission) dans laquelle se fait la réception des entêtes paquets (les paquets de signalisation sont entièrement contenus dans la zone entête).

Comme indiqué plus haut la réception des données ne pouvant pas être contenues dans la zone entête se fait directement dans le buffer application : il n'y a pas de récopie entre les niveaux. Cela nécessite de connaître l'adresse buffer où ranger les données. Un sous-programme du niveau voie (RECPKC pour les paquets entièrement contenus dans l'entête, RECHBUF pour les paquets plus longs) analyse l'entête paquet et fournit éventuellement l'adresse buffer. Dans le cas où ce buffer n'existe pas (la voie n'étant plus en phase données par exemple) le sous-programme retourne un compte rendu d'échec et le niveau inférieur doit ignorer le reste du paquet.

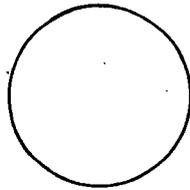
Le niveau inférieur doit également indiquer au niveau voie la fin (correcte : AKREC, incorrecte NAKREC) de la réception pour que celui-ci la répercute éventuellement au niveau application par l'événement ERECDON (réception de paquet).

3) Connexion - déconnexion du niveau inférieur

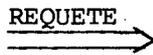
Le niveau voie n'est utilisable que si le niveau inférieur est connecté. Celui-ci répercute les événements connexion réussie, connexion ratée, déconnexion par l'appel de sous-programmes (VCNX, VDISC, VIHS). Ces événements agissent sur l'automate global et sont transmis à l'application (événements globaux) (ETCNX, ETDISC, ETHS).

IV.4: Les différents automates

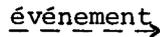
Symbolisme utilisé dans les automates :



état



requête application
(cf annexe)



événement de niveau inférieur

On distingue :

- reçu-pk : réception paquet
- ack-pk : acquittement émission
- ackrec : acquittement réception positif
- nackrec : acquittement réception négatif



événement répercuté à l'application
(cf annexe)



sonnerie d'horloge de garde

Passage d'un automate à l'autre :

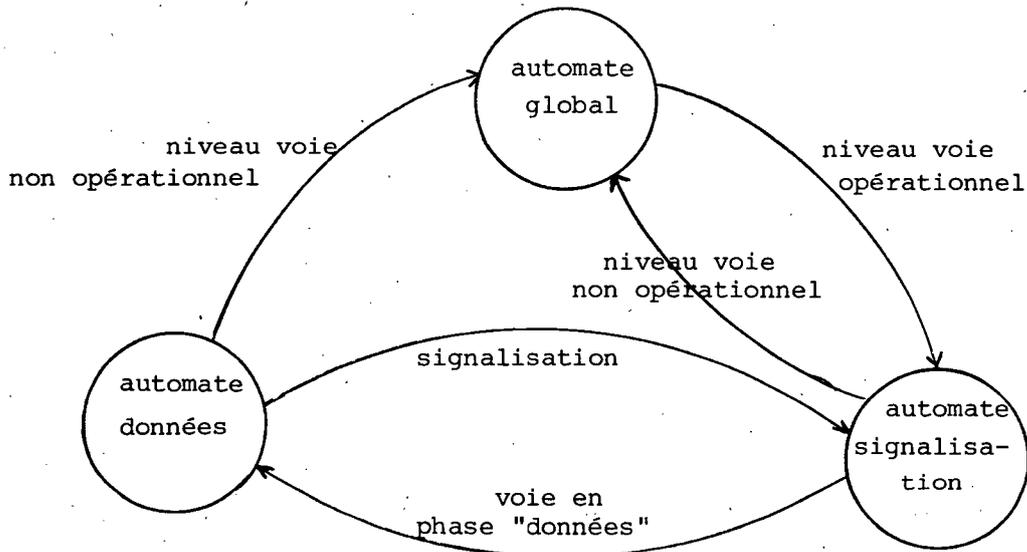


Figure IV.16

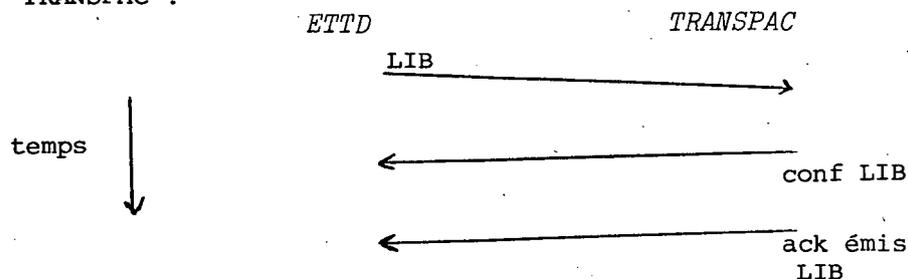
Chaque automate est relatif a une phase de même nom. Le passage de l'un à l'autre se fait sur :

- arrivée dans un état établi (niveau voie opérationnel pour l'automate global, "voie en phase données" pour l'automate signalisation)
- réception de paquet ou prise en compte de requête signalisation : fait passer de l'automate données à l'automate signalisation
- arrivée d'un paquet Reprise, déconnexion du niveau inférieur, prise en compte d'une requête Reprise : fait passer à l'automate global que l'on soit en phase données ou en phase signalisation (niveau voie non opérationnel).

Notons que la prise en compte des acquittements émission dans l'automate global et l'automate signalisation provient des considérations suivantes :

- les acquittements émission sont de toute façon nécessaires pour les paquets de données pour que l'application puisse réutiliser le buffer
- les prendre en compte permet :
 - . de mieux ajuster les horloges de garde : l'armement d'une horloge de garde est fait à l'acquittement émission donc l'overhead dû au franchissement des niveaux inférieurs (TRAME ou RIM) ne compte pas
 - . d'avoir au niveau de l'application la certitude qu'il n'y a plus de buffers engagés au niveau inférieur.

Les automates tiennent compte du fait que les acquittements émission et les réceptions faisant office de "réponse" à ces émissions peuvent s'intercaler dans le temps. Comme le montre l'exemple suivant sur TRANSPAC : l'incrémentement de la fenêtre réception trame côté TRANSPAC (ce qui est en fait l'acquittement d'émission) pour l'ETTD est indépendante des émissions de TRANSPAC :



Les actions prises aux changements d'état ne sont pas détaillées hormis les répercussions sur l'application (événements). Pour avoir plus de détails se référer aux spécifications de réalisation du niveau voie.

Figure IV.17. - AUTOMATE GLOBAL

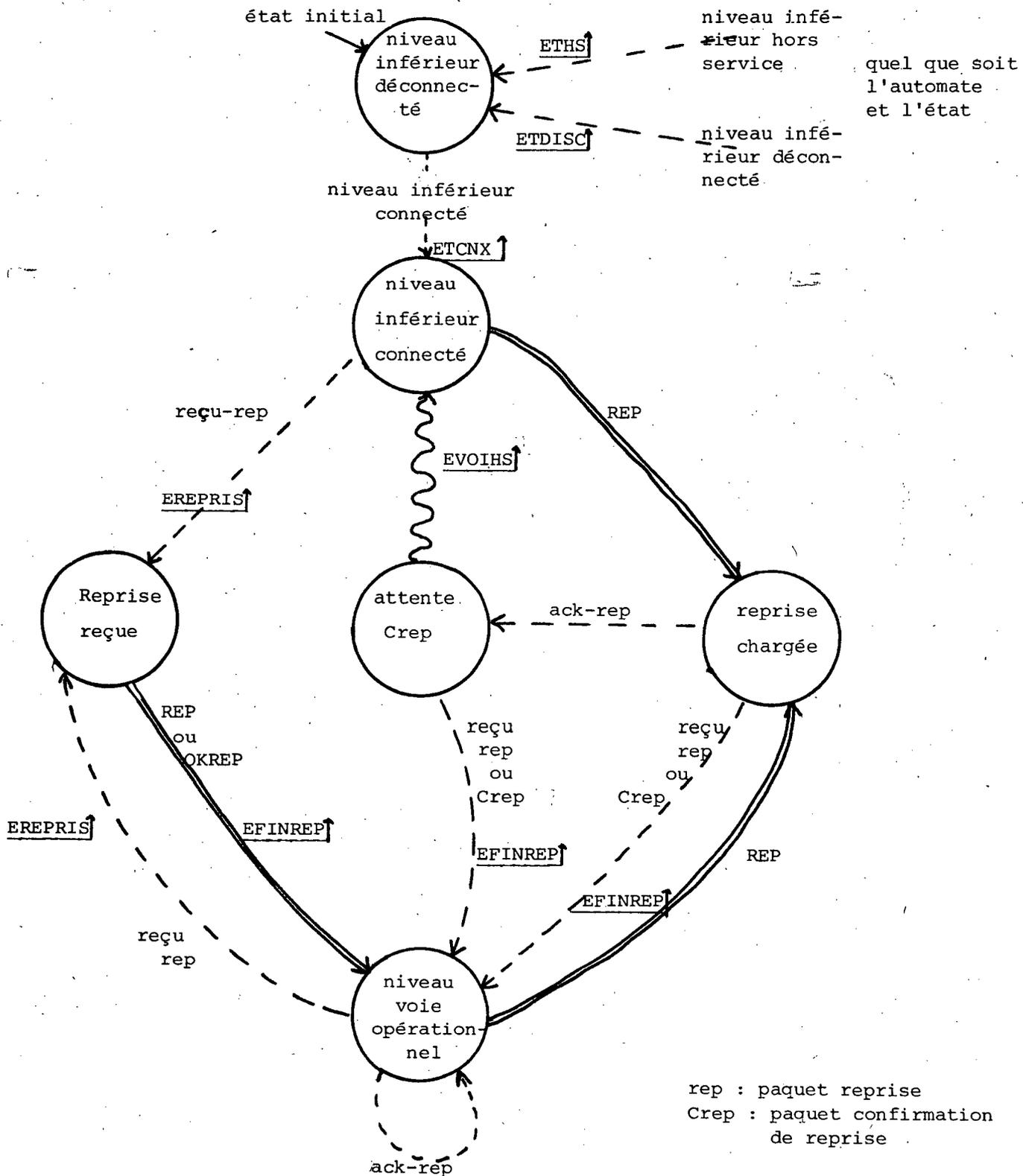
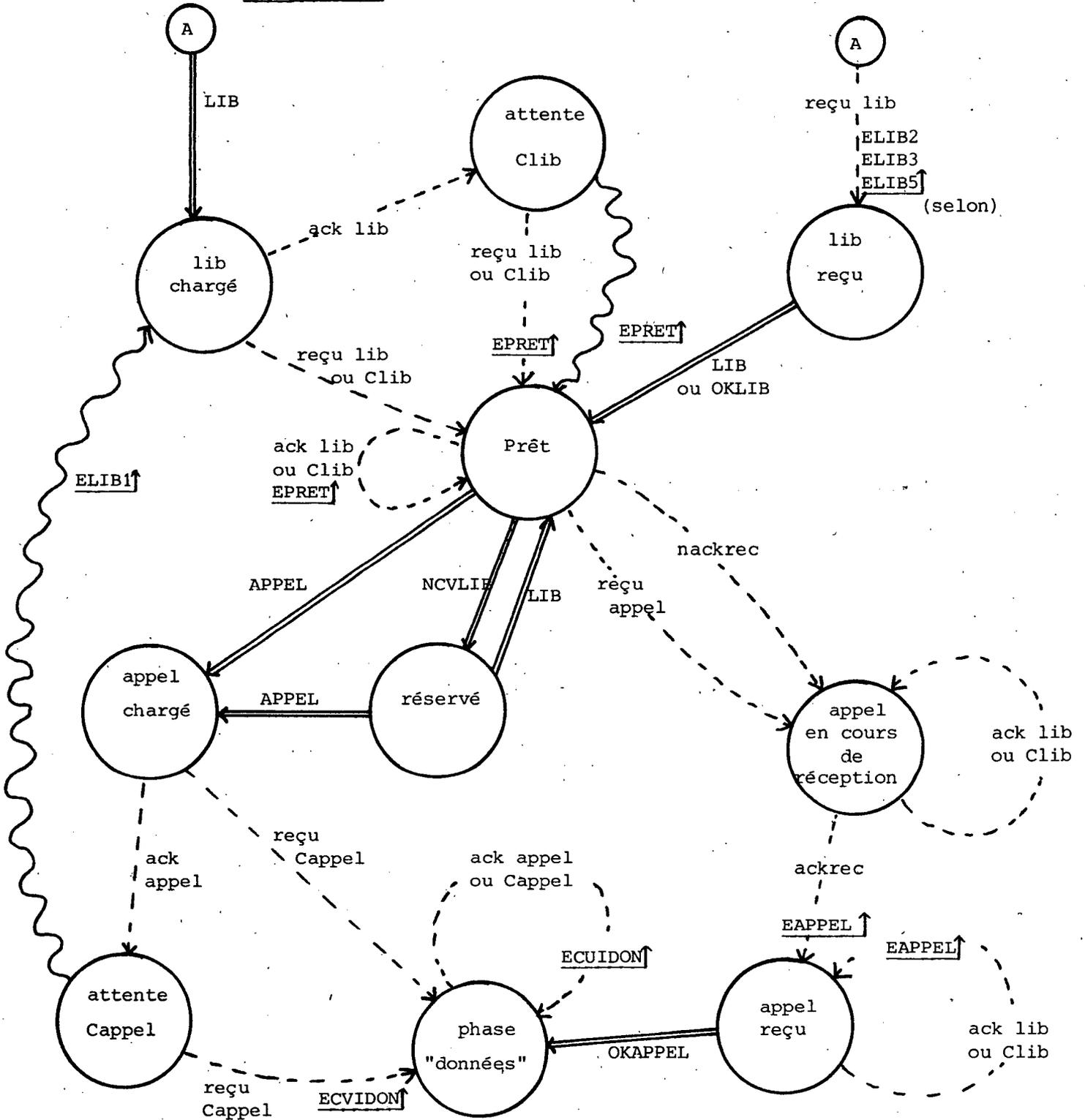


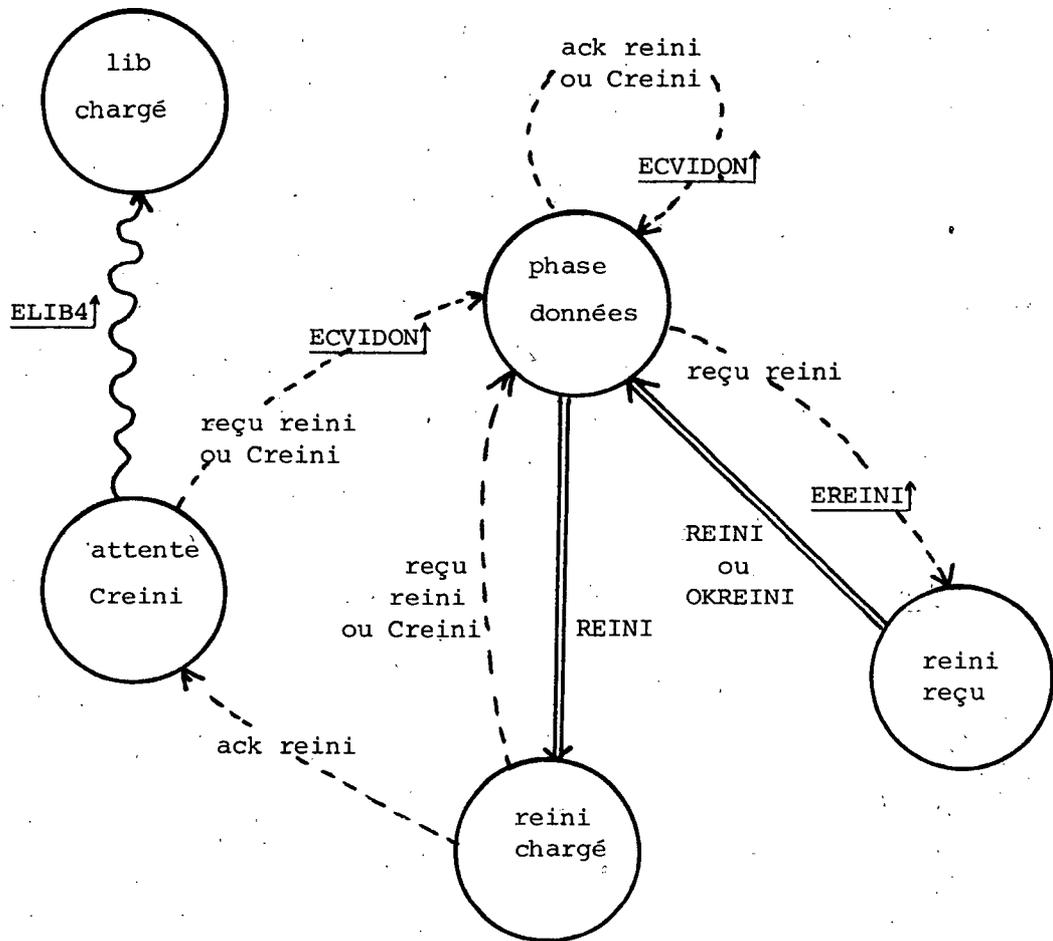
Figure IV.18. - AUTOMATE SIGNALISATION : ETABLISSEMENT ET LIBERATION.



A : tout état sauf : prêt, réservé, libchargé, libreçu
 appel en cours de réception, attente Clib.

L'état initial est "prêt". Lorsqu'on est dans l'automate global l'état est "bloqué".

Figure IV.19. - AUTOMATE SIGNALISATION : REINITIALISATION.



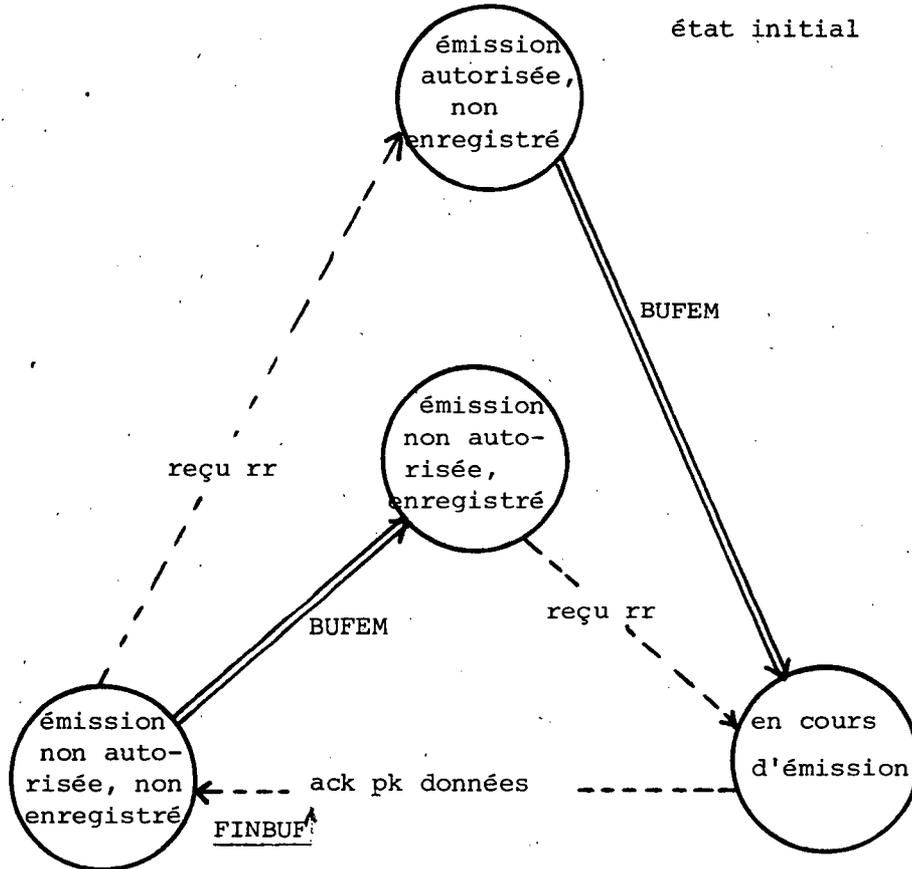
Paquets intervenant dans l'automate signalisation :

- appel : paquet d'appel
- cappel : paquet de confirmation d'appel
- lib : paquet de libération
- clib : paquet de confirmation de libération
- reini : paquet de réinitialisation
- creini : paquet de confirmation de réinitialisation.

Signalons que certains événements (EREPRIS, ELIB2, ELIB3, ELIB5) correspondant aux états reprise reçue, libération reçue nécessitent une confirmation de l'application. L'absence de cette confirmation n'est pas surveillée par le niveau voie. La voie reste donc gelée tant que l'application n'a pas fait cette confirmation.

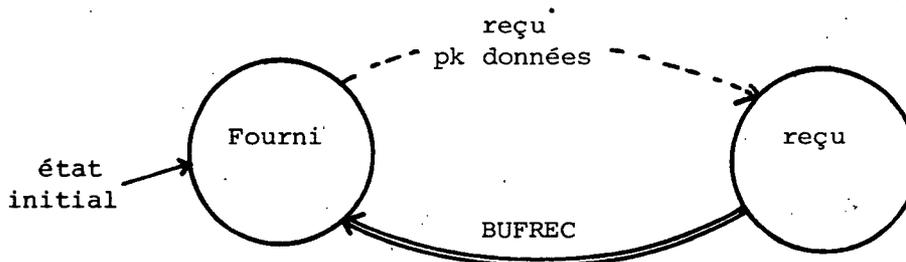
Figure IV.20. - AUTOMATE Phase données.

Emission



rr : paquet rr (receive ready) ou progression du P(R) dans un paquet de données.

Réception



Les automates concernant simplement un buffer.

Ils ne sont pas implémentés sous la forme explicite d'automates (voir les spécifications de réalisation du niveau voie).

IV.5. Caractéristiques de la réalisation effectuée

Vis à vis de l'application :

- mono application
- formattage des données d'appel à la charge de l'application.

Vis à vis de TRANSPAC :

- nombre de voies logiques : 255 (numérotées de 0 à 254)
- fenêtre paquet quelconque mais la même pour tous les CV
- interruptions non traitées
- uniquement les circuits virtuels commutés
- taille des paquets limitée à 128 octets de données.

V - LE DRIVER RIM

L'objet de ce chapitre n'est pas de décrire le fonctionnement de la RIM ; il a été vu de manière sommaire au chapitre II.2. Pour la conception et la réalisation, nous nous sommes appuyés sur les documents [1] et [5].

V.1. Rôle

- Offrir un jeu de primitives plus évolué que les instructions d'E/S pour échanger des messages. Ces primitives devront tenir compte des contraintes et choix exposés au chapitre III (durée d'un traitement inférieur à 1 ms, répercussion directe des événements, etc...).

- Gérer les buffers RIM pour le niveau supérieur (appelé niveau U) utilisant le driver RIM.

- Signaler au niveau U les événements (fin d'émission d'un message, réception d'un message) et les défaillances RIM au réseau.

- Profiter au maximum du parallélisme de fonctionnement entre la RIM et le calculateur hôte c'est-à-dire permettre à la RIM d'émettre (resp recevoir) pendant que le calculateur écrit le message suivant à émettre (resp lit le message arrivé).

Le driver RIM a été conçu uniquement pour gérer la transmission de messages en mode adressé d'une RIM à une autre.

V.2. Structuration

Le driver RIM comprend (cf fig. V.21.) :

- un processus prioritaire appelé gérant RIM (GR)
- un ensemble de procédures appelées par le module U correspondant aux primitives
- un ensemble de procédures correspondant aux sonneries des réveils armés par le driver RIM. Ces réveils sont gérés par le processus Horloge chargé des réveils de tous les niveaux.

Niveau U

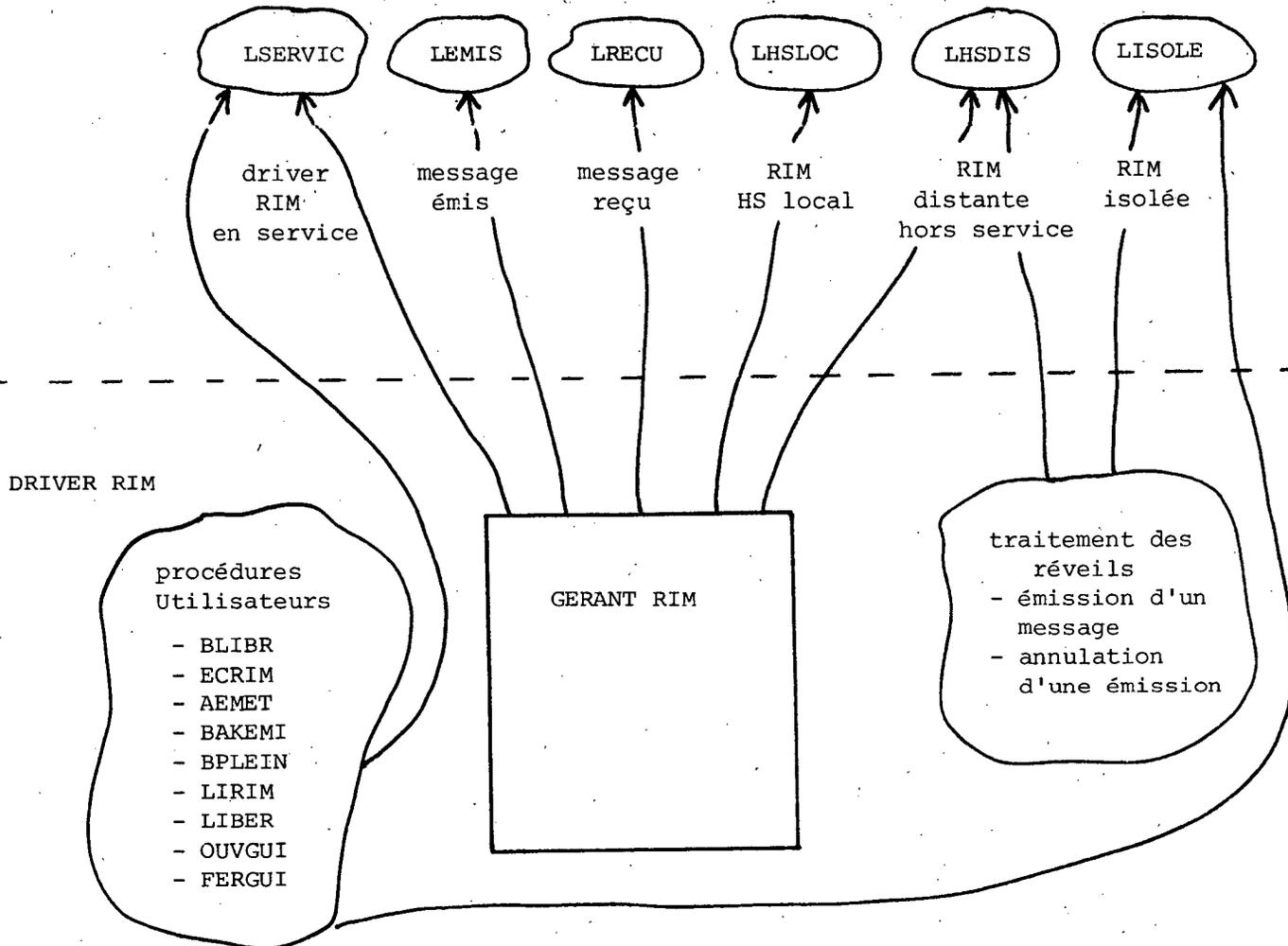


Figure V.21. - Structuration du driver RIM.

V.3. Primitives utilisateur

a) Pour émettre des messages, les primitives à employer sont :

- BLIBR pour acquérir un buffer libre en émission
- une fois le buffer accordé, ECRIM pour écrire tout ou partie* du message à écrire. En effet l'écriture d'un paquet long (128 octets) dure plus d'1 ms et nécessite sous MULTILINK d'être fractionnée
- Le message entièrement écrit, indiquer que le message doit être émis. Le niveau U sera prévenu de l'émission correcte du message ou non, respectivement par l'activation des procédures LEMIS et LHDIS.

b) Pour la réception des messages, le niveau U est prévenu par l'activation de sa procédure LRECU.

- le niveau U pourra lire tout ou partie du message arrivé par LIRIM
- par LIBER, U indiquera que le buffer peut être réutilisé pour une nouvelle réception.

c) Les autres primitives offertes sont :

- OUVGUI qui initialise le RIM et le DRIVER RIM
- FERGUI qui ferme le DRIVER RIM : le récepteur de la RIM sera alors fermé et aucune procédure événement ne sera activée
- BPLEIN (facultative) permet de retrouver le numéro du buffer contenant le message arrivé le moins récent et non libéré par LIBER (utile si le n° de buffer n'est pas mémorisé dans la procédure-événement LRECU)
- BAKEMI permet de retrouver le numéro de buffer contenant le message émis le moins récent et le rend disponible pour une autre émission.

* La lecture d'un message RIM en plusieurs étapes est importante; en effet, elle permet :

- de lire l'entête du message, de déterminer à la lecture de cet entête où ranger la suite (contrainte de non recopie des données)
- de pouvoir fractionner la lecture d'un message long (contrainte de temps).

V.4. Gestion des buffers

L'ensemble des buffers RIM est divisé en 2 sous-suites :

- 0 à NE pour l'émission
- NE+1 à 3 pour la réception.

NE étant paramétrable $1 \leq NE \leq 3$, le driver peut être configuré en favorisant un sens de transmission mais il est accordé au minimum un buffer pour chaque sens de transmission (NE = 2 par défaut).

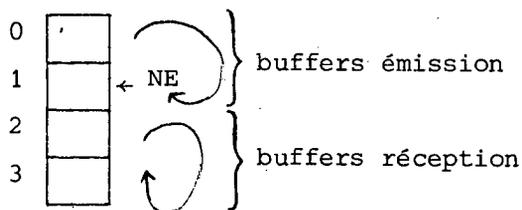


Figure V.21.

Chaque sous-suite est gérée de manière cyclique afin d'émettre les buffers pleins ou de délivrer les messages à U dans l'ordre.

La gestion des buffers est entièrement assurée par le driver RIM qui mémorise pour son propre compte et pour le niveau U l'état des buffers. Un buffer émission passe successivement par les états suivants :

- libre en émission
- rempli à émettre
- en cours d'émission (affecté au "transmitter")
- émis correctement (acquitté) ou non (acquiescement non reçu, refus de la RIM réceptrice de recevoir).

Un buffer réception peut être :

- libre en réception
- affecté à la réception (affecté au "receiver")
- plein : message reçu et à exploiter.

V.5. Le gérant RIM

Nous avons vu que la RIM ne générerait aucune interruption pour prévenir le calculateur. Seule l'erreur de parité, lorsque le calculateur envoie un octet

à la RIM, provoque un déroutement.

En conséquence, un processus du DRIVER RIM appelé GERANT RIM (GR) examine périodiquement le mot d'état de la RIM et avertit le niveau supérieur par l'appel de procédures-événements.

A chaque activation, le gérant RIM vérifie que :

- la RIM se déclare toujours valide (bits DA et IPE)
- il n'y a pas eu de défaut secteur (bit POR).

Ces erreurs sont fatales et entraînent la fermeture du driver RIM, le niveau U sera prévenu par LHSLOC.

V.5.a) Emission de messages

Lorsque l'émission d'un message est lancée, le gérant RIM examine si l'émetteur de la RIM redevient disponible, c'est-à-dire si l'émission est terminée. L'émission terminée, plusieurs cas peuvent se présenter :

α) l'émission s'est bien passée : on prévient l'utilisateur U par l'activation de LEMIS

β) l'émission a été avortée suite à une erreur de parité détectée par l'émetteur RIM. En cas d'erreur de parité persistante, GR essaie au maximum n fois ($n = 3$) au bout desquelles la RIM est déclarée hors service (procédure événement LHSLOC).

γ) le message est non acquitté : le message émis n'a pas été reçu par la RIM réceptrice ou l'acquiescement de réception positif (ou négatif) a été perdu. Pour éviter d'émettre un double, dans le cas de l'accusé de réception positif perdu, le gérant RIM avertit U par LHSDIS.

Dans les situations α et γ, si un buffer émission contient un message prêt à émettre, GR relancé aussitôt l'émission sur ce buffer.

V.5.b) Réception de messages

GR testera si un message vient d'arriver : c'est le cas où le récepteur ouvert sur un buffer vide vient de se fermer. Auquel cas, le niveau U sera prévenu par l'activation de LRECU et si un buffer est libre pour la réception,

le récepteur RIM sera immédiatement ouvert.

V.6. Horloges de garde

Deux horloges de garde sont nécessaires :

a) pour l'émission d'un message.

Dans le dialogue inter-RIM, la RIM voulant émettre un message attend l'accord de la RIM réceptrice. Il faut se protéger d'une attente indéfinie par une horloge. La durée d'attente est paramétrable et est fixée à 2s.

Lorsque l'horloge sonne, le driver RIM considère la RIM distante comme hors service et avertit le module U par la procédure-événement LHSDIS. Après il doit annuler cette émission en cours pour rendre disponible l'émetteur RIM.

b) pour annuler une émission.

Une émission ne peut être annulée que lorsque la RIM reçoit son droit de parole et si la RIM n'est pas isolée, l'émetteur RIM doit redevenir disponible dans un délai inférieur à 400 ms.

NB. Nous voyons donc que lorsqu'une RIM réceptrice distante refuse de recevoir, un test d'isolement est effectué par le driver RIM. Si la RIM est isolée, le niveau U sera prévenu par la procédure-événement LISOLE.

V.7. Test d'isolement

Il s'appuie sur la propriété de l'émetteur vu au paragraphe précédent. Ce test est fait systématiquement :

- à l'initialisation du driver RIM : on lance une émission que l'on annule aussitôt
- lorsqu'une RIM distante refuse de recevoir un message.

Le test d'isolement peut être aussi lancé à l'initiative du niveau U. Deux cas sont à considérer :

- une émission est en cours, si elle réussit cela prouve que la RIM n'est pas isolée, si la RIM distante refuse de recevoir le test sera obligatoirement lancé ; on ne fait donc rien dans ce cas.

- aucune émission n'est en cours, on lance une émission que l'on annule aussitôt.

Dans tous les cas, le niveau U n'est prévenu qu'en cas d'isolement par la procédure LISOLE.

V.8. Gestion efficace des échanges

Le driver RIM a été conçu pour utiliser les 4 buffers RIM :

- ainsi en réception, quand l'on détecte la fermeture du récepteur, le récepteur est réouvert aussitôt (si un autre buffer est libre pour la réception, cf gérant RIM) où dès que le niveau U libérera un buffer par la procédure LIBER. Favoriser la réception doit être aussi le souci des niveaux supérieurs au driver RIM. En effet si le trafic se concentre sur un RIM, le pilotage de la réception de cette RIM doit être rapide pour ne pas trop ralentir les émissions sur les autres RIM (une RIM ne traitant qu'une émission à la fois).

- en émission, on aura intérêt aussi à lancer l'ordre d'émettre à la RIM dès que le message est prêt. Ainsi, le processeur local pourra remplir un buffer pendant qu'un autre sera en cours d'émission.

VI - NIVEAU LIENS

VI.1. Rôle

Comme décrit au chapitre II, ce niveau se justifie par le fait que le réseau local doit se présenter fonctionnellement à l'ETTD (niveau voie) comme TRANSPAC.

Il est donc nécessaire de réaliser les fonction "réseau" :

- la gestion des voies logiques locales vues du réseau
- de routage et d'adressage dans le réseau
- de surveillance du réseau
- traitement des paquets reprise.

Ces fonctions sont réalisées dans un sous niveau appelé "accès réseau" (ou "ETCD" dans les listings). Certaines caractéristiques de TRANSPAC sont exclues à priori à cause des choix faits : en particulier il n'y a pas de mémorisation des paquets propre à ce niveau puisqu'on utilise directement les buffers de l'application.

Ce sous niveau s'appuie sur un protocole de communication entre calculateurs de réseau local : le LIEN. L'existence de ce protocole spécifique LIEN est liée essentiellement à la volonté de vérifier le bon fonctionnement du réseau local (transmission sans double ni perte).

Bâti au-dessus du driver RIM un lien est représenté par l'adresse "réseau local" (= adresse RIM) du calculateur distant.

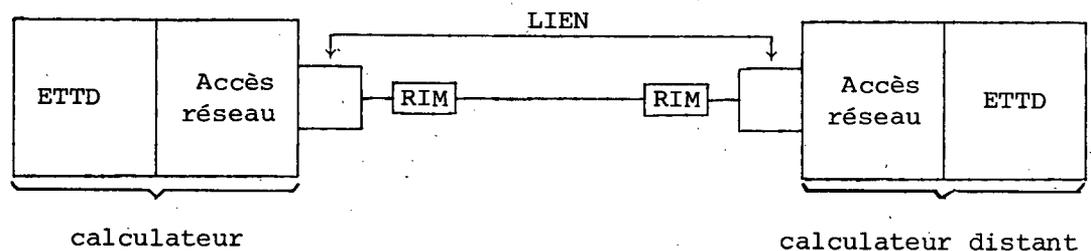


Figure IV.22.

VI.2. "Accès Réseau"

VI.2.1. Gestion des voies logiques et signalisation

Le problème au niveau "accès réseau" est de pouvoir aiguiller un APPEL entrant sur une voie logique locale libre : il faut donc avoir un état voie pouvant prendre 2 valeurs : libre / occupé.

Un tel état est déjà mémorisé par l'automate voie du niveau paquet (chapitre IV). L'état "prêt" est "libre" et tout autre état est "occupé". La simplification consistant à confondre l'état voie niveau "accès réseau" et l'état voie niveau paquet peut se faire compte tenu de la structuration en processus prise (chapitre III) et du fait que les traitements "accès réseau" et niveau paquet se font en séquence sans interruption possible entre les deux.

Le rôle de l'accès réseau en ce qui concerne les APPELS entrants est donc de rechercher une voie "libre" (i.e. "prête") en accédant à l'état voie du niveau paquet. En cas d'insuccès (aucune voie disponible) il génère une libération vers le distant.

Signalisation :

Sur TRANSPAC la signalisation s'adresse au réseau : c'est le réseau qui fait les confirmations. On pourrait penser qu'en CVI ces confirmations peuvent être faites par l'"accès réseau" sans attendre un signal venant de l'autre bout et donc d'éviter de transmettre ce signal sur le "réseau local".

L'exemple suivant montre que cela peut conduire à un mauvais fonctionnement : croisement de libérations.

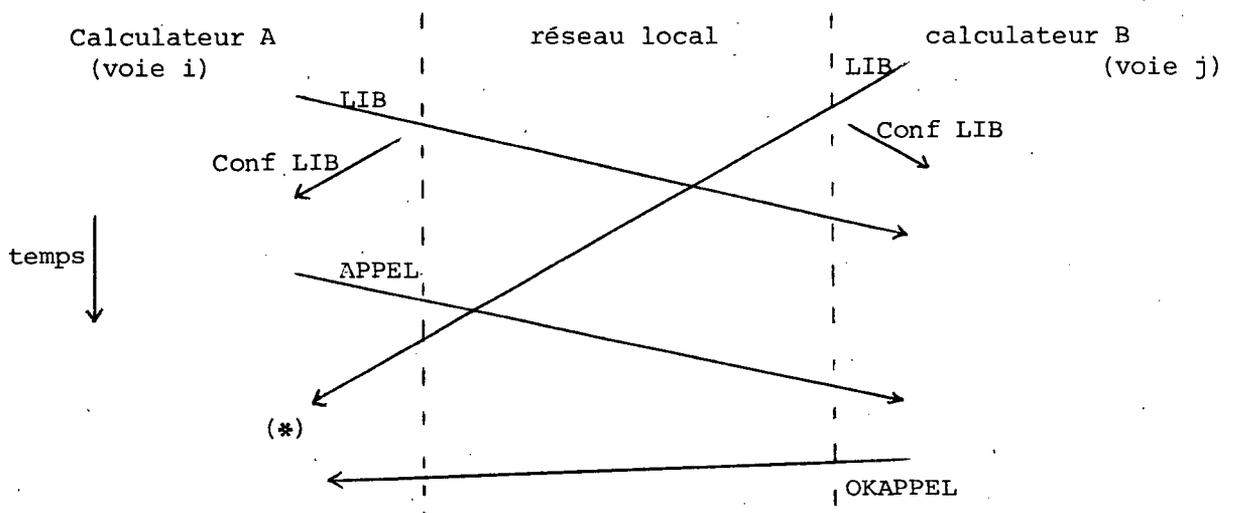


Figure VI.23.

(*) cette libération est prise en A pour une réponse à l'appel. La confirmation d'appel venant après est alors refusée. Le CVI est établi en B alors qu'il n'existe pas en A.

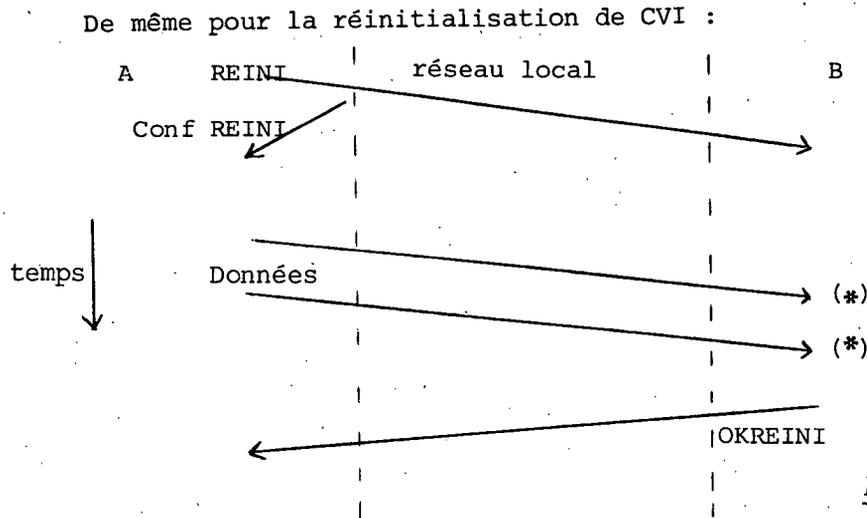


Figure VI.24.

(*) les données arrivant en B avant la confirmation de réinitialisation de l'application sont ignorées, l'"accès réseau" n'ayant pas de buffer pour mémoriser les paquets.

Ces deux exemples montrent la nécessité de transmettre de bout en bout les signaux de confirmation et d'attendre ces signaux avant de répercuter la confirmation à l'application.

VI.2.2. Routage, adressage

L'"accès réseau" doit d'autre part gérer des informations d'adressage relatives à l'autre bout du CVI ; en particulier le n° de lien (adresse "réseau local" du distant). Le souci :

- 1) d'avoir l'accès le plus direct possible pour trouver la voie logique locale en réception
 - 2) de permettre des vérifications de correspondance entre n° voie locale et n° voie distante / n° lien (ces vérifications sont utiles dans la phase de mise au point par exemple)
- a conduit à transmettre les deux numéros de voie logique (locale et distante) dans tous les paquets. Ne transmettre que le n° de voie de l'émetteur oblige

le récepteur à une recherche dans une table de correspondance ce qui est coûteux ; Ne transmettre que le n° de voie du récepteur n'est pas toujours possible (APPEL par exemple).

Transmettre les deux numéros de voie conduit l'accès réseau à gérer une table de correspondance : n° voie locale → n° voie distante, lien donc à filtrer tous les paquets qui peuvent changer cette table : les paquets d'établissement et de libération de CVI : APPEL sortant et entrant (réserve un n° voie locale) confirmation d'APPEL (récupérer le n° de voie distante), libération et confirmation de libération (annuler une entrée de table). On aurait donc pu transmettre les deux numéros de voie uniquement dans les paquets de signalisation. Les transmettre dans tous les paquets permet une vérification systématique et un format unique d'entête.

L'"accès réseau" prend en compte tous les cas particuliers où la correspondance voie locale ↔ voie distante n'est pas encore établie (il y a alors une valeur NIL (255) dans l'entête pour le n° de voie qui n'est pas connu). Ce sont par exemple les cas :

- LIB arrivant après un APPEL entrant et avant que la confirmation d'appel ait transité : le n° de voie locale n'est pas connu du distant ; il faut donc rechercher dans la table à partir du n° de voie distant et du n° de lien.

- Conf LIB arrivant suite à un LIB de l'accès réseau (cf IV.2.1.). Il faut le reconnaître et ne pas le répercuter au-dessus.

En émission l'accès réseau vérifie si le lien est ACTIF (cf VI.3.). Si ce n'est pas le cas il n'émet pas le paquet et répercute vers l'application une libération pour certains paquets (APPEL par exemple).

VI.2.3. Surveillance

La surveillance est ici à 3 niveaux :

- surveillance du réseau local par le Driver RIM. Celui-ci détecte les cas d'isolement et de mauvais fonctionnement (parités) du communicateur et les répercute au niveau LIENS.

- surveillance des LIENS : une impossibilité du Driver RIM à émettre un paquet indique que le communicateur distant n'est pas en mesure de recevoir.

Cela se traduit au niveau lien par 'lien cassé' et purge de tous les CVI bâtis sur ce lien. D'autre part une numérotation est faite à l'émission pour chaque paquet de niveau LIEN et vérifiée à la réception. Cela permet de détecter les doubles ou pertes. Actuellement il n'est pas prévu de corriger ces erreurs, elles sont considérées comme irréparables.

- surveillance par l'Application : les deux niveaux précédents de surveillance ne permettent pas de détecter tous les cas de "disparition" du distant : par exemple lorsque le calculateur local n'émet pas. C'est pourquoi il a été prévu un mécanisme permettant à l'application de déclencher des émissions de paquets sur demande (en l'occurrence des paquets RR). Cette émission permet de détecter si le communicateur distant est en service ou non. Le choix d'avoir mis cette surveillance à la charge de l'application plutôt qu'au niveau lien est basé sur :

1) le souci d'éviter une surcharge du réseau local par un trafic minimal sur chaque lien et d'alourdir le protocole lien par des échanges de surveillance

2) le fait que les besoins de surveillance sont plutôt spécifiques à l'application.

VI.2.4. Reprise

La Reprise paquet venant de l'application nécessite un traitement particulier : la Reprise provoque la réinitialisation de tous les liens, ce qui a pour effet de libérer tous les CV établis. L'accès réseau simule la réception d'une confirmation de Reprise (la répercute vers l'application) lorsque tous les liens sont dans un des états suivants : ACTIF, PAUSE ou OFF (cf automate lien).

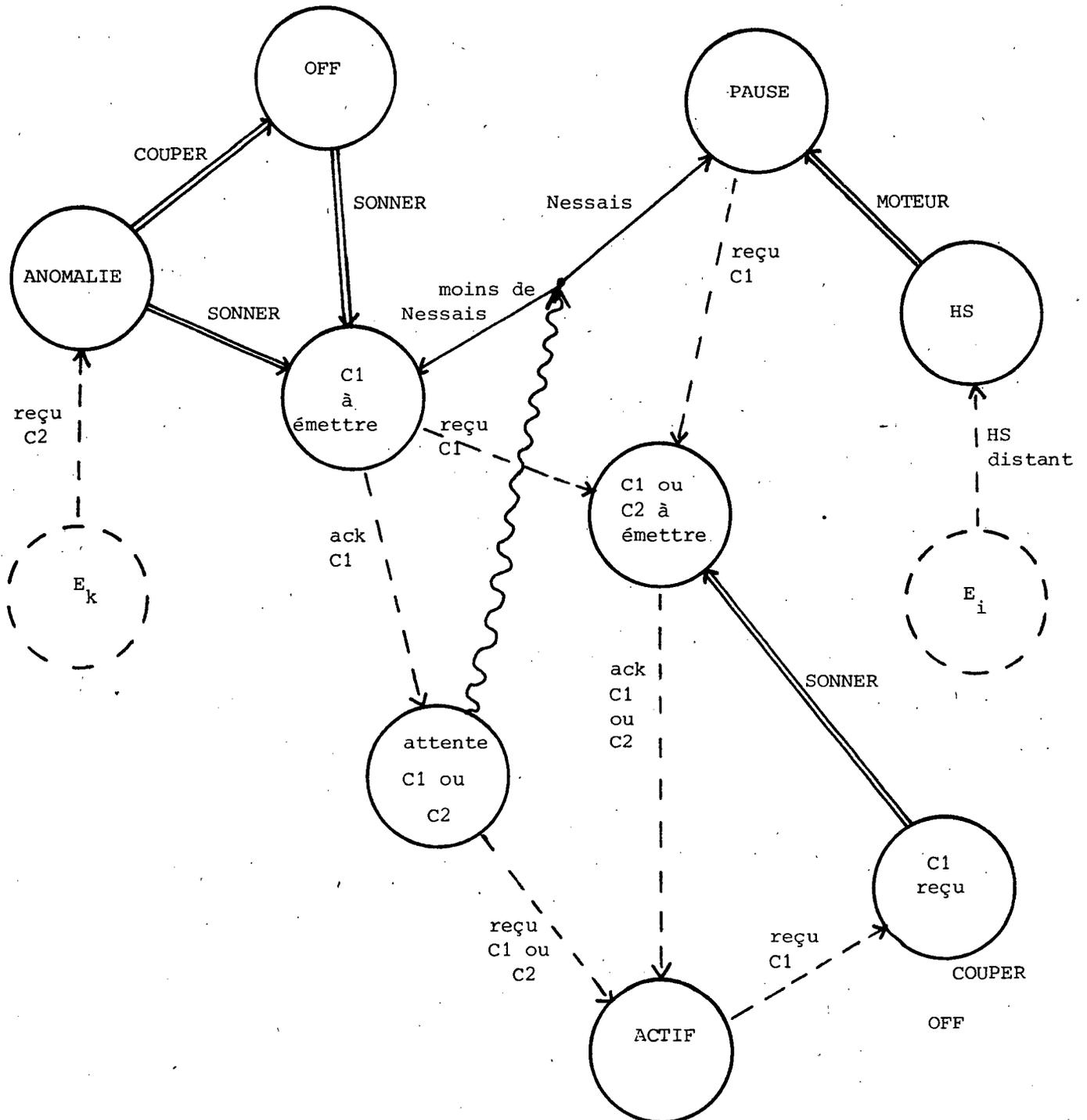
VI.3. Protocole LIEN

Ce protocole permet d'initialiser et de réinitialiser un LIEN pour :

- synchroniser les deux extrémités du LIEN en ce qui concerne le début de la numérotation
- purger les CVI établis sur ce lien.

Il y a deux types de paquets de signalisation-lien clear1 (C1) et clear2 (C2), ce dernier étant la confirmation à C1.

Figure IV.25. - Automate signalisation lien



E_i : tout état où une émission est en cours

E_k : tout état sauf : ANOMALIE, HS, attente C1 ou C2, OFF

Trois requêtes :

MOTEUR : permet après un HS distant ("lien cassé") de se remettre en attente de connexion du distant (état PAUSE)

SONNER : lancer la phase d'initialisation du lien

COUPER : se mettre en état 'OFF' : rien n'est alors possible sur le lien.

Le symbolisme utilisé dans l'automate de signalisation lien est le même que celui des automates voie.

L'état initial est 'OFF', la transition SONNER est faite lors de l'initialisation du niveau voie (émission d'une REPRISE paquet par l'Application). En CVI l'application doit obligatoirement "émettre" une REPRISE. On revient dans l'état "OFF" suite à un comportement anormal du distant : répétition de réception de C1 ou C2. Un compteur incrémenté à chaque SONNER permet de tester ce cas de figure.

L'état PAUSE a la signification suivante :

- après un HS distant et un nettoyage au-dessus (purge des CVI) pendant lequel aucune signalisation n'est acceptée, on se met à l'écoute des C1 distants mais on ne prend pas l'initiative de réinitialiser le lien (sauf REPRISE venant de l'Application).

- on vient également dans cet état après N tentatives d'initialisation non réussies : cas où le driver RIM est présent (puisque'il n'y a pas HS distant) mais pas le niveau lien distant (puisque'il n'y a pas de réponse).

Initialisation d'un LIEN : (quand faire l'initialisation ?)

Les choix envisagés étaient :

- 1) à l'initialisation du système (REPRISE paquet)
- 2) lors du premier appel sur le lien.

La première solution a été choisie pour les raisons suivantes :

- elle permet de définir avant le chargement du système les liens qui pourront être ouverts et donc de limiter à un sous-ensemble du réseau des calculateurs avec lesquels on autorise une communication.

- elle permet d'avoir la liste des calculateurs distants sous CVI même s'il n'y a pas eu de CVI d'établi.

- un calculateur qui se reconnecte purge par cette réinitialisation tous les CVI qui étaient restés établis lors de sa déconnexion (il peut y en avoir : cf surveillance) même s'il n'émet pas d'appel.

Les inconvénients de cette solution tiennent à son aspect figé :

- la liste des calculateurs pouvant être atteints est définie une fois pour toutes et ne peut donc pas être mise-à-jour dynamiquement
- le processus d'initialisation lien prend du temps au lancement du système surtout si peu de calculateurs sont présents.

Des solutions plus efficaces et moins figées étaient peut-être possibles en utilisant la diffusion RIM mais ce point n'a pas été étudié.

VI.4. Structuration et Interfaces

L'interface haut (avec le niveau voie) a été décrit au chapitre V. Le niveau lien se présente de la même façon que le niveau TRAME vis à vis du niveau voie : pseudo-fenêtre trame en émission, et répercussion directe des événements. On retrouve donc la même structuration (cf chapitre III) : un processus Emission LIEN (EL) et un processus Réception Lien (RL). Comme pour le cas TRAME les acquittements émission sont traités par le processus Réception. Pour écrire et lire dans le communicateur les processus disposent d'un ensemble de primitives du Driver RIM permettant d'opérer en plusieurs étapes (cf Driver RIM).

Processus Emission Lien (EL) :

Il est rendu activable lorsqu'il y a un buffer RIM disponible et au moins un paquet à émettre. Il prend les paquets suivant la priorité décroissante suivante :

- paquet de signalisation lien
- paquet LIB généré par l'ETCD sur appel entrant lorsqu'il n'y a plus de voie logique libre
- paquet CVI (entête formaté dans la pseudo fenêtre trame).

L'entête des paquets CVI est complété par les informations relatives à l'autre bout du CVI (n° voie logique distante) (cf annexe). On vérifie que le lien est utilisable (état ACTIF), on numérote le paquet puis on recopie dans le buffer RIM, d'abord l'entête puis les données directement à partir du buffer application. La recopie peut se faire en plusieurs étapes si le paquet est long (traitement dépassant le quantum de temps imparti). Le buffer application peut être libéré dès que la copie dans la RIM est faite.

Processus Réception Lien (RL)

Il est rendu activable par le gérant Rim (GR) lorsqu'il y a un buffer plein dans la RIM ou une fin d'émission RIM.

Il comporte en séquence les traitements des niveaux LIEN voie et procédure événement s'il y a lieu. Il peut rendre activable EL (paquet de signalisation LIEN à émettre, buffer RIM disponible).

Le traitement lien en réception est le traitement automate lien pour les paquets de signalisation lien et la vérification de l'absence de doubles et de perte pour les paquets CVI. L'entête est lu dans une zone spéciale puis après analyse de l'entête pour déterminer le buffer le reste du paquet est lu directement dans le buffer application. La lecture des données peut se faire en plusieurs étapes si le paquet est long.

VI.5. Format des paquets lien

Paquets de signalisation-lien :

octet 0	C1 ou C2
------------	----------

Paquets de données-lien :

octet 0	octet 1	octet 2	octet 3	octet 4	octet 5	octet 6	octet 7
0	NUMERO de PK-LIEN	VLE	VLR	TYPE de PK		

VLE : Voie logique émetteur

VLR : Voie logique récepteur.

VII - LA PASSERELLE

La passerelle doit permettre à tout processeur du réseau local d'entretenir des conversations via TRANSPAC avec des terminaux ou calculateurs externes au réseau local. La passerelle, supportée par un processeur local, sera chargée de gérer toutes ces conversations externes.

VII.1. Intérêt de la passerelle

L'intérêt de concentrer l'accès TRANSPAC sur un seul processeur est :

- d'avoir une seule connexion physique et la présence du logiciel Transpac en un seul exemplaire
- un seul abonnement suffit pour l'ensemble du réseau local considéré comme un seul ETTD
- le processeur supportant la passerelle pourra être adapté à cette fonction.

VII.2. Rôle de la passerelle

Une conversation externe va donc être la concaténation d'un circuit virtuel Transpac (CV) et un circuit virtuel interne (CVI). Les CV et CVI obéissent au même protocole X25 niveau 3, la concaténation des deux aboutit à un circuit virtuel ayant toujours les mêmes fonctions. Assurer la concaténation de deux circuits revient donc à retransmettre les paquets du CV sur le CVI et vice-versa du CVI sur le CV.

Pour assurer cette retransmission des CV sur les CVI, la passerelle doit :

- a) tout d'abord gérer la liaison passerelle-TRANSPAC pour échanger des paquets avec TRANSPAC ; fonction réalisée par le niveau TRAME et le driver HDLC.
- b) échanger des paquets avec les processeurs locaux en respectant le protocole interne de transport : fonction réalisée par le niveau LIEN et le driver RIM

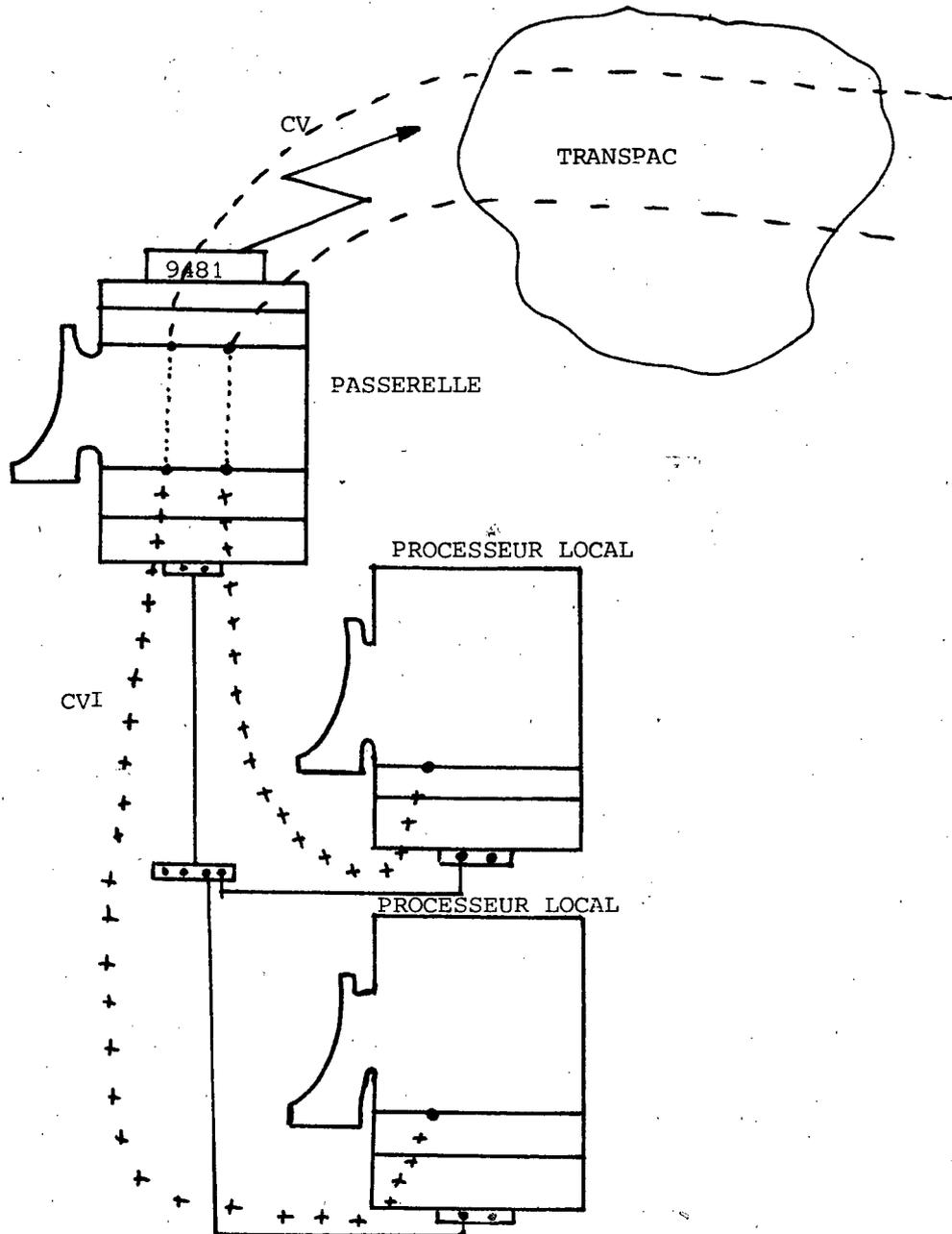


Figure VI.26.

c) aller chercher les paquets délivrés au niveau TRAME pour les répercuter sur le niveau LIENS et inversement.

La retransmission des paquets à ce niveau ne sera pas aveugle, il faudra :

d) changer le codage des paquets : le format des paquets CVI diffère des paquets TRANSPAC

e) aiguiller chaque paquet d'appel entrant TRANSPAC vers le processeur local supportant l'application appelée.

Sur TRANSPAC, une application est identifiée par son adresse externe : <numéro ETTD>, <numéro complémentaire> et sur le réseau local par son adresse interne : <n° RIM>, <n° application propre au processeur local>.

Grâce à une table de correspondance (adresse externe, adresse interne) des applications externes, la passerelle pourra retransmettre chaque paquet d'appel entrant TRANSPAC vers la RIM indiquée dans l'adresse interne.

f) Si l'application appelée n'existe pas, refuser l'appel par l'envoi à TRANSPAC d'une libération qui sera confirmée ; laquelle confirmation ne devra pas être retransmise.

g) à l'établissement des circuits virtuels, choisir un numéro de voie logique respectant la convention TRANSPAC valable aussi pour CVI : appel entrant sur le plus petit numéro disponible, appel sortant sur le plus grand disponible. Ce qui nous conduit à associer un numéro de voie CVI à un numéro de voie TRANSPAC à l'appel et à mémoriser cette association pour convertir les numéros dans la retransmission des paquets.

h) gérer les files des paquets en attente d'être émis sur TRANSPAC (pas encore acceptés par le niveau TRAME) et sur les processeurs locaux (niveau LIENS).

VII.3. Réalisation de la passerelle

En utilisant au maximum les logiciels développés par ailleurs, deux solutions sont possibles pour réaliser la passerelle :

VII.3.a) le rôle de la passerelle est réduit à retransmettre les paquets en filtrant les paquets d'appels, rôle décrit dans le paragraphe VII.2.

Auquel cas, la passerelle comprendrait : (cf figure VII.27)

- le driver HDLC et le niveau TRAME (fonction a de VII.2.)
- le driver RIM et le niveau LIENS (fonction b de VII.2.)
- un niveau RELAI émettant sur le niveau LIENS tous les paquets reçus de TRANSPAC (niveau TRAME) et vice-versa en émettant sur le niveau TRAME tous les paquets des processeurs locaux tout en assurant les fonctions d, e, f, g, h.

Notons en passant les fonctions réalisées par le niveau LIENS :

- α) routage des paquets (mémorisation pour chaque CVI du processeur local distant)
- β) traitement des paquets de reprise reçus de TRANSPAC
- γ) refus des appels provenant du réseau local si la passerelle est saturée.

Ces fonctions α, β, γ auraient dû être rajoutées si le niveau LIENS n'était pas présent.

L'inconvénient dans cette solution est que la passerelle constitue un intermédiaire pouvant ralentir le débit tout particulièrement dans le cas d'une fenêtre de longueur $W = 1$. En effet, dans le sens TRANSPAC réseau local, TRANSPAC avant d'émettre un paquet doit attendre le prêt à recevoir transitant du processeur local à la passerelle, de la passerelle à TRANSPAC ; le phénomène est identique dans le sens processeur local TRANSPAC. L'autre inconvénient est que pour un prêt à recevoir envoyé à TRANSPAC, deux tampons sont gelés pour recevoir le paquet : un sur la passerelle et un sur le processeur local.

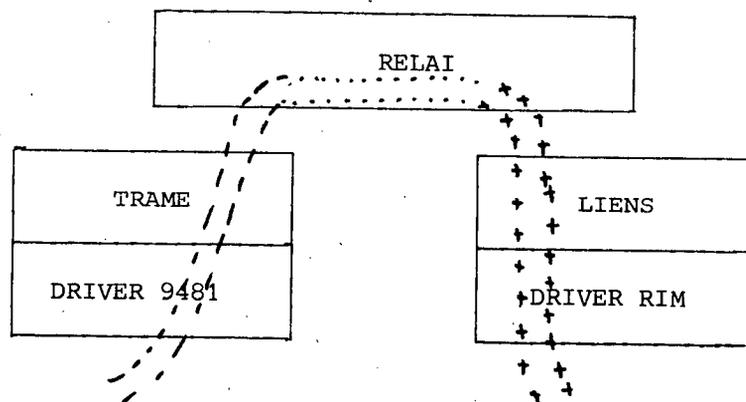


Figure VII.27.

VII.3.b) La passerelle anticipe la réception des paquets de données. Elle joue le rôle de mémoire tampon vis à vis de TRANSPAC et vis à vis des processeurs locaux.

Toutes les fonctions de la solution précédente sont assurées mais en plus au lieu d'avoir un seul contrôle de flux entre TRANSPAC et un processeur local pour un CV donné, nous avons deux contrôles de flux : un entre TRANSPAC et la passerelle, un autre entre la passerelle et TRANSPAC.

Cette solution peut être facilement réalisée en assurant la JONCTION non plus au niveau TRAME-LIENS mais au niveau voie (fig. VII.28).

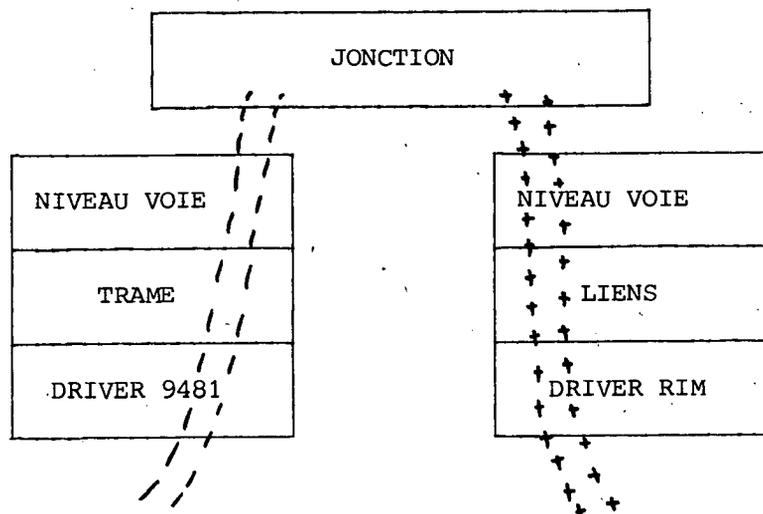


Figure VII.28.

Les deux niveaux voies assurent donc :

- le rôle de mémoire tampon ; chaque niveau voie gère un contrôle de flux vis à vis de TRANSPAC ou d'un processeur local
- la gestion des files des paquets en attente d'être émis sur TRANSPAC (en attente d'être transmis au niveau TRAME) et sur les processeurs locaux (en attente d'être transmis au niveau LIENS) par les processus 'ALT' ; l'émission est faite de manière équitable entre toutes les voies.
- le codage et le décodage des paquets (fonction d) de VII.2.). La réalisation du module JONCTION se trouve donc facilité d'autant plus que :
 - les deux niveaux voie présentent un interface unique

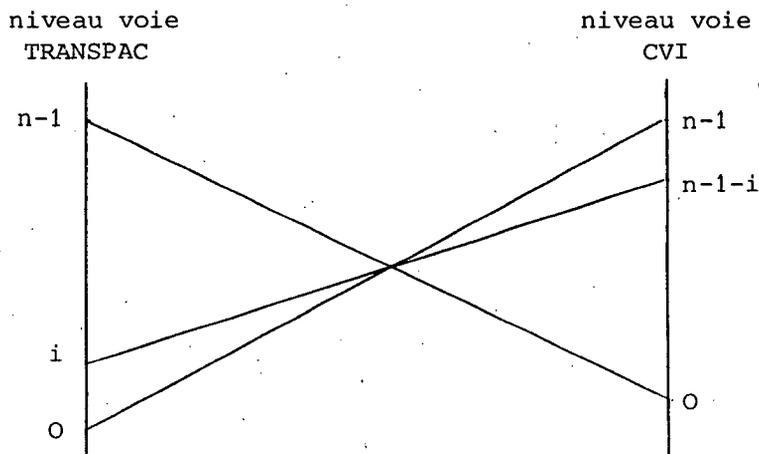
- la discrimination entre paquets est faite grâce aux procédures-événements de niveau utilisateur
- seules restent les fonctions e, f, g restent à faire.

C'est donc cette solution en définitive qui a été choisie pour réaliser la passerelle.

VII.4. Choix effectués dans la solution retenue

VII.4.a) Association des numéros voie CVI et TRANSPAC

Si n est le nombre maximum de circuits virtuels susceptibles d'être ouverts avec TRANSPAC, le n° de voie CVI i sera associé de manière statique* au n° de voie TRANSPAC $n-1-i$.



Ainsi chaque appel entrant CVI utilisant le plus petit numéro libre i sur CVI est bien répercuté sur le plus grand numéro de voie $n-i+1$ TRANSPAC : le module JONCTION évite donc les collisions d'appels avec TRANSPAC.

* L'association statique nous économise une recherche de voies libres à l'établissement du circuit virtuel. En contrepartie, cela implique que la voie associée soit libre pour transmettre l'appel. Dans le cas contraire, voie associée non disponible, il faut attendre qu'elle le devienne : ce qui est généralement très rapide.

VII.4.b) Non recopie des données

. Nous avons voulu éviter toute recopie de données au niveau de la passerelle. C'est-à-dire dans le sens réseau local → TRANSPAC, lire les données du buffer RIM dans une zone mémoire et puis à partir de cette même zone mémoire écrire ces données dans le buffer du coupleur HDLC.

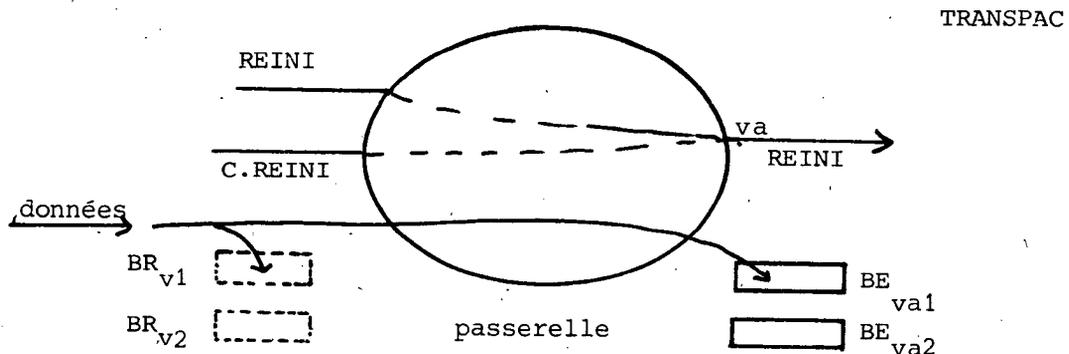
Pour réaliser cette non recopie des données dans les deux sens (TRANSPAC → réseau local et réseau local → TRANSPAC), nous avons confondu les buffers de réception de la voie v (resp va) avec les buffers émission de la voie associée va (resp v). Ainsi pour un couple de voies (v, va), w étant la taille de la fenêtre paquet, 2 w buffers suffiront et pour le module JONCTION n * 2 x w buffers.

Exemple : une passerelle pouvant supporter 32 circuits virtuels avec une taille de paquets de 128 octets et w = 2, 16 k octets suffiraient.

Toutefois, cette confusion des buffers va se répercuter sur le traitement de la signalisation.

VII.4.c) Traitement de la signalisation

Lorsque le module JONCTION reçoit un paquet de libération ou de réinitialisation, il pourrait anticiper la confirmation. Par exemple, pour une réinitialisation reçue d'un processeur local sur la voie v, le module JONCTION pourrait répercuter cette réinitialisation sur la voie associée va et confirmer aussitôt sur la voie v.



Le cas défavorable suivant peut alors arriver :

- des paquets de données de la voie va sont en cours d'émission ou à émettre au niveau trame.
- la configuration de réinitialisation est immédiatement envoyée au processeur local, lequel processeur peut immédiatement envoyer des données.
- les données reçues devront être rangés dans les buffers de réception de la voie v (BRv1) qui sont aussi les buffers (BEva1) contenant le paquet en cours d'émission au niveau trame, ce qui peut aboutir à l'envoi de paquets de données incohérent au niveau voie. Un phénomène identique pourrait se passer si la libération était immédiatement confirmée ; en effet un des buffers de réception sert à contenir les données du paquet d'appel.

La durée des temporisateurs de surveillance fixée par TRANSPAC pour les confirmations locales étant de 40 s, n'oblige pas à anticiper les confirmations.

VII.4.d) Répercussion directe des événements

Le module JONCTION est composé uniquement de procédures événements des deux niveaux voies. Le rôle d'une procédure événement sera de retransmettre le paquet arrivé sur la voie associée.

Cependant la voie associée va peut ne pas être prête à prendre en compte le paquet, auquel cas il faudra attendre qu'elle soit prête par exemple :

- réception d'un paquet d'appel sur la voie v alors que la voie va est en cours de libération.
- réception d'un paquet de données sur v alors que la voie va est en cours de réinitialisation.

VII.5. Traitement des appels

VII.5.a) Réception des appels TRANSPAC

Le traitement de l'arrivée d'un paquet d'appel ne sera fait que si la voie associée va est dans l'état prêt. Si va n'est pas dans l'état prêt,

le traitement du paquet d'appel sera fait lorsque va passera à l'état "prêt". Le module JONCTION examinera si l'application appelée existe grâce à la table de correspondance.

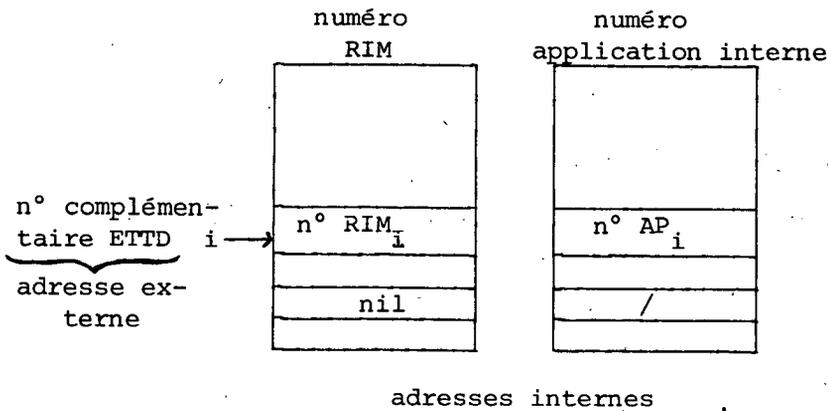
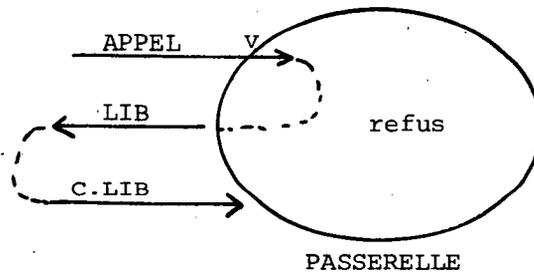


table de correspondance des adresses applications.

Deux cas peuvent se produire :

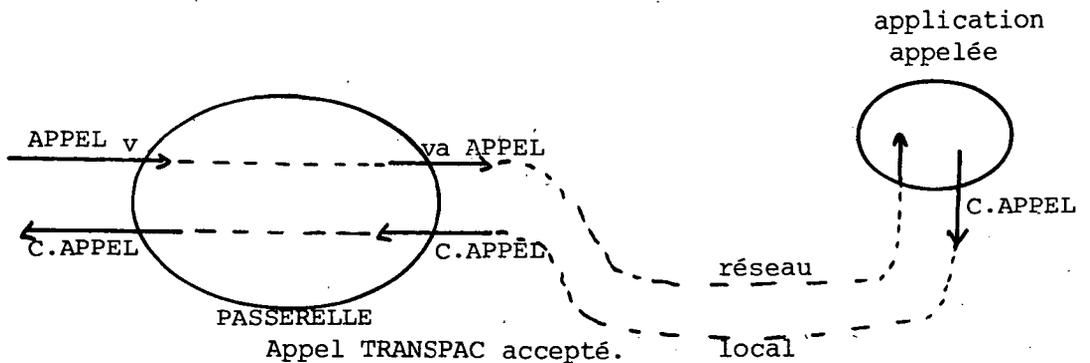
- l'application n'existe pas : l'appel est alors refusé.

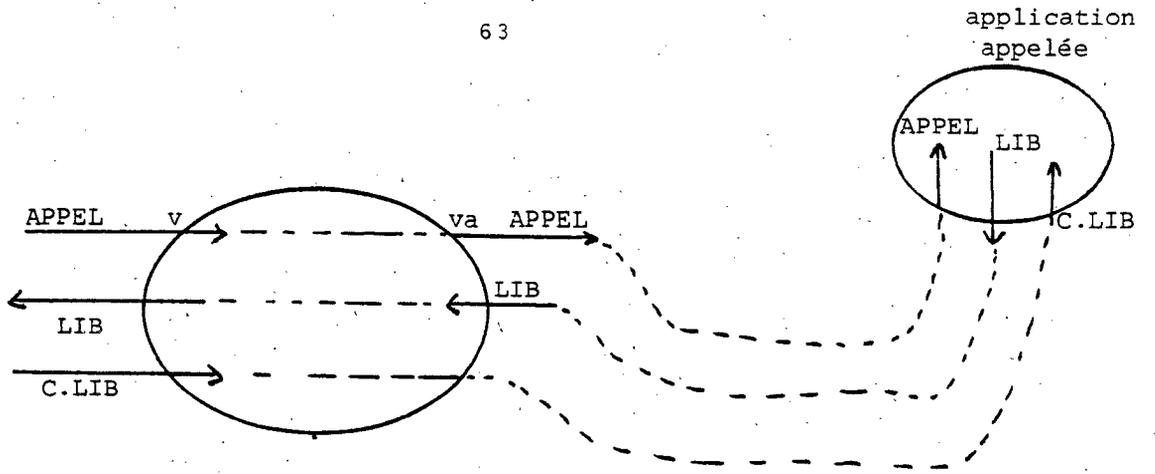
TRANSPAC



- l'application existe : les données d'appel TRANSPAC sont traduites en données d'appel CVI et l'appel reçu sur la voie v est transmis sur la voie va : l'initiative du refus ou de l'acceptation de l'appel va donc dépendre maintenant du processeur local supportant l'application appelée.

TRANSPAC



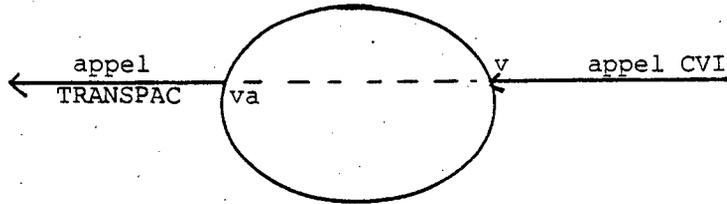


Appel TRANSPAC refusé par application locale.

VII.5.b) Réception des appels CVI

Aucun filtrage n'est fait par le module JONCTION pour les appels CVI. Dès que la voie associée va est prête les données d'appel CVI sont traduites en appel TRANSPAC et le paquet est transmis sur la voie va.

TRANSPAC



VII.5.c) Format des données d'appel

Format des données appel TRANSPAC

4	5	
LE/LR	NR/NE	DONNEES APPEL TRANSPAC

Format des données appel CVI

4	5	6	7
n° RIM récepteurs	AP	n° RIM émetteurs	n° AP

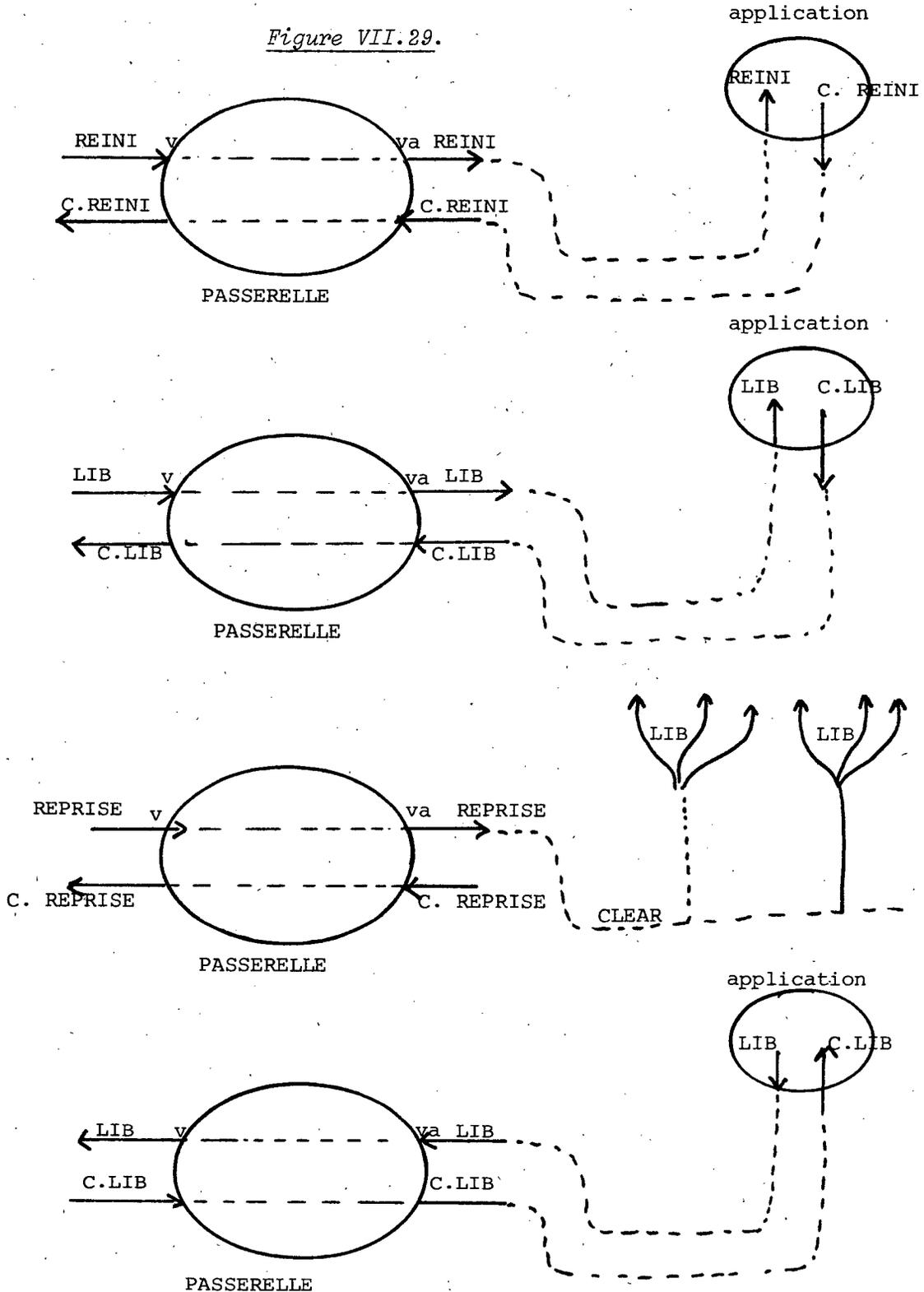
8			
longueur données CVI	LE/LR	NE/NR	DONNEES APPELTRANSPAC

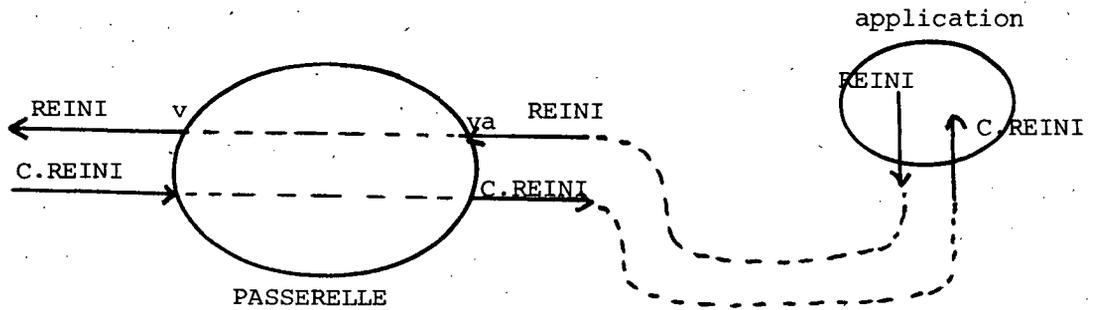
VII.6. Traitement des autres paquets de signalisation

Les autres paquets de signalisation : libération, réinitialisation, confirmation de libération, confirmation de réinitialisation, reprise et confirmation de reprise sont répercutés sur la voie associée dès qu'ils arrivent.

Figure VII.29.

TRANSPAC





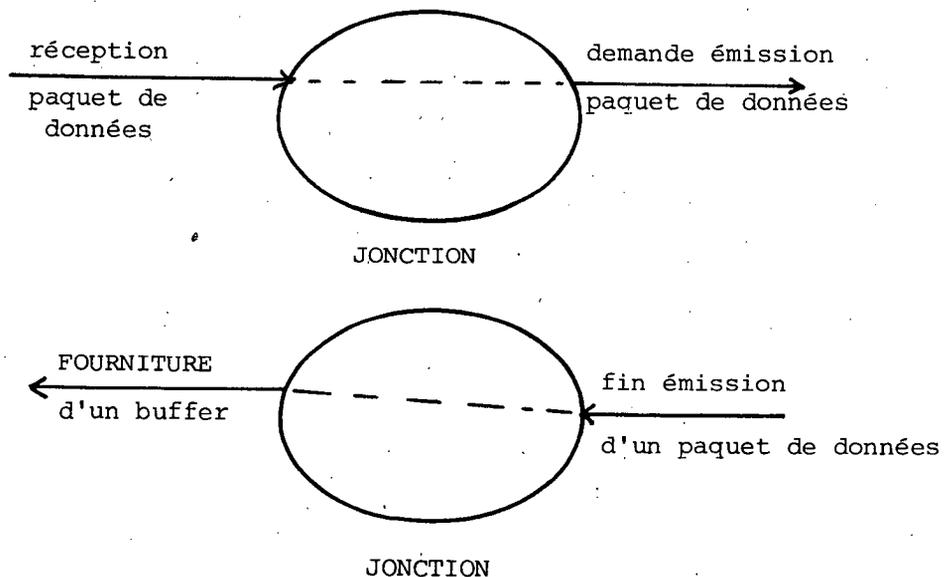
Remarques sur la figure ci-dessus :

a) le traitement d'une réinitialisation (resp d'une libération) est identique qu'elle soit à l'initiative de TRANSPAC ou d'une application du réseau local.

b) il n'y a pas de reprise faite à l'initiative du réseau local.

c) une reprise faite par TRANSPAC se traduit par une libération sur tous les circuits ouverts qui est transportée au niveau LIENS par des "clear" (cf chapitre LIENS).

VII.7. Traitement des paquets de données



Tout paquet reçu est immédiatement mis prêt à émettre sur la voie associée si elle est en phase de données.

Dans le cas contraire : la voie va associée n'est pas en phase de données (réinitialisation ou établissement non terminé). Lorsque cette voie va passera en phase de données, le module JONCTION examinera si des paquets de données ne sont pas en attente sur la voie v et auquel cas, ces paquets seront émis sur va.

Dès qu'un paquet sera émis, le buffer le contenant sera mis à la disposition de la voie associée pour recevoir le paquet de données suivant.

VIII - CONCLUSION

Les réalisations de l'IRISA sont terminées et testées : il s'agit des circuits virtuels internes (CVI) et de la passerelle TRANSPAC (sans niveau trame). Les tests ont été effectués avec des applications permettant d'une part de valider tous les mécanismes des circuits virtuels dans toutes les configurations, d'autre part de générer un trafic suffisamment important pour faire fonctionner les circuits virtuels à leur débit maximum. Le test de la passerelle TRANSPAC a été réalisé en interconnectant 2 réseaux locaux en faisant intervenir un module spécifique réalisant les fonctions essentielles de TRANSPAC.

La société MATRA Informatique réalise pour sa part le logiciel auquel elle est directement intéressée, à partir de spécifications IRISA : l'interface entre le niveau transport CV et son système d'exploitation DATASHARE, le niveau Trame.

L'expérience a montré que la décomposition en modules pouvant être utilisée indifféremment dans les configurations CV Transpac, CVI réseau local, passerelle, si elle conduit à un logiciel d'encombrement mémoire un peu plus important et d'efficacité moindre que des réalisations spécifiques, est très intéressante au point de vue développement et test de logiciel.

REFERENCES

- [1] Resource Interface Module RIM 9483.
DATAPOINT Corp.
- [2] Connexion d'un réseau local à TRANSPAC - CCETT.
Projet de 5ème année INSA (1980), D. Malorey, T. Pichon.
- [3] Spécifications techniques d'utilisation du réseau TRANSPAC.
- [4] Attached Resource Computer (ARC).
Simplified User's guide.
- [5] Product specification and hardware ref. Manual 5500.
DATAPOINT Corp.

Notes de travail

- . Spécifications de réalisation du niveau voie.
- . Spécifications de réalisation Trame.
- . Programmes de tests, mode d'emploi.

26 JUIN 81

INTERFACE UTILISATEUR - NIVEAU VOIE

L'UTILISATEUR DU NIVEAU VOIE (NIVEAU PAQUET) DISPOSE D'UN ENSEMBLE DE REQUETES POUR DEMANDER UNE ACTION AU NIVEAU VOIE.

INVERSEMENT UN EVENEMENT DU NIVEAU VOIE SE TRADUIT PAR L'APPEL D'UNE PROCEDURE EVENEMENT.

REQUETE: SOUS-PROGRAMME DU NIVEAU VOIE APPELE PAR L'UTILISATEUR.
PROCEDURE EVENEMENT: SOUS PROGRAMME UTILISATEUR APPELE PAR LE NIVEAU VOIE.

26 JUIN 81

 LISTE DES REQUETES

EN ENTREE: LES PARAMETRES SONT PASSES DANS LES REGISTRES.
 IL N'Y A PAS DE VERIFICATION SUR LES PARAMETRES (NUM. DE VOIE, TYPE DE PK ...)
 EN SORTIE: LES REGISTRES SONT INDEFINIS SAUF CEUX QUI RENDENT UN RESULTAT.
 LE FLAG ZERO INDIQUE L'ACCEPTATION (TRUE) OU LE REFUS (FALSE) DE LA REQUETE.

1) REQUETES GENERALES

.....
 REP

FONCTION: DEMANDER L'EMISSION D'UN PAQUET 'REPRISE'
 ENTREE: NEANT
 SORTIE: FLAG ZERO

.....
 OKREP

FONCTION: DEMANDER L'EMISSION D'UN PAQUET 'CONFIRMATION DE REPRISE'
 A UTILISER SUITE A LA RECEPTION D'UNE 'REPRISE'.
 ENTREE: NEANT
 SORTIE: FLAG ZERO

.....
 INIBUFR

FONCTION: INITIALISER LES ADRESSES DE BUFFERS RECEPTION.
 A UTILISER UNE FOIS AVANT TOUTE ACTION SUR LA VOIE.
 ENTREE: B: NUMERO DE VOIE LOGIQUE
 DE: ADRESSE D'UNE TABLE CONTENANT LES W ADRESSES DE BUFFERS.
 (W: LARGEUR DE LA FENETRE PAQUET)
 SORTIE: NEANT

.....
 NCVLIB

FONCTION: OBTENIR LE PREMIER NUMERO DE VOIE LOGIQUE LIBRE POUR FAIRE UN
 APPEL ; LA RECHERCHE SE FAIT A PARTIR DES PLUS GRANDS NUMEROS.
 ENTREE: NEANT
 SORTIE: FLAG ZERO
 B: NUMERO DE VOIE (NON SIGNIFICATIF SI FLAG ZERO = FALSE)

.....

16 JUIN 81

2) REQUETES SIGNALISATION

.....
APPEL

FONCTION: DEMANDE D'EMISSION D'UN PAQUET D'APPEL
ENTREE: B: NUMERO DE VOIE LOGIQUE
DE: ADRESSE DES DONNEES D'APPEL (CF 'FORMAT DES DONNEES D'APPEL')
C: LONGUEUR DES DONNEES D'APPEL (EN NOMBRE D'OCTETS)
SORTIE: FLAG ZERO

.....
OKAPPEL

FONCTION: ACCEPTATION D'UN APPEL ENTRANT RECU (CF 'EVENEMENTS')
ENTREE: B: NUMERO DE VOIE LOGIQUE
SORTIE: FLAG ZERO

.....
LIB

FONCTION: LIBERATION DU CIRCUIT VIRTUEL ETABLI OU EN COURS D'ETABLISSEMENT
SUR LA VOIE LOGIQUE INDIQUEE; CAUSE ET DIAGNOSTIC SONT TRANSMIS
DANS LE PAQUET 'LIBERATION'.
ENTREE: B: NUMERO DE VOIE LOGIQUE
D: DIAGNOSTIC
E: CAUSE
SORTIE: FLAG ZERO

.....
OKLIB

FONCTION: CONFIRMATION DE LIBERATION OBLIGATOIRE APRES LA RECEPTION D'UNE
LIBERATION (CF 'EVENEMENTS').
ENTREE: B: NUMERO DE VOIE LOGIQUE
SORTIE: FLAG ZERO

.....
REINI

FONCTION: DEMANDE DE REINITIALISATION DU CIRCUIT VIRTUEL.
ENTREE: B: NUMERO DE VOIE LOGIQUE
D: DIAGNOSTIC
E: CAUSE
SORTIE: FLAG ZERO

.....
OKREINI

FONCTION: CONFIRMATION DE REINITIALISATION DU CIRCUIT VIRTUEL OBLIGATOIRE
APRES RECEPTION D'UNE REINITIALISATION (CF 'EVENEMENTS').

26 JUIN 81

ENTREE:B:NUMERO DE VOIE
SORTIE:FLAG ZERO

.....
*
3) REQUETES EN "PHASE DONNEES"
.....

.....
BUFEM
.....

FONCTION: DEMANDE D'EMISSION D'UN PAQUET DE DONNEES
(CF NOTE 1)
ENTREE:B:NUMERO DE VOIE LOGIQUE
DE:ADRESSE DU BUFFER DE DONNEES
C:LONGUEUR DES DONNEES (EN NOMBRE D'OCTETS)
A:TYPE DONNEES (CF LISTE DES TYPES DE PAQUETS)
SORTIE:FLAG ZERO

.....
BUFREC
.....

FONCTION: FOURNITURE D'UN BUFFER EN RECEPTION.ACQUITTEMENT IMPLICITE DU
PREMIER BUFFER RECU ET NON ACQUITTE.
(CF NOTE 2)
ENTREE:B:NUMERO DE VOIE
DE:ADRESSE DU BUFFER
SORTIE:FLAG ZERO

.....
GETADR
.....

FONCTION:OBTENTION DES CARACTERISTIQUES DU PREMIER PAQUET DE DONNEES RECU.
(CF NOTE 2)
ENTREE:B:NUMERO DE VOIE LOGIQUE
SORTIE:FLAG ZERO (=FALSE SI PAS DE PAQUET RECU)
DE:ADRESSE DU BUFFER OU ONT ETE RANGEES LES DONNEES RECUES
C:LONGUEURS DONNEES (EN NOMBRE D'OCTETS)
A:TYPE DONNEES (CF LISTE DES TYPES DE PAQUETS)

.....
CONSON
.....

FONCTION:ACQUITTEMENT DU PREMIER BUFFER RECU ET NON ACQUITTE.
(CF NOTE 2)
ENTREE:B:NUMERO DE VOIE LOGIQUE
SORTIE:FLAG ZERO

.....
BUFRD
.....

FONCTION:FOURNITURE D'UN BUFFER EN RECEPTION
(CF NOTE 2)

6 JUIN 81

ENTREE:B#NUMERO DE VOIE LOGIQUE
DE:ADRESSE BUFFER RECEPTION
SORTIE:FLAG ZERO

.....

26 JUIN 81

LISTE DES EVENEMENTS

EN ENTREE LE NIVEAU VOIE FOURNIT LES PARAMETRES DANS LES REGISTRES.
L'UTILISATEUR DOIT TERMINER LA PROCEDURE 'EVENEMENT' PAR 'RET'.

1) EVENEMENTS GLOBAUX

ILS CONCERNENT LES CONNEXIONS/DECONNEXIONS DU NIVEAU INFERIEUR (TRAME OU
LIENS-RIM) ET LES REINITIALISATIONS DU NIVEAU VOIE (REPRISE).

.....
ETCNX

FONCTION:INDICATION QUE LE NIVEAU TRAME (OU LIENS-RIM) EST CONNECTE:
L'UTILISATEUR PEUT EMETTRE UNE REPRISE (REQUETE REF).
ENTREE:RIEN

.....
ETHS

FONCTION:INDICATION QUE LE NIVEAU TRAME (OU LIENS-RIM) N'A PAS PU
SE CONNECTER.
ENTREE:RIEN

.....
ETDISC

FONCTION:INDICATION QUE LE NIVEAU TRAME (OU LIENS-RIM) S'EST DECONNECTE:
L'UTILISATEUR PEUT ATTENDRE 'ETCNX' OU 'ETHS'.
ENTREE:RIEN

.....
EREPRIS

FONCTION:RECEPTION D'UN PAQUET REPRISE (TRANSFAC UNIQUEMENT):
L'UTILISATEUR DOIT CONFIRMER (REQUETE 'OKREP')
ENTREE:RIEN

.....
EVOIHS

FONCTION:INDICATION D'ECHEC DE LA PHASE DE REPRISE (EMISSION DE REPRISE
OU DE CONFIRMATION DE REPRISE):ARRET
ENTREE:RIEN

.....
EFINREP

6 JUIN 81

FONCTION:INDICATION DE LA FIN CORRECTE DE LA PHASE DE REPRISE:
L'UTILISATEUR PEUT EMETTRE OU RECEVOIR DES APPELS.
ENTREE:RIEN

.....
2) EVENEMENTS SIGNALISATION
.....

.....
EAPPEL
.....

FONCTION:RECEPTION D'UN APPEL:L'UTILISATEUR PEUT SOIT CONFIRMER (REQUETE 'OKAPPEL' SOIT REFUSER (REQUETE 'LIB').
ENTREE:B:NUMERO DE VOIE LOGIQUE
DE:ADRESSE DONNEES D'APPEL (CF 'FORMAT DES DONNEES D'APPEL')
C:LONGUEUR DES DONNEES D'APPEL (EN NOMBRE D'OCTETS)

.....
ECVIDON
.....

FONCTION:PASSAGE DE LA VOIE LOGIQUE EN "PHASE DONNEES":L'UTILISATEUR PEUT EMETTRE DES PAQUETS DE DONNEES (REQUETE 'BUFEM').
ENTREE:B:NUMERO DE VOIE LOGIQUE

.....
EREINI
.....

FONCTION:RECEPTION D'UNE REINITIALISATION:L'UTILISATEUR DOIT CONFIRMER (REQUETE 'OKREINI') OU LIBERER (REQUETE 'LIB')
ENTREE:B:NUMERO DE VOIE
D:DIAGNOSTIC
E:CAUSE

.....
ELIB1
.....

FONCTION:INDICATION QU'IL Y A EU UNE LIBERATION EMISE SUITE A NON REPONSE DE L'APPEL LORS D'UN APPEL.RIEN A FAIRE.
ENTREE:B:NUMERO DE VOIE LOGIQUE

.....
ELIB2
.....

FONCTION:RECEPTION D'UNE LIBERATION EN REPONSE A UN APPEL:L'UTILISATEUR DOIT OBLIGATOIREMENT CONFIRMER (REQUETE 'OKLIB').
ENTREE:B:NUMERO DE VOIE LOGIQUE
D:DIAGNOSTIC
E:CAUSE

.....
ELIB3
.....

26 JUIN 81

 FONCTION:RECEPTION D'UNE LIBERATION EN "PHASE DONNEES":L'UTILISATEUR
 DOIT OBLIGATOIREMENT CONFIRMER (REQUETE 'OKLIB').
 ENTREE:B:NUMERO DE VOIE LOGIQUE
 D:DIAGNOSTIC
 E:CAUSE

.....
ELIB4

 FONCTION:INDICATION QU'IL Y A EU UNE LIBERATION EMISE SUITE A NON REPONSE
 DU CORRESPONDANT LORS D'UNE REINITIALISATION:RIEN A FAIRE.
 ENTREE:B:NUMERO DE VOIE LOGIQUE

.....
ELIB5

 FONCTION:RECEPTION D'UNE LIBERATION EN REPONSE A UNE REINITIALISATION:
 L'UTILISATEUR DOIT OBLIGATOIREMENT CONFIRMER (REQUETE 'OKLIB').
 ENTREE:B:NUMERO DE VOIE LOGIQUE
 D:DIAGNOSTIC
 E:CAUSE

.....
EPRET

 FONCTION:INDICATION QUE LA VOIE LOGIQUE EST REDEVENUE PRETE:L'UTILISATEUR
 PEUT EMETTRE UN APPEL.
 ENTREE:B:NUMERO DE VOIE LOGIQUE

.....
3) EVENEMENTS EN "PHASE DONNEES"

.....
RECDON

 FONCTION:RECEPTION D'UN PAQUET DE DONNEES
 L'UTILISATEUR A LE CHOIX ENTRE 2 TYPES D'INTERFACE RECEPTION
 (CF NOTE 2)
 ENTREE:B:NUMERO DE VOIE LOGIQUE
 DE:ADRESSE BUFFER DONNEES RECU
 C:LONGUEUR DONNEES (EN NOMBRE D'OCTETS)
 A:TYPE DONNEES (CF 'LISTE DES TYPES DE PAQUETS')

.....
FINBUF

 FONCTION:INDICATION QUE LE BUFFER A ETE EMIS:L'UTILISATEUR PEUT LE
 REUTILISER. (CF NOTE 1)
 ENTREE:B:NUMERO DE VOIE LOGIQUE

6 JUIN 81

DE:ADRESSE BUFFER DONNEES EMIS

.....

26 JUIN 81

+
NOTE 1

=====

EN EMISSION UN BUFFER DE DONNEES NE DOIT PAS ETRE UTILISE NI ECRASE AVANT L'INDICATION DE BONNE EMISSION (EVENEMENT 'FINBUF'). EN EFFET IL N'Y A PAS RECOPIE DE BUFFER AU NIVEAU INFERIEUR.

NOTE 2

=====

EN RECEPTION L'UTILISATEUR DISPOSE DE 2 JEUX DE REQUETES (EXCLUSIFS L'UN DE L'AUTRE): JEU 1 : BUFREC

ACQUITTEMENT IMPLICITE DU PREMIER BUFFER RECU ET FOURNITURE D'UN BUFFER RECEPTION.

SI LE TRAITEMENT DU PAQUET RECU (EVENEMENT 'RECDON') DOIT ETRE DIFFERE, IL FAUT DANS RECDON MEMORISER LES PARAMETRES DU PAQUET RECU.

JEU 2 : GETADR+CONSUM+BUFRD

GETADR PERMET D'OBTENIR LES CARACTERISTIQUES DU PREMIER PAQUET RECU, CONSUM ACQUITTE LE BUFFER BUFRD FOURNIT UN BUFFER RECEPTION.

CET INTERFACE PERMET DE NE RIEN MEMORISER DANS 'RECDON'.

LA RECEPTION D'UN PAQUET DE DONNEES SE TRADUIT PAR L'EVENEMENT 'RECDON' QUEL QUE SOIT LE JEU DE REQUETES UTILISE.

ANNEXE 2

26 JUIN 81

INTERFACE NIVEAU VOIE AVEC LE NIVEAU TRAME (CAS TRANSPAC)
OU LE NIVEAU LIENS-RIM (CAS CUI)

CET INTERFACE DECOULE:

- 1): DE LA STRUCTURATION EN PROCESSUS (PROCESSUS ALIMENTATION):
GESTION DE LA 'FENETRE TRAME'
- 2): DU FAIT DE NON RECOPIE DE DONNEES EN EMISSION ET EN RECEPTION:
D'OU LA NECESSITE DE FOURNIR AU NIVEAU INFERIEUR LES COORDONNEES DES
BUFFERS A UTILISER POUR LES DONNEES.
D'OU LA NECESSITE D'AVOIR CONNAISSANCE DES EVENEMENTS 'BONNE RECEPTION'
POUR VALIDER LE BUFFER RECU, 'BONNE EMISSION' POUR LIBERER LE BUFFER EMIS.

L'INTERFACE IMPOSE AU NIVEAU TRAME (OU LIENS-RIM):
 DE DISPOSER EN RECEPTION DE BUFFERS POUVANT CONTENIR L'ENTETE PAQUET
 (OCTETS 0 A 4, CF 'FORMAT DES PAQUETS').
 DE GERER EN EMISSION N ZONES DE 8 OCTETS (N = VALEUR DE LA 'FENETRE TRAME')
 FORMATTES AINSI:
 OCTET 0 A 4: ENTETE PAQUET (CF 'FORMAT DES PAQUETS')
 OCTET 5: LONGUEUR TOTALE DU PAQUET Y COMPRIS L'ENTETE.
 OCTETS 6 ET 7: ADRESSE DE LA SUITE DU PAQUET (S'IL Y A UNE SUITE)

LISTE DES SOUS-PROGRAMMES DU NIVEAU VOIE

=====

INITIALISATION/TERMINAISON

VCNX

FONCTION: INDICATION QUE LE NIVEAU TRAME (OU LIENS-RIM) S'EST CONNECTE
ENTREE: NEANT

VDISC

FONCTION: INDICATION DE LA DECONNEXION DU NIVEAU TRAME (OU LIENS-RIM)
ENTREE: NEANT

VTHS

FONCTION: INDICATION QUE LE NIVEAU TRAME N'A PAS PU SE CONNECTER APRES
PLUSIEURS ESSAIS.
ENTREE: NEANT

5 JUIN 81

RECEPTION

=====

.....
RECPCD

FONCTION: RECEPTION D'UN PAQUET COURT (ENTIEREMENT CONTENU DANS LE BUFFER TRAME (OU LIENS-RIM))

CAS TRAME:

ENTREE:DE:ADRESSE ENTETE PAQUET
C:LONGUEUR DU PAQUET

CAS LIENS-RIM:

ENTREE:DE:ADRESSE ENTETE PAQUET (POINTE SUR L'OCTET 2:TYPE)
B:LONGUEUR ENTETE
A:NUMERO DE VOIE

SORTIE:NEANT

.....
RECHBUF

FONCTION: RECEPTION D'UN PAQUET LONG: ACQUISITION DE L'ADRESSE BUFFER OU IL FAUT RANGER LE RESTE DU PAQUET.

CAS TRAME:

ENTREE:DE:ADRESSE ENTETE PAQUET

CAS LIENS-RIM:

ENTREE:DE:ADRESSE ENTETE PAQUET (POINTE SUR L'OCTET 2:TYPE)
A:NUMERO DE VOIE

SORTIE:SI FLAG ZERO=TRUE HL:ADRESSE BUFFER
SI FLAG ZERO=FALSE IL FAUT IGNORER LE RESTE DU PAQUET.

.....
AKREC

FONCTION: RECEPTION D'UN PAQUET LONG: ACQUITTEMENT POSITIF;LA RECEPTION S'EST TERMINEE CORRECTEMENT.

ENTREE:DE:ADRESSE ENTETE PAQUET

C:LONGUEUR TOTALE DU PAQUET Y COMPRIS L'ENTETE

SORTIE:NEANT

.....
NAKREC

FONCTION: RECEPTION D'UN PAQUET LONG:ACQUITTEMENT NEGATIF;LA RECEPTION NE S'EST PAS TERMINEE CORRECTEMENT.

ENTREE:NEANT

SORTIE:NEANT

.....

26 JUIN 81

*
EMISSION
=====.....
VAKEMI

FONCTION: ACQUITTEMENT EMISSION PAQUET
 CAS TRAME:
 ENTREE: DE: ADRESSE ENTETE PAQUET EMIS
 CAS LIENS-RIM:
 ENTREE: DE: ADRESSE ENTETE PAQUET EMIS (POINTE SUR L'OCTET 2:TYPE)
 A: NUMERO DE VOIE
 SORTIE: NEANT

.....
VLIBUF

FONCTION: LIBERATION DU BUFFER DE DONNEES (UTILISE UNIQUEMENT DANS LE
 CAS LIENS-RIM)
 ENTREE: DE: ADRESSE DE BUFFER DE DONNEES A LIBERER
 B: NUMERO DE VOIE
 SORTIE: NEANT

*
GESTION DE LA 'FENETRE TRAME'
=====

- 1) OBTENTION DE L'ADRESSE ENTETE OU DOIT ETRE FORMATTE LE PAQUET.
 (SOUS-PROGRAMME DU NIVEAU TRAME (OU LIENS-RIM) APPELE PAR LE NIVEAU VOIE)

.....
FTLIBR

FONCTION: FOURNIR LE PREMIER BUFFER ENTETE DISPONIBLE POUR L'EMISSION.
 ENTREE: NEANT
 SORTIE: DE: ADRESSE ENTETE OU FORMATTER LE PROCHAIN PAQUET

- 2) INDICATION DE L'INCREMENTATION DE LA 'FENETRE TRAME' EMISSION.
 (SOUS-PROGRAMME DU NIVEAU VOIE APPELE PAR LE NIVEAU TRAME (OU LIENS-RIM))

.....
FENTRA

FONCTION: INDICATION DE L'INCREMENTATION DE LA 'FENETRE TRAME' EMISSION.
 ENTREE: C: NOMBRE DE TRAMES ACQUITTEES

ANNEXE 3

6 JUIN 81

 FORMAT DES PAQUETS

 1) PAQUETS CUI (NON COMPRIS ENTETE LIEN)

OCTET 0: VLE:NUMERO DE VOIE LOGIQUE EMETTRICE
 OCTET 1: VLR:NUMERO DE VOIE LOGIQUE RECEPTRICE
 OCTET 2: TYPE DE PAQUET (CF LISTE DES TYPES DE PAQUET)
 LES OCTETS 0 ET 1 SONT EXPLOITES PAR LE NIVEAU LIENS-RIM

	0	1	2	3	4	5	6
APPEL	VLE	0377	013		DONNEES D'APPEL (CF FORMAT)		
CONF.APPEL	VLE	VLR	017				
LIB	VLE	VLR	013		CAUSE		DIAG.
CONF.LIB	VLE	VLR	017				
REINI	VLE	VLR	033		CAUSE		DIAG.
CONF.REINI	VLE	VLR	037				
DONNEES	VLE	VLR	TYPD	P(R)	P(S)		DONNEES
RR	VLE	VLR	001	P(R)			

TYPD EST UN DES TYPES 'DONNEES' DECRIT DANS 'LISTE DES TYPES DE PAQUETS'.
 P(R) ET P(S) ONT LA MEME SIGNIFICATION QUE POUR TRANSPAC MAIS SONT CODES
 SUR UN OCTET.
 LA LONGUEUR DES DONNEES EST LIMITEE A 246 OCTETS.

 *
 2) FORMAT TRANSPAC

 VOIR LA DOCUMENTATION TRANSPAC

6 JUIN 81

 LISTE DES TYPES DE PAQUETS

 PAQUETS DE DONNEES

CES TYPES DE PAQUETS SONT CEUX QUI SONT MANIPULES PAR L'UTILISATEUR
 (REQUETE 'BUFEM' , EVENEMENT 'RECDON').
 LEUR SIGNIFICATION CORRESPOND AUX PAQUETS DE DONNEES TRANSPAC AVEC OU SANS
 BIT Q ET/OU M (CF DOCUMENTATION TRANSPAC).
 EN PROTOCOLE CUI (RESEAU LOCAL) ILS SONT TRANSPORTES SANS MODIFICATION
 EN PROTOCOLE TRANSPAC ILS SE TRADUISENT PAR LES BITS Q ET M DANS LES PAQUETS
 (CF 'FORMAT DES PAQUETS').

CODAGE OCTAL

000 : DONNEES SANS BIT Q NI BIT M
 020 : DONNEES AVEC BIT M
 200 : DONNEES AVEC BIT Q
 220 : DONNEES AVEC BIT Q ET BIT M

 AUTRES TYPES DE PAQUETS

ILS NE SONT PAS MANIPULES EXPLICITEMENT PAR L'UTILISATEUR.
 CE SONT LES TYPES DE PAQUETS TRANSPAC CODES SUR 1 OCTET.

CODAGE OCTAL

. 013 : APPEL
 . 017 : CONFIRMATION D'APPEL
 . 023 : LIBERATION
 . 027 : CONFIRMATION DE LIBERATION
 . 033 : REINITIALISATION
 . 037 : CONFIRMATION DE REINITIALISATION
 . 373 : REPRISE
 . 377 : CONFIRMATION DE REPRISE
 . 001 : FRET A RECEVOIR (RR)

. REMARQUE: L'INTERRUPTION ET LA CONFIRMATION D'INTERRUPTION NE SONT PAS
 . IMPLEMENTEES.

 INTERFACE UTILISATEUR - DRIVER RIM

 LISTE DES REQUETES

EN ENTREE#LES PARAMETRES SONT PASSES DANS DES REGISTRES.
 IL N'Y A PAS DE VERIFICATION (NUM. DE BUFFER,NOMBRE D'OCTETS...)

EN SORTIE LES REGISTRES SONT INDEFINIS SAUF CEUX QUI RENDENT UN RESULTAT.

1) REQUETES RELATIVES A L'EMISSION

BLIBR

FONCTION#OBTENIR UN NUMERO DE BUFFER RIM LIBRE POUR L'EMISSION
 ENTREE#NEANT
 SORTIE#A#NUMERO DE BUFFER RIM LIBRE (0377 SI AUCUN BUFFER LIBRE)

ECRIM

FONCTION#ECRITURE DANS UN BUFFER RIM POUR L'EMISSION
 ENTREE#A#NUMERO DE BUFFER RIM
 B#NUMERO DE RIM DESTINATAIRE
 C#NOMBRE D'OCTETS A ECRIRE
 D#DEPLACEMENT DE L'ECRITURE (PAR RAPPORT AU DEBUT DES DONNEES)
 E#NOMBRE TOTAL D'OCTETS DE DONNEES
 HL#ADRESSE BUFFER OU SONT LES DONNEES
 SORTIE#NEANT

AEMET

FONCTION#DEMANDER L'EMISSION D'UN BUFFER RIM
 ENTREE#A#NUMERO DE BUFFER RIM A EMETTRE
 SORTIE#NEANT

BAKEMI

FONCTION#OBTENIR LE NUMERO DU PREMIER BUFFER RIM EMISSION ACQUITTE
 OU REFUSE PAR LE DISTANT ,ET LE RENDRE REUTILISABLE.
 ENTREE#NEANT
 SORTIE#A#NUMERO DE BUFFER RIM (0377 SI PAS DE BUFFER)
 B#0 POUR 'ACQUITTE' , 1 POUR 'REFUSE PAR LE DISTANT'

2) REQUETES RELATIVES A LA RECEPTION

.....
BPLEIN

.....
FONCTION:OBTENIR LE NUMERO DU BUFFER RIM RECU LE PREMIER.
ENTREE:NEANT
SORTIE:A:NUMERO DE BUFFER RIM RECU (0377 SI PAS DE BUFFER)
.....

.....
LIRIM

.....
FONCTION:LECTURE DANS UN BUFFER RIM
ENTREE:A:NUMERO DE BUFFER RIM
C:OMBRE D'OCTETS A LIRE
D:DEPLACEMENT POUR LA LECTURE (PAR RAPPORT AU DEBUT DES DONNEES)
HL:ADRESSE BUFFER
SORTIE:B:NUMERO DE RIM DISTANTE
E:OMBRE TOTAL D'OCTETS DE DONNEES
.....

.....
LIBER

.....
FONCTION:LIBERER UN BUFFER RIM RECEPTION
ENTREE:A:NUMERO DE BUFFER RIM
.....

3) REQUETES D'INITIALISATION ET DE TERMINAISON

.....
OUVGUI

.....
FONCTION:LANCER LE DRIVER RIM
ENTREE:NEANT
SORTIE:NEANT
.....

.....
FERMGUI

.....
FONCTION:ARRETER LE DRIVER RIM
ENTREE:NEANT
SORTIE:NEANT
.....

.....
ISOLEM

.....
FONCTION:LANCER LE TEST D'ISOLEMENT
ENTREE:NEANT
.....

LISTE DES EVENEMENTS

LEMIS

FONCTION: INDICATION DE LA BONNE EMISSION D'UN BUFFER RIM
ENTREE: A: NUMERO DE BUFFER RIM

LHSDIS

FONCTION: INDICATION QUE LE DISTANT EST HORS SERVICE (REFUSE DE RECEVOIR)
ENTREE: A: NUMERO DE BUFFER RIM

LRECU

FONCTION: INDICATION QU'UN MESSAGE RIM A ETE RECU
ENTREE: A: NUMERO DE BUFFER RIM

LSERVIC

FONCTION: INDICATION QUE LE DRIVER RIM S'EST CORRECTEMENT INITIALISE
ENTREE: NEANT

LISOLE

FONCTION: INDICATION QUE LA RIM EST ISOLEE DANS LE RESEAU
ENTREE: NEANT

LHSLOC

FONCTION: INDICATION QUE LA RIM EST HORS SERVICE
ENTREE: NEANT

Liste des Publications Internes IRISA

- PI 142 **Un lemme général de stabilité pour la commande adaptative en déterministe de systèmes non nécessairement à minimum de phase:**
Cl. Samson , 40 pages ; *Novembre 1980*
- PI 143 **Détection, Estimation de l'orientation et saisie d'une cible mobile par proximité optique**
B. Espiau , 142 pages ; *Janvier 1981*
- PI 144 **Une contribution à l'étude de l'impact de l'informatique sur les organisations**
L. Breton, A. Prod'homme, J. Villard , 58 pages ; *Décembre 1980*
- PI 145 **Rupture de modèles statistiques**
M. Basseville, A. Benveniste , 130 pages ; *Mars 1981*
- PI 146 **Traitement des questionnaires avec non réponse, analyse des correspondances avec marge modifiée et analyse multicanonique avec contrainte**
B. Escoffier , 38 pages ; *Mars 1981*
- PI 147 **Deux files d'attente à capacité limitée en tandem**
J. Pellaumail, J. Boyer , 19 pages ; *Juillet 1981*
- PI 148 **Programme de classification hiérarchique : 1) Méthode de la vraisemblance des liens, 2) Méthode de la variance expliquée**
I.C. Lerman , 113 pages ; *Juin 1981*
- PI 149 **Convergence des méthodes de commande adaptative en présence de perturbations aléatoires**
J.J. Fuchs , 46 pages ; *Juillet 1981*
- PI 150 **Construction automatique et évaluation d'un graphe d'«implication» issu de données binaires, dans le cadre de la didactique des mathématiques**
H. Rostam , 112 pages ; *Juin 1981*
- PI 151 **Réalisation d'un outil d'évaluation de mécanismes de détection de pannes]-Projet Pilote SURF**
B. Decouty, G. Michel, C. Wagner, Y. Crouzet , 59 pages ; *Juillet 1981*
- PI 152 **Règle maximale**
J. Pellaumail , 18 pages ; *Septembre 1981*
- PI 153 **Corrélation partielle dans le cas « qualitatif »**
I.C. Lerman , 125 pages ; *Octobre 1981*
- PI 154 **Stability analysis of adaptively controlled not-necessarily minimum phase systems with disturbances**
Cl. Samson , 40 pages ; *Octobre 1981*
- PI 155 **Analyses d'opinions d'instituteurs à l'égard de l'appropriation des nombres naturels par les élèves de cycle préparatoire**
R. Gras , 37 pages ; *Octobre 1981*
- PI 156 **Récursion induction principe revisited**
G. Boudol, L. Kott , 49 pages ; *Décembre 1981*
- PI 157 **Loi d'une variable aléatoire à valeur R^+ réalisant le minimum des moments d'ordre supérieur à deux lorsque les deux premiers sont fixés**
M.Kowalowka, R. Marie , 8 pages ; *Décembre 1981*
- PI 158 **Réalisations stochastiques de signaux non stationnaires, et identification sur un seul échantillon**
A. Benveniste J.J. Fuchs , 33 pages ; *Mars 1982*
- PI 159 **Méthode d'interprétation d'une classification hiérarchique d'attributs-modalités pour l'«explication» d'une variable ; application à la recherche de seuil critique de la tension artérielle systolique et des indicateurs de risque cardiovasculaire**
B. Tallur , 34 pages ; *Janvier 1982*
- PI 160 **Probabilité stationnaire d'un réseau de files d'attente multiclasse à serveur central et à routages dépendant de l'état**
L.M. Le Ny , 18 pages ; *Janvier 1982*
- PI 161 **Détection séquentielle de changements brusques des caractéristiques spectrales d'un signal numérique**
M. Basseville, A. Benveniste , pages ; *Mars 1982*
- PI 162 **Actes regroupés des Journées de Classification de Toulouse (Mai 1980), et de Nancy (Juin 1981)**
I.C. Lerman , 304 pages ;
- PI 163 **Modélisation et Identification des caractéristiques d'une structure vibratoire : un problème de réalisation stochastique d'un grand système non stationnaire**
M. Prévosto, A. Benveniste, B. Barnouin , 46 pages ; *Mars 1982*
- PI 164 **An enlarged definition and complete axiomatization of observational congruence of finite processes**
Ph. Darondeau , 45 pages ; *Avril 1982*
- PI 165 **Accès vidéotex à une banque de données médicales**
A. Chauffaut, M. Dragone, R. Rivoire, J.M. Roger , 25 pages ; *Mai 1982*
- PI 166 **Comparaison de groupes de variables définies sur le même ensemble d'individus**
B. Escoffier, J. Pages , 115 pages ; *Mai 1982*
- PI 167 **Transport en circuits virtuels internes sur réseau local et connexion Transpac**
M. Tournois, R. Trépos , 90 pages ; *Mai 1982*
- PI 168 **Impact de l'intégration sur le traitement automatique de la parole**
P. Quinton , 14 pages ; *Mai 1982*
- PI 169 **A systolic algorithm for connected word recognition**
J.P. Banâtre, P. Frison, P. Quinton , 13 pages ; *Mai 1982*
- PI 170 **A network for the detection of words in continuous speech**
J.P. Banâtre, P. Frison, P. Quinton , 24 pages ; *Mai 1982*

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique