



**HAL**  
open science

# Le traitement par lots dans un réseau hétérogène - implémentation du serveur OS/MVT sur IBM 360/67 pour le réseau CYCLADES

Robert Fournier

► **To cite this version:**

Robert Fournier. Le traitement par lots dans un réseau hétérogène - implémentation du serveur OS/MVT sur IBM 360/67 pour le réseau CYCLADES. Réseaux et télécommunications [cs.NI]. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1976. Français. NNT: . tel-00010533

**HAL Id: tel-00010533**

**<https://theses.hal.science/tel-00010533>**

Submitted on 11 Oct 2005

**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.

# THESE

*présentée à*

**Université Scientifique et Médicale de Grenoble  
Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*  
**DOCTEUR INGENIEUR**

*par*

**Robert FOURNIER**



**LE TRAITEMENT PAR LOTS  
DANS UN RESEAU HETEROGENE**

**IMPLEMENTATION DU SERVEUR OS/MVT  
SUR «I.B.M. 360/67»  
POUR LE RESEAU CYCLADES**



Thèse soutenue le 20 décembre 1976 devant la Commission d'Examen

Président : N. GASTINEL

Examineurs : L. BOLLIET  
F. ANCEAU

Rapporteur : H. ZIMMERMANN



UNIVERSITE SCIENTIFIQUE  
ET MEDICALE DE GRENOBLE

---

Monsieur Gabriel CAU : Président  
Monsieur Pierre JULLIEN : Vice-Président

---

**MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.**

---

PROFESSEURS TITULAIRES

MM. ARNAUD Paul	Chimie
AUBERT Guy	Physique
AYANT Yves	Physique approfondie
Mme BARBIER Marie-Jeanne	Electrochimie
MM. BARBIER Jean-Claude	Physique Expérimentale
BARBIER Reynold	Géologie appliquée
BARJON Robert	Physique nucléaire
BARNOUD Fernand	Biosynthèse de la cellulose
BARRA Jean-René	Statistiques
BARRIE Joseph	Clinique chirurgicale
BEAUDOING André	Clinique de Pédiatrie et Puériculture
BERNARD Alain	Mathématiques Pures
Mme BERTRANDIAS Françoise	Mathématiques Pures
MM. BERTRANDIAS Jean-Paul	Mathématiques Pures
BEZES Henri	Pathologie chirurgicale
BLAMBERT Maurice	Mathématiques Pures
BOLLIET Louis	Informatique (IUT B)
BONNET Georges	Electrotechnique
BONNET Jean-Louis	Clinique ophtalmologique
BONNET-EYMARD Joseph	Clinique gastro-entérologique
Mme BONNIER Marie-Jeanne	Chimie générale
MM. BOUCHERLE André	Chimie et toxicologie
BOUCHEZ Robert	Physique nucléaire
BOUSSARD Jean-Claude	Mathématiques Appliquées
BOUTET DE MONTVEL Louis	Mathématiques Pures
BRAVARD Yves	Géographie
CABANEL Guy	Clinique rhumatologique et hydrologique
CALAS François	Anatomie
CARLIER Georges	Biologie végétale
CARRAZ Gilbert	Biologie animale et pharmacodynamie
CAU Gabriel	Médecine légale et toxicologie
CAUQUIS Georges	Chimie organique
CHABAUTY Claude	Mathématiques Pures
CHARACHON Robert	Clinique Oto-rhino-laryngologique
CHATEAU Robert	Clinique de neurologie
CHIBON Pierre	Biologie animale
COEUR André	Pharmacie chimique et chimie analytique
CONTAMIN Robert	Clinique gynécologique
COUDERC Pierre	Anatomie pathologique
Mme DEBELMAS Anne-Marie	Matière médicale
MM. DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELORMAS Pierre	Pneumophtisiologie

MM. DEPORTES Charles	Chimie minérale
DESRE Pierre	Métallurgie
DESSAUX Georges	Physiologie animale
DODU Jacques	Mécanique appliquée (IUT A)
DOLIQUE Jean-Michel	Physique des plasmas
DREYFUS Bernard	Thermodynamique
DUCROS Pierre	Cristallographie
DUGOIS Pierre	Clinique de dermatologie et syphiligraphie
GAGNAIRE Didier	Chimie physique
GALLISSOT François	Mathématiques Pures
GALVANI Octave	Mathématiques Pures
GASTINEL Noël	Analyse numérique
GAVEND Michel	Pharmacologie
GEINDRE Michel	Electroradiologie
GERBER Robert	Mathématiques Pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
JANIN Bernard	Géographie
KAHANE André	Physique générale
KLEIN Joseph	Mathématiques Pures
KOSZUL Jean-Louis	Mathématiques Pures
KRAVTCHENKO Julien	Mécanique
KUNTZMANN Jean	Mathématiques Appliquées
LACAZE Albert	Thermodynamique
LACHARME Jean	Biologie végétale
Mme LAJZEROWICZ Janine	Physique
MM. LAJZEROWICZ Joseph	Physique
LATREILLE René	Chirurgie générale
LATURAZE Jean	Biochimie pharmaceutique
LAURENT Pierre-Jean	Mathématiques Appliquées
LEDRU Jean	Clinique médicale B
LLIBOUTRY Louis	Géophysique
LOISEAUX Pierre	Sciences nucléaires
LONGEQUEUE Jean-Pierre	Physique nucléaire
LOUP Jean	Géographie
Mlle LUTZ Elisabeth	Mathématiques Pures
MM. MALGRANGE Bernard	Mathématiques Pures
MALINAS Yves	Clinique obstétricale
MARTIN-NOEL Pierre	Clinique cardiologique
MAZARE Yves	Clinique médicale A
MICHEL Robert	Minéralogie et Pétrographie
MICOUD Max	Clinique maladies infectieuses
MOURIQUAND Claude	Histologie
MOUSSA André	Chimie nucléaire
MULLER Jean-Michel	Thérapeutique (Néphrologie)
NEEL Louis	Physique du Solide
OZENDA Paul	Botanique
PAYAN Jean-Jacques	Mathématiques Pures
PEBAY-PEYROULA Jean-Claude	Physique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
REVOL Michel	Urologie
RINALDI Renaud	Physique
DE ROUGEMONT Jacques	Neuro-chirurgie
SEIGNEURIN Raymond	Microbiologie et Hygiène
SENGEL Philippe	Zoologie

MM. SIBILLE Robert	Construction mécanique (IUT A)
SOUTIF Michel	Physique générale
TANCHE Maurice	Physiologie
TRAYNARD Philippe	Chimie générale
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire
VAUQUOIS Bernard	Calcul électronique
Mme VERAÏN Alice	Pharmacie galénique
MM. VERAÏN André	Physique
VEYRET Paul	Géographie
VIGNAIS Pierre	Biochimie médicale
YOCCOZ Jean	Physique nucléaire théorique

#### PROFESSEURS ASSOCIES

MM. CLARK Gilbert	Spectrométrie physique
CRABBE Pierre	CERMO
ENGLMAN Robert	Spectrométrie physique
HOLTZBERG Frédéric	Basses températures
DEMBICKI Eugéniuz	Mécanique
MATSUSHIMA Yozo	Mathématiques Pures

#### PROFESSEURS SANS CHAIRE

Mlle AGNIUS-DELORD Claudine	Physique pharmaceutique
ALARY Josette	Chimie analytique
MM. AMBROISE-THOMAS Pierre	Parasitologie
BELORIZKY Elie	Physique
BENZAKEN Claude	Mathématiques Appliquées
BIAREZ Jean-Pierre	Mécanique
BILLET Jean	Géographie
BOUCHET Yves	Anatomie
BRUGEL Lucien	Energétique (IUT A)
BUISSON René	Physique (IUT A)
BUTEL Jean	Orthopédie
COHEN ADDAD Pierre	Spectrométrie physique
COLOMB Maurice	Biochimie
CONTE René	Physique (IUT A)
DEPASSEL Roger	Mécanique des fluides
FONTAINE Jean-Marc	Mathématiques Pures
GAUTHIER Yves	Sciences Biologiques
GAUTRON René	Chimie
GIDON Paul	Géologie et Minéralogie
GLENAT René	Chimie organique
GROULADE Joseph	Biochimie médicale
HACQUES Gérard	Calcul numérique
HOLLARD Daniel	Hématologie
HUGONOT Robert	Hygiène et Médecine préventive
IDELMAN Simon	Physiologie animale
JOLY Jean-René	Mathématiques Pures
JULLIEN Pierre	Mathématiques Appliquées
Mme KAHANE Josette	Physique
MM. KRAKOWIAK Sacha	Mathématiques Appliquées
KUHN Gérard	Physique (IUT A)
LE ROY Philippe	Mécanique (IUT A)
LUU DUC Cuong	Chimie organique

MM. MAYNARD Roger	Physique du solide
Mme MINIER Colette	Physique (IUT A)
MM. PELMONT Jean	Biochimie
PERRIAUX Jean-Jacques	Géologie et Minéralogie
PFISTER Jean-Claude	Physique du solide
Mlle PIERY Yvette	Physiologie animale
MM. RAYNAUD Hervé	M.I.A.G.
REBECQ Jacques	Biologie (CUS)
REYMOND Jean-Charles	Chirurgie générale
RICHARD Lucien	Biologie végétale
Mme RINAUDO Marguerite	Chimie macromoléculaire
MM. ROBERT André	Chimie papetière
SARRAZIN Roger	Anatomie et chirurgie
SARROT-REYNAULD Jean	Géologie
SIROT Louis	Chirurgie générale
Mme SOUTIF Jeanne	Physique générale
MM. STREGLITZ Paul	Anesthésiologie
VIALON Pierre	Géologie
VAN CUTSEM Bernard	Mathématiques Appliquées

MATTRES DE CONFERENCES ET MATTRES DE CONFERENCES AGREGES

MM. AMBLARD Pierre	Dermatologie
ARMAND Gilbert	Géographie
ARMAND Yves	Chimie (IUT A)
BACHELOT Yvan	Endocrinologie
BARGE Michel	Neuro-chirurgie
BARJOLLE Michel	M.I.A.G.
BEGUIN Claude	Chimie organique
Mme BERIEL Hélène	Pharmacodynamie
MM. BOST Michel	Pédiatrie
BOUCHARLAT Jacques	Psychiatrie adultes
Mme BOUCHE Liane	Mathématiques (CUS)
MM. BRODEAU François	Mathématiques (IUT B)
CHAMBAZ Edmond	Biochimie médicale
CHAMPETIER Jean	Anatomie et organogénèse
CHARDON Michel	Géographie
CHERADAME Hervé	Chimie papetière
CHIAVERINA Jean	Biologie appliquée (EFP)
CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
CORDONNIER Daniel	Néphrologie
COULOMB Max	Radiologie
CROUZET Guy	Radiologie
CYROT Michel	Physique du solide
DELOBEL Claude	M.I.A.G.
DENIS Bernard	Cardiologie
DOUCE Roland	Physiologie végétale
DUSSAUD René	Mathématiques (CUS)
Mme ETERRADOSSI Jacqueline	Physiologie
MM. FAURE Jacques	Médecine légale
FAURE Gilbert	Urologie
GAUTIER Robert	Chirurgie générale
GENSAC Pierre	Botanique
GIDON Maurice	Géologie
GROS Yves	Physique (IUT A)

MM. GUITTON Jacques	Chimie
HICTER Pierre	Chimie
IVANES Marcel	Electricité
JALBERT Pierre	Histologie
JUNIEN-LAVILLAVROY Claude	O.R.L.
KOLODIE Lucien	Hématologie
LE NOC Pierre	Bactériologie-virologie
LEROY Philippe	IUT A
MACHE Régis	Physiologie végétale
MAGNIN Robert	Hygiène et médecine préventive
MALLION Jean-Michel	Médecine du travail
MARECHAL Jean	Mécanique (IUT A)
MARTIN-BOUYER Michel	Chimie (CUS)
MICHOULIER Jean	Physique (IUT A)
NEGRE Robert	Mécanique (IUT A)
NEMOZ Alain	Thermodynamique
NOUGARET Marcel	Automatique (IUT A)
PARAMELLE Bernard	Pneumologie
PECCOUD François	Analyse (IUT B)
PEFFEN René	Métallurgie (IUT A)
PERRET Jean	Neurologie
PERRIER Guy	Géophysique - Glaciologie
PHELIP Xavier	Rhumatologie
RACHAIL Michel	Médecine interne
RACINET Claude	Gynécologie et obstétrique
RAMBAUD André	Hygiène et hydrologie
RAMBAUD Pierre	Pédiatrie
Mme RENAUDET Jacqueline	Bactériologie
MM. ROBERT Jean-Bernard	Chimie Physique
ROMIER Guy	Mathématiques (IUT B)
SHOM Jean-Claude	Chimie générale
STOEBNER Pierre	Anatomie pathologique
VROUSOS Constantin	Radiologie

MAITRE DE CONFERENCES ASSOCIES

M. COLE Antony

Sciences nucléaires

Fait à SAINT MARTIN D'HERES, AVRIL 1976.



INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M. Philippe TRAYNARD

Vice-Président : M. Pierre-Jean LAURENT

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BLOCH Daniel	Physique du solide
BONNETAIN Lucien	Chimie Minérale
BONNIER Etienne	Electrochimie et Electrometallurgie
BOUDOURIS Georges	Radioélectricité
BRISSONNEAU Pierre	Physique du solide
BUYLE-BODIN Maurice	Electronique
COUMES André	Radioélectricité
DURAND Francis	Métallurgie
FELICI Noël	Electrostatique
FOULARD Claude	Automatique
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie-Physique
PAUTHENET René	Physique du solide
PERRET René	Servomécanismes
POLOUJADOFF Michel	Electrotechnique
SILBER Robert	Mécanique des Fluides

PROFESSEUR ASSOCIE

M. ROUXEL Roland	Automatique
------------------	-------------

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuel	Electronique
BOUVARD Maurice	Génie Mécanique
COHEN Joseph	Electrotechnique
LACOUME Jean-Louis	Géophysique
LANCIA Roland	Electronique
ROBERT François	Analyse numérique
VEILLON Gérard	Informatique Fondamentale et Appliquée
ZADWORNY François	Electronique

MATTRES DE CONFERENCES

MM. ANCEAU François	Mathématiques Appliquées
CHARTIER Germain	Electronique
GUYOT Pierre	Chimie Minérale
IVANES Marcel	Electrotechnique
JOUBERT Jean-Claude	Physique du solide
MORET Roger	Electrotechnique Nucléaire
PIERRARD Jean-Marie	Mécanique
SABONNADIÈRE Jean-Claude	Informatique Fondamentale et Appliquée
Mme SAUCIER Gabrièle	Informatique Fondamentale et Appliquée

MATRE DE CONFERENCES ASSOCIE

M. LANDAU Ioan

Automatique

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

MM. FRUCHART Robert

Directeur de Recherche

ANSARA Ibrahim

Maître de Recherche

CARRE René

Maître de Recherche

DRIOLE Jean

Maître de Recherche

MATHIEU Jean-Claude

Maître de Recherche

MUNIER Jacques

Maître de Recherche



*Je tiens à remercier Monsieur GASTINEL, Président du Jury, qui a constamment encouragé la participation active du Centre de Calcul au développement des réseaux d'ordinateurs, ainsi que Messieurs les membres du Jury et plus particulièrement Monsieur BOLLIET et Monsieur ZIMMERMANN.*

*Je tiens à préciser que ce travail s'insère dans un ensemble d'activités réalisées sous le contrôle de notre regretté collègue Jean du MASLE qui m'a initialement conseillé d'entreprendre ce travail et qui m'a constamment guidé et encouragé.*

*Je remercie également les chercheurs avec qui j'ai travaillé, plus particulièrement Messieurs ANSART, QUINT et DANG, ainsi que le personnel du laboratoire.*

*Je remercie aussi les opérateurs qui mettent chaque jour à la disposition des utilisateurs les services que nous avons développés, ainsi que la secrétaire qui a effectué la frappe et le service de reprographie.*



# S O M M A I R E

---

	Page
1. INTRODUCTION	1
Historique	
Choix du mode d'implémentation. Justifications.	
2. MISE EN PLACE D'UN SOUS-SYSTÈME DE MULTIPROGRAM- MATION DU SYSTÈME ØS-MVT	12
2.1 Chargement et initialisation du système TELCØM.	13
2.2 Le commutateur de processus.	16
2.3 La gestion des opérations d'entrée-sortie.	17
2.4 SVC affectée au traitement des services devant se dérouler en mode maître.	22
2.5 Gestion d'horloges au niveau des processus	23
3. LA GESTION DES RÉVEILS	24
3.1 Etude d'un cas simple : une seule demande de réveil.	26
3.2 Etude du cas général de prise en considération d'une demande de réveil.	31
4. COMPATIBILITÉ DES VERSIONS CP ET ØS DU SYSTÈME TELCØM. LES SERVICES TRANSPORTABLES.	45
4.1 La procédure de télétransmission.	46
4.2 La station de transport CYCLADES.	48
4.3 Le concentrateur de terminaux.	48
4.3.1 La console opérateur	49
4.3.2 Les clients conversationnels du réseau Cyclades.	51

	Page
5. LE SERVEUR DE TRAITEMENTS PAR LOTS	53
5.1 Mécanisme d'accès aux fichiers séquentiels du système OS.	57
5.1.1 La méthode d'accès (module DRDSK)	57
5.1.2 Les tâches de travail.	61
5.1.3 Les processus de traitement (DSK)	63
5.1.4 Les commandes affectées à la gestion des fichiers.	65
5.1.5 Considérations générales sur le mécanisme implémenté.	67
5.2 Les moyens de communication avec le système ASP.	67
5.2.1 La soumission des travaux.	67
5.2.2 La réception des listes.	72
5.2.3 Les pseudo consoles opérateur ASP.	81
5.2.3.1 Support de pseudo consoles par le système ASP.	81
a) Lecture des commandes opérateur	82
b) Envoi de messages à l'opérateur	83
5.2.3.2 Comportement d'un processus TELCØM utilisateur d'une pseudo console ASP.	84
a) Attention sur une pseudo console	85
b) "Device end" sur une pseudo console.	85
5.3 Gestion d'une file d'attente réseau sur une unité à accès direct.	88
5.3.1 Justification et logique de conception	88
5.3.2 Structure des informations cont nues dans la file d'attente.	90
5.3.2.1 La table d'allocation des pistes	90
5.3.2.2 Les blocs descripteurs de la file d'attente.	91
5.3.2.3 Les fichiers dans la file d'attente.	92
5.3.2.4 Exemple et schéma	93

	Page
5.3.3 Les requêtes des utilisateurs et leur traitement.	94
5.3.3.1 Les requêtes d'ouverture de fichier.	94
a) Ouverture d'un fichier pour lecture de cartes.	95
b) Ouverture d'un fichier pour écriture de cartes.	95
c) Ouverture d'un fichier pour lecture de lignes.	95
d) Ouverture d'un fichier pour écriture de lignes.	96
5.3.3.2 Requête de lecture.	96
5.3.3.3 Requête d'écriture.	96
5.3.3.4 Les requêtes de fermeture	97
a) Requête $\text{\textcircled{X}}\text{CL}\text{\textcircled{O}}\text{SE}$	97
b) Requête $\text{\textcircled{X}}\text{AB}\text{\textcircled{O}}\text{RT}$	98
5.4 Les abonnés serveurs	99
5.4.1 Le processus de contact (ABC)	99
a) ST1	99
b) ST2	100
5.4.2 Les processus RMCØNS	100
5.4.3 Les processus IMPR	102
5.4.4 Les processus LECR	104
6. UN CONCENTRATEUR DE TERMINAUX LOURDS SOUS TELCØM	106
6.1 Les moyens de communication avec le terminal lourd (gestion du 'multi-leaving')	107
6.1.1 Les processus nécessaires à la gestion du 'multi-leaving'.	107
6.1.2 Les listes nécessaires aux échanges d'information entre les processus chargés du 'multi-leaving'.	108



	Page	
6.1.3	Traitement des informations reçues sur la ligne.	110
6.1.4	Emission d'informations vers le terminal lourd.	111
6.2	La gestion des appareils virtuels	115
6.2.1	La gestion des appareils de type console	118
6.2.1.1	Traitement des demandes de connexion	118
6.2.1.2	Exploitation des voies virtuelles associées aux consoles.	120
6.2.2	La gestion des appareils de type lecteur de cartes.	125
6.2.3	La gestion des appareils de type imprimante.	126
6.2.3.1	Retrait et impression des listes de la file d'attente du système TELCØM.	127
6.2.3.2	Réception des listes des serveurs connectés.	129
	a) Réception des listes du serveur local.	129
	b) Réception des listes des serveurs distants.	129
	Schéma général	132
	7. CONCLUSION	133
	ANNEXE 1	136
	Macro-instructions disponibles sous la version ØS du système SYNCØP, associées aux services devant se dérouler en mode maître.	

# PLAN GENERAL

<><><>

## 1. INTRODUCTION

Historique

Choix du mode d'implémentation

## 2. MISE EN PLACE D'UN SOUS-SYSTÈME DE MULTIPROGRAMMATION

(Telcom puis Syncop)

## 3. LA GESTION DES RÉVEILS

## 4. MISE EN PLACE DE LA PROCÉDURE DE LIGNE, DE LA STATION DE TRANSPORT ET DU CONCENTRATEUR DE TERMINAUX

## 5. LE SERVEUR DE TRAITEMENT PAR LOTS

- Mécanisme d'accès aux fichiers ØS utilisé pour les premiers tests
- Moyens de communications avec le système ASP
- Gestion d'une file d'attente réseau.
- Les abonnés serveurs

## 6. UN CONCENTRATEUR DE TERMINAUX LOURDS

## 7. CONCLUSION

ANNEXES.



## 1. INTRODUCTION

Le Centre de Calcul de l'Université de Grenoble possède un ordinateur IBM 360/67 depuis 1968. Il est mis à la disposition des utilisateurs sous le contrôle de deux systèmes d'exploitation distincts : un système de temps partagé (CP) qui permet le support en mode conversationnel d'un nombre d'utilisateurs simultanés pouvant aller jusqu'à 50 avec des temps de réponse très satisfaisants et un système de traitements par lots qui permet l'exécution d'une moyenne de 5 travaux en parallèles entre 12 h et 15 h et la nuit.

On s'intéresse ici au travail qui a permis de mettre le système de traitements par lots à la disposition d'autres Centres de Calcul par l'intermédiaire du réseau Cyclades.

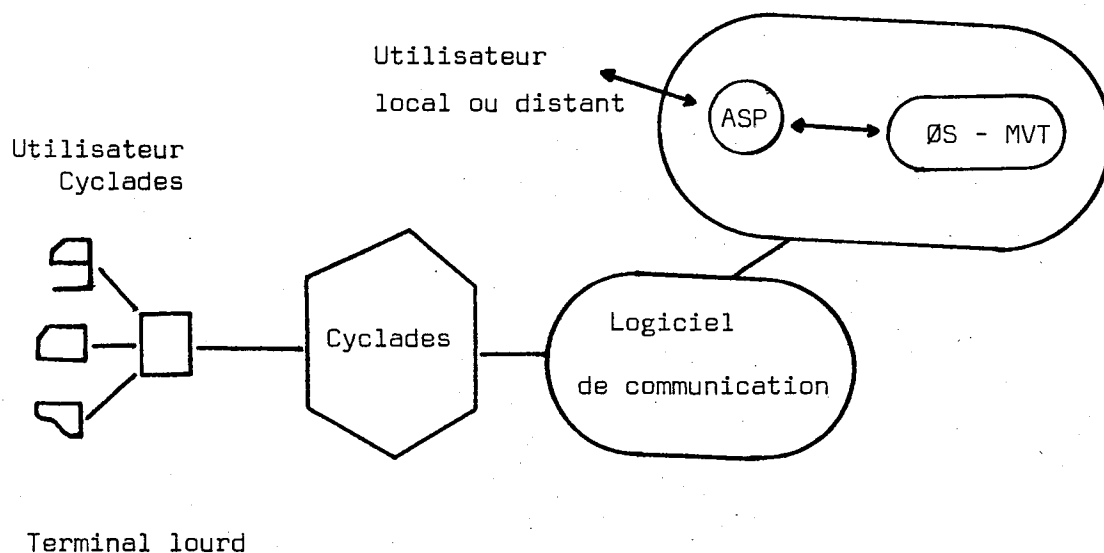
Le traitement par lots se fait sous le contrôle du système de multiprogrammation ØS-MVT (Operating System - Multiprogramming with a variable number of tasks) qui supporte dans l'une de ses régions le système ASP (Attached Support Processor) en mode local.

Le rôle du système ASP est d'effectuer une gestion programmée de l'activation des travaux destinés au système ØS-MVT, plus précisément :

- Demander le montage des volumes non résidents nécessaires à un travail donné avant de le soumettre au système ØS.
- Effectuer la lecture des travaux et la sortie des listes soit sur des unités locales connectées directement au 360, soit sur les unités d'un terminal lourd relié par liaison téléphonique au 360.
- Effectuer la gestion d'une file d'attente sur une unité à accès direct pour les travaux en attente d'exécution et les listes en attente d'impression.
- Effectuer un support programmé de la console opérateur du système ØS et gérer un ensemble de consoles opérateurs permettant d'accéder aux services fournis par le système ASP.

On voit ainsi que le système ASP joue un rôle de communications entre l'utilisateur local ou distant (l'utilisateur désignant ici aussi bien le programmeur que l'opérateur) et le système d'exploitation.

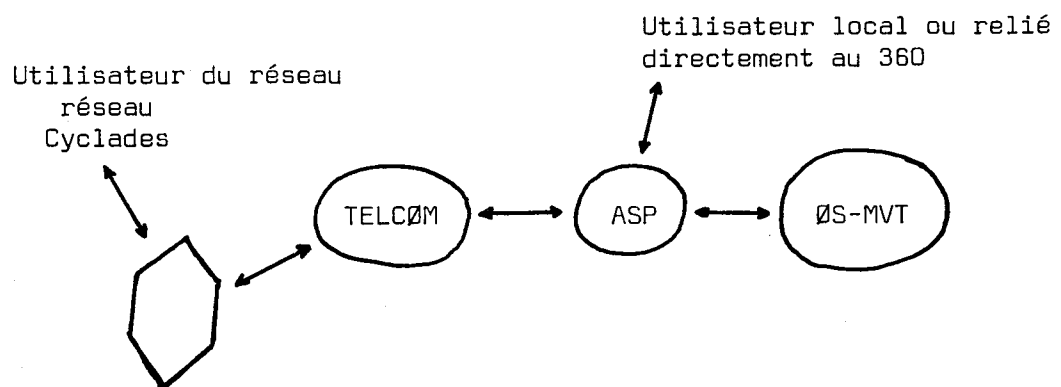
On peut considérer que le travail réalisé pour mettre les services du système de multiprogrammation à la disposition des utilisateurs du réseau Cyclades est la mise en place d'un moyen de communications entre l'utilisateur Cyclades et le système d'exploitation (ensemble ASP et ØS/MVT) analogue au rôle de communication entre l'utilisateur et le système ØS-MVT joué par le système ASP.



La mise en place de ce logiciel de communication pouvait se faire soit en insérant au système ASP des programmes réalisant ces fonctions de communication soit en implémentant un autre sous-système de multiprogrammation dans une région du système ØS-MVT et en établissant des moyens de communications entre ce système et ASP.

C'est la deuxième solution qui a été choisie pour plusieurs raisons :

- On voulait minimiser les modifications apportées au système d'exploitation livré par le constructeur. La solution choisie permet en effet d'implémenter la plupart du logiciel nouveau dans une région associée à un travail d'utilisateur standard.
- On voulait réaliser un logiciel qui ne soit pas lié de façon trop étroite au système ASP, ceci dans le but de faciliter la connexion éventuelle au réseau Cyclades d'un autre Centre de Calcul utilisant le système ØS et également dans un esprit de remplacement du système ASP (trop encombrant pour notre installation) par des tâches ØS réalisant les fonctions du système ASP dont on ne saurait se priver. Il s'agit essentiellement de la gestion du montage des volumes (disques et bandes) pour les travaux des utilisateurs et du support des terminaux lourds distants. Ce travail, réalisé à un niveau expérimental n'a pas atteint un niveau de développement suffisant pour sa mise en exploitation.
- Un logiciel analogue développé sur une machine virtuelle du système CP67 devient en grande partie transportable d'un système vers l'autre dans la mesure où l'on s'impose de réaliser un sous-système de multiprogrammation respectant de façon assez rigoureuse les mêmes conventions de programmation que le système utilisé sur machine virtuelle.



Ce choix nous a donc emmené à implémenter sous ØS, comme un travail utilisateur, un sous-système de multiprogrammation, appelé TELCØM, qui gère des processus susceptibles de dérouler des programmes en parallèle et capables de se synchroniser entre eux.

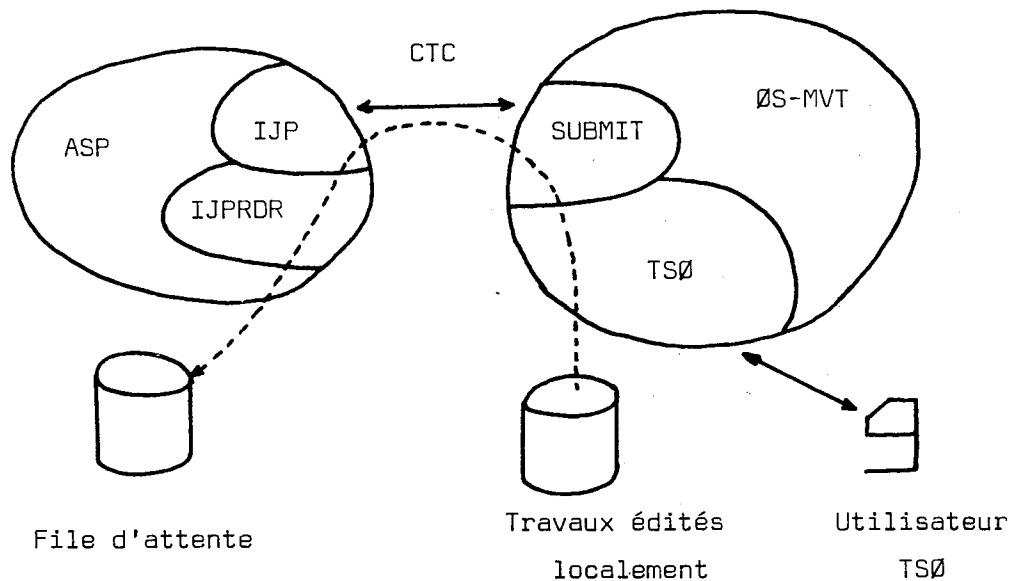
Le système TELCØM a été développé initialement sur une machine virtuelle gérée par CP dans le but de supporter les terminaux lourds connectés au 360. Il a été repris pour implémenter le logiciel nécessaire à la connexion du système CP au réseau Cyclades et c'est cette deuxième version du système TELCØM qui a été reprise pour une implémentation sous le contrôle d'une tâche OS.

Dès que le système TELCØM a été mis en place sous le contrôle d'une tâche ØS et testé, une grosse partie des modules nécessaires à la connexion du système d'exploitation au réseau Cyclades a pu être transportée pratiquement sans modifications de la version du système TELCØM développée sur machine virtuelle à celle implémentée sous ØS. Il s'agit des modules chargés de la gestion de la procédure de ligne reliant le 360 au réseau de télétransmission de Cyclades, de ceux permettant le support des protocoles de communications entre les calculateurs participants du réseau (appelés la "station de transport") et d'un concentrateur de terminaux. Le logiciel ainsi transporté a pu s'améliorer par des mises au point successives sur les deux systèmes qui le supportaient, ce qui impliquait que toute modification de ce logiciel effectuée sur l'un des deux systèmes soit reportée sur l'autre.

Le système TELCØM ainsi complété par le logiciel transporté depuis CP pouvait communiquer avec le réseau grâce à la procédure de ligne et à la station de transport mais n'avait pas de liens avec le système d'exploitation. Il a alors fallu implémenter un serveur de traitements par lots et pour cela établir des moyens de communications entre le système ASP et le système TELCØM. On a pu à cet effet utiliser le logiciel développé dans le système ASP pour permettre aux systèmes ØS qu'il supporte d'utiliser l'option de temps partagé TSØ (Time Sharing Option).

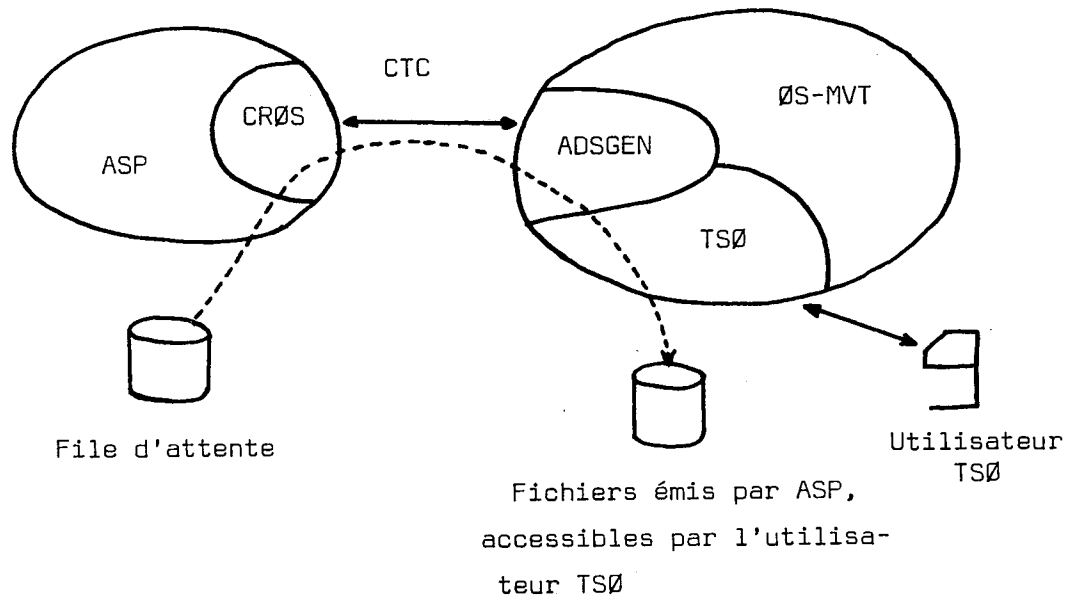
Ce logiciel permet :

- à une tâche OS d'émettre un flot de travaux vers le système ASP qui le reçoit à l'aide des modules IJP (Internal Job Processing) et le soumet au DSP (Dynamic Support Program) spécifié par l'émetteur (IJPRDR dans le cas où ce flot de travaux doit être interprété par ASP et rangé dans la file d'attente de la même façon que les travaux lus localement). Ce service est celui qui permet à l'utilisateur conversationnel de TSØ de soumettre, depuis son terminal, au système ASP des travaux qu'il a édités localement.



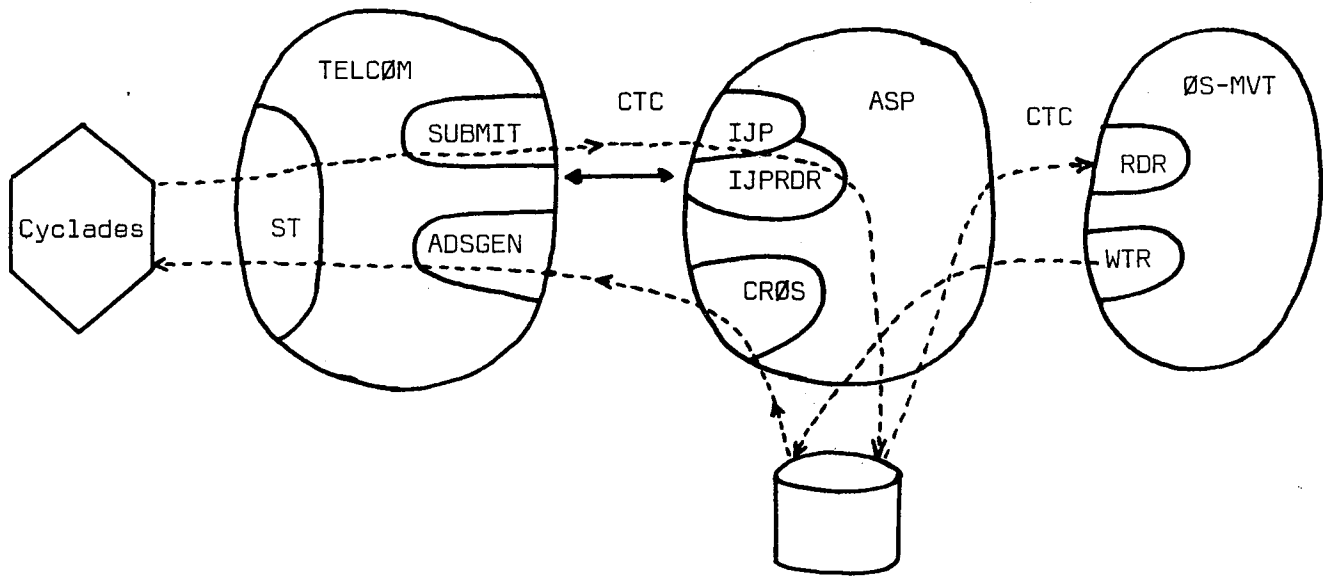
- au système ASP de retourner à une tâche OS un fichier produit par l'exécution d'un travail. Ce service permet à l'utilisateur de TSØ de récupérer sous forme de fichiers éditables certains résultats de travaux qu'il a précédemment soumis à l'exécution, ou bien les résultats de travaux soumis par d'autres utilisateurs qui ont spécifié par des cartes de contrôle ASP le nom de l'utilisateur TSØ destinataire.





Ces deux mécanismes de communication entre ASP et OS-MVT utilisent l'adaptateur de canal à canal (CTC) simulé par logiciel ou réel suivant que ces deux systèmes se trouvent sur la même machine ou sur deux machines distinctes.

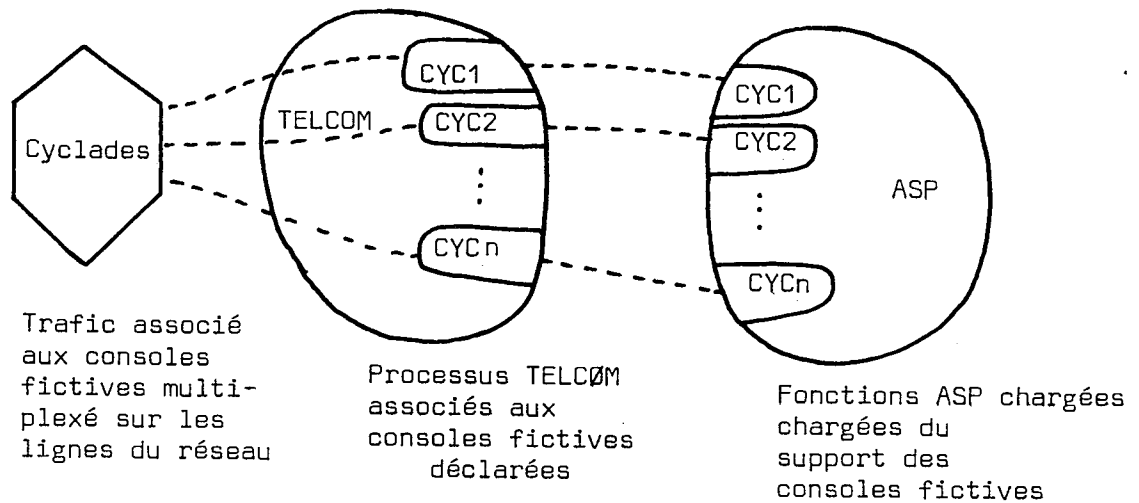
Ils ont été repris pour permettre au système TELCØM de soumettre au système ASP les travaux reçus du réseau et de retourner les listes de ces travaux vers celui qui les a émis, ainsi que pour envoyer sur le réseau des listes produites par l'exécution de travaux ne provenant pas forcément du réseau.



File d'attente ASP

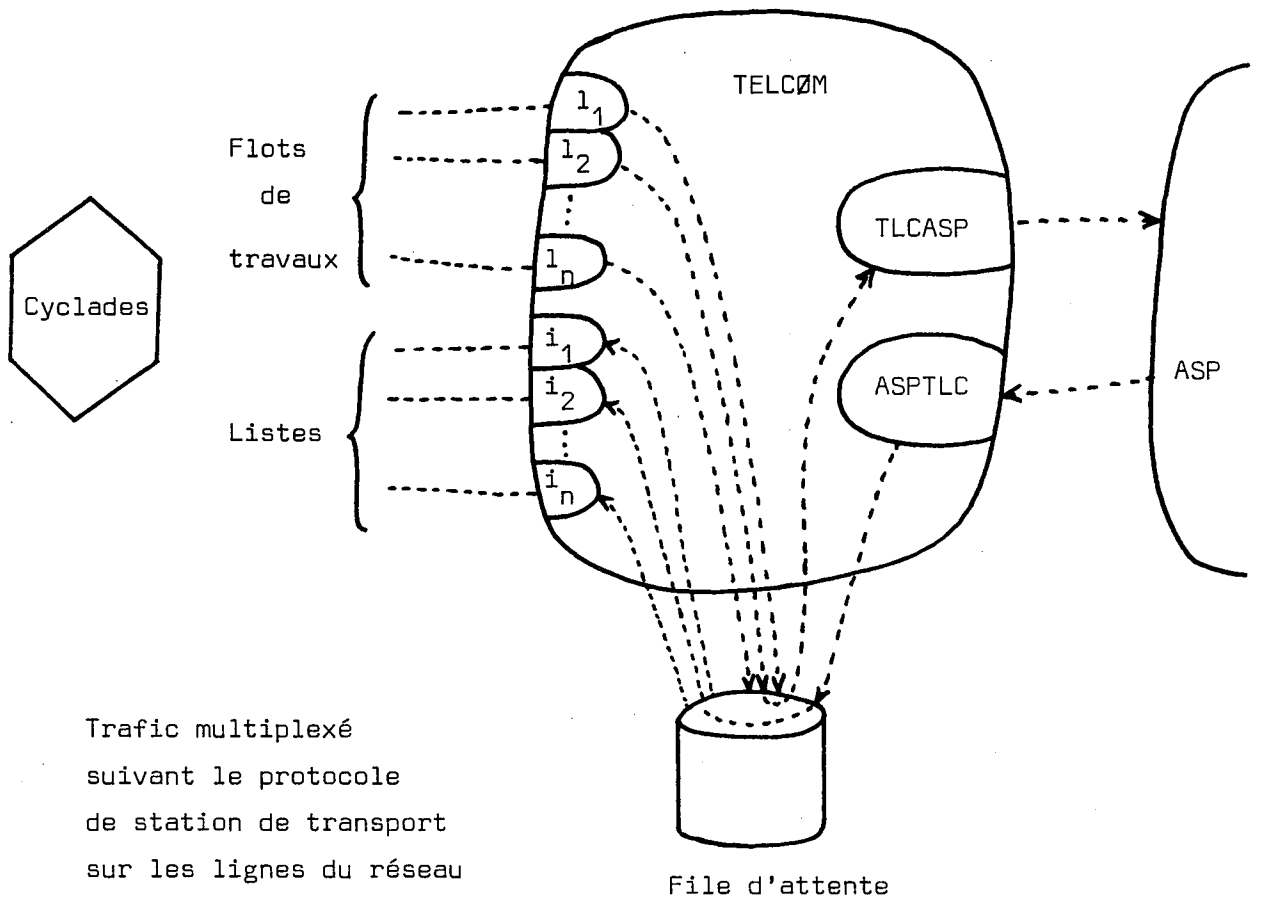
Un autre mécanisme de communications entre ASP et TELCØM a du être implémenté pour permettre à des processus de TELCØM de se comporter comme des consoles opérateur du système ASP, ceci dans le but de mettre à la disposition des terminaux lourds du réseau Cyclades l'accès aux commandes opérateur du système ASP.

Il a fallu pour cela modifier quelques modules du système ASP pour déclarer des consoles fictives pour lesquelles des programmes canaux sont construits comme pour des consoles réelles mais le lancement de l'opération d'entrée-sortie est remplacé par l'exécution d'un service particulier appelé  $\$PEXCP$  qui permet, à l'aide d'une SVC, de réveiller le processus TELCØM éventuellement en attente de traitement de l'opération d'entrée-sortie lancée sur la console fictive à laquelle il est associé. Ce processus TELCØM est lui-même capable de simuler la fin normale de l'opération d'entrée-sortie dès qu'il a récupéré les informations qui le concerne.



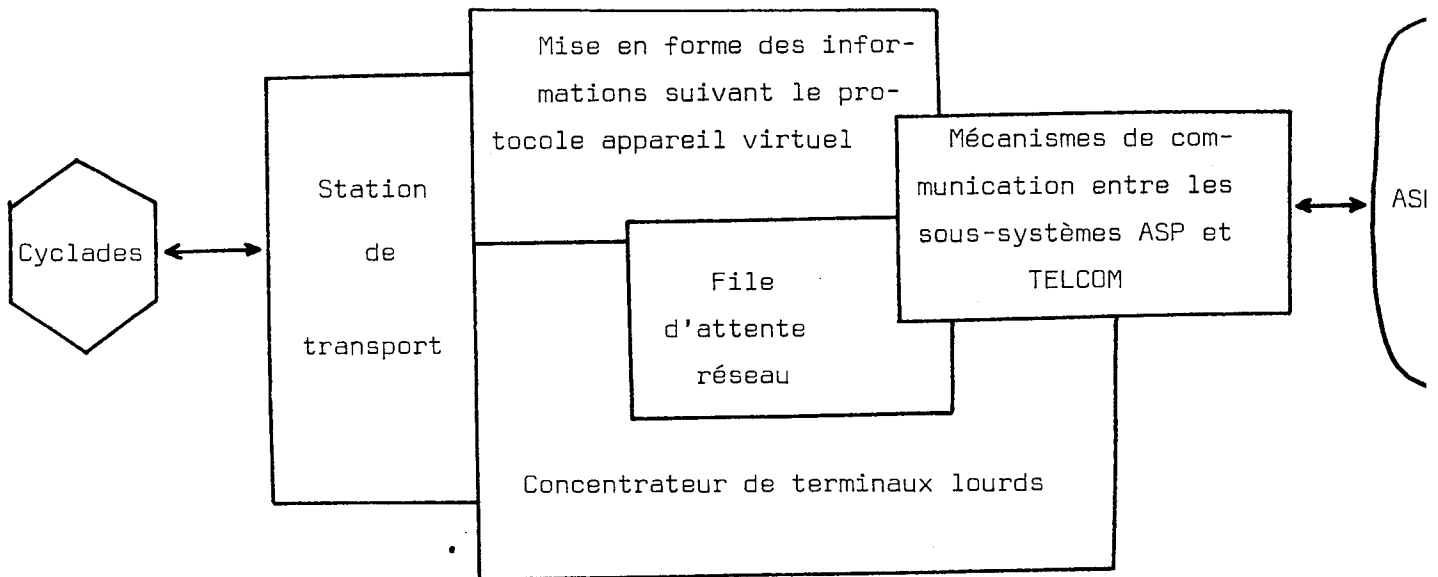
Le nombre de voies de communications entre ASP et TELCØM n'est pas limité par la logique du mécanisme en ce qui concerne le support des consoles opérateur fictives mais pour les lecteurs et les imprimantes fictives (soumission des flots de travaux et récupération des listes) une seule voie séquentielle est disponible dans chaque sens. La solution qui consisterait à augmenter ce nombre de voies séquentielles serait coûteuse en programmation et en taille mémoire au niveau du système ASP. Il faut pourtant qu'un serveur de traitements par lots soit capable d'accepter et de satisfaire les demandes des clients qui désirent se connecter jusqu'à un nombre d'utilisateurs simultanés limité par les moyens physiques mis en jeu et non par la logique de la conception.

Il a fallu pour cela développer une file d'attente réseau sur une unité d'entrée-sortie à accès direct. Cette file d'attente contient deux types de fichiers : des flots de travaux et des fichiers de type listes. Chaque fichier est affecté à un "usager" du réseau et il est possible de mettre n processus TELCØM associés à n "usagers" distincts du réseau simultanément en lecture de fichiers de type listes ainsi que m autres processus associés à m "usagers" distincts (non forcément différents des n "usagers" en lecture) en écriture de flots de travaux. Deux processus TELCØM spécialisés peuvent alimenter la file d'attente séquentiellement par des fichiers de type listes et récupérer séquentiellement par fichiers les flots de travaux.



Une fois ces mécanismes de communication entre ASP et TELCOM mis en place, ainsi que le mécanisme de gestion de la file d'attente dont le rôle est celui de multiplexer et démultiplexer les voies d'accès aux informations au niveau des fichiers, il suffit de savoir accéder aux services offerts par la station de transport transportée depuis le système TELCOM développé sur machine virtuelle pour réaliser le multiplexage et le démultiplexage des informations associées aux différents appareils supportés, la liaison entre le 360 et le réseau de télécommunication se faisant par une seule ligne suivant la procédure de transmission BSC (Basic Synchronous Communication).

Les informations à transmettre au moyen de la station de transport doivent avoir un format défini par le "protocole appareil virtuel" pour permettre aux correspondants de les interpréter sans ambiguïté.

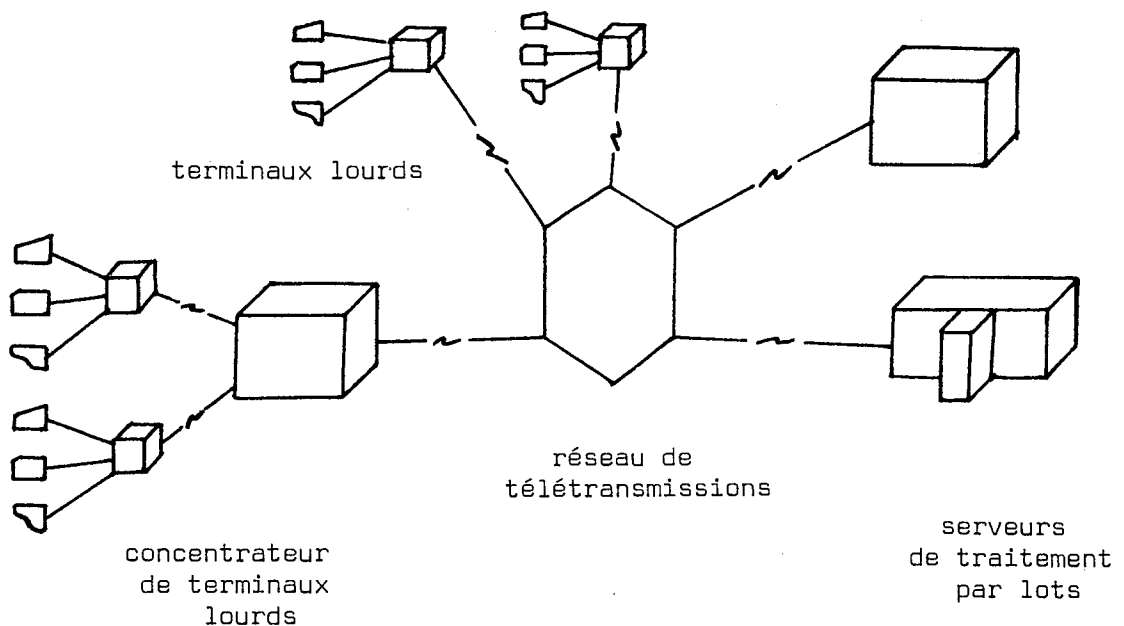


Pour permettre d'effectuer des tests sur le réseau avant que les mécanismes de communication entre ASP et TELCOM soient totalement opérationnels, une gestion de fichiers OS par des processus TELCOM a été développée. Ceci a permis aux clients de traitements par lots du réseau Cyclades, en cours de développement à ce moment là, d'effectuer des essais en accédant à des fichiers du système OS. Le protocole "client-serveur de terminaux" était alors utilisé pour permettre au terminal lourd de choisir le fichier OS à lister sur son imprimante.

Ce qu'apporte un réseau de télétransmission dans le type d'applications auquel on s'est intéressé ici (service de traitement par lots) c'est essentiellement de permettre à un terminal lourd d'obtenir non plus l'accès à une seule machine qui offre des services de traitement par lots par une liaison directe à cette machine mais l'accès à un ensemble de serveurs du réseau par une liaison à un noeud du réseau de télétransmission.

Dans cet esprit, on a développé un concentrateur de terminaux lourds sous le système TELCOM pour permettre aux terminaux lourds du 360 d'accéder au réseau Cyclades à travers ce concentrateur et leur permettre de se comporter comme des clients de tout serveur de traitements par lots du réseau. Ce travail a été réalisé en tenant compte des logiciels existants sur les terminaux lourds supportés, ce qui a permis de ne modifier ni les logiciels, ni les modes de connexion de ces machines. On a pu également profiter de la file d'attente réseau développée sous TELCOM pour permettre à chacun de ces terminaux lourds de se comporter simultanément comme clients de plusieurs serveurs. Les listes reçues des différents serveurs sont stockées dans la file d'attente et émises vers l'imprimante du client concerné dès la connexion du terminal lourd correspondant et la disponibilité de son imprimante.

Cette application semble mettre encore plus en évidence l'intérêt d'un réseau de télétransmission puisque l'on peut ainsi sur l'imprimante unique d'un terminal lourd relié à une seule machine obtenir des listes en provenance de différentes machines au fur et à mesure de la disponibilité des fichiers correspondants.



## 2. MISE EN PLACE D'UN SOUS-SYSTÈME DE MULTIPROGRAMMATION DU SYSTÈME ØS-MVT.

Il est possible, pour développer une application de multiprogrammation comme un travail utilisateur du système ØS, d'utiliser les facilités de multiprogrammation offertes par le système ØS : création de tâches OS à l'aide du service "ATTACH". Mais la mise en place d'un contrôleur de processus élémentaire dans une tâche ØS qui n'offre que les services minimum requis pour l'application considérée est moins onéreuse en temps CPU consacré à la commutation des processus et aux services système, et c'est la solution que nous avons choisie. De plus, dans le but de tirer profit du développement d'une application semblable dans une machine virtuelle gérée par le système CP ou s'est imposé de réaliser ce sous-système de multiprogrammation dans une tâche OS de telle façon que tout programme se déroulant sous le contrôle d'un processus de ce système soit transportable sans modification de (ou vers) le système élémentaire de multiprogrammation développé sur machine virtuelle : TELCOM.

Pour cela on a établi des analogies entre les services de programmation offerts par le système CP à l'utilisateur d'une machine virtuelle (services très voisins de ceux offerts par le matériel d'une machine réelle) et les services de programmation offerts par le système ØS au programmeur sous le contrôle d'une tâche ØS : ces analogies sont essentiellement les suivantes :

- l'horloge en mémoire basse (adresse 80) d'une machine virtuelle offre des services analogues à ceux offerts par la gestion des pseudo horloges au niveau des tâches ØS (services STIMER et TTIMER).
- l'interruption externe qui se réalise lorsque l'horloge en mémoire basse atteint la valeur 0 est comparable à l'activation de la "routine d'exit" spécifiée à l'appel de "STIMER" qui prend le contrôle lorsque la pseudo horloge associée à la tâche concernée atteint la valeur 0.

- les interruptions d'entrée-sortie reflétées à la machine virtuelle par le système CP trouvent leur équivalent sous ØS dans l'activation par le superviseur d'entrée-sortie du système ØS des "normal end appendage" et "abnormal end appendage" dont un et un seul se déroule en fin de toute opération d'entrée-sortie lancée par le superviseur.  
 Cette analogie ne peut être respectée que si l'on impose à tous les programmes se déroulant sous le système TELCØM sous ØS de ne lancer des opérations d'entrée-sortie que par appel au superviseur d'entrée-sortie du système ØS.
- le chargement d'un PSW avec le bit "wait" à 1 dans la machine virtuelle est le moyen d'informer CP que le système qui travaille dans la machine virtuelle n'a plus de code à dérouler de la même façon que le service "WAIT" du système ØS permet de signaler au système que le programme qui travaille sous le contrôle de la tâche qui effectue l'appel à ce service n'a plus de code à dérouler.

On va s'attacher à présenter les particularités de l'implémentation du système TELCØM sous ØS plutôt que la logique du système lui-même qui est spécifiée par la thèse de Z.PAPACHRISTØDØULØU [P.1] pour sa version initiale, par celle de J.P.ANSART [A.1] qui présente les modifications apportées pour le support d'une station de transport du réseau CYCLADES et enfin la version suivante qui a pour nom SYNCØP est documentée par [S.1], [S.2] et [S.3].

## 2.1 Chargement et initialisation du système TELCØM

Alors que la version sur machine virtuelle nécessitait l'écriture d'un chargeur, qui a été prévu initialement sur cartes (lecteur de cartes virtuel) puis sur disque (mini-disque CP), pour la version sous ØS on a pu utiliser l'éditeur de liens du système pour regrouper en un seul module



chargeable l'ensemble des programmes objets produits par les assemblages des différents modules sources constituant le système. Le module ainsi produit est chargé comme un programme utilisateur et son point d'entrée est le début du module INIT.

Le module INIT commence par s'assurer que le système TELCØM n'est pas déjà actif sous le même système ØS à l'aide des services ENQ et DEQ, ceci pour éviter des conflits d'utilisation d'une SVC propre à TELCØM ainsi que des unités d'entrée-sortie gérées par les applications développées sous TELCØM. Puis il active une tâche OS chargée de transmettre les flots de travaux au système ASP par l'intermédiaire de l'adaptateur de canal à canal (la tâche SUBMIT). Ce service de soumission de travaux à ASP ne peut pas se dérouler sous le contrôle d'un processus TELCØM car il bloquerait les autres processus pendant l'exécution des services ØPEN et WTØR qui mettent la tâche appelante en attente pendant leur exécution.

Le module INIT est également chargé d'initialiser la mémoire libre pour la gestion de l'espace mémoire au niveau des processus du système TELCØM (services ØGETMN et ØFREEMN). Alors que dans la version sur machine virtuelle la mémoire initialement considérée comme libre est la totalité de la mémoire de la machine virtuelle moins la mémoire occupée par les modules du système, dans la version ØS c'est la totalité de la mémoire disponible dans la région affectée au système TELCØM après chargement des modules du système moins 4K octets pour permettre le chargement de la méthode d'accès et des "buffers" nécessaires pour sortir un "dump" en cas de fin anormale. Pour trouver la taille et l'emplacement de cette zone mémoire initialement libre, il faut faire une demande conditionnelle d'acquisition de mémoire au système ØS avec des valeurs de taille minimale et maximales telles que la taille disponible soit dans cet intervalle, le système fournit en retour l'adresse et la taille de la plus grande zone mémoire libre dans l'intervalle requis, il suffit alors de retrancher 4K, de libérer la mémoire acquise puis de faire une nouvelle demande d'acquisition de mémoire au système ØS (GETMAIN) pour une taille correspondant à la valeur ainsi obtenue.

Le module INIT doit exécuter une instruction sous le contrôle d'un PSW de clé de protection nulle pour remplacer le PSW en mémoire basse chargé à toute interruption SVC (SVC NEW PSW) par un PSW propre à TELCØM. Le PSW qui se trouvait préalablement à cet emplacement est sauvegardé dans une zone de données du système et sera chargé par le programme de filtre des SVC adressé par le nouveau PSW activé sur interruption SVC dans le cas où l'instruction SVC qui a provoqué l'interruption n'a pas le numéro de code attendu par TELCØM. Ce mécanisme demande d'exécuter une instruction (MVC) et une seule sous le mode non standard du programmeur (clé non nulle et mode programme) à l'aide d'un service propre à l'implémentation du système ØS/MVT (utilisation d'une SVC résidente dans le cas du C.I.C.G.) pour l'initialiser et il permet alors, au coût de l'exécution de 7 instructions supplémentaires pour tout appel SVC, d'exécuter pour le système TELCØM tous les services qui doivent se dérouler en clé 0 et mode système. Les deux octets qui suivent l'instruction SVC sont utilisés pour définir des codes internes au système TELCØM. Ce mécanisme est commode pour développer des services qui doivent se dérouler en mode maître, on peut envisager une fois que ces services sont au point de les inclure au système sous forme d'une ou plusieurs SVC résidentes (Edition de Lien avec le Noyau du Système) ou non (rangement dans la librairie des SVC) selon leur fréquence d'utilisation.

Le module INIT utilise le service STAE (Specify Task Abnormal Exit) du système ØS pour spécifier au système l'adresse d'une séquence d'instructions qui sera exécutée en cas de fin anormale de la tâche sous le contrôle de laquelle s'exécute TELCØM. Le rôle de cette séquence est essentiellement de restaurer le PSW en mémoire basse, activé sur interruption SVC, à la valeur qu'il avait avant l'initialisation de TELCØM. Cette restauration doit aussi être réalisée en cas de fin normale du système TELCØM (arrêt sur commande opérateur) car tout arrêt de TELCØM sans cette restauration provoquerait une fin anormale du système ØS dès que la zone mémoire contenant le filtre SVC serait détruite par un autre utilisateur du système.

En cas de fin normale ou anormale, du système TELCØM il faut également informer les processus ASP chargés de la gestion des pseudo consoles opérateur [F.4] .

Le module INIT est également chargé d'initialiser une pseudo horloge ØS à l'aide de la SVC "STIMER" à sa valeur maximale pour permettre la gestion d'horloges au niveau des processus du système TELCØM [F.5].

Puis il initialise les blocs de contrôle associés aux processus résidents et enfin active le commutateur de processus du système TELCØM à l'aide de la macro instruction ØXCTL (appel\_sans retour).

## 2.2 Le commutateur de processus

Il ne diffère de celui du système TELCØM développé sur machine virtuelle que sur les points suivants :

- la mise en attente du système lorsque aucun processus n'est activable se fait par appel du service "WAIT" du système ØS, au lieu d'un LPSW (load PSW) adressant un PSW avec le bit WAIT à 1.

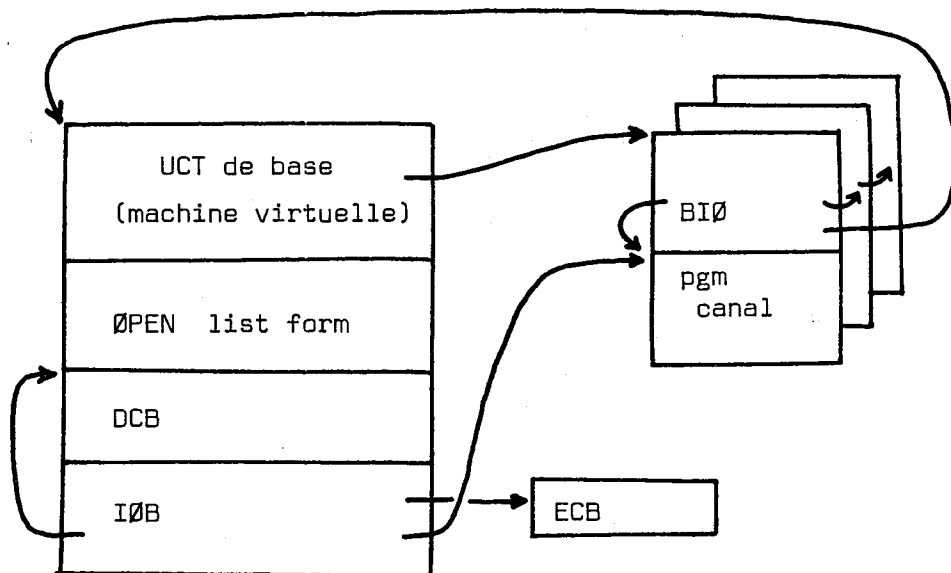
Ce "WAIT" est réalisé sur un ECB spécifique pour lequel un "PØST" sera affecté chaque fois qu'un évènement externe au système TELCØM doit réveiller un processus TELCØM. Dans ce cas le "PØST" ØS doit être accompagné d'un "ØPOST" TELCØM qui rend activable le processus concerné si nécessaire.

- la consultation des blocs de contrôle associés aux processus se déroule masquée et en mode maître à l'aide d'un appel SVC. Quand un processus activable est trouvé, le contrôle lui est donné par un LPSW qui l'active en mode programme, interruptible, et sous la clé de protection associée à la tâche TELCØM.

### 2.3 La gestion des opérations d'entrée-sortie

Toujours par souci de compatibilité avec le système analogue développé sur machine virtuelle et également dans le but de ne pas perturber les autres utilisateurs du système ØS nous avons rajouté aux blocs de contrôle (UCT) associés à chaque unité d'entrée-sortie supportée par le système TELCØM sous CP les informations nécessaires pour travailler au niveau EXCP (Execute Channel Program) sous ØS. Ces informations sont :

- Une expression 'list form' de la macro instruction ØPEN.
- Un DCB pour travailler au niveau EXCP qui contient le "ddname" qui permet de référencer la carte contrôle qui spécifie l'unité d'entrée-sortie effectivement utilisée.
- Un IØB (Input Output Block) qui adresse l'ECB sur lequel le commutateur de processus se met en attente quand aucun processus du système TELCØM n'est activable.



Ceci permet de ne pas modifier les blocs de contrôle (BIØ) gérés par les méthodes d'accès développées dans la version du système sur machine virtuelle.

On peut ainsi transporter les "device routines" associées à des unités d'entrée-sortie spécifiques d'un système vers l'autre avec un minimum de modifications.

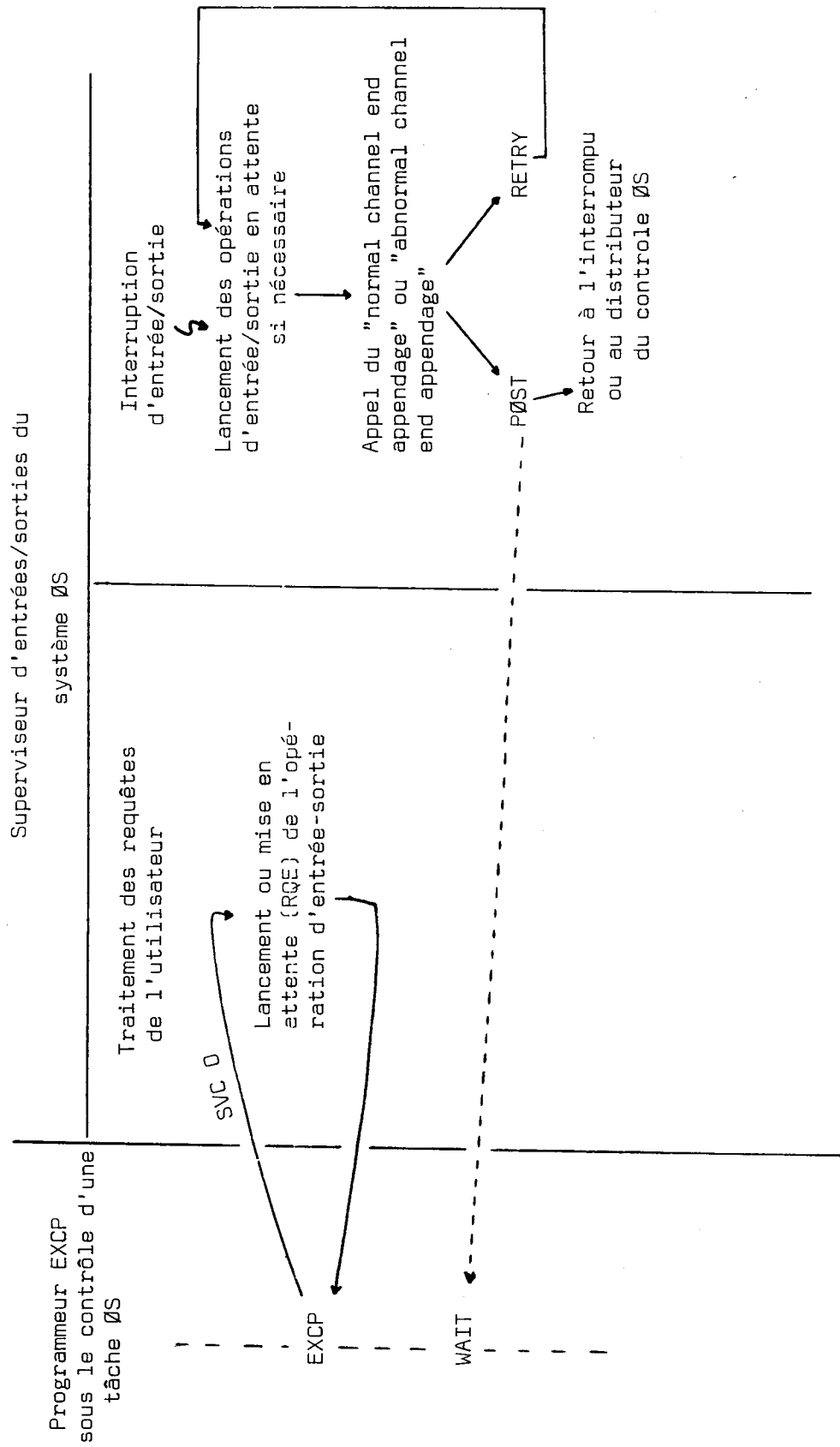
Le module EXCP ne lance pas lui-même les opérations d'entrée-sortie par des SIØ comme dans la version sur machine virtuelle mais il met à jour l'IØB à l'aide des informations contenues dans le BIØ (adresse du programme canal) et appelle le superviseur d'entrée-sortie du système ØS (SVC 0).

Pour continuer l'analogie avec le système développé sur machine virtuelle on a remplacé le module (IØINT) chargé du traitement des interruptions d'entrée-sortie par un module (IØAPPN) activé par le superviseur d'entrée-sortie du système ØS à la fin de toute opération d'entrée-sortie, soit en tant que "normal channel end appendage" soit en tant que "abnormal channel end appendage". Pour cela il a fallu rajouter au système TELCOM un service (ØIØAPPN) capable de demander au système ØS pour chaque fichier ouvert, l'utilisation de ces routines particulières.

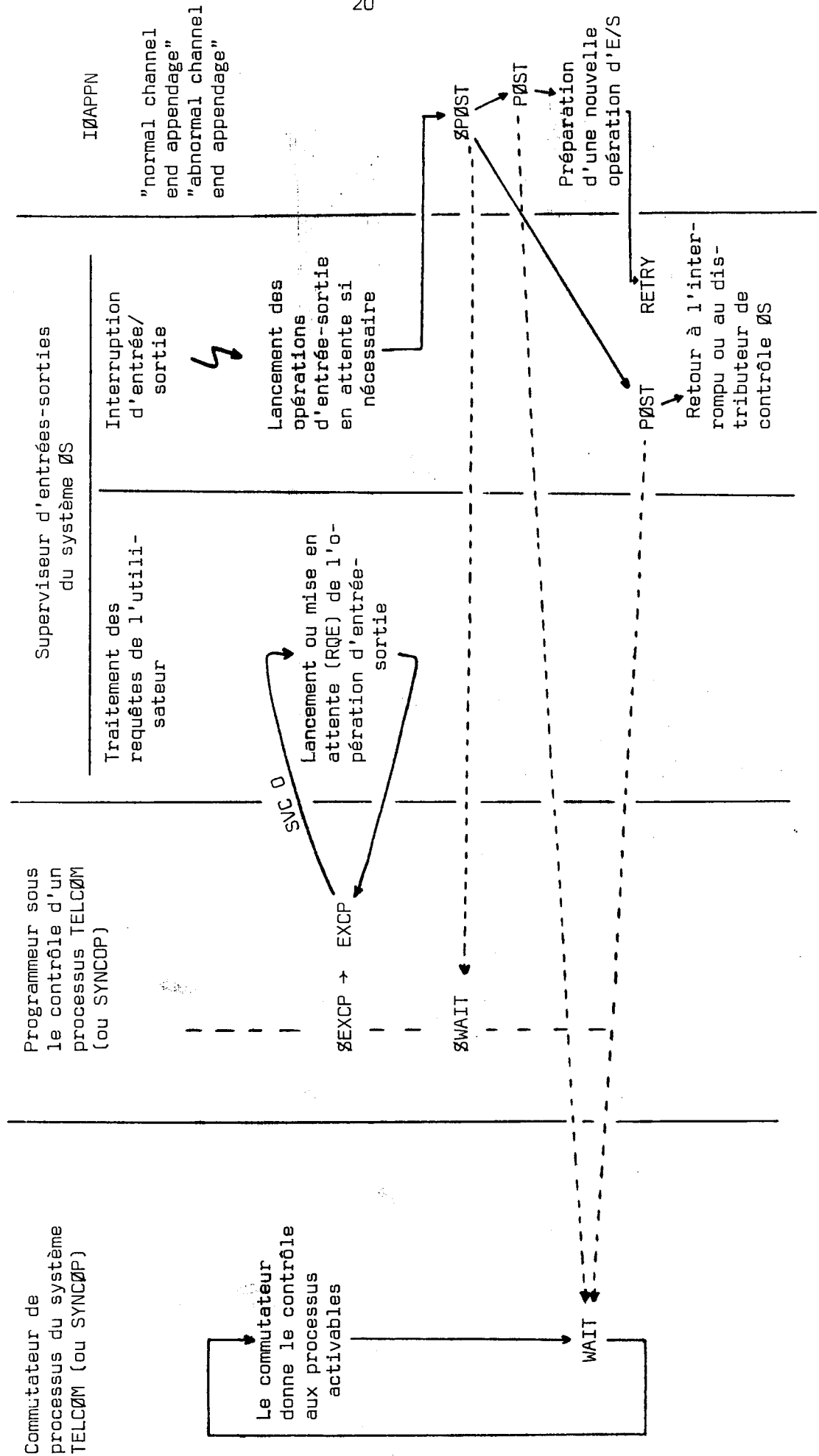
On voit ainsi que les "devices routines" du système TELCOM ne sont pas tout à fait transportables sans modifications d'un système vers l'autre puisqu'il faut, dans les séquences d'initialisation rajouter les appels à ØPEN et ØIØAPPN dans la version ØS, et dans les séquences de fermeture, l'appel à CLØSE.

Une difficulté dans la mise en place du module IØAPPN est apparue du fait que le module IØINT dans la version sur machine virtuelle effectue des appels à EXCP. Ces appels ont été conservés mais il a fallu apporter quelques modifications pour que, dans ce cas le module EXCP du système TELCOM ne fasse pas appel à EXCP (SVC 0) du système ØS mais mémorise sa requête de telle façon que le module IØAPPN rende, dans ce cas précis, le contrôle au superviseur d'entrée-sortie du système ØS à une adresse demandant de relancer une opération d'entrée-sortie.

La programmation des entrées-sorties au niveau EXCP (Execute Channel Program)  
 sous le contrôle du système ØS

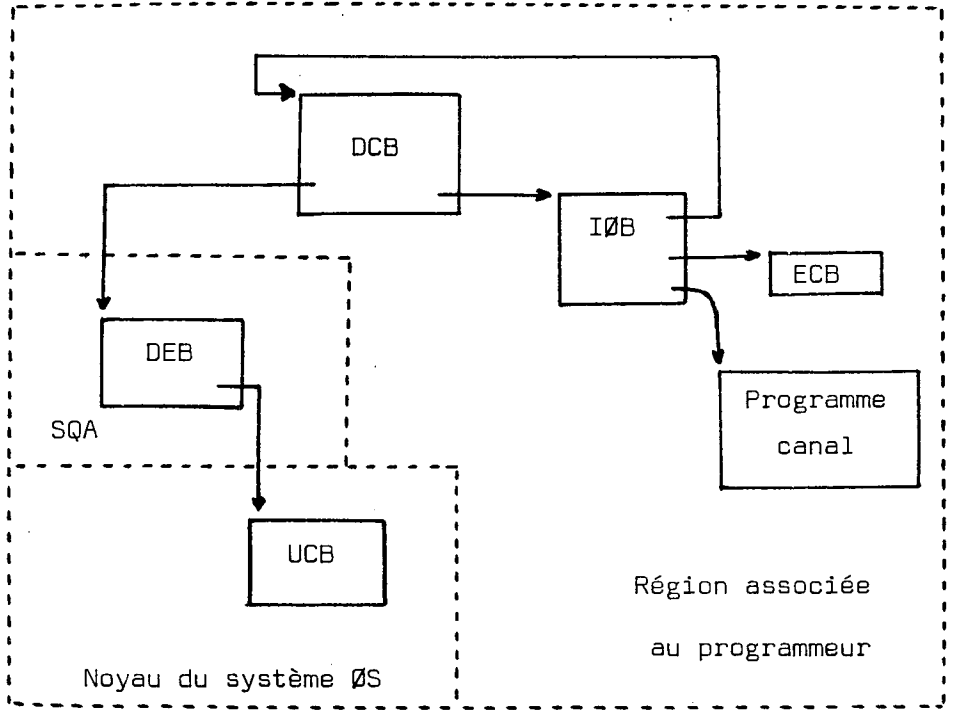


La programmation des entrées-sorties sous le contrôle du système TELCØM  
 (et du système SYNCØP) sous ØS

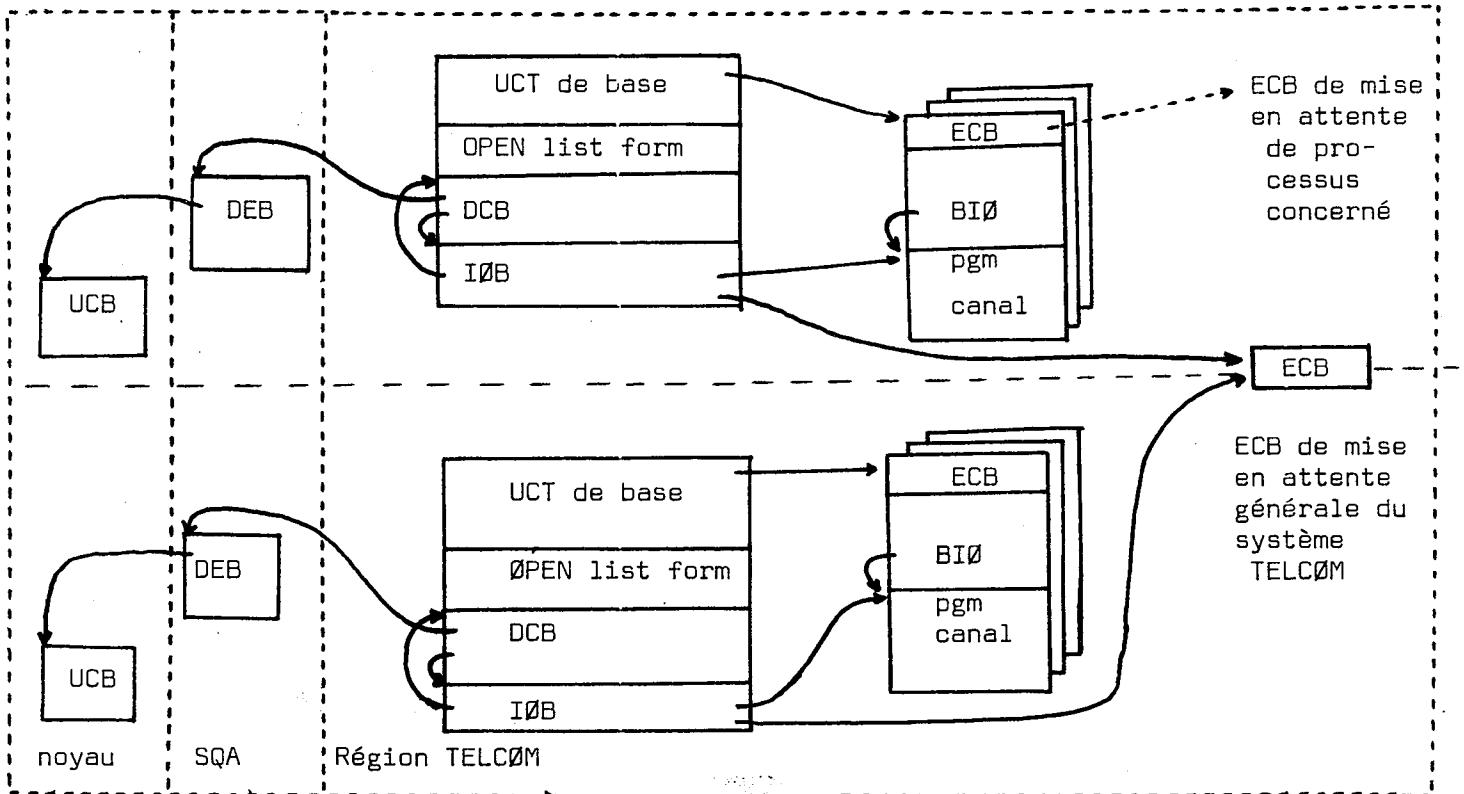


Les blocs de contrôle associés à la programmation  
au niveau EXCP (Execute Channel Program) sous ØS

- DCB : Data Control Block
- IØB : Input Output Block
- ECB : Event Control Block
- DEB : Data Extent Block
- UCB : Unit Control Block
- SQA : System Queue Area



Les blocs de contrôle associés à la programmation  
des entrées-sorties sous TELCØM/ØS





## 2.4 SVC affectée au traitement des services devant se dérouler en mode-maitre

Tous les programmes ne pouvant pas se dérouler sous le mode standard du programmeur sous ØS sont regroupés en une seule SVC dont le numéro est spécifié par une variable globale d'assemblage. Les deux octets qui suivent permettent respectivement d'effectuer un contrôle de validité sur la clé de protection de l'appelant et de spécifier un code associé au programme appelé.

Le premier octet identifie l'appelant autorisé de l'une des façons suivantes :

- utilisation exclusive par la tâche TELCØM
- tout utilisateur permis (pas de contrôle de validité)
- tout utilisateur permis sauf la tâche TELCØM
- les utilisateurs permis sont ceux ayant la clé de protection du système TELCØM ou la clé 0.
- utilisateurs travaillant sous la clé de protection 0 seuls utilisés.

Le deuxième octet identifie le service requis par l'appelleur :

- simulation d'une interruption de type attention pour la gestion des pseudo consoles opérateur ASP
- simulation d'une interruption d'entrée-sortie de type 'channel end' sur une pseudo console opérateur ASP
- simulation de la fin normale d'une opération d'entrée-sortie pour la gestion des pseudo unités ('channel end' et 'device end' dans le CSW)
- masquage des interruptions d'entrée-sortie, externes, machine check et passage en mode superviseur
- démasquage des interruptions d'entrée-sortie, externes, machine check et retour en mode programme.
- exécution d'une instruction, sans modification, en mode superviseur, sous le contrôle d'un PSW ayant une clé de protection nulle
- exécution d'une instruction, modifiée par un masque, en mode superviseur, sous le contrôle d'un PSW de clé de protection nulle.
- demande de définition, pour un fichier Ø\$ ouvert, du module IØAPPN comme "normal channel end appendage" et comme "abnormal channel end appendage".

- simulation d'une opération d'entrée-sortie sur une pseudo unité. Ce travail consiste à informer le processus chargé de la gestion de cette pseudo unité et à lui fournir les informations nécessaires au traitement de l'opération d'entrée-sortie.
- demande d'une trace qui permet de compter le nombre de passages pour chaque appeleur et de mémoriser l'ordre des derniers appels effectués.

Il existe un ensemble de macro instructions permettant d'effectuer l'appel de ces services. Leur mode d'appel est fourni en annexe 1.

## 2.5 Gestion d'horloges au niveau des processus

On peut aussi effectuer des comparaisons entre les services offerts sur une machine virtuelle gérée par CP et ceux offerts sur une tâche gérée par ØS-MVT pour effectuer une gestion d'horloge au niveau des processus du système TELCØM. Il faut alors comparer l'horloge en mémoire basse de la machine virtuelle à celle gérée par ØS, pour une tâche, qui peut être initialisée par STIMER et remise à zéro par TTIMER. Il faut aussi comparer les programmes de traitement des interruptions externes à la routine d'exit spécifiée à l'appel de STIMER et activée quand l'horloge correspondante atteint la valeur 0. Mais une difficulté importante apparaît du fait qu'il n'est pas permis de réinitialiser cette horloge dans la routine d'exit alors qu'il est possible de réinitialiser l'horloge d'une machine virtuelle dans un programme de traitement d'interruption externe. Il faut alors remplacer cette initialisation par le réveil d'un processus du système TELCØM (STIME) qui est chargé de réinitialiser l'horloge de la tâche ØS, si un autre processus n'a pas été amené à la réinitialiser lui-même (suite à une demande de réveil) avant l'activation du processus STIME.

Ces difficultés, ainsi que le souhait d'apporter quelques améliorations nous ont conduits, après une première version, à reprendre complètement la logique et la programmation de la gestion des horloges au niveau des processus tout en gardant la plus grande compatibilité possible avec la première version, au moins du point de vue de l'utilisateur.

Une description du mécanisme utilisé est fournie par la note technique du CICG n°35 [F.5], nous allons le présenter dans le chapitre suivant car son étude présente un intérêt plus général que le cas particulier d'implémentation qui nous intéresse directement.

### 3. LA GESTION DES RÉVEILS

Bien que présentée dans le cadre de la gestion de réveils sous le contrôle de processus du système TELCØM développé sous ØS l'étude faite dans ce chapitre peut s'appliquer à tout système qui dispose d'une horloge unique (qu'il est possible d'initialiser, de consulter, d'annuler et qui peut activer une séquence d'instructions lorsqu'elle atteint la valeur 0) et pour lequel on veut mettre à la disposition des utilisateurs de ce système autant d'horloges (ou réveils)-de même type que celle dont dispose le système- qu'ils le souhaitent.

On verra en fin de ce chapitre dans quels cas la méthode présentée ici se justifie le mieux.

La précision requise dans les demandes de réveil effectuées par les processus du système TELCØM ne dépasse jamais le centième de seconde, cependant, du fait que le système ØS n'est pas spécialement conçu pour des applications en temps réel et ne fournit pas une valeur précise du délai maximum pendant lequel il peut travailler sans activer de tâche utilisateur, on a utilisé l'unité la plus fine disponible au programmeur sous le contrôle d'une tâche ØS tout en tenant compte d'un écart minimum  $\epsilon$ , exprimé dans cette unité, qui fournit une valeur telle qu'on puisse considérer comme égales deux heures qui diffèrent de moins de cet écart minimum.

L'unité ainsi utilisée est le 'timer Unit' noté TU

$$1 \text{ TU} = 26,04166 \mu\text{s}$$

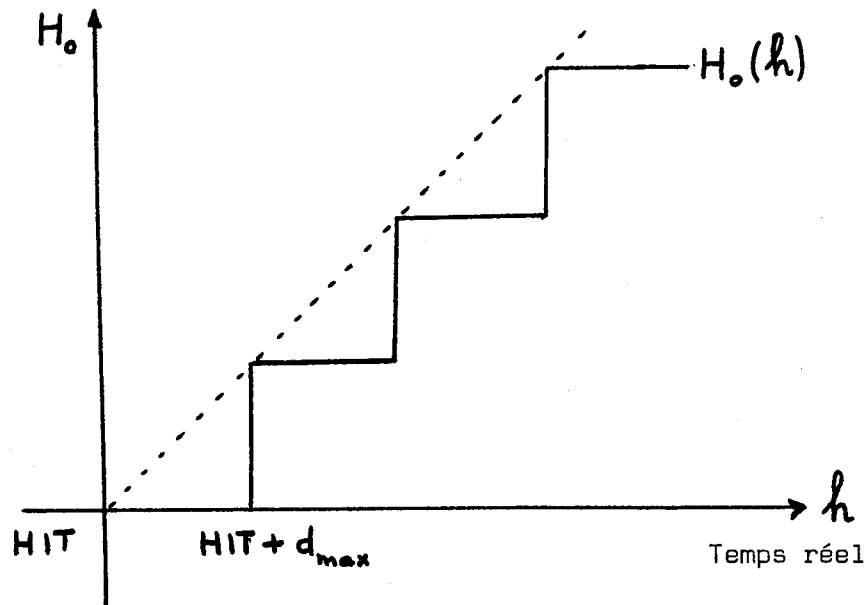
et la valeur choisie pour l'écart minimum  $\epsilon$  est :

$$\epsilon = 30 \text{ TU} \approx 780 \mu\text{s}$$

Le délai maximum qui peut être demandé à ØS se représente en TU dans un mot de 32 bits non signé, mais on s'est limité à l'utilisation d'un nombre positif représenté sur un mot en virgule fixe pour faciliter les opérations arithmétiques. On appelle  $d_{\max}$  le délai maximum utilisable qui vaut ainsi :

$$d_{\max} = 2^{31} - 1 \text{ TU} \approx 14 \text{ h}$$

On désigne par  $H_o$  une variable à laquelle on affecte à l'initialisation du système TELCØM l'heure demandée à ØS par la SVC TIME, et qui est incrémentée périodiquement de  $d_{\max}$ , toutes les fois qu'un intervalle de temps de durée  $d_{\max}$  est écoulé.



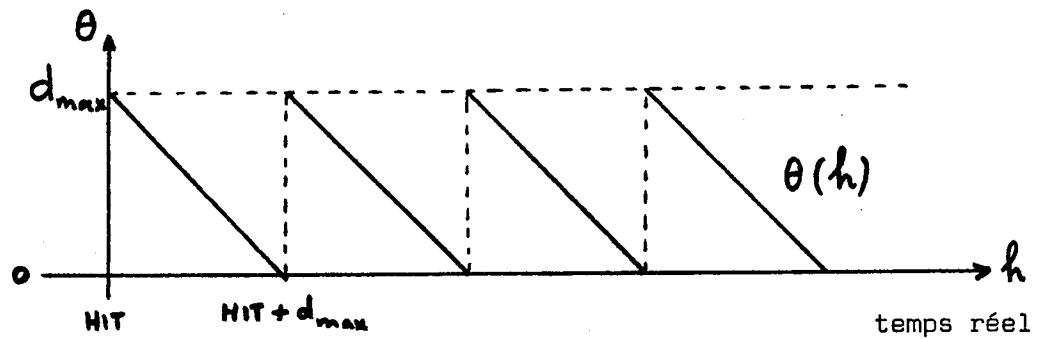
$HIT$  désigne ici l'heure d'initialisation du système TELCØM.

Pour effectuer la mise à jour de cette variable  $H_o$ , après l'avoir initialisée à la valeur  $HIT$ , on effectue une demande à l'horloge ØS de la tâche TELCØM (appel de la SVC STIMER) pour un réveil dans un délai  $d_{\max}$ . On suppose, pour l'instant, que les processus de TELCØM n'effectuent pas de demande de réveil. Il suffit alors, toutes les fois que l'horloge ØS atteint la valeur 0 (routine TEXT activée), d'incrémenter  $h_o$  de  $d_{\max}$  et de relancer l'horloge ØS pour un délai de  $d_{\max}$ .

Désignons par  $\theta$  le temps restant dans l'horloge à l'heure  $h$  :

$$h = H_o + d_{\max} - \theta$$

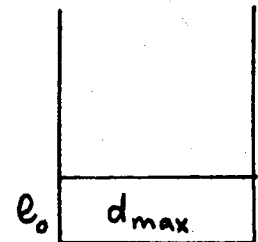
$$\forall h \geq HIT$$



Ce mécanisme simple permet donc, à l'aide de la variable  $H_0$  et du temps restant dans l'horloge OS (qui peut être obtenu par la SVC TTIMER) de calculer l'heure  $h$  à tout moment.

Mais il faut pouvoir gérer des demandes de réveil effectuées par les processus du système TELCØM. Pour cela il faut gérer une pile des heures relatives de réveil (PHR) dans laquelle on met la valeur  $d_{\max}$  à l'initialisation du système TELCØM. Les valeurs insérées dans les entrées de cette pile sont les heures auxquelles les processus demandeurs devront être réveillés, relatives à la variable  $H_0$ .

L'entrée  $e_0$  insérée initialement dans la pile doit pouvoir se différencier des entrées associées à des demandes de réveil par des processus puisque à l'heure spécifiée par cette entrée il faudra incrémenter  $H_0$  de  $d_{\max}$  et non réveiller le processus demandeur.



Pile des heures relatives de réveil

On va s'intéresser maintenant aux entrées insérées dans la pile dans le cas d'une demande de réveil effectuée par un processus. On traitera d'abord un cas simple où une demande de réveil et une seule est effectuée, puis on s'intéressera au cas le plus général.

### 3.1 Etude d'un cas simple : une seule demande réveil

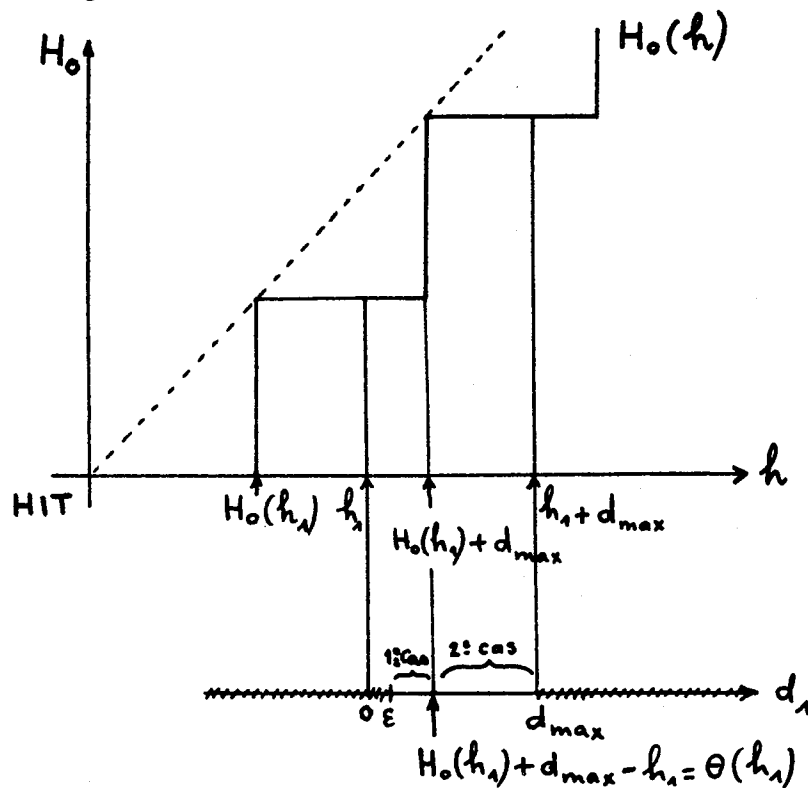
On suppose qu'à l'heure  $h_1$  un processus effectue une demande réveil pour l'heure  $h_1 + d_1$ .

Si  $d_1 \leq \epsilon$  on considère qu'il est inutile d'initialiser un réveil pour une durée si brève, on réalise aussitôt la requête spécifiée par le processus demandeur et on lui rend le contrôle.

De plus, on impose  $d_1 \leq d_{\max}$  (sinon retour au demandeur avec un code d'erreur).

On se limite ainsi à prendre effectivement en considération les requêtes pour lesquelles  $d_1 \in ]\epsilon, d_{\max}]$ .

On va différencier deux cas selon que  $h_1$  et  $h_1 + d_1$  se trouvent sur le même palier de  $H_0$  ou pas.



1er CAS :  $h_1 + d_1 \leq H_0(h_1) + d_{\max}$

Suivant les notations précédemment utilisées  $\theta(h_1)$  désigne le temps restant dans l'horloge  $\emptyset S$  à l'heure  $h_1$ . On rappelle que  $\theta(h_1)$  est directement accessible par programme à l'heure  $h_1$ , alors que la valeur  $h_1$  ne l'est pas, mais est calculable par :

$$h_1 = H_0(h_1) + d_{\max} - \theta(h_1)$$

Le premier cas s'identifie alors simplement par :

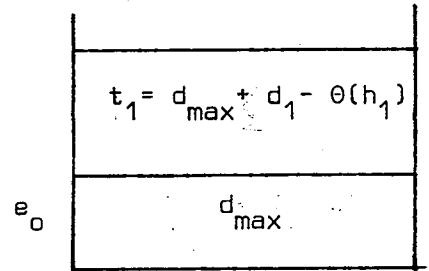
$$d_1 \leq \theta(h_1)$$

On considère qu'il y a égalité quand  $\theta(h_1)$  et  $d_1$  diffèrent de moins de  $\epsilon$ . On insère au sommet de la pile des heures relatives de réveil, la valeur  $t_1 = h_1 + d_1 - H_0(h_1)$  qui fournit l'heure de réveil relative à la valeur de la variable  $H_0$  au moment du réveil qui ne diffère pas, dans ce cas de sa valeur au moment de la demande. La valeur  $t_1$  est calculable en remplaçant  $h_1$  par son expression en fonction de variables accessibles par programme, soit :

$$t_1 = d_{\max} + d_1 - \theta(h_1)$$

Par la définition même de  $t_1$  et du fait qu'on se trouve dans le cas  $d_1 \leq \theta(h_1)$  on voit que :

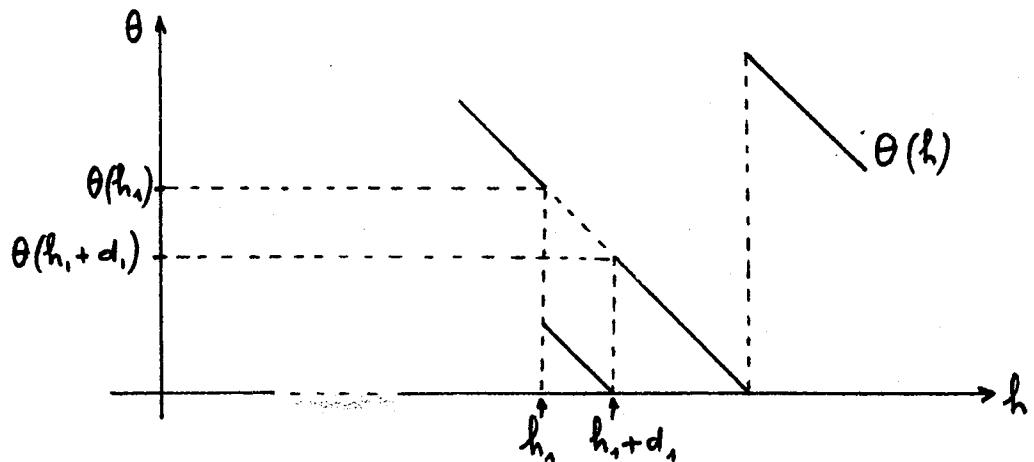
$$0 < t_1 \leq d_{\max}$$



Pile des heures relatives de réveil (PHR)

La valeur  $t_1$  est donc exprimable en notation en complément à 2 sur un mot de 32 bits.

Après l'insertion de ce nouvel élément dans la pile des heures relatives de réveil, on réinitialise l'horloge  $\emptyset S$  à la valeur  $d_1$  (si  $\theta_1(h_1)$  et  $d_1$  ne diffèrent pas de plus de  $\epsilon$  on la laisse inchangée).



Quand l'horloge ØS atteint la valeur 0, il faut supprimer l'entrée qui se trouve en sommet de pile, réaliser l'action correspondante à ce réveil, et relancer l'horloge ØS à la valeur qu'elle aurait si la demande n'avait pas eu lieu, soit :

$$\theta(h_1 + d_1) = H_0(h_1 + d_1) + d_{\max} - (h_1 + d_1)$$

$$\theta(h_1 + d_1) = H_0(h_1 + d_1) + d_{\max} - (H_0(h_1) + d_{\max} - \theta(h_1)) - d_1$$

$h_1$  et  $h_1 + d_1$  se trouvant sur le même palier de  $H_0$  :

$$H_0(h_1 + d_1) = H_0(h_1)$$

d'où 
$$\theta(h_1 + d_1) = \theta(h_1) - d_1$$

On voit que cette nouvelle valeur de  $\theta$  s'obtient facilement quand l'horloge ØS atteint la valeur 0 en retranchant la valeur qui se trouve au sommet de la pile PHR à la valeur précédente.

Si  $\theta(h_1) = d_1$  (ou ne diffèrent pas de plus de  $\epsilon$ ) il est clair qu'il ne faut pas réinitialiser  $\theta$  à 0 (ni à  $\epsilon$ ) mais à  $d_{\max}$  et incrémenter  $H_0$  de  $d_{\max}$ .

2e CAS : 
$$h_1 + d_1 > H_0(h_1) + d_{\max}$$

Ce qui peut aussi s'écrire, de façon directement identifiable par programme :

$$d_1 > \theta(h_1)$$

Pour une telle demande de réveil, on insère en bas de la pile des heures relatives de réveil la valeur :



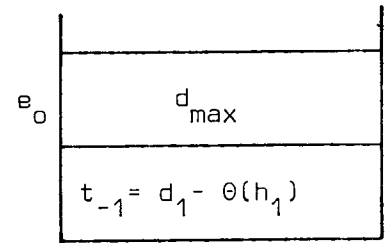
$$t_{-1} = h_1 + d_1 - H_0(h_1 + d_1)$$

qui fournit l'heure de réveil relative à la valeur de la variable  $H_0$  au moment du réveil.

Dans le cas présent :

$$H_0(h_1 + d_1) = H_0(h_1) + d_{\max}$$

On va, comme précédemment, exprimer  $t_{-1}$  en fonction de variables directement accessibles par programme :



PHR

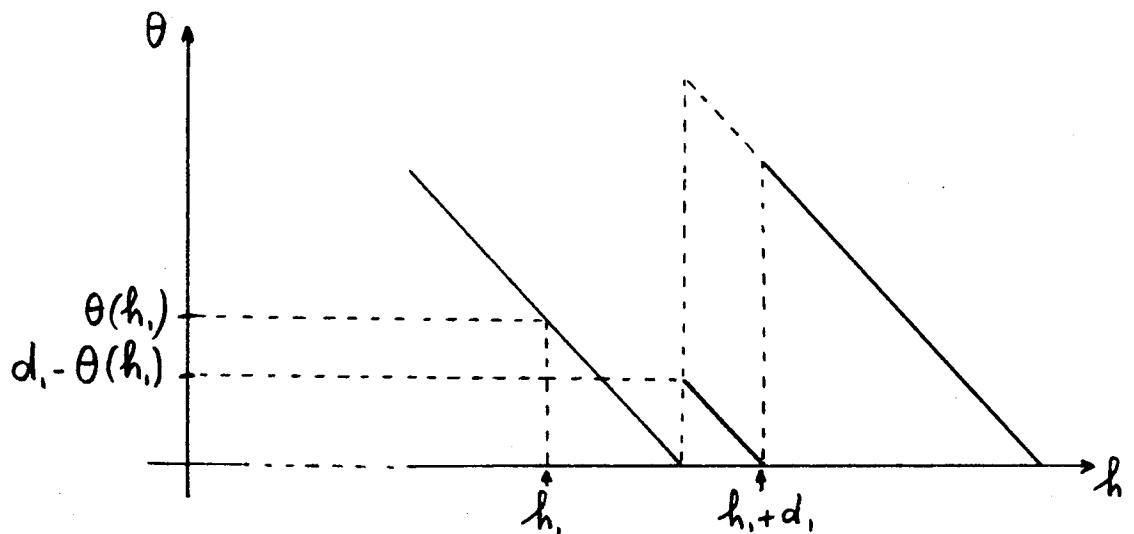
$$t_{-1} = H_0(h_1) + d_{\max} - \theta(h_1) + d_1 - H_0(h_1) - d_{\max}$$

$$t_{-1} = d_1 - \theta(h_1)$$

Du fait que  $d_1 \leq d_{\max}$  (imposé à l'appelleur), que  $\theta$  est toujours positif ou nul et que l'on se trouve dans le cas  $d_1 > \theta(h_1)$ , on en déduit que :

$$0 < t_{-1} \leq d_{\max}$$

On se retrouve bien dans les mêmes limites que pour  $t_1$  dans le premier cas. Au moment de la demande de réveil on insère ce nouvel élément dans la pile mais on ne modifie pas l'horloge  $\emptyset S$ .



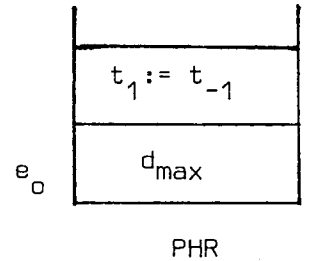
Quand l'horloge  $\emptyset S$  atteint la valeur 0 l'entrée  $e_0$  se trouve en sommet de pile. On incrémente  $H_0$  de  $d_{\max}$ , on supprime l'entrée  $e_0$  du sommet de la pile et on insère une nouvelle entrée  $e_0$  en bas de pile contenant la valeur  $d_{\max}$ .

Il faut alors réinitialiser l'horloge  $\emptyset S$

à une valeur telle qu'elle atteindra 0 à

l'heure  $h_1 + d_1$ , soit :

$$\begin{aligned} h_1 + d_1 - H_0(h_1 + d_1) &= H_0(h_1) + d_{\max} - \theta(h_1) \\ &\quad + d_1 - (H_0(h_1) + d_{\max}) \\ &= d_1 - \theta(h_1) \end{aligned}$$



C'est la valeur  $t_{-1}$  qui se trouve maintenant en sommet de pile et que l'on nomme  $t_1$ .

On constate que cette valeur  $t_1$  est bien l'heure de réveil (demandée à l'heure  $h_1$  pour  $h_1 + d_1$ ) relative à la valeur actuelle de  $H_0$  puisque  $H_0$  vient d'être incrémentée de  $d_{\max}$  et ne sera plus modifiée jusqu'au moment du réveil.

On se retrouve donc dans la même situation que précédemment.

### 3.2 Etude du cas général de prise en considération d'une demande de réveil

Supposons maintenant que la pile des heures relatives de réveil contienne de bas en haut les valeurs suivantes :

$$t_{-n}, t_{-(n-1)}, \dots, t_{-2}, t_{-1}, t_0 = d_{\max}, t_1, t_2, \dots, t_m$$

telles que :

$$\epsilon < t_{-1} \leq t_{-2} \leq \dots \leq t_{-n} \leq d_{\max}$$

$$\epsilon < t_m \leq t_{m-1} \leq \dots \leq t_2 \leq t_1 \leq t_0 = d_{\max}$$

Ces valeurs représentant les heures de réveil relatives à la valeur de la variable  $H_0$  au moment où le réveil concerné arrive à expiration.

Si on désigne par  $h_i$  les heures de demande de réveil pour les heures  $h_i + d_i$  pour  $i$  allant de  $-n$  à  $+m$ , on peut écrire :

$$h_i + d_i = t_i + H_0(h_i + d_i) \text{ pour } i \text{ allant de } -n \text{ à } +m.$$

Cette notation implique que l'on considère l'entrée  $e_0$  comme correspondant à une demande de réveil effectuée à l'heure  $h_0$  pour l'heure  $h_0 + d_{\max}$  et que la variable  $H_0$  prend la valeur  $h_0$  à l'instant  $h_0 + d_{\max}$ .

$$t_0 = d_{\max}$$

$$d_0 = d_{\max}$$

$$H_0(h_0 + d_{\max}) = h_0$$

Ce qui permet alors d'écrire :

$$H_0(h_i + d_i) = h_0 \text{ pour } i \text{ allant de } 0 \text{ à } m.$$

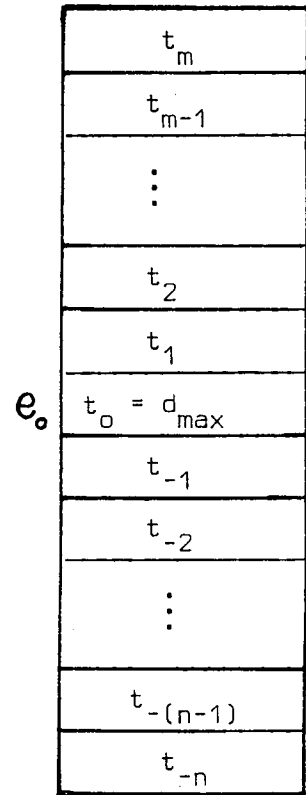
$$H_0(h_i + d_i) = h_0 + d_{\max} \text{ pour } i \text{ allant de } -1 \text{ à } -n.$$

On suppose aussi que l'horloge  $\emptyset S$  contient une valeur telle qu'elle atteindra 0 à l'heure  $t_m + H_0(h_m + d_m) = t_m + h_0$  c'est-à-dire à l'heure d'expiration du réveil associé à l'entrée se trouvant en sommet de pile, si aucune nouvelle demande ne nécessite de la modifier d'ici là.

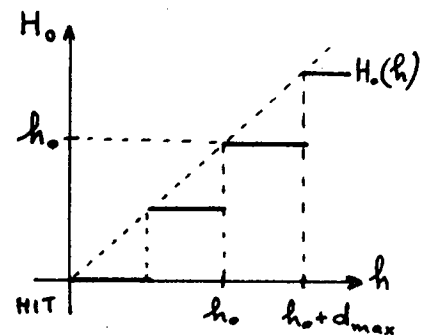
On suppose maintenant qu'à l'heure  $h_k$  telle que :

$$h_k > h_i$$

$$\forall i \in [-n, +m]$$



Pile des heures relatives de réveil



un processus fait une demande réveil pour l'heure

$$h_k + d_k \quad \text{avec} \quad 0 < d_k \leq d_{\max}$$

Dans le cas où  $d_k \leq \epsilon$  on réalise aussitôt la requête demandée pour le réveil, sinon il faudra insérer une nouvelle entrée dans la pile des heures relatives de réveil. Sa position sera déterminée par comparaison de l'heure de réveil demandée avec celles des réveils précédemment attendus.

On va examiner la façon de déterminer cette position par comparaison de variables accessibles par programme, en effectuant les analyses successives des différentes situations possibles.

$$\textcircled{a} \quad \underline{h_k + d_k < h_m + d_m - \epsilon}$$

Ceci signifie que le réveil demandé doit être considéré à l'heure  $h_k$  comme le prochain à se réaliser et qu'il se réalisera avant celui précédemment prévu comme le prochain.

Autrement dit, la nouvelle entrée sera à rajouter en sommet de pile.

Si on désigne par  $\theta_k$  le temps restant dans l'horloge ØS à l'heure  $h_k$ , on sait que  $\theta_k$  est le temps qu'il faut ajouter à  $h_k$  pour obtenir  $h_m + d_m$ , soit :

$$h_k + \theta_k = h_m + d_m = t_m + H_0(h_m + d_m) = t_m + h_0$$

Ce qui permet d'écrire la relation qui permet de déterminer que l'on se trouve dans le cas  $\textcircled{a}$  sous la forme :

$$t_m + h_0 - \theta_k + d_k < t_m + H_0(h_m + d_m) - \epsilon$$

$$\boxed{d_k < \theta_k - \epsilon}$$

Il est alors facile d'identifier ce cas.

On va déterminer la valeur  $t_{m+1}$  qu'il faut rajouter au sommet de la pile des heures relatives de réveil :

$$t_{m+1} = h_k = d_k - H_0(h_k + d_k)$$

C'est bien l'heure du réveil relative à la valeur de la variable  $H_0$  au moment où le réveil se réalise.

$$t_{m+1} = t_m + h_0 - \theta_k + d_k - h_0$$

$$t_{m+1} = t_m + d_k - \theta_k$$

On rappelle l'hypothèse initiale :

$$h_k > h_i \quad \forall i \in [-n, +m]$$

On en déduit :

$$h_k > h_0$$

$$t_{m+1} + h_0 - d_k > h_0$$

$$t_{m+1} > d_k > \epsilon$$

$t_{m+1} = t_m + d_k - \theta_k$
$t_m$
$t_{m-1}$
$\vdots$
$t_1$
$t_0 = d_{\max}$
$t_{-1}$
$\vdots$
$t_{-n}$

PHR

D'autre part, du fait que  $d_k < \theta_k - \epsilon$ , ce qui caractérise le cas (a), on peut écrire  $t_{m+1} < t_m - \epsilon$

D'où :  $\epsilon < t_{m+1} < t_m - \epsilon$

Après insertion de cette nouvelle entrée dans la pile des heures relatives de réveil, on réinitialise l'horloge  $\emptyset S$  à la valeur  $d_k$  de telle façon qu'elle atteigne la valeur 0 à l'heure d'expiration du nouveau réveil associé à l'entrée se trouvant en sommet de pile. Il faut signaler que l'on considère ainsi que l'heure à laquelle on réinitialise l'horloge  $\emptyset S$  est toujours  $h_k$ , c'est-à-dire que l'on considère comme négligeable le temps écoulé entre le moment où le processus demandeur a effectué sa requête et le moment où l'horloge  $\emptyset S$  est réinitialisée.

Après ces opérations on se retrouve bien dans la même situation qu'avant la demande de réveil avec une entrée de plus dans la table des heures relatives de réveil.

(b) Il existe  $i \in [-n, +m]$  tel que :

$$h_i + d_i - \epsilon \leq h_k + d_k \leq h_i + d_i + \epsilon$$

Ceci signifie que le réveil demandé à l'heure  $h_k$  peut être considéré comme devant se réaliser en même temps que celui demandé à l'heure  $h_i$ .

La nouvelle entrée sera rajoutée au dessus de l'entrée  $i$ .

En désignant, comme précédemment, par  $\theta_k$  le temps restant dans l'horloge  $\emptyset S$  à l'heure  $h_k$  :

$$h_k = t_m + h_0 - \theta_k$$

On peut écrire la relation qui identifie cette situation en utilisant des variables accessibles par programme au moment de la requête :

$$t_i + H_0(h_i + d_i) - \epsilon \leq t_m + h_0 - \theta_k + d_k \leq t_i + H_0(h_i + d_i) + \epsilon$$

Le cas (b) peut alors s'identifier comme suit :

$$\left\{ \begin{array}{l} \exists i \in [0, m] \text{ tel que } t_i - \epsilon \leq t_m - \theta_k + d_k \leq t_i + \epsilon \\ \text{ou} \\ \exists i \in [-1, -n] \text{ tel que } t_i + d_{\max} - \epsilon \leq t_m - \theta_k + d_k \leq t_i + d_{\max} + \epsilon \end{array} \right.$$

Il est alors facile, en consultant la pile des heures relatives de réveil, de déterminer si une telle valeur  $i$  existe.

Si elle est trouvée, il suffit de rajouter sur l'entrée  $i$  de la pile PHR une entrée pour la demande faite à l'heure  $h_k$  (pour un réveil à l'heure  $h_k + d_k$ ) contenant la valeur  $t_i$ . On se retrouve à nouveau dans la même situation qu'avant la demande de réveil avec une entrée de plus dans la pile des heures relatives de réveil (PHR).

(c) Il existe  $i \in [-n, m-1]$  tel que :

$$h_{i+1} + d_{i+1} + \epsilon < h_k + d_k < h_i + d_i - \epsilon$$

Ceci signifie que le réveil demandé à l'heure  $h_k$  ne peut pas être considéré comme devant se réaliser à la même heure qu'un réveil précédemment demandé et qu'il faudra insérer une nouvelle heure relative de réveil dans la pile PHR entre l'entrée  $i$  et l'entrée  $i+1$ .

Cette relation peut aussi s'écrire :

$$t_{i+1} + H_0(h_{i+1} + d_{i+1}) + \epsilon < t_m + h_0 - \theta_k + d_k < t_i + H_0(h_i + d_i) - \epsilon$$

On sait que  $H_0(h_i + d_i) = h_0$  pour  $i \in [0, m]$

$$H_0(h_i + d_i) = h_0 + d_{\max} \text{ pour } i \in [-1, -n]$$

On voit alors que le cas (c) peut s'identifier de la façon suivante :

$$\left\{ \begin{array}{l} \exists i \in [0, m-1] \text{ tel que } t_{i+1} + \epsilon < t_m - \theta_k + d_k < t_i - \epsilon \\ \text{ou} \\ t_0 + \epsilon < t_m - \theta_k + d_k < t_{-1} + d_{\max} - \epsilon \\ \text{ou} \\ \exists i \in [-n, -2] \text{ tel que } t_{i+1} + d_{\max} + \epsilon < t_m - \theta_k + d_k < t_i + d_{\max} - \epsilon \end{array} \right.$$

Si, en consultant les valeurs  $t_i$  de la pile PHR on arrive à satisfaire l'une de ces 3 relations (la deuxième correspondant à  $i = -1$ ), il suffit alors d'insérer entre l'entrée  $i$  et l'entrée  $i+1$  de PHR une entrée contenant l'heure du réveil demandé relative à la valeur de  $H_0$  au moment où le réveil doit se réaliser, soit :

$$t_k = h_k + d_k - H_0(h_k + d_k)$$

$$t_k = t_m + h_0 - \theta_k + d_k - H_0(h_k + d_k)$$

La valeur de  $H_o(h_k + d_k)$  dépend des valeurs de  $H_o(h_{i+1} + d_{i+1})$  et  $H_o(h_i + d_i)$  qui l'encadrent :

$$H_o(h_k + d_k) = h_o \quad \text{pour } i \in [0, m-1]$$

$$H_o(h_k + d_k) = h_o + d_{\max} \quad \text{pour } i \in [-n, -1]$$

d'où :

$$\begin{cases} t_k = t_m - \theta_k + d_k & \text{pour } i \in [0, m-1] \\ t_k = t_m - \theta_k + d_k - d_{\max} & \text{pour } i \in [-n, -1] \end{cases}$$

Les relations qui ont permis de trouver  $i$  prennent alors la forme suivante :

$$\begin{cases} t_{i+1} + \epsilon < t_k < t_i + \epsilon & \text{pour } i \in [0, m-1] \\ \epsilon < t_k < t_{-1} - \epsilon & \text{pour } i = -1 \\ t_{i+1} + \epsilon < t_k < t_i - \epsilon & \text{pour } i \in [-n, -2] \end{cases}$$

On se retrouve bien dans la même situation qu'avant la demande de réveil, avec une entrée de plus dans la pile des heures relatives de réveil.

$$\textcircled{d} \quad \underline{h_k + d_k > h_{-n} + d_{-n} + \epsilon}$$

Ce qui signifie que le réveil demandé doit être considéré à l'heure  $h_k$  comme le dernier à se réaliser parmi ceux en attente. La nouvelle entrée sera rajoutée au bas de la pile PHR.

On exprime la relation qui permet d'identifier cette situation à l'aide de variables accessibles par programme :



$$t_m + h_o - \theta_k + d_k > t_{-n} + H_o(h_{-n} + d_{-n}) + \epsilon$$

$$t_m - \theta_k + d_k > t_{-n} + d_{\max} + \epsilon$$

Cette situation étant reconnue, il faut alors rajouter en bas de la pile PHR une entrée contenant l'heure  $t_k$  du réveil demandé relative à la valeur de  $H_o$  au moment où le réveil doit se réaliser.

$$t_k = h_k + d_k - H_o(h_k + d_k)$$

Du fait qu'à l'heure  $h_k$  aucun des réveils associés aux différentes entrées de la pile PHR n'est arrivé à expiration (l'entrée associée à un réveil étant supprimée de la pile lorsque celui-ci arrive à expiration),

$$h_k < h_i + d_i \quad \forall i \in [-n, +m] \quad \text{en particulier : } h_k < h_o + d_o = h_o + d_{\max}$$

On a également imposé  $d_k \leq d_{\max}$

On en déduit  $h_k + d_k < h_o + 2 d_{\max}$

On voit alors que  $h_k + d_k$  est le sur le même palier de  $H_o$  que  $h_{-n} + d_{-n}$ :

$$h_{-n} + d_{-n} + \epsilon < h_k + d_k < h_o + 2 d_{\max}$$

d'où :

$$H_o(h_k + d_k) = h_o + d_{\max}$$

$$t_k = h_k + d_k - h_o - d_{\max}$$

$$t_k = t_m + h_o - \theta_k + d_k - h_o - d_{\max}$$

$$t_k = t_m - \theta_k + d_k - d_{\max}$$

Il faut maintenant montrer que  $t_{-n} \leq t_k \leq d_{\max}$  pour s'assurer qu'on se retrouve bien dans la même situation qu'avant la demande de réveil, avec l'entrée  $k$  en plus en bas de la pile PHR.

On a vu que le cas (d) s'identifie par :

$$t_m - \theta_k + d_k > t_{-n} + d_{\max} + \varepsilon$$

Ce qui peut aussi s'écrire :

$$t_k > t_{-n} + \varepsilon$$

On a aussi montré que :

$$t_k = h_k + d_k - h_o - d_{\max} < (h_o + 2 d_{\max}) - h_o - d_{\max}$$

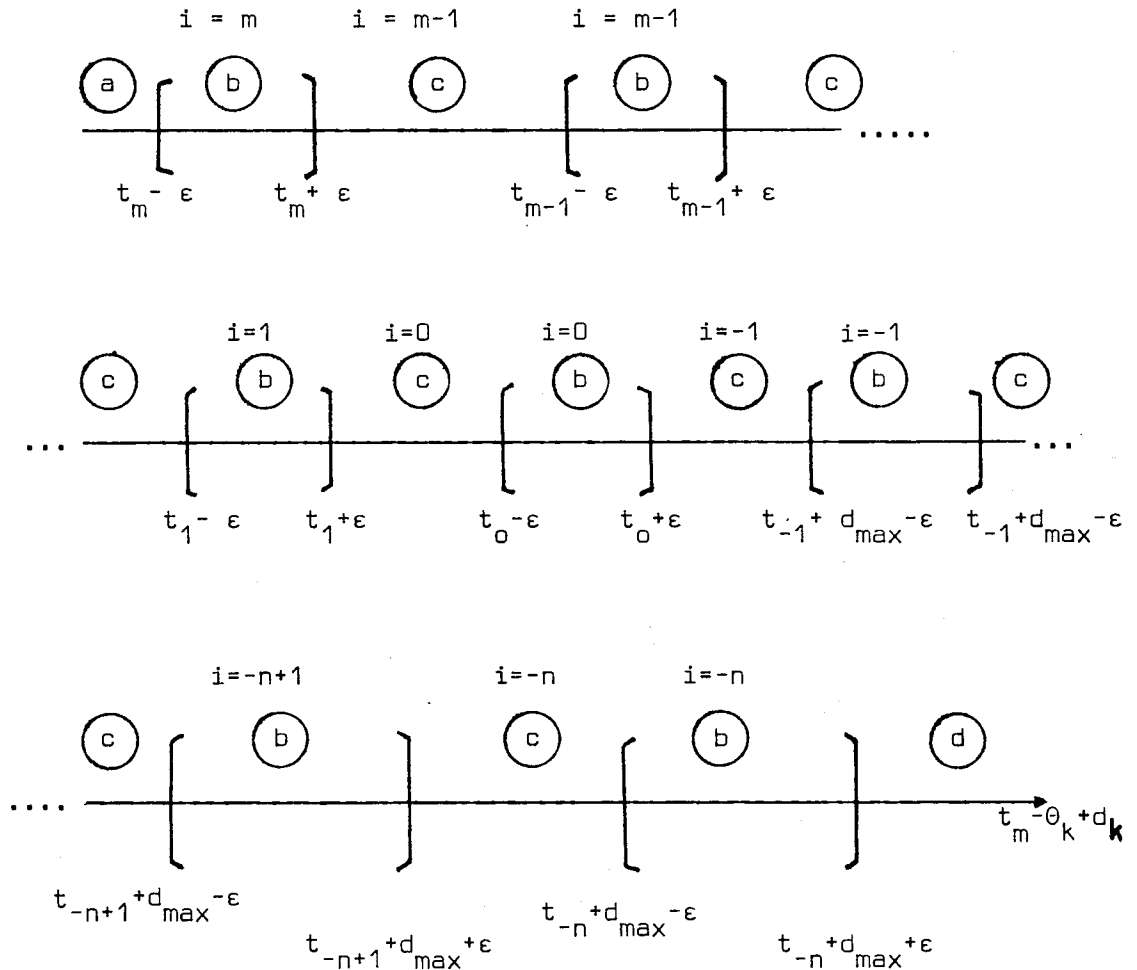
$$t_k < d_{\max}$$

Ce qui prouve bien que  $t_{-n} \leq t_k \leq d_{\max}$

Après l'analyse des différentes situations possibles on peut dresser le tableau récapitulatif suivant :

Cas	Comment déterminer le cas correspondant	où insérer la nouvelle entrée de la pile PHR	Valeur de $t_k$ à spécifier dans la nouvelle entrée
(a)	$d_k < \theta_k - \epsilon$	En sommet de pile	$t_m + d_k - \theta_k$
(b)	$\exists i \in [0, m]$ tel que : $t_i - \epsilon \leq t_m - \theta_k + d_k \leq t_i + \epsilon$ ou $\exists i \in [-1, -n]$ tel que : $t_i + d_{\max} - \epsilon \leq t_m - \theta_k + d_k \leq t_i + d_{\max} + \epsilon$	Au dessus de l'entrée $i$	$t_i$
(c)	$\exists i \in [0, m-1]$ tel que : $t_{i+1} + \epsilon < t_m - \theta_k + d_k < t_i - \epsilon$ ou $t_0 + \epsilon < t_m - \theta_k + d_k < t_{-1} + d_{\max} - \epsilon$ ou (i = -1) $\exists i \in [-n, -2]$ tel que : $t_{i+1} + d_{\max} + \epsilon < t_m - \theta_k + d_k < t_i + d_{\max} - \epsilon$	Entre l'entrée $i$ et l'entrée $i+1$	$t_m - \theta_k + d_k$ pour $i \in [0, m-1]$  $t_m - \theta_k + d_k - d_{\max}$ pour $i \in [-n, -1]$
(d)	$t_m - \theta_k + d_k > t_{-n} + d_{\max} + \epsilon$	En bas de pile	$t_m - \theta_k + d_k - d_{\max}$

Ce tableau nous montre que pour déterminer rapidement dans quelle situation on se trouve, il faut, dès la requête de l'utilisateur, calculer  $t_m - \theta_k + d_k$  et le comparer aux heures relatives de réveil de la pile PHR comme le montre le schéma suivant:



Le module chargé de prendre en considération les demandes de réveil devra effectuer ces comparaisons successives pour déterminer l'action à réaliser. Il s'agit du module STIMER. Une description en pseudo algol de ce module est fournie dans le document [F.5].

On va maintenant s'attacher à examiner plus précisément dans quels cas on pourra considérer qu'une demande de réveil arrive à expiration et quelles sont alors les actions à effectuer.

On considère qu'un réveil se réalise :

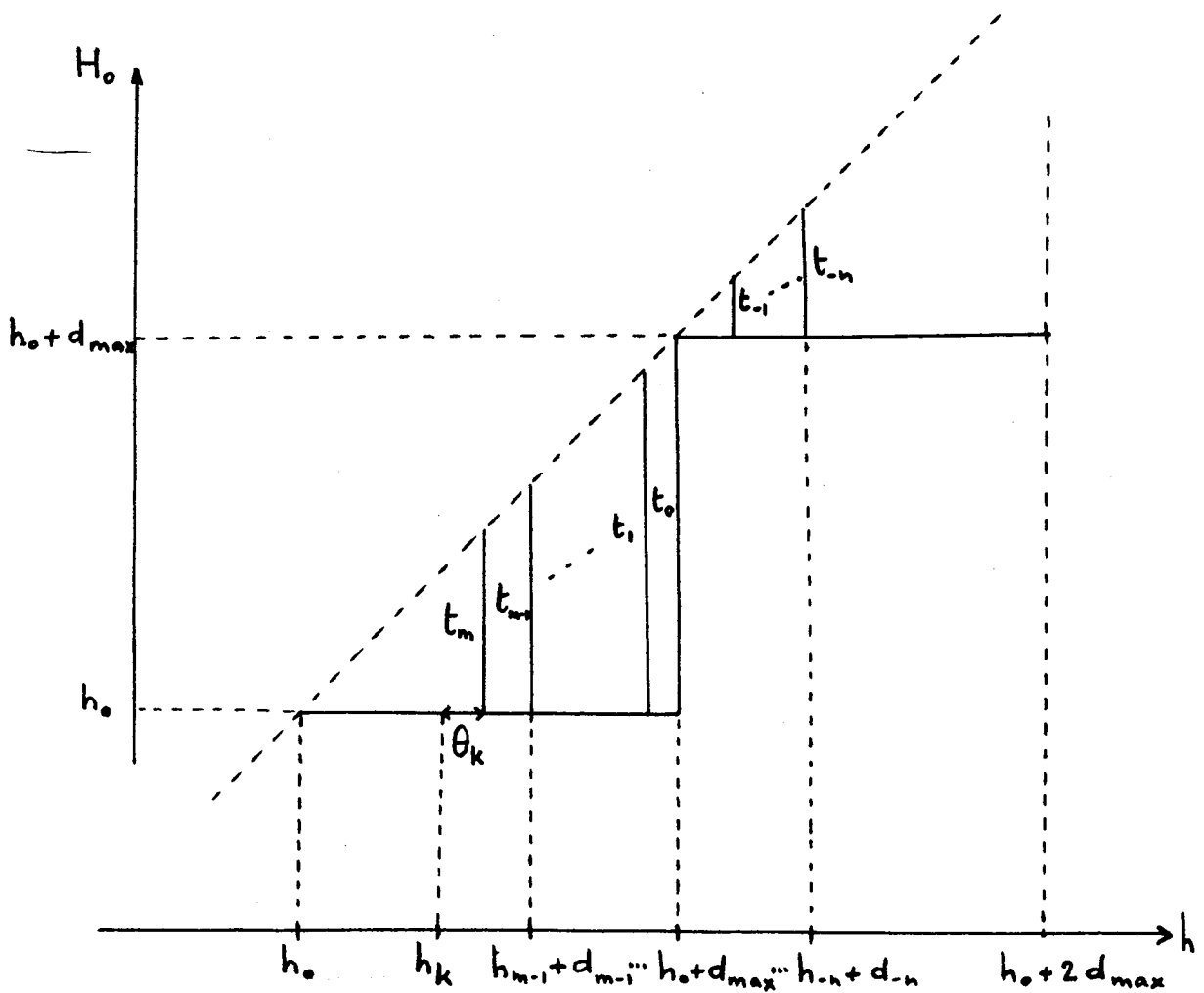
- Soit lorsque le temps restant  $\theta$  dans l'horloge  $\emptyset S$  consultée à une demande de réveil a une valeur inférieure ou égale à  $\epsilon$ . C'est alors le module chargé de prendre en considération les demandes de réveil (STIMER) qui devra effectuer les actions correspondantes à l'arrivée à expiration d'une demande de réveil.
- Soit lorsque l'horloge  $\emptyset S$  a effectivement atteint la valeur 0, et la routine d'exit (TEXIT) activé dans ce cas par les modules de traitement d'interruptions externes du système  $\emptyset S$  a rendu activable ( $\$P\emptyset ST$ ) le processus STIME du système TELCØM et celui-ci a été effectivement activé avant qu'une nouvelle demande de réveil ait eu lieu. Une description en pseudo algol du processus STIME est également fournie dans le document [F.5].

Dans ces deux cas :

- a) Si l'entrée  $e_o$  ne se trouve pas un sommet de pile, on effectue l'action correspondante à l'entrée se trouvant en sommet de pile, on supprime cette entrée. On regarde si l'entrée suivante contient la même valeur, auquel cas (s'il ne s'agit pas de l'entrée  $e_o$ ) on effectue l'action correspondante à cette nouvelle entrée et on la supprime. Ceci jusqu'à l'occurrence d'une entrée  $j$  telle que  $t_j \neq t_m$  ou jusqu'à l'entrée  $e_o$ . Si l'entrée  $e_o$  arrive ainsi en sommet de pile, on effectue le même traitement que dans le cas où l'entrée  $e_o$  se trouve initialement en sommet de pile, défini en b) ci-dessous sinon on initialise l'horloge  $\emptyset S$  à la valeur  $t_j - t_m$ .
- b) Si l'entrée  $e_o$  se trouve en sommet de pile, on la supprime du sommet de la pile et on l'insère en bas de pile. On incrémente  $H_o$  de  $d_{max}$ . On initialise l'horloge  $\emptyset S$  à la valeur se trouvant dans l'entrée en sommet de pile.

Une représentation imagée du mécanisme décrit dans ce chapitre peut être obtenue sur le graphique de la fonction  $H_0(h)$  en considérant que les heures relatives de réveil  $t_i$  de la pile PHR sont fournies par l'écart à l'heure  $h_i + d_i$  (à laquelle le réveil  $i$  doit arriver à expiration) entre la diagonale et la représentation de la fonction  $H_0$  :

$$t_i = h_i + d_i - H_0(h_i + d_i) \quad \forall i \in [-n, +m]$$



Le mécanisme de gestion d'horloges au niveau des processus du système TELCOM décrit ici diffère de celui implémenté sur la première version développée sur une machine virtuelle du système CP essentiellement sur les deux points suivants :

- La variable  $H_0$  est incrémentée périodiquement de la valeur  $d_{\max}$ , alors que dans la version précédente elle était soit incrémentée de  $d_{\max}$  lorsque le temps  $d_{\max}$  est écoulé depuis le dernier incrément de  $H_0$ , soit forcée à l'heure courante lorsqu'un utilisateur effectue une demande de réveil pour une heure qui dépasse de plus de  $d_{\max}$  la valeur de  $H_0$  au moment de la demande de l'utilisateur. Dans ce cas il faut alors remettre à jour la pile des heures relatives de réveil. On voit que cette situation sera assez rarement réalisée si les délais  $d_i$  des demandes de réveil des utilisateurs sont tous petits comparativement à  $d_{\max}$ .
- On tient compte de l'écart minimum  $\epsilon$  entre deux heures de réveil, au dessous duquel on peut considérer les heures comme égales, ce qui permet de diminuer de façon appréciable le nombre d'instructions à dérouler dans le cas où la fréquence moyenne des demandes de réveil est suffisamment élevée (ce qui augmente la probabilité d'obtenir des réveils à des heures voisines), sinon le nombre d'instructions déroulées peut, au contraire, être augmenté, du fait qu'il faut exécuter des instructions supplémentaires pour déceler les écarts inférieurs à  $\epsilon$ .

En conclusion, ce type de gestion d'horloges, un peu plus élaboré que le premier implémenté, ne se justifie pleinement que dans le cas où les processus utilisateurs du système effectuent un nombre tel de demandes de réveil que le temps consacré à leur traitement devient important comparativement à l'unité de temps utilisée pour effectuer les demandes de réveil. On voit même qu'il serait souhaitable de faire augmenter la valeur de l'écart minimum  $\epsilon$  quand on augmente le nombre de processus effectuant des demandes de réveil. Autrement dit, quand on augmente le nombre de demandes on perd de la précision mais le fait d'en tenir compte permet de diminuer le nombre d'instructions exécutées.

#### 4. COMPATIBILITÉ DES VERSIONS CP ET ØS DU SYSTÈME TELCØM.

##### LES SERVICES TRANSPORTABLES.

L'effort réalisé dans la mise en place du système TELCØM en tant que sous-système de multiprogrammation du système ØS-MVT pour permettre à tout programme se déroulant sous le contrôle d'un processus utilisateur d'être transporté, au coût d'un simple réassemblage, depuis la version du système TELCØM développée sur une machine virtuelle gérée par CP a permis d'effectuer ce transport pour la plupart des modules concernant les trois services successifs suivants :

- la procédure de télétransmission chargée de la gestion de la ligne téléphonique reliant le noeud CIGALE au 360,
- la station de transport du réseau CYCLADES offrant ses services d'accès au réseau à des processus du système TELCØM,
- un concentrateur de terminaux légers clients des services de temps partagé du réseau CYCLADES.

Ces transferts successifs ont été réalisés dès que les modules correspondants étaient écrits et les premiers tests effectués, ce qui a permis d'effectuer une mise au point plus élaborée en travaillant simultanément sur les deux systèmes d'exploitation. Cette méthode de travail présente quand même une contrainte assez sévère du fait que toute modification effectuée sur les programmes sources de l'un des systèmes TELCØM doit aussitôt être reportée sur l'autre, mais elle a permis une évolution simultanée des applications développées sur chacun de ces deux systèmes.

Une méthode de travail différente a été choisie pour la mise en place de la version suivante (système SYNCØP et niveau 2 de la station de transport CYCLADES; elle consiste à ne développer qu'un seul programme source pour les modules comportant d'importantes séquences de code communes aux versions CP et ØS et à différencier les deux cas à l'aide d'instructions d'assemblage conditionnel.



Pour mieux justifier cet effort de compatibilité entre les versions CP et ØS du système TELCOM on va fournir, pour chacun des 3 services auxquels on s'intéresse ici, le nombre de modules concernés et leur taille totale en octets, pour l'implémentation de la station de transport de niveau 1.

service	nombre de modules	taille en octets
Procédure de ligne	2	3528
Station de transport	26	20232
Concentrateur de terminaux	1	4345
Total	29	28105

Une description assez précise de ces services est fournie par la thèse de J.P. ANSART [A.1]. On va en faire ici une présentation sommaire en s'attachant aux quelques particularités de l'implémentation sous ØS-MVT.

#### 4.1 La procédure de télétransmission

La réseau de commutation de paquets CIGALE ne permettait initialement que des connexions à ses noeuds suivant la procédure de télétransmission TMM-UC.

Pour la connexion du 360 du CICG il a fallu développer une procédure BSC (Basic Synchronous Communication) assez voisine de celle utilisée par la méthode d'accès RTAM (Remote Terminal Acces Method) des systèmes HASP et ASP et l'implémenter sur les noeuds de CIGALE ainsi que sur le système TELCOM. L'expérience acquise par la mise en place d'un support de terminaux lourds sous CP (documentée par [P.1]) a été particulièrement profitable à ce travail.

Après la définition de la procédure de télétransmission choisie, décrite sous forme d'automate, par le document [I.5] son implémentation a été réalisée simultanément à l'IRIA sur CIGALE et au CICG sur les versions CP et ØS du système TELCOM. La mise au point a pu se faire sans trop de difficultés du fait que les logiciels correspondants étaient prêts pour des tests sensiblement à la même date et qu'un mécanisme assez complet de trace a été prévu dans TELCOM qui disparaît de plus de mémoire libre que CIGALE. Il s'agit de deux types de trace :

- celle des états successifs de l'automate mémorisés dans une zone mémoire de taille fixe allouée à cet effet,
- celle des paquets émis et des paquets reçus qui est enregistrée dans les F-espaces gérés par CP dans la version de TELCOM sur machine virtuelle et dans une zone mémoire de taille fixe dans la version de TELCOM sous ØS. Ces traces sont supprimées dans la version d'exploitation pour éviter de mobiliser trop d'espace mémoire sous ØS. La possibilité de les obtenir à nouveau si nécessaire, a cependant été conservée en utilisant des variables d'assemblage conditionnel qui permettent de spécifier si on désire un assemblage avec ou sans trace.

Les autres particularités de la version ØS concernent le bloc de contrôle associé à l'unité d'entrée-sortie (UCT), la définition du "normal channel en appendage" et "abnormal channel end appendage" et l'appel du superviseur d'entrée-sortie du système ØS. Il ne s'agit là que de respecter le mécanisme de gestion des opérations d'entrée-sortie choisi dans l'implémentation du système TELCOM sous ØS et décrit dans le paragraphe 2.3.

## 4.2 La station de transport CYCLADES

La station de transport du réseau CYCLADES constitue la plus importante partie de code qui a pu être transportée depuis CP sans modifications.

Le rôle essentiel d'une station de transport est de multiplexer à l'émission sur le réseau de commutation et démultiplexer à la réception des voies logiques de communications entre paires d'utilisateurs distants. Ces voies logiques de communication s'appellent des voies virtuelles dans le protocole ST-ST version 1 défini par le document [I.1] et des flots dans le protocole version 2 défini par [I.2]. Les programmes utilisateurs de ces voies de communication s'appellent des abonnés. L'écriture de ces programmes ne peut se faire qu'à l'aide d'une description précise des services mis à la disposition des abonnés par la station de transport et de leurs conventions d'appel. Cette description est fournie par les documents [A.4] pour la version A et [D.1] pour la version 2.

Actuellement les abonnés ne peuvent être implémentés que comme des processus du système TELCOM (pour le niveau 1) ou du système SYNCØP (pour le niveau 2). Une extension envisagée consiste à permettre leur implémentation, dans la version CP sur toute machine virtuelle gérée par CP et dans la version ØS sous le contrôle de toute tâche activée pour un programme utilisateur.

## 4.3 Le concentrateur de terminaux

Un concentrateur de terminaux développé sous CP a également pu être transporté sous ØS avec quelques modifications concernant en particulier la gestion des opérations d'entrée-sortie.

Il s'agit en fait d'un ensemble de processus, associés chacun à un terminal du 360 (soit un 2741, soit un télétype) qui sont initialement en attente de

réception d'une commande "Login" sur le terminal qui leur est associé. Selon le nom de l'utilisateur spécifié dans la commande "Login" le terminal correspondant devient soit une console opérateur du système TELCØM soit un terminal client du réseau CYCLADES pouvant travailler avec tout serveur conversationnel du réseau respectant le protocole appareil virtuel [I.3].

Dans les deux cas la méthode d'accès utilisée est la même (module DRCØNS), seule diffère la logique de traitement des messages reçus et des réponses émises (module CØNS dans le cas de la console opérateur et module CT dans le cas d'un client conversationnel CYCLADES).

**4.3.1 La console opérateur** accepte un ensemble de commandes qui a évolué en cours de développement du système et qui est constitué des commandes suivantes dans la version d'exploitation :

- Display permet d'obtenir le contenu, en hexadécimal de un ou plusieurs mots successifs de la mémoire centrale sur la console opérateur.
- Dump permet d'arrêter le système TELCØM en remettant dans leur état initial les informations externes au système TELCØM qui ont été modifiées à son initialisation et d'obtenir un dump de TELCOM sur imprimante (listé par le système ASP).
- Echo permet d'émettre un pli écho à une station du réseau Cyclades.
- Local permet le passage de la station de transport en mode local, c'est-à-dire que tout paquet dont le destinataire est aussi l'émetteur sera directement remis au processus de réception de la station de transport, sans passer par un noeud du réseau de commutation.
- Logout permet à l'utilisateur qui a fait "Login" de libérer son terminal qui devient alors à nouveau disponible soit en temps que console opérateur, soit en temps que client conversationnel du réseau Cyclades.
- Map permet d'obtenir l'adresse de début de tout module du système TELCØM.
- Msg permet d'envoyer un message sur un autre terminal du concentrateur sur lequel un utilisateur s'est déclaré à l'aide de la commande 'Login'.

- Nlocal Permet le passage de la station de transport en mode non local, c'est-à-dire que les paquets seront émis sur le réseau de commutation même si le destinataire est l'émetteur.
- Ordre Permet d'émettre un message vers la console opérateur d'une autre station de transport du réseau Cyclades en suivant des conventions établies au cours des premières implémentations de la station de transport de niveau spécifiant que l'opérateur est représenté par l'abonné 0.
- Return Arrêt normal du système TELCOM, de même que pour la commande 'dump' mais sans sortie de dump.
- Size Permet d'obtenir des informations sur la mémoire disponible dans la partie allouée et libérée de façon dynamique, sous la forme suivante :
  - nombre de trous
  - nombre d'octets libres
  - taille du trou le plus grand
- Start Permet de créer et d'activer (ou d'activer seulement, s'ils sont déjà créés) les processus chargés de la gestion des lignes synchrones, soit pour la liaison avec le noeud du réseau, soit pour le support d'un terminal lourd.
- Stat Permet d'obtenir des statistiques concernant la gestion d'une ligne synchrone. Il s'agit du nombre total de transmissions effectuées et du nombre d'anomalies détectées en fin d'opération d'entrée-sortie soit dans les octets d'état du canal, soit dans les octets de "sense" de l'unité de contrôle de télétransmission (2701)
- Stop Permet de mettre en attente un processus chargé de la gestion d'une ligne synchrone. Il pourra être réactivé à l'aide d'une commande "Start" nommant la même ligne.
- Store Permet de ranger dans un mot se trouvant sous la clé de protection associée au système TELCOM une valeur spécifiée en hexadécimal.
- z Permet d'envoyer un message sur la console opérateur d'un client de traitement par lots du réseau Cyclades connecté au serveur OS. Une forme de cette commande permet également d'émettre un message sur toutes les consoles des clients connectés.

### 4.3.2 Les clients conversationnels du réseau Cyclades

La disponibilité des terminaux du 360 pendant les heures d'exploitation réservées au système ØS-MVT a facilité la mise en place de clients conversationnels du réseau Cyclades et a permis d'affecter quelques uns de ces terminaux aux utilisateurs des serveurs conversationnels du réseau, essentiellement au serveur SIRIS8 sur IRIS80.

Le mode d'utilisation de ces terminaux, clients conversationnels Cyclades, est très voisin de celui des terminaux attachés au concentrateur de terminaux développé sur machine virtuelle à l'aide de la commande DIAL du système CP, lequel est décrit par la note technique n°12 [A.3].

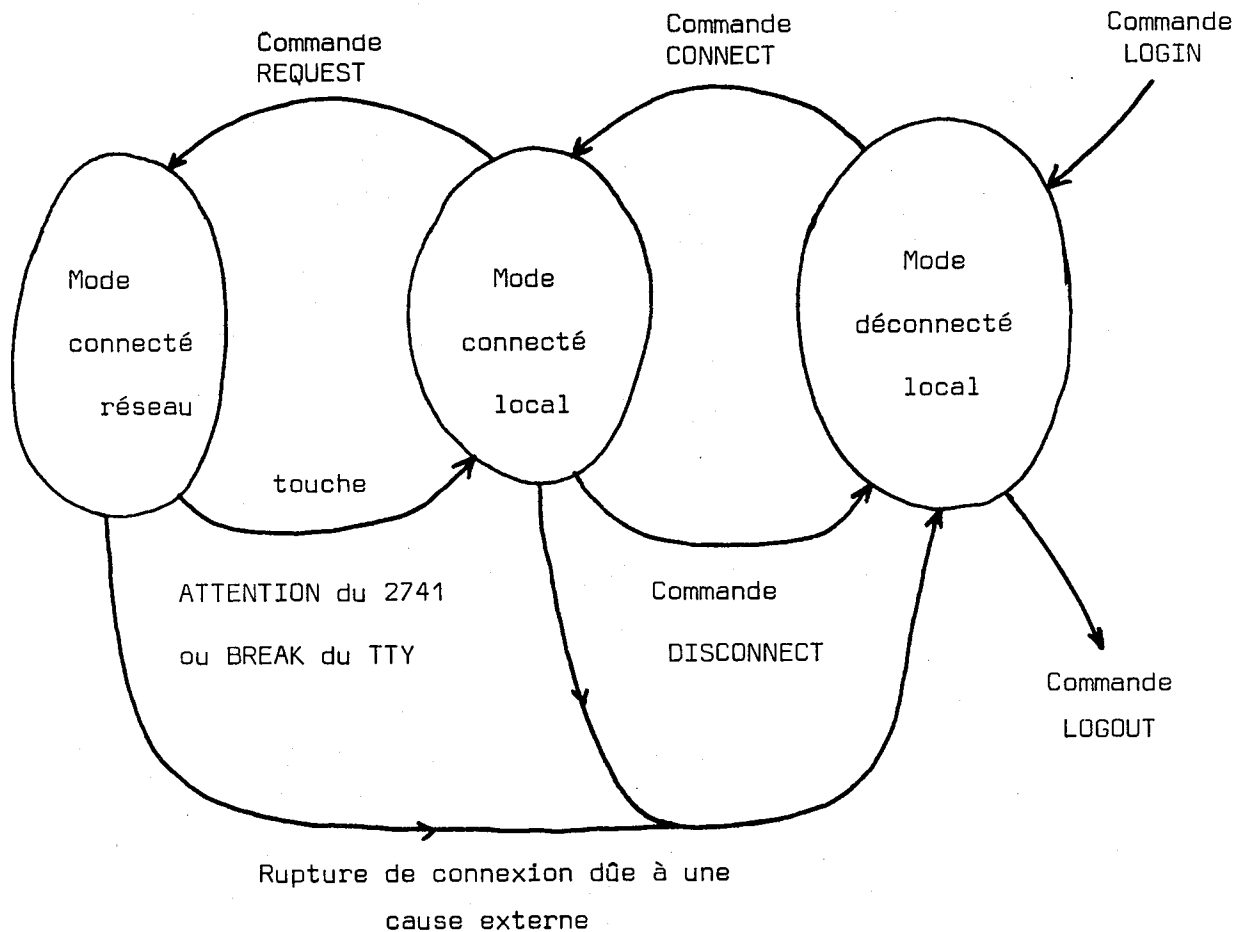
Il n'en diffère que par les points suivants :

- a) La correspondance Appareil virtuel-Appareil réel gérée par le système CP, n'existe pas sous le système ØS, ce qui fait que seuls les quelques terminaux réels (5 ou 6 selon les besoins) déclarés dans les cartes de contrôle utilisées pour lancer le système TELCOM sous ØS sont disponibles pour les utilisateurs.

Un utilisateur qui dispose de l'un de ces terminaux après avoir reçu le message l'informant que Cyclades est accessible par ce terminal, ou après la commande 'Logout' de l'utilisateur précédant, peut utiliser la commande 'Login' en spécifiant un nom prédéclaré pour passer en mode déconnecté local du concentrateur de terminaux. Les noms prédéclarés sont les suivants :

CLIENT1,CLIENT2,...,CLIENT5

Il se trouve alors dans une situation analogue à l'utilisateur du concentrateur de terminaux disponible sous CP, c'est-à-dire qu'il dispose d'un ensemble de commandes analysées localement par le concentrateur (connect, request, attn, disconnect, upcase, def), qu'il peut passer en mode connecté local à l'aide de la commande 'CONNECT' puis en mode connecté réseau à l'aide de la commande 'REQUEST', qu'il peut quitter ces modes à l'aide de la touche attention du 2741 ou BREAK du télétype, de la commande 'DISCONNECT' comme le spécifie le graphe suivant :



- b) Une correspondance est établie entre certains terminaux du concentrateur et des pseudo consoles opérateurs du système ASP.
- Le mécanisme de gestion des pseudo consoles opérateurs ASP a été développé pour permettre à un client de traitements par lots du réseau Cyclades d'accéder aux commandes du système ASP et est décrit par la note technique n°34 [F.4].
- Dans le but de mettre à la disposition des opérateurs des consoles supplémentaires en cas de pannes des consoles utilisées normalement on considère que toute commande émise par un utilisateur du concentrateur de terminaux en mode local (déconnecté local ou connecté local) et qui débute par le caractère \* est destinée au système ASP. Cette commande est donc transmise à une pseudo console opérateur ASP associée à l'utilisateur

qui a fait "login" en simulant une attention puis une lecture et la réponse du système ASP est récupérée par simulation d'une ou plusieurs écritures, les messages correspondants sont alors écrits sur le terminal de l'utilisateur.

On voit que ce mécanisme permet d'utiliser le même terminal aussi bien comme console opérateur du système ASP que comme client d'un serveur conversationnel, ce qui fournit la plus grande disponibilité possible du terminal.

## 5. LE SERVEUR DE TRAITEMENTS PAR LOTS

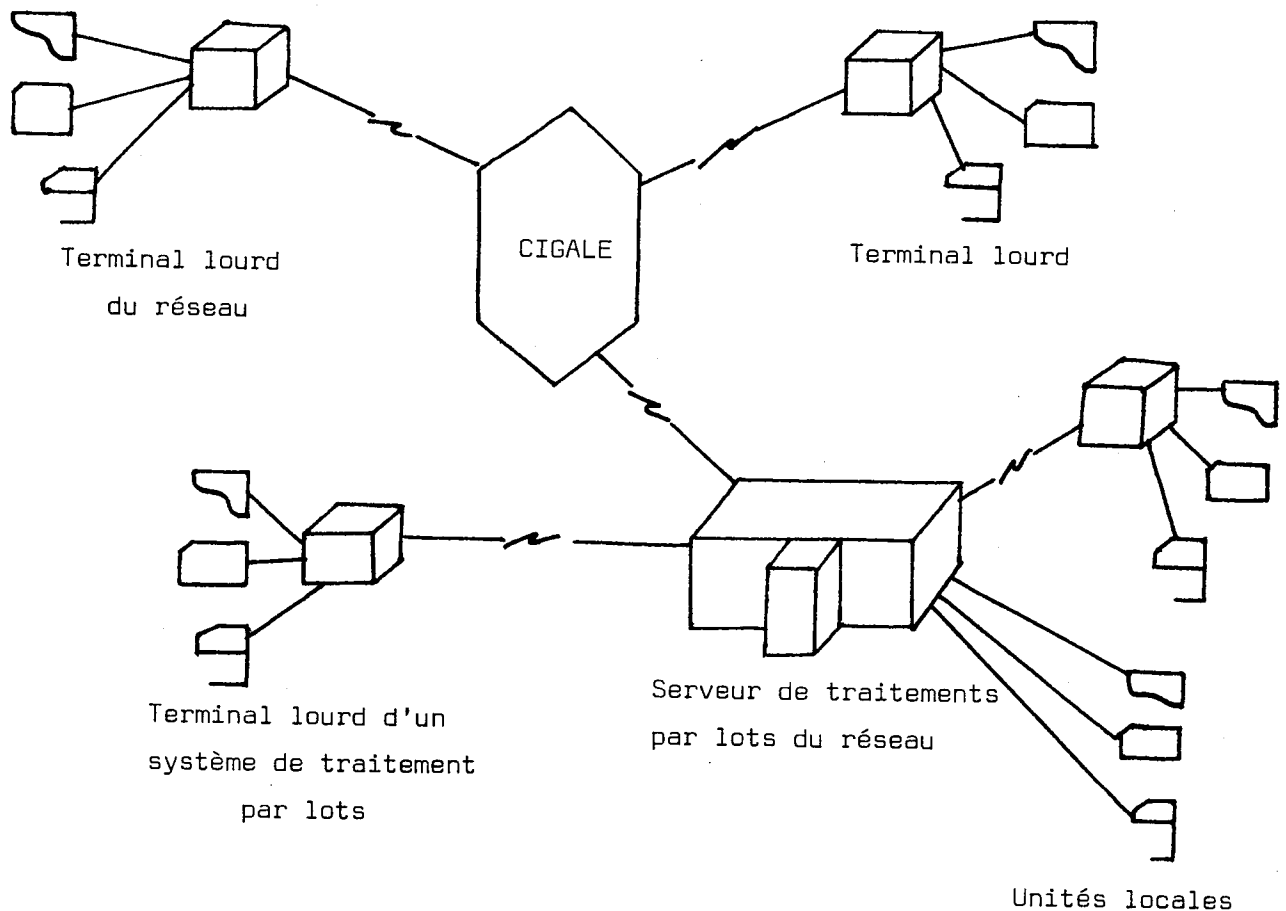
Le but essentiel du logiciel développé sous le système ØS pour lui permettre de communiquer avec les autres centres participants du réseau Cyclades est de mettre les ressources accessibles par l'utilisateur local (ou relié par liaison téléphonique directement au 360) à la disposition des utilisateurs du réseau Cyclades.

On peut considérer que l'application développée est une extension du support de terminaux lourds géré par le système ASP. Un terminal lourd du système ASP est un petit ordinateur possédant une console, un (ou plusieurs) lecteur de cartes, une (ou plusieurs) imprimantes, relié au 360 par une liaison téléphonique gérée par une procédure de télécommunication synchrone (Basic Synchronous Communication) et disposant d'un logiciel qui, en respectant des conventions de multiplexage et démultiplexage sur la ligne synchrone des informations lues et à écrire sur les unités d'entrée-sortie du terminal lourd, appelées "multi-leaving" [U.1], permet de gérer celle-ci comme des unités supportées de façon locale.

On appelle maintenant terminal lourd d'un réseau d'ordinateurs, un petit ordinateur analogue à celui qui vient d'être défini du point de vue matériel,



mais dont le logiciel inclut une station de transport et, pour chaque unité d'entrée-sortie, une mise en correspondance de l'appareil réel avec un appareil virtuel suivant un protocole d'appareil virtuel [I.3] commun à tous les utilisateurs du réseau de telle façon que les unités d'entrée-sortie du terminal lourd soient gérées de manière analogue aux unités locales d'un serveur de traitements par lots.



On voit que l'avantage essentiel d'un terminal lourd d'un réseau d'ordinateurs par rapport à un terminal lourd connecté directement à un ordinateur tient au fait que l'utilisateur peut accéder à chacun des serveurs de traitements

par lots du réseau avec une seule liaison téléphonique au réseau de commutation. De plus il peut connecter ses différents appareils à différents serveurs.

Deux extensions possibles de ces services consistent à :

- permettre la connexion simultanée d'un appareil à plusieurs serveurs
  - réaliser des concentrateurs de terminaux lourds
- et seront présentées plus loin.

On s'intéresse ici à la mise en place sur le système ØS-MVT géré par ASP d'un serveur de traitements par lots du réseau Cyclades permettant la connexion simultanée de plusieurs terminaux lourds.

Du fait que la station de transport du réseau Cyclades a été implémentée sous le contrôle du système TELCØM et que ses services ne sont accessibles que par des processus de TELCØM, les premiers problèmes qui se posent consistent à trouver des moyens de communication entre des processus du système TELCØM et des fonctions du système ASP. Il s'agit, en particulier, de pouvoir gérer depuis un processus du système TELCØM une pseudo console opérateur du système ASP, de pouvoir émettre à partir d'un processus du système TELCØM un flot de travaux dans la file d'attente du système ASP, de pouvoir récupérer dans un processus du système TELCØM une liste émise par le système ASP après l'exécution d'un travail destinant des listes à un terminal lourd du réseau.

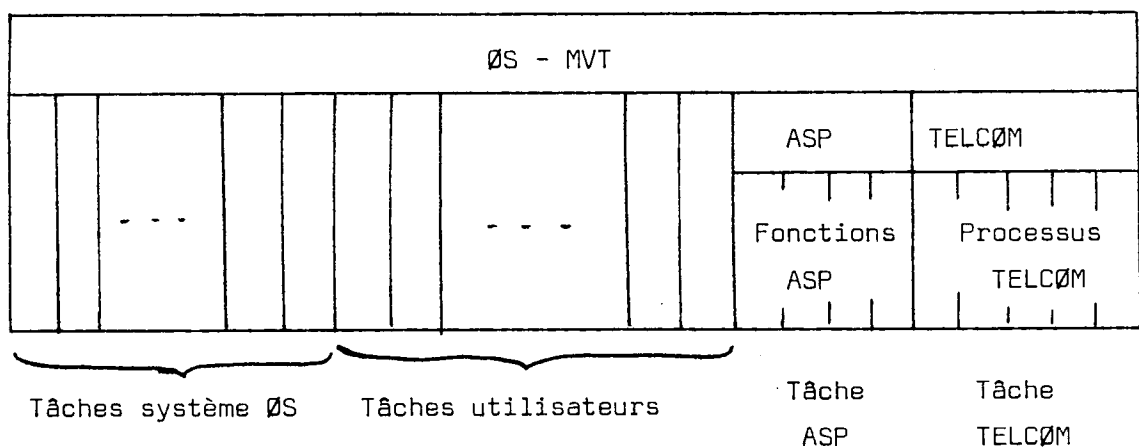
Les deux derniers points (soumission de travaux et réception de listes) ont été résolus, dans un contexte tout à fait analogue, par le support d'une option de temps partagé du système ØS, TSO (Time Sharing Option). Nous avons utilisé le même mécanisme de communication : l'adaptateur de canal à canal, ou du moins sa simulation par logiciel du fait que l'on utilise le système ASP en mode local.

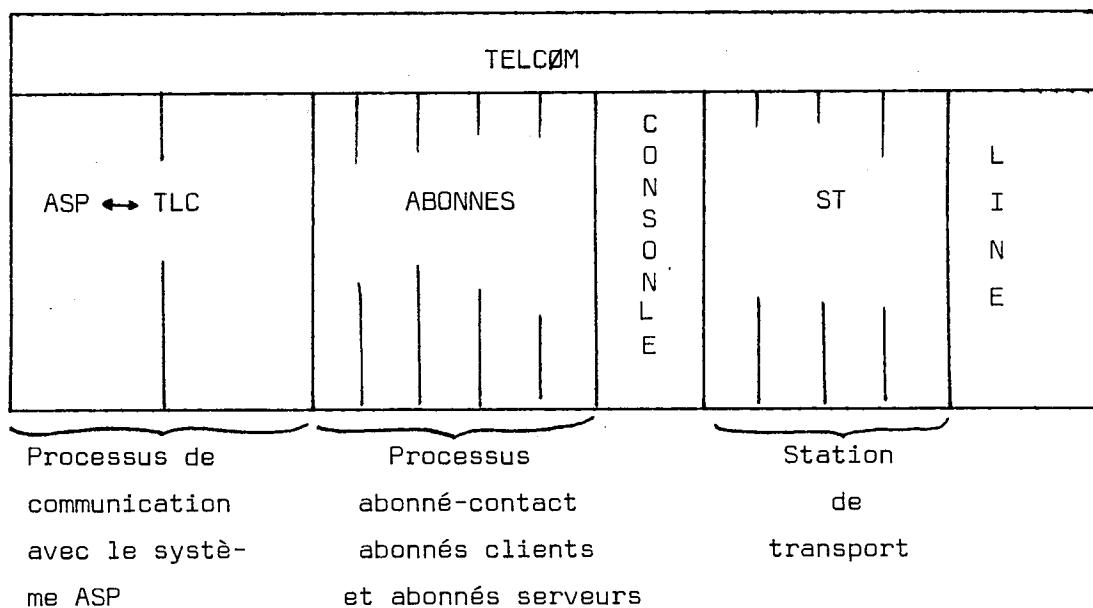
Mais ce mécanisme ne permet pas de gérer un nombre variable de voies de communications au gré des demandes de connexion des utilisateurs, ce qui nous a conduits à développer une file d'attente sur unité à accès direct.

C'est cet ensemble de moyens (station de transport sous TELCØM, moyens de communications avec ASP, file d'attente réseau) qui a permis de développer, sous le contrôle de processus du système TELCOM, un ensemble d'abonnés serveurs de traitement par lots activables dynamiquement.

Pour permettre des tests avant d'arriver à cette solution assez longue à élaborer, nous avons développé un mécanisme d'accès à des fichiers séquentiels du système ØS par des processus du système TELCØM, ce qui a permis de réaliser des abonnés serveurs capables de lister sur l'imprimante d'un terminal lourd client du réseau certains fichiers choisis par le client. Pour réduire l'encombrement du système implémenté, nous avons été conduits à supprimer ce mécanisme de la version d'exploitation mais on le présentera tout de même ci-dessous.

Les deux schémas suivants rappellent les environnements de multiprogrammation sous lesquels les applications présentées ici ont été développées :





## 5.1 Mécanisme d'accès aux fichiers séquentiels du système ØS

Une méthode d'accès a été développée sous TELCØM pour permettre à un ensemble de processus d'utiliser des fichiers séquentiels du système ØS pour différents services.

### 5.1.1 La méthode d'accès (module DRSK)

L'implémentation de la méthode d'accès respecte la logique de conception du système TELCØM : c'est un ensemble de sous-programmes constituant un seul module et dont les différents points d'entrées sont fournis par un vecteur de transfert se trouvant au début du module. Les opérations d'entrée-sortie se font par appel au superviseur d'entrée-sortie du système ØS, ou, pour certaines opérations nécessitant la consultation ou la modification de données propres à la gestion des fichiers par le système ØS (telles

que consultation du catalogue, allocation d'un nouveau fichier ...) par appel de tâches de travail qui sont des tâches filles du système TELCØM développées à cet effet.

Il existe des processus spécialisés qui accèdent aux sous-programmes constituant la méthode d'accès. Chacun de ces processus est associé à un et un seul fichier ØS à un instant donné et les caractéristiques du fichier correspondant ainsi que les opérations en cours sont fournies par un UCT et un bloc de contrôle supplémentaire appelé ØDEFIC.

On va présenter successivement les différents sous-programmes constituant la méthode d'accès.

#### a) Initialisation

Selon le type d'appel effectué, spécifié par des octets d'état dans le bloc descripteur de fichier ØDEFIC, l'initialisation se réalise de façon différente.

Si nécessaire une consultation du catalogue ØS est effectuée par appel de la tâche de travail CTASK. Le nom du fichier ØS est fourni à l'appel et le nom du volume sur lequel se trouve le fichier est fourni en retour. Puis les entrées de la TIØT (Task Input Output Table) sont consultées pour trouver celle associée au volume contenant le fichier concerné. On rappelle que sous ØS une entrée de la TIØT est associée à une carte contrôle de définition de fichier (carte DD). Il faut donc, au démarrage du système TELCØM prévoir une carte DD par volume susceptible d'être utilisé et une carte DD par processus chargé de la gestion des fichiers. Quand l'entrée de la TIØT concernée a été trouvée, son contenu permet de mettre à jour l'entrée de la TIØT associée au processus demandeur ainsi que l'adresse de l'UCB (Unit Control Block) dans l'UCT.

On peut alors acquérir de la mémoire pour initialiser un JFCB (Job File Control Block) et, selon le type d'appel effectué :

- on active la tâche de travail FTASK pour une demande d'allocation de fichier et on met le processus appeleur en attente. La tâche FTASK active les routines d'allocation de fichier du système ØS par appel de la SVC 32 puis réveille le processus demandeur.

- ou bien on active la tâche de travail ØTASK pour effectuer un open de type J du fichier concerné. Dans ce cas on continue l'initialisation par appel de la macro instruction \$IØAPPN pour insérer les "appendages" comme convenu pour la mise en place des méthodes d'accès sous TELCØM sous ØS. Puis, dans le cas où l'initialisation se fait pour une lecture le même BIØ sera constamment réutilisé et un programme canal ayant la structure suivante est initialisé :

```

    ↻ search id eq
      tic
      read data
  
```

dans lequel la première commande adresse l'identifieur qui a été initialisée dans l'IØB à l'adresse de la première piste du fichier avec un numéro d'enregistrement égal à 1 et qui servira aussi au superviseur d'entrée-sortie du système ØS pour lancer une commande "seek" permettant de positionner le mécanisme d'accès sur le cylindre adressé et de sélectionner la tête de lecture écriture correspondant à la piste concernée.

#### b) Lecture

La lecture consiste à récupérer l'enregistrement logique suivant dans l'enregistrement physique en cours de traitement et précédemment lu s'il existe, sinon on acquiert de la mémoire pour un nouvel enregistrement physique, on insère son adresse dans la commande de lecture du programme canal précédemment initialisé et on appelle le superviseur d'entrée-sortie du système ØS pour effectuer une lecture, on range en tête de l'enregistrement physique l'adresse du premier octet libre, et on se retrouve dans la situation qui permet de récupérer un enregistrement logique dans l'enregistrement physique précédemment lu. Quand un enregistrement physique est épuisé, on prépare la lecture du suivant en incrémentant l'adresse d'enregistrement dans l'IØB puis on libère la mémoire occupée par l'enregistrement totalement lu.

c) Ecriture

En écriture il s'agit de construire progressivement un enregistrement physique à partir des enregistrements logiques fournis par l'appelant et de l'écrire sur disque dès qu'il est complet.

La mémoire nécessaire pour la construction d'un enregistrement physique est acquise à l'appel lorsqu'il n'y a pas d'enregistrement physique en cours de construction et est libérée après son écriture sur disque.

La mémoire acquise par l'appelant pour fournir un enregistrement logique est libérée dès l'insertion de l'enregistrement logique dans l'enregistrement physique.

Le BIØ est fourni par l'appelant et le programme canal est reconstruit toutes les fois qu'un enregistrement physique doit être écrit sur disque.

Il a la structure suivante :

```

    ↪ search ID eq
    TIC
    write count key & data
  
```

Après l'opération d'entrée-sortie la mémoire occupée par le BIØ est libérée et l'adresse courante d'écriture sur disque est incrémentée dans l'IØE

Un code retour est fourni à l'appelant pour lui indiquer que le dernier enregistrement du fichier vient d'être écrit si nécessaire.

d) Fermeture

Dans le cas d'un fichier qui vient d'être utilisé en écriture, si un enregistrement physique est en cours de construction il est écrit sur disque, éventuellement tronqué, puis la zone mémoire qu'il occupait est libérée.

Une tâche de travail, ØTAST, est activée pour effectuer le "close" ØS du fichier puis la zone mémoire contenant le JFCB qui avait été acquise à l'initialisation est libérée.

### 5.1.2 Les tâches de travail

Pour éviter de mettre la totalité du système TELCOM, qui se déroule sous le contrôle d'une seule tâche ØS, en attente il a fallu activer certains services du système ØS nécessitant l'exécution d'opérations d'entrée-sortie sous le contrôle de tâches spécialisées. Ces tâches sont activées quand le processus du système TELCOM demandeur du service correspondant effectue un "POST" sur un ECB associé au service. Quand le service a fini de s'exécuter la tâche qui en a fait la demande au système ØS réveille le processus demandeur qui a pu ainsi se mettre en attente sans bloquer les autres processus.

Ces tâches de travail créées par le module d'initialisation du système TELCOM sont les suivantes :

a) ØTASK

Cette tâche est chargée d'effectuer les appels des SVC ØPEN et CLØSE du système ØS pour l'ensemble des processus qui en ont fait la demande.

b) CTASK

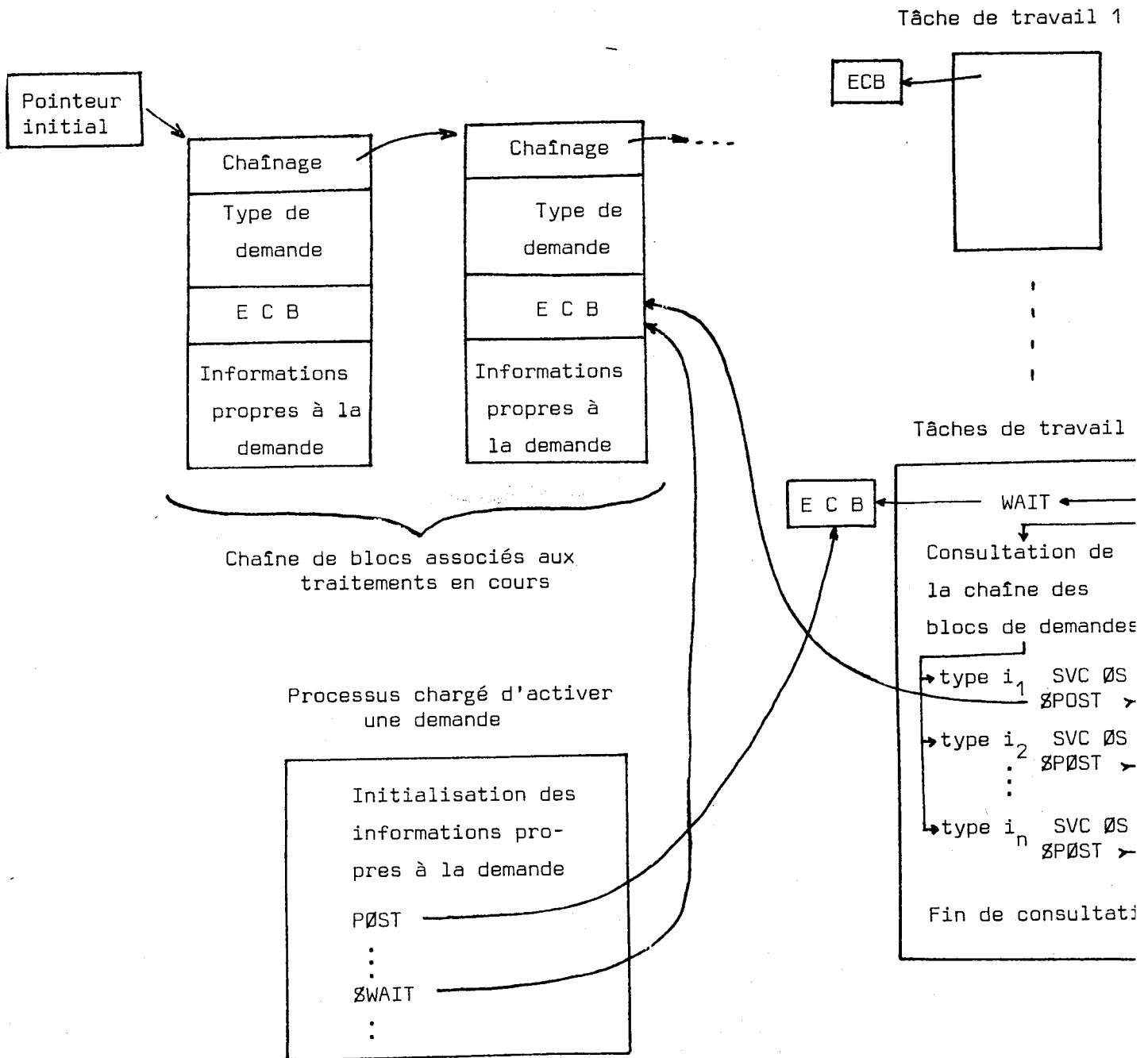
Cette tâche est chargée d'effectuer l'appel des services de maintenance du catalogue ØS. Elle effectue les cataloguages, décataloguages et consultations du catalogue (SVC 26) pour l'ensemble des processus qui en ont fait la demande.

c) FTASK

Cette tâche est chargée d'effectuer la maintenance de la VTØC (Volume Table of Contents) d'un volume à accès direct géré par le système ØS. Elle effectue les allocations (SVC 32) et suppressions (SVC 29) des fichiers ØS ainsi que la consultation de l'espace libre sur un volume demandé (SVC 78), pour l'ensemble des processus qui ont fait une demande de ce type.



Le schéma ci-dessous montre le mécanisme d'activation de ces tâches de travail.



### 5.1.3 Les processus de traitement (DSK)

Il existe un ensemble de processus destinés à manipuler des fichiers du système ØS auxquels aucune fonction particulière n'est affectée initialement. Lorsque l'un de ces processus est activé l'action qu'il doit réaliser lui est spécifiée par des octets d'état qui ont été initialisés par le processus qui a pris l'initiative de l'activer. Ces différentes actions sont les suivantes :

a) Emettre un fichier ØS sous forme de lettres régulières

Ce service est l'un des premiers implémentés pour permettre des tests de compatibilité entre les stations de transport de niveau 1 implémentées sur IRIS 80 et sur 360. Il permet des échanges de fichiers, images de cartes, sous forme de lettres régulières, entre des abonnés particuliers de chaque station de transport, sur l'initiative de l'opérateur de la station émettrice.

b) Recevoir un fichier ØS sous forme de lettres régulières.

C'est le processus de réception d'un fichier séquentiel associé au service défini ci-dessus.

c) Allouer un fichier

L'initiative d'allouer un fichier ØS est prise par l'opérateur et c'est le processus chargé de la gestion de la console opérateur qui transmet à un processus DSK la demande d'allocation avec les informations nécessaires (nom du volume sur lequel le fichier doit être alloué, nom du fichier et nombre de pistes). Après un appel de la séquence d'initialisation de la méthode d'accès décrite plus haut, il suffit d'informer l'opérateur de la fin normale d'allocation ou de l'impossibilité d'allouer le fichier demandé.

d) Supprimer un fichier

De même que pour l'allocation, l'initiative de supprimer un fichier est prise par l'opérateur du système TELCØM. Le processus opérateur recherche, s'il n'est pas spécifié explicitement par l'opérateur, le nom du

volume sur lequel se trouve le fichier spécifié dans le catalogue du système ØS à l'aide de la tâche de travail CTASK puis transmet ces informations à un processus DSK qui demande à la tâche de travail FTASK d'effectuer la suppression du fichier concerné. Il ne reste alors plus qu'à informer l'opérateur de la fin normale ou anormale de la suppression du fichier.

e) Consulter le catalogue

Ici aussi il s'agit d'une demande de l'opérateur qui fournit un nom de fichier ØS et auquel on indique en retour (après appel de la tâche CTASK pour consultation du catalogue) soit que le fichier concerné n'est pas catalogué, soit le nom du volume sur lequel se trouve ce fichier.

f) Demander l'espace disponible sur un volume

Il s'agit du traitement d'une demande de l'opérateur qui fournit un nom de volume et auquel on indique en retour (après appel de la tâche FTASK pour consultation de la VTØC du volume) le nombre de trous, la taille du plus grand trou et le nombre total de pistes disponibles sur le volume concerné.

g) Effectuer le cataloguage d'un fichier

L'opérateur fournit le nom du fichier et le nom du volume et le processus DSK effectué, par appel de CTASK, le cataloguage correspondant puis informe l'opérateur de la fin normale ou anormale de l'opération.

h) Effectuer le décataloguage d'un fichier

De même que pour le cataloguage mais l'opérateur ne fournit que le nom du fichier.

i) Emettre un fichier sur voie virtuelle

Il s'agit là d'un service qui a été mis en place pour permettre au plus vite des tests de compatibilité de la station de transport de niveau 1 implémentée sur le système ØS avec le client terminal lourd en cours de développement sur MITRA 15.

Un processus permanent ABC (abonné contact) gère un abonné de la station de transport qui reste en attente de lettres régulières provenant d'autres stations de transport du réseau. Sur réception d'une lettre de demande de contact respectant le protocole client-serveur de terminaux [I.4] pour une connexion de type imprimante un processus DSK est activé et sera chargé d'activer un abonné d'exploitation, d'ouvrir une voie virtuelle, d'initialiser le fichier ØS dont le nom se termine par les 4 caractères de l'entrée du service spécifié dans la demande de contact (les caractères précédents sont un préfixe standard qui affecte le fichier aux utilisateurs de Cyclades) puis d'émettre ce fichier sur la voie virtuelle jusqu'à sa fin ou jusqu'à rupture de la connexion, et enfin fermer le fichier, couper la voie virtuelle et désactiver l'abonnement de l'abonné exploitation. On voit que ce mécanisme permet de lister sur l'imprimante d'un terminal lourd client de traitements par lots du réseau Cyclades un fichier catalogué du système ØS (au choix de l'utilisateur) qui se trouve sur un volume résident et dont le nom commence par un préfixe imposé.

#### 5.1.4 Les commandes affectées à la gestion des fichiers

On a vu que la plupart des services réalisés par les processus chargés de la gestion des fichiers sont activables sur commande opérateur. On va dresser la liste de ces commandes en spécifiant leur syntaxe et leur fonction.

a(bort) r name

Permet d'arrêter la réception du fichier de nom "name" si elle est en cours.

Le champ "name" indique, dans toutes ces commandes, soit le nom complet d'un fichier ØS (il doit alors être spécifié entre quote et est limité à 24 caractères), soit le nom simple du fichier auquel est affecté un préfixe standard affecté aux utilisateurs de Cyclades (il est alors limité à 8 caractères).

al{locate) volser name trks

Permet d'allouer pour le fichier de nom "name" sur le volume "volser" un nombre de pistes égal à "trks".

c{atlg) volser name

Permet de cataloguer le fichier de nom "name" en spécifiant qu'il se trouve sur le volume "volser"

locat(e) name

Permet de consulter le catalogue pour obtenir le volume sur lequel doit se trouver le fichier de nom "name".

r{receive) name abloc abdis

Active un processus DSK pour la réception du fichier de nom "name" sous le contrôle de l'abonné "abloc" en provenance de l'abonné "abdis".

Scratch valser name

Permet de supprimer le fichier de nom "name" sur le volume "volser".

S{end) name abloc abdis

Active un processus DSK pour l'émission du fichier de nom "name" par l'abonné "abloc" à destination de l'abonné "abdis".

Sp{ace) volser

Permet d'obtenir des informations sur l'espace disponible du volume "volser" (nombre de trous, taille du plus grand trou et nombre total de pistes libres).

u{ncatlg) name

Permet de décataloguer le fichier de nom "name".

### 5.1.5 Considérations générales sur le mécanisme implémenté

La mise en place de ce mécanisme d'accès aux fichiers ØS par des processus du système TELCØM s'est justifié par le fait qu'il a permis rapidement des tests de compatibilité entre client et serveur de traitements par lots. Si l'encombrement en mémoire centrale de l'ensemble des applications développées sous TELCØM sous ØS n'avait pas conduit à supprimer ce mécanisme dans la version d'exploitation on aurait pu réaliser sans grosses difficultés les extensions suivantes :

- accès aux commandes de gestion des fichiers spécifiés ci-dessus pour tout terminal du concentrateur de terminaux,
- mise en place d'un éditeur conversationnel permettant de faire la mise à jour d'un fichier ØS, image de cartes, depuis un terminal du concentrateur,
- émission d'un fichier ØS, considéré comme un flot de travaux, vers le système ASP sur l'initiative d'un utilisateur du concentrateur de terminaux.

Bien que ce travail de gestion de fichiers ØS dans un environnement réseau n'ait pu évoluer comme on le souhaitait initialement, on peut considérer qu'il s'agit d'une toute première approche du problème de transfert de fichiers sur un réseau d'ordinateurs.

## 5.2 Les moyens de communications avec le système ASP

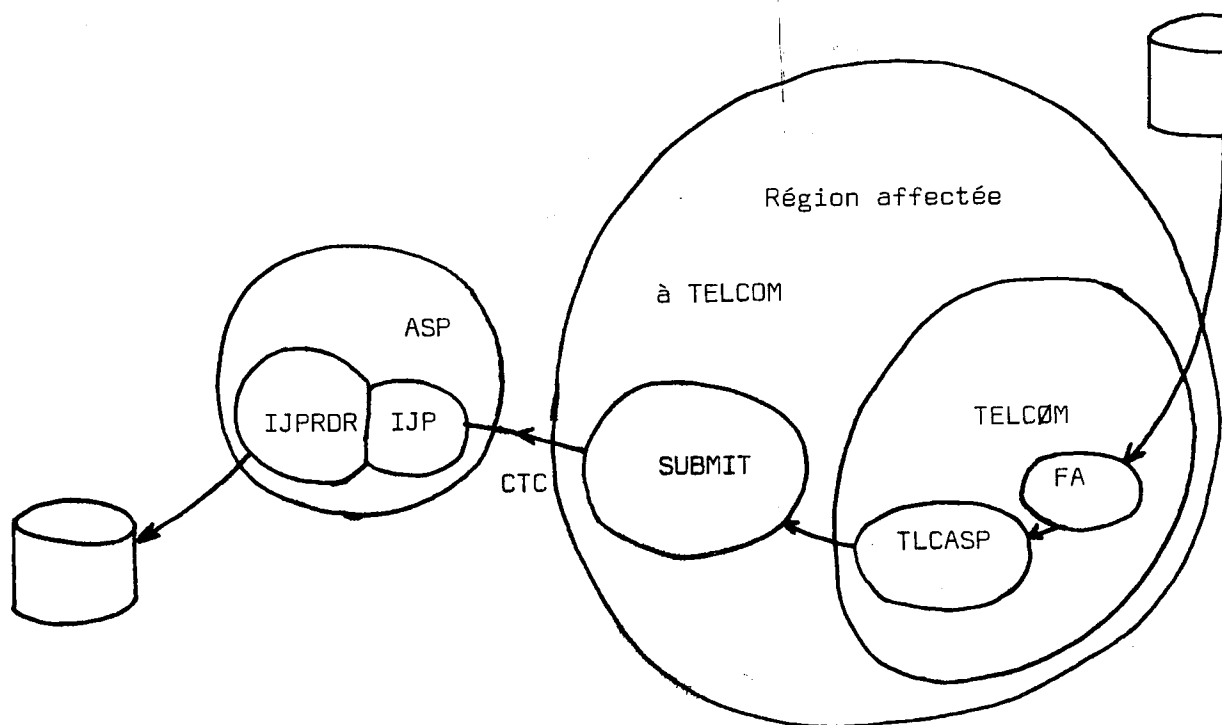
### 5.2.1 La soumission des travaux

Un mécanisme de réception de flots de travaux émis par une tâche du système ØS a été mis en place par l'équipe chargée du développement du système ASP. Il s'agit du service IJP (Internal Job Processing) qui peut recevoir un flot de travaux sur l'une des voies de l'adaptateur de canal à canal (ou de sa simulation par logiciel en mode local) et le transmettre au module IJPRDR qui se charge de l'analyse des cartes de contrôle dans le but de séparer les

différents travaux et d'effectuer une préparation des phases successives du traitement de chacun d'eux. Le travail effectué par le module IJPRDR est en fait tout à fait analogue à celui effectué par le module RDR du système ASP qui lit les flots de travaux sur les lecteurs de cartes (locaux ou distants) ou par le module TØS qui les lit sur bande magnétique, il n'en diffère que par le fait qu'il est activé dynamiquement par IJP qui lui transmet les images de carte qu'il reçoit d'une tâche OS. La soumission des travaux au service IJP du système ASP ne peut pas se faire depuis un processus du système TELCØM car elle nécessite l'utilisation de services ØS tels que WTØR et ØPEN qui seraient bloquants pour tous les autres processus (mise en attente de la tâche TELCØM jusqu'à la fin de l'exécution du service requis).

Ce service de soumission des travaux au système ASP a donc été implémenté sous le contrôle d'une tâche de travail, appelée SUBMIT, qui est une tâche fille de la tâche sous laquelle est développé le système TELCØM, créé par le module d'initialisation de TELCØM.

Le fait d'implémenter ce service sous le contrôle d'une tâche ØS rajoute le problème de communication entre un processus du système TELCØM et cette tâche



Du fait que la tâche SUBMIT se trouve dans la région affectée à TELCOM, c'est-à-dire qu'elle travaille sous la même clé de protection que le système TELCOM, il est facile, pour cette communication de prévoir deux ECB et une zone d'échange accessibles par les deux tâches. Il suffit alors de considérer que la zone d'échange est initialement disponible pour le processus émetteur (TLCASP), dès qu'il possède des informations à transmettre il les range dans cette zone puis en informe la tâche SUBMIT par un PØST(ØS) sur le premier ECB. Dès que la tâche SUBMIT a fini de traiter les informations reçues, elle informe le processus TLCASP que la zone d'échange est à nouveau à la disposition de l'émetteur par un ØPØST(TELCØM) sur le deuxième ECB, et on se retrouve dans la situation initiale. On travaille en fait alternativement sur deux zones d'échanges distinctes pour pouvoir en libérer une pendant le traitement de l'autre. Quand à la communication entre la tâche SUBMIT et le système ASP elle est gérée suivant les conventions imposées par le logiciel IJP du système ASP, c'est-à-dire que la tâche SUBMIT commence par informer le système ASP qu'elle est prête à émettre un flot de travaux en émettant un message à l'opérateur avec attente de réponse (WTØR) sur la console primaire du système ØS. Du fait que l'on travaille avec ASP, la console effectivement utilisée est la console alternée de la console primaire, c'est-à-dire l'adaptateur de canal à canal. Le message respecte une syntaxe définie par ASP et est transmis à IJP. En cas de réponse favorable par ASP la tâche SUBMIT peut alors effectuer un ØPEN (ØS) sur l'unité CTC (unité déclarée comme un support de bande magnétique pour les méthodes d'accès du système ØS mais affectée à une position logique de l'adaptateur de canal à canal par logiciel) qui lui a été affectée par une carte de contrôle du système ØS (carte DD). Il est alors possible d'utiliser la méthode d'accès QSAM du système ØS ("Queued Sequential Access Method") pour émettre par carte le flot de travaux en provenance de TELCOM sur le CTC, à destination du service IJP.

La fin de l'émission d'un flot de travaux a lieu soit anormalement, sur impossibilité d'émettre sur le CTC par exemple, auquel cas on informe ASP par un message opérateur codifié qui spécifie à IJP qu'il faut abandonner le traitement du flot en cours, soit normalement sur réception de fin de fichier



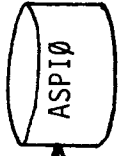
émise par le processus TLCASP, auquel cas la tâche SUBMIT effectue un CLOSE(ØS) sur l'unité CTC affectée à la liaison, puis informe ASP par un message opérateur codifié qui demande au système ASP de prendre en considération le flot de travaux émis.

La tâche SUBMIT effectue alors une nouvelle demande à ASP pour l'émission d'un flot de travaux (par WTØR) et, dès réponse favorable, se remet en attente de réception de cartes par le processus TLCASP en provenance de la file d'attente du système TELCØM.

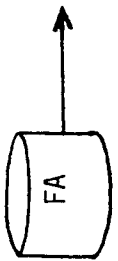
Le schéma ci-dessous explicite les mécanismes de communication entre les processus TELCØM et le système ASP par l'intermédiaire de la tâche de travail SUBMIT. Les requêtes qui concernent la gestion de la file d'attente du système TELCØM sont symbolisées par :

§ OPEN	FA	Ouverture en lecture d'un flot de travaux
§ READ	FA	Lecture du flot de travaux
§ ABØRT	FA	Fermeture du flot de travaux avec perte des informations déjà lues.

et sont commentées avec plus de précision dans le paragraphe concernant la gestion de la file d'attente réseau du système TELCØM.



SOUSSION D'UN FLOT DE TRAVAUX AU SYSTEME ASP



T L C A S P

Processus du système TELCØM

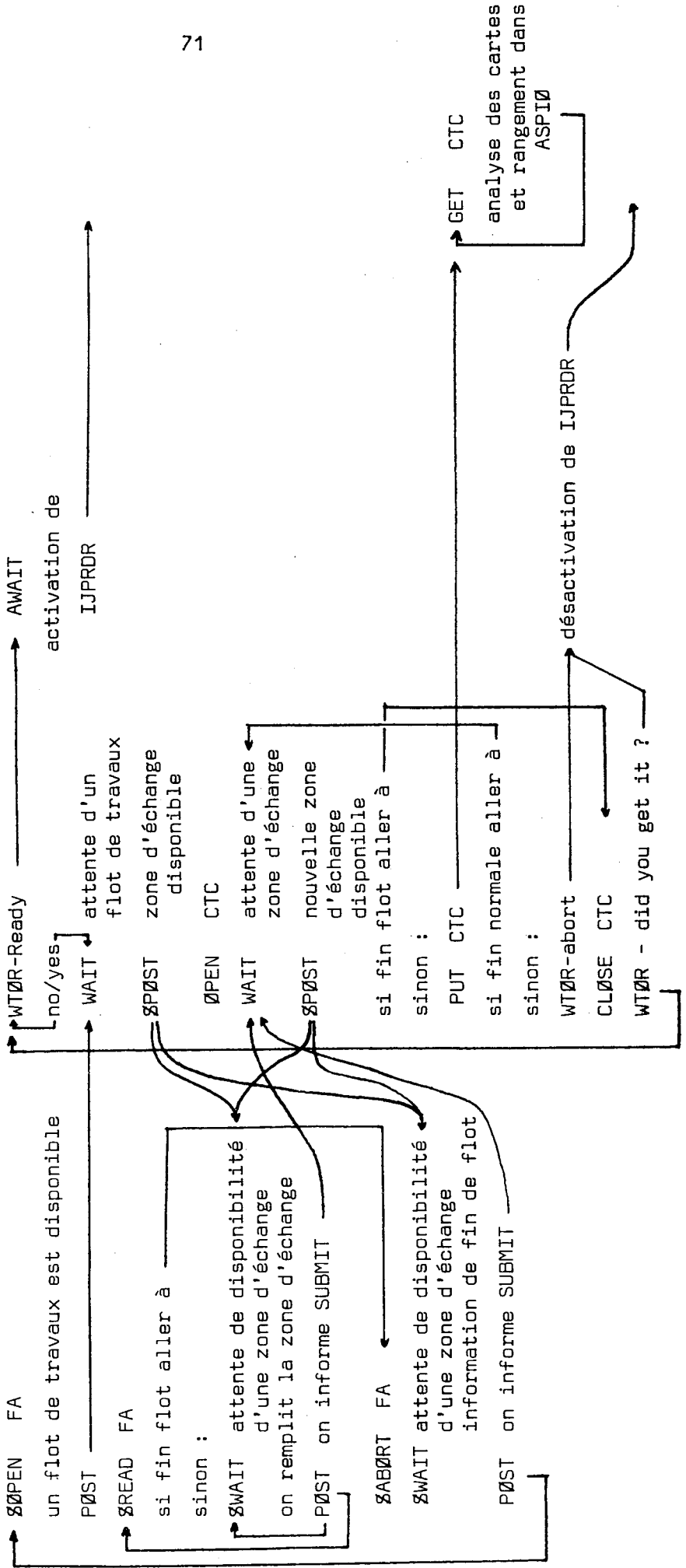
S U B M I T

Tâche du système ØS

A S P

I J P

I J P R D R



### 5.2.2 La réception des listes

Un DSP (Dynamic Support Program) a été développé par le système ASP pour permettre l'émission de fichiers de type listes à destination des utilisateurs de l'option de temps partagé TSØ du système ØS.

L'activation de ce DSP, CRØS, se fait sur demande de l'utilisateur qui peut spécifier à l'aide de cartes contrôles du système ASP (les cartes /\*FØRMAT) les fichiers qu'il veut émettre, pour un travail donné, à destination d'un utilisateur de TSØ.

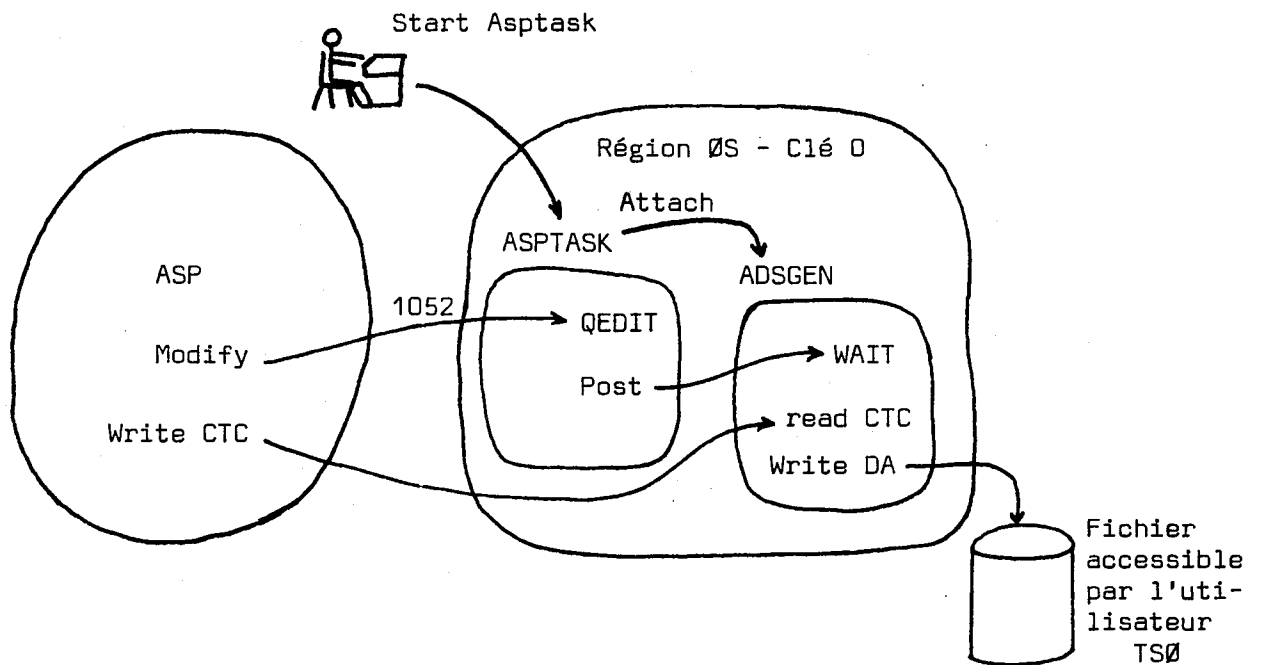
Du fait que le système ØS exploité au CIGG ne peut pas supporter l'option TSØ (qui défavoriserait trop les utilisateurs de traitement par lot, vue la configuration possédée) le mécanisme ainsi développé est disponible pour permettre l'émission des listes, non plus à destination d'un utilisateur nommé de TSØ, mais à destination de l'imprimante d'un terminal lourd du réseau Cyclades, identifié par son nom d'usager.

Le transfert de fichier du système ASP vers l'utilisateur TSØ se fait par l'intermédiaire d'une tâche de travail, ASPTASK, qui se déroule en mode système et travaille sous la clé de protection 0 (tâche système, activable par commande opérateur).

Cette tâche reçoit une demande d'émission de fichier à l'aide de la commande opérateur "Modify" du système ØS qui lui est adressée par l'intermédiaire de la console alternée gérée par ASP, et qui lui demande d'activer la tâche ADSDGEN :

```
Modify ASPTASK,ADSDGEN,START
```

La tâche ADSDGEN, activée par ASPTASK sur réception de cette commande est alors chargée d'effectuer la réception du fichier émis par ASP sur une position CTC qui lui est affectée et de ranger ce fichier sur une unité à accès direct, sous une forme accessible par l'utilisateur TSØ.



Il a donc suffi d'écrire une nouvelle version de la tâche ADSGEN telle qu'elle émette vers un processus du système TELCØM le fichier reçu qui sera ensuite rangé dans la file d'attente réseau.

Pour que la tâche ASPTASK puisse reconnaître les commandes "Modify" qui lui sont adressées elle doit être déclarée en tant que "tâche système" du système ØS et lancée par commande opérateur. La tâche ADSGEN, tâche fille de ASPTASK se trouve donc affectée à la même région ØS de clé de protection nulle, ce qui fait qu'il n'est pas possible d'utiliser le même mécanisme de communication entre ADSGEN et un processus du système TELCØM que celui entre SUBMIT et le processus TLCASP.

Il a fallu développer à cet effet une gestion de pseudo unités dont une extension a permis une gestion de pseudo consoles opérateurs par le système ASP.

A chaque pseudo unité doit être associé un UCB (Unit Control Block) dans le noyau du système ØS qui correspond à une unité déclarée à la génération du système ØS. En pratique on a utilisé des UCB associés à des terminaux non

utilisés en exploitation sous ØS. Il suffit alors que la tâche (ADSGEN ici) qui veut communiquer avec un processus du système TELCØM s'effectue l'une de ces unités à l'aide d'une carte de contrôle (carte DD) et la gère par une méthode d'accès en déclarant un "SIØ Appendage" pour sa gestion. Cette routine est activée par le superviseur d'entrée-sortie du système ØS et effectue un appel au service ØPEXCP (voir annexe 1) du système TELCØM qui permet de réveiller le processus TELCØM associé à cette pseudo unité et retourne au superviseur d'entrée-sortie avec demande de ne pas lancer le SIØ réel. Le processus TELCØM ainsi réveillé qui se comporte en fait comme une unité d'entrée-sortie simulée par programme, peut analyser le programme canal qui a été construit par la tâche utilisatrice de la pseudo unité et effectue les traitements des informations transmises (write) ou charger les buffers de lecture (read) puis devra réaliser un travail analogue à celui effectué par le superviseur d'entrée-sortie sur réception d'une interruption signalant la fin normale d'une opération d'entrée-sortie. Le service ØCEDE (voir annexe 1) a été développé à cet effet, il permet de forcer à 1 les bits "Channel end" et "device end" dans les octets de "Status" du CSW (Channel Status Word) sauvegardé dans l'IØB, et d'effectuer un PØST(ØS) sur l'ECB adressé par l'IØB qui permet d'informer la méthode d'accès de la fin de l'opération d'entrée-sortie.

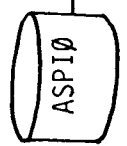
Une fois ce moyen de communication entre la tâche ADSGEN et le processus ASPTLC du système TELCØM développé il a suffit d'utiliser, dans la tâche ADSGEN le mécanisme de lecture de fichiers dans la file d'attente du système ASP à travers le CTC et le DSP CRØS développé pour le support de TSØ par le système ASP et le moyen de communication par pseudo unité pour émettre le fichier vers le processus ASPTLC.

Le rôle du processus ASPTLC est alors de reconnaître toutes les commandes d'écriture émises sur la pseudo unité gérées par ADSGEN, de regrouper les lignes adressées par ces commandes dans des blocs suivant le protocole appareil virtuel et d'émettre des requêtes vers le processus FA du système TELCØM pour l'écriture de ces blocs dans la file d'attente réseau.

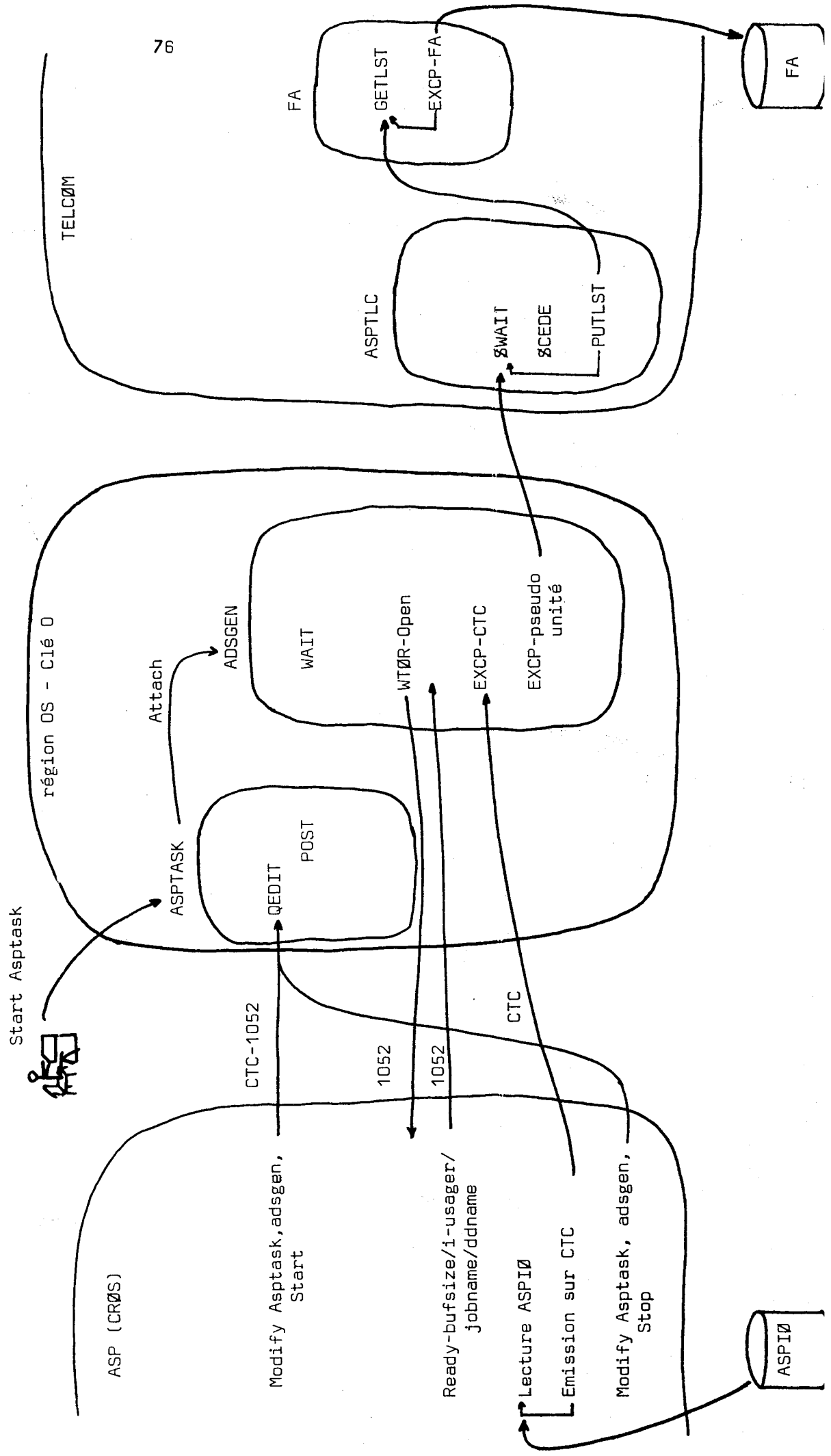
Le mécanisme de gestion de cette file d'attente est explicité plus loin. On remarque ici que la première ligne d'un fichier reçu par ASPTLC est particularisée. Elle contient des informations caractérisant le fichier qui ont été émises par CRØS en réponse au WTØR émis par ADGEN avant d'ouvrir un fichier CTC. Ces informations :

I-USAGER/JØBNAME/DDNAME

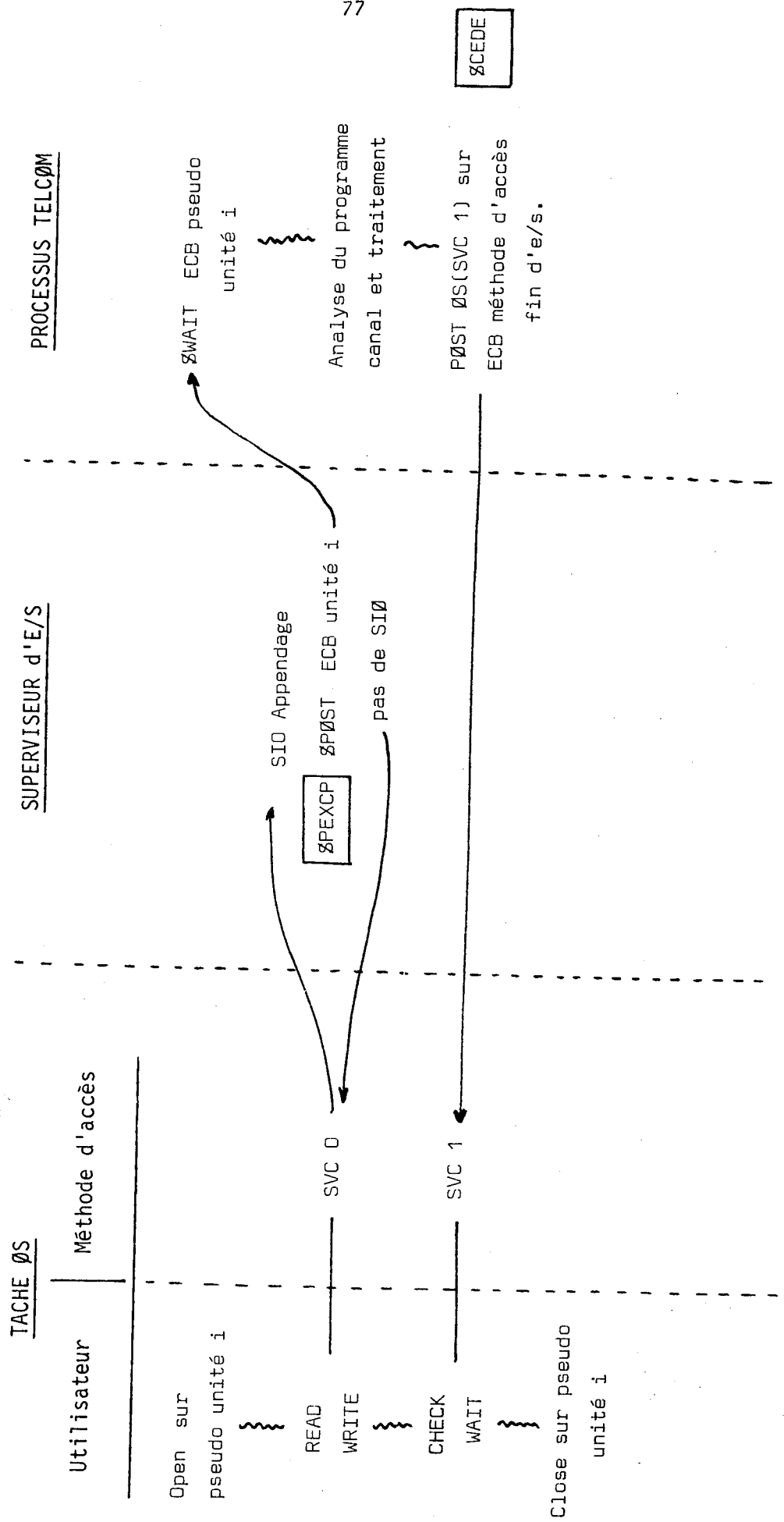
ne sont pas écrites dans le fichier inséré dans la file d'attente réseau mais serviront à l'identifier. Elles sont en fait écrites dans les blocs descripteurs de la file d'attente.



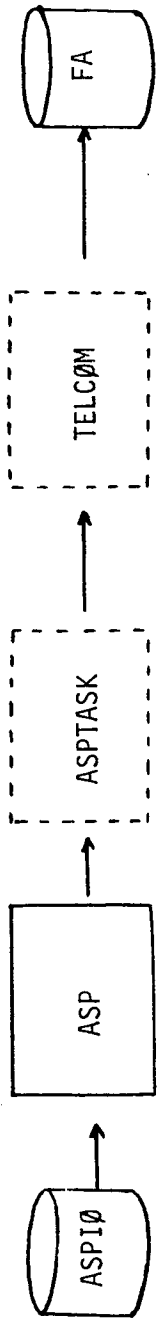
STOCKAGE DES FICHIERS DANS LA FILE D'ATTENTE DU SYSTEME TELCØM



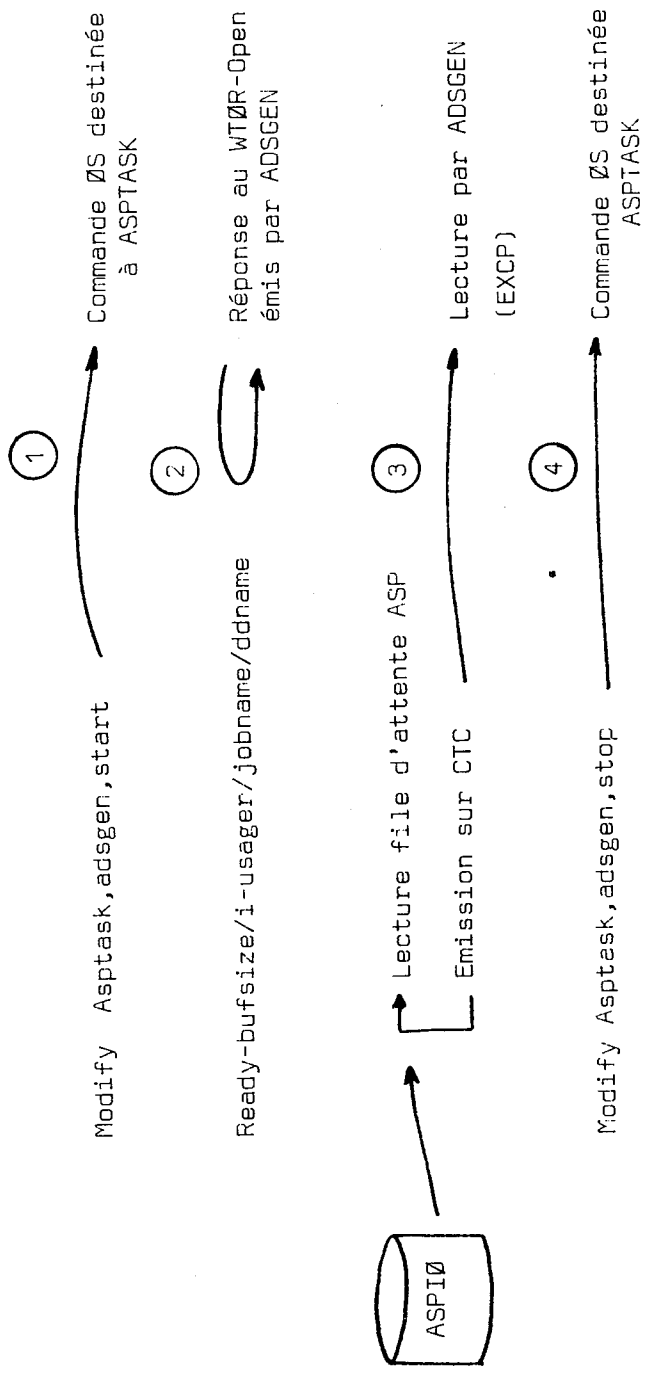
COMMUNICATION ENTRE UNE TACHE ØS ET UN PROCESSUS TELCØM PAR GESTION DE PSEUDO-UNITES

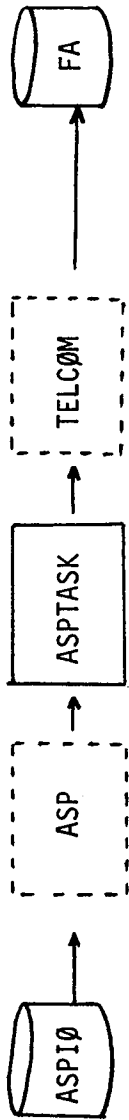






A S P





A S P T A S K

Vérification TELCØM actif et

ASPTASK non actif;

ATTACH ADSCEN ; - - - - -

Attente commande opérateur

si tâche ADSCEN terminée;

DETACH ADSCEN;

1 → si commande "Modify Asptask,adsgen,start"

reque;

PØST Start ECB;

si tâche ADSCEN active aller à

ATTACH ADSCEN; - - - - -

4 → si commande "Modify Asptask,adsgen,stop"

reque;

si tâche ADSCEN active :

PØST Stop ECB;

aller à

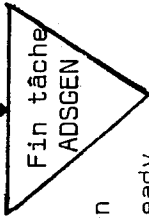
WTØ - INCØRRECT CØMMAND - MODIFY ASPTASK

aller à

A D S G E N

WAIT sur les ECB Stop et Start

Start



Emission WTØR-Open

2 → Attente réponse Ready

Open pseudo unité 1

Getmain CTC buffer

WTØR - Open CTC

3 → EXCP - lecture CTC

fin

GET - déblocage buffer CTC

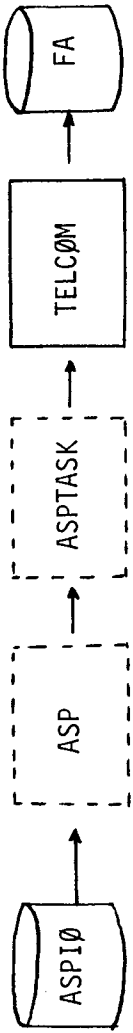
PUT - pseudo unité 1

eof sur pseudo unité 1

freemain CTC buffer

CLOSE pseudo unité 1

5



Processus FA

Processus ASPTLC

Attente d'une ligne sur pseudo unité 1  
 la première ligne d'un fichier contient :

I-USAGER/JOBNAME/DDNAME

ØPEN écriture a'un fichier (lignes) dans FA

Rangement des informations sous forme de message VTP  
 dans un bloc de travail

5

Attente d'une ligne sur la pseudo unité 1

si bloc plein

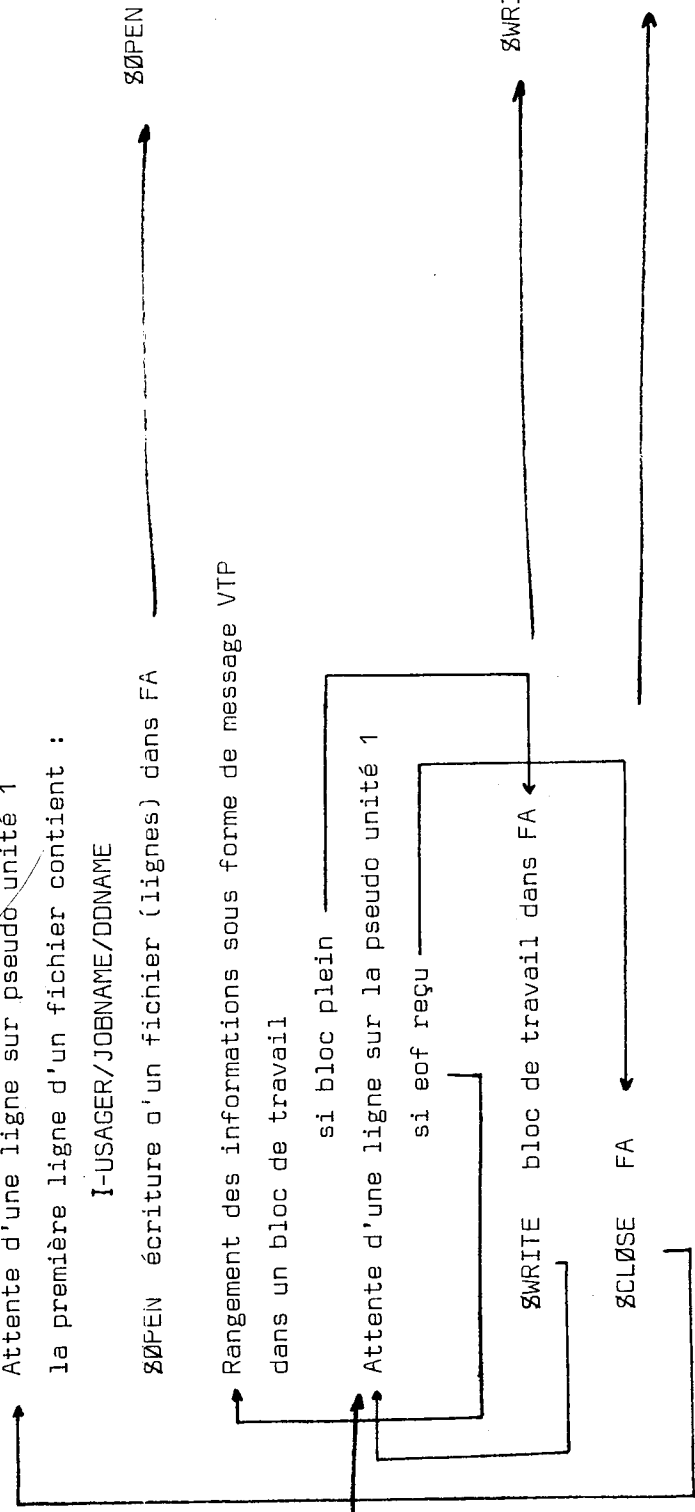
si eof reçu

ØWRITE bloc de travail dans FA

ØCLOSE FA

ØWRITE

ØCLOSE



### 5.2.3 Les pseudo consoles opérateur ASP

Il s'agit de déclarer un ensemble de pseudo consoles au système ASP dont la gestion soit conçue de telle façon qu'un processus du système TELCØM puisse se comporter comme un opérateur utilisant l'une de ces consoles. Ce mécanisme étant réalisé, il est alors possible de permettre à un client du réseau CYCLADES d'accéder aux commandes ASP en lui associant une pseudo console.

On va s'intéresser à la réalisation de ce mécanisme aussi bien du côté du système ASP qui supporte la pseudo console que de celui du système TELCØM qui l'utilise.

#### 5.2.3.1 Support de pseudo consoles par le système ASP

On a voulu que la gestion d'une pseudo console se rapproche le plus possible de celle d'une unité réelle, simple, le 1052. Pour éviter de déclarer un nouveau type de console au système ASP, on a choisi de déclarer des pseudo consoles en utilisant un type d'unités reconnues par le système ASP et n'existant pas dans notre configuration : les 1053.

Le principe de fonctionnement d'une pseudo console est le suivant :

- L'utilisateur, qui est en fait un processus TELCØM, doit "prendre la main" en émettant une attention lorsqu'il veut envoyer une commande opérateur au système ASP.

On voit alors que le système ASP doit réveiller le processus TELCØM utilisateur lorsque le programme canal de lecture est prêt; le processus TELCØM range alors sa commande dans la zone de lecture adressée par le CCW et simule la fin normale d'une opération de lecture.

- Lorsque le système ASP doit émettre un message sur une pseudo console, il prépare un programme canal d'écriture et réveille le processus TELCØM utilisateur de cette pseudo console qui doit alors récupérer le message et simuler la fin normale d'une opération d'écriture.

a) Lecture des commandes opérateur

Un module du système ASP (CØNTRAP) est chargé de traiter les interruptions asynchrones d'entrée-sortie pour les consoles opérateur ASP. Dans le cas d'une attention sur une pseudo console, le processus ASP chargé de la gestion de la pseudo console est réveillé, de la même façon qu'une attention sur un 1052 réveille le processus ASP chargé du support de ce 1052. De plus un bit associé à la console ASP (qui se trouve en fait dans l'octet inutilisé du CCW de lecture ou d'écriture) indique que le processus TELCØM utilisateur de la pseudo console est vivant. Si ce bit est à 0 à la réception d'une attention, il faut signaler à ASP que l'utilisateur est là (mise à 1 du bit) et simuler éventuellement la fin d'une précédente opération d'entrée-sortie lancée par ASP sur cette pseudo console et qui n'a pas pu se terminer du fait que le précédent processus TELCØM correspondant à cette pseudo console est mort avant d'avoir pu signaler la fin de l'opération d'entrée-sortie. Cette précaution est nécessaire pour éviter le blocage d'une pseudo console à la suite de la fin anormale d'un processus TELCØM utilisateur de la pseudo console.

La routine chargée du traitement des attentions (dans le module CØNS1053) déroulée sous le contrôle du processus ASP chargé de la gestion de la pseudo console, prépare alors un programme canal d'écriture (un seul CCW de code commande X'02') et un appel au superviseur d'entrée-sortie (EXCP) est alors fait par ASP pour l'exécution de cette opération d'entrée-sortie.

Du fait que la pseudo unité n'existe pas réellement un "appendage de SIØ" a été codé dans le module CØNS1053 et est activé par le superviseur d'entrée-sortie du système ØS avant de lancer l'instruction SIØ. Dans cet "appendage" un test est fait pour savoir s'il existe un processus TELCØM correspondant à cette pseudo console (une attention déjà reçue).

. Si un tel processus n'existe pas on simule aussitôt la fin de l'opération d'entrée-sortie pour ASP. Ceci correspond au cas où il y a des messages à émettre sur une pseudo console alors que le processus TELCOM utilisateur de cette pseudo console n'existe pas (ou plus).

- . S'il existe, un appel au service TELCOM  $\$PEXCP$  (pseudo EXCP) permet de le réveiller en effectuant un  $\$POST$  sur l'ECB associé à la pseudo console et un  $P\emptyset ST$  sur l'ECB de TELC $\emptyset M$ .

Dans les deux cas le retour au superviseur d'E/S du système OS se fait avec demande de ne pas lancer l'opération d'E/S.

Pour que le superviseur d'E/S du système  $\emptyset S$  puisse accepter les requêtes concernant les pseudo consoles il faut qu'à chacune de ces pseudo consoles corresponde un UCB (Unit Control Block) dans le système OS, même si l'unité réelle n'existe pas, c'est-à-dire qu'il a fallu déclarer des unités non existantes aux adresses des pseudo consoles au moment de la génération du système OS.

Du fait que le superviseur d'E/S d'OS n'a pas lancé effectivement l'opération d'E/S, il ne sera pas capable de reconnaître la fin de l'opération par une interruption CE+DE, il faut donc que le processus TELC $\emptyset M$  simule la fin de l'opération par une interruption asynchrone DE.

Ce type d'interruption sur une pseudo console est traité dans le module C $\emptyset NTRAP$ , il suffit alors de réveiller ASP en lui signalant la fin d'une E/S sur la pseudo console.

#### b) Envoi de messages à l'opérateur

Le sous-programme du module C $\emptyset NS1053$  activé pour émettre un message sur une pseudo console construit un programme canal constitué d'un seul CCW d'écriture. ASP appelle le superviseur d'E/S du système  $\emptyset S$  pour l'exécution de ce programme canal.

De même que pour une lecture un "appendage de SI $\emptyset$ " propre aux pseudo consoles est activé par le superviseur d'E/S avant l'exécution de l'instruction SI $\emptyset$ . Le processus TELCOM associé à la pseudo console, s'il existe, est alors réveillé par appel du service  $\$PEXCP$  (pseudo EXCP) et le retour au superviseur d'E/S  $\emptyset S$  se fait avec demande de non exécution du SI $\emptyset$ .

Le processus TELCØM ainsi activé peut analyser le programme canal construit par ASP, récupérer le message et simuler la fin de l'opération d'Entrée/Sortie de la même façon que pour une opération de lecture, c'est-à-dire en mettant le CSW à jour dans l'IØB et en simulant une interruption d'E/S. "Device end" qui sera récupérée comme une interruption asynchrone par CØNTRAP.

### 5.2.3.2. Comportement d'un processus TELCØM utilisateur d'une pseudo console ASP.

Pour qu'un processus TELCØM puisse utiliser une pseudo console ASP il doit d'abord choisir l'adresse d'une pseudo console disponible et la réserver.

A chaque pseudo console est associée une entrée de 3 mots dans le module DATA1 contenant un ECB, une adresse d'IØB et une adresse de TCB.

Initialement ces trois champs sont à zéro. L'ECB est associé à l'évènement : On a lancé une opération d'entrée-sortie sur la pseudo unité. C'est-à-dire qu'un SØST sera fait sur cet ECB par SØEXCP appelé par "l'appendage de SIØ" lorsque le système ASP demande au superviseur d'E/S d'ØS de lancer une opération d'E/S sur une pseudo console. De plus SØEXCP range dans le 2e et le 3e mot de l'entrée associée à la pseudo unité l'adresse de l'IØB et l'adresse du TCB correspondant à la requête d'E/S.

Ces deux adresses permettent au processus TELCØM lorsqu'il sera réveillé de retrouver le programme canal et de simuler l'exécution de l'opération d'entrée-sortie (ranger une commande dans une zone de lecture ou récupérer un message émis par ASP) et sa fin (interruption d'E/S simulée).

Pour permettre à un processus TELCØM d'utiliser une pseudo console facilement deux services TELCØM sont mis à sa disposition :

SATTN

SØEND

a) Attention sur une pseudo console

Un processus qui veut émettre une attention sur une pseudo console ASP doit charger dans le registre 1 l'adresse de la pseudo console qu'il s'est affectée et appeler une SVC propre à TELCOM à l'aide de la macro instruction  $\$ATTN$ .

Cette SVC se déroule totalement masquée. Elle remet le CSW à zéro, puis force le bit "attention" du CSW à 1. Elle range dans le "I/O OLD PSW" en mémoire basse l'adresse de l'instruction qui suit l'appel à la SVC  $\$ATTN$  et l'adresse de la pseudo console dans la zone de code interruption puis effectue un LOAD PSW du "I/O new PSW" en mémoire basse. Ceci simule une interruption d'entrée-sortie provoquée par une attention sur l'unité à l'adresse de la pseudo console qui serait prise en compte entre l'exécution de l'instruction qui précède  $\$ATTN$  et de celle qui la suit.

b) "Device end" sur une pseudo console

Un processus TELCOM qui veut signaler la fin d'une opération d'entrée sortie sur une pseudo console opérateur ASP doit charger dans le registre 1 l'adresse de la pseudo console qu'il utilise puis appeler une SVC à l'aide de la macro instruction  $\$DEND$ .

Cette SVC est tout à fait analogue à celle qui simule une attention, à la différence près que le bit mis à 1 dans le CSW est "device end", au lieu de "attention".

Un processus TELCOM qui veut utiliser une pseudo console est alors capable d'émettre une attention, d'être réveillé lorsqu'une opération d'E/S est lancée sur la pseudo console et de simuler la fin de l'opération d'E/S. Pour effectuer le traitement de l'opération d'E/S il a à sa disposition l'adresse de l'IØB à partir duquel il peut accéder à toutes les informations qui concernent l'opération d'E/S. Dans le cas d'une lecture, il lui faut ranger dans la zone de lecture, qui n'est pas sous la même clé de protection que le système TELCOM, la commande à émettre. Il a pour cela



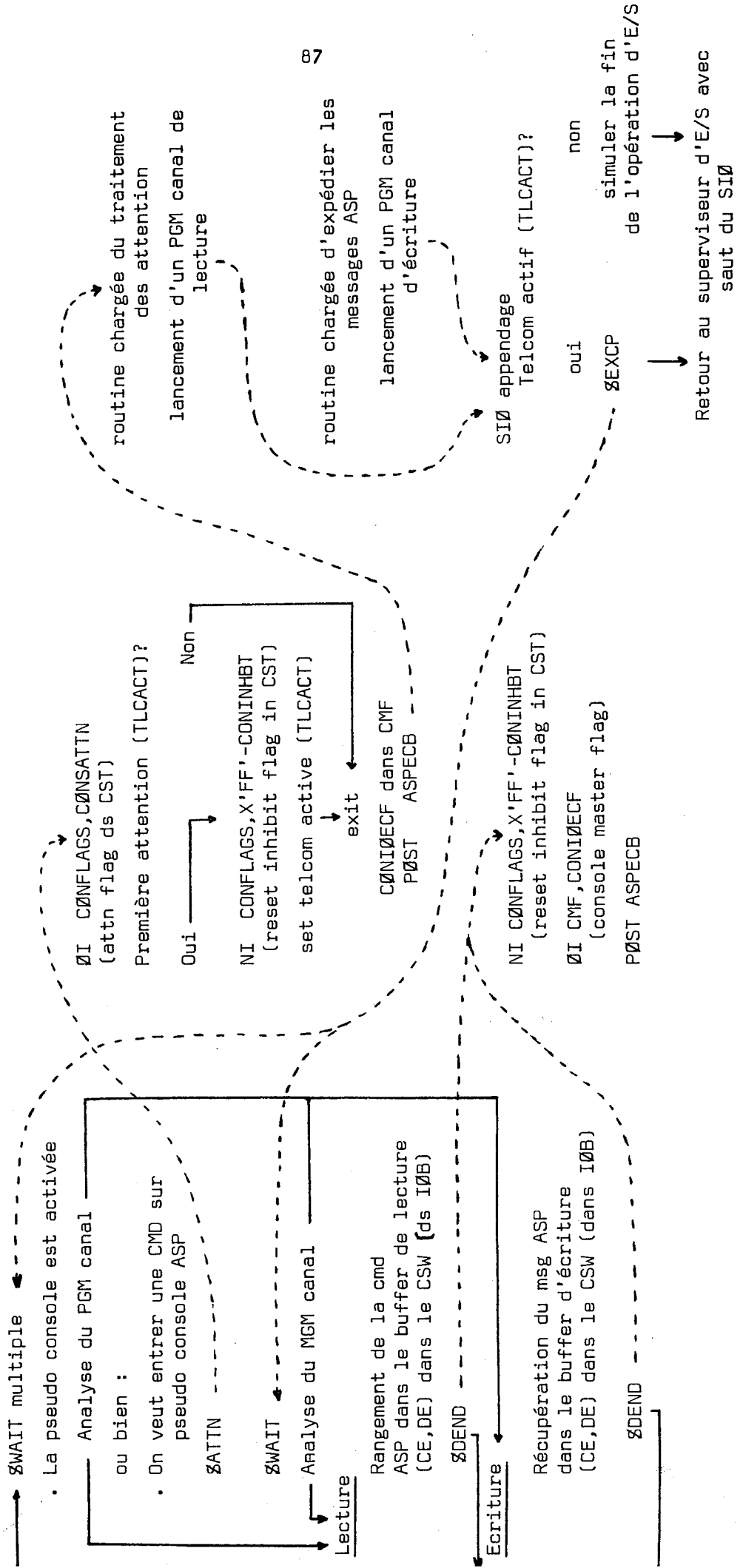
à sa disposition une macro instruction  $\Sigma$ EX qui permet d'exécuter en mode maître, clé de protection 0, une instruction EXECUTE adressant l'instruction adressée par le registre 14 et utilisant le masque contenu de R1 avant l'appel de  $\Sigma$ EX.

A l'aide de ces services, un processus TELCØM peut alors se comporter comme l'utilisateur d'une console ASP.

Processus TELCOM chargé de la gestion d'une pseudo console ASP

Module C0NTRAP du système ASP chargé du traitement des interruptions d'E/S asynchrones

Module C0NS1053 du système ASP chargé de la gestion des pseudo consoles



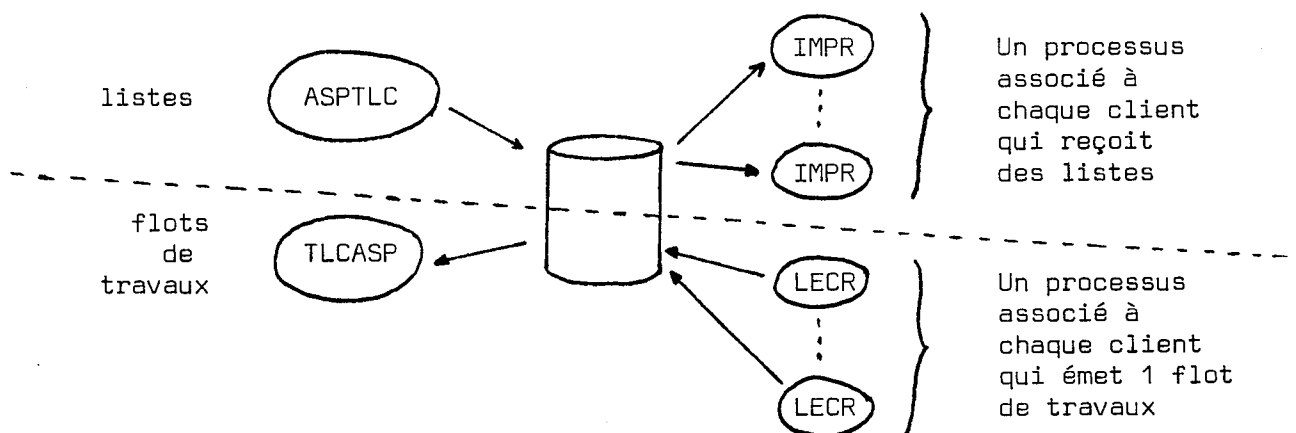
### 5.3 Gestion d'une file d'attente réseau sur une unité à accès direct

#### 5.3.1 Justification et logique de conception

Le fait qu'une seule voie séquentielle de communication avec le système ASP soit établie aussi bien pour la soumission des travaux que pour la réception des listes nous a conduits à mettre au point la gestion d'une file d'attente sur une unité à accès direct pour permettre :

- à plusieurs processus du système TELCØM d'enregistrer simultanément à leur vitesse de réception des fichiers séquentiels correspondant à des flots de travaux reçus des différents clients connectés. Ce nombre de processus doit être variable et limité seulement par la charge du système. Un processus unique doit pouvoir lire ces fichiers en séquence dans leur ordre de réception, un fichier étant considéré comme reçu quand il a été totalement enregistré.
- à un processus unique d'enregistrer en séquence des fichiers correspondant à des listes reçues du système ASP et d'identifier ces fichiers de telle façon qu'un ensemble de processus activés sur l'initiative des clients connectés puissent lire simultanément les listes du client qui leur est associé dans leur ordre de réception.

Ce qui nous a conduits à définir deux types de fichiers : flots de travaux (cartes) ou listes (lignes).



Le support utilisé pour enregistrer ces fichiers est un "data set ØS" initialement formaté en enregistrements de longueur fixe ayant un champ clé de longueur nulle et un champ donné de 258 octets.

Seul le champ donnée de ces enregistrements est modifiable en cours d'exploitation.

Deux séquences particulières d'initialisation qui peuvent être demandées par un paramètre spécifié sur les cartes de contrôle utilisées pour démarrer le système TELCØM permettent soit de reformatter\* la totalité de la file d'attente, soit de la modifier pour qu'elle soit considérée comme vide, à l'initialisation du système. Le formatage complet est assez long et ne doit être refait que s'il est nécessaire de changer le disque support de la file d'attente.

Un processus du système TELCØM, le processus FA, est chargé d'effectuer les opérations d'entrée-sortie concernant la gestion de la file d'attente par appel de sous-programmes du module DRFA constituant la méthode d'accès développée en respectant les règles générales définies précédemment pour la gestion des opérations d'entrée-sortie sous le système TELCØM sous ØS.

Ce processus retire dans une liste qui lui est affectée des requêtes émises par les processus utilisateurs de la file d'attente et effectue les traitements correspondants.

L'unité d'allocation d'espace disque est la piste. Un mécanisme de gestion de l'espace libre par table d'allocation de pistes de conception analogue à celui utilisé par le système HASP a été implémenté. Chaque processus utilisateur de la file d'attente possède sa propre table d'allocation de piste qui identifie les pistes utilisées par ce processus et qui permet ainsi à tout moment, sur demande du processus utilisateur de libérer ou d'allouer l'ensemble des pistes correspondantes.

Une table d'allocation statique est maintenue en mémoire centrale et fournit une image de la table d'allocation enregistrée sur disque, elle n'est mise à jour, en mémoire et sur disque simultanément, que sur une demande explicite d'un processus utilisateur de la file d'attente désireux de libérer ou allouer un ensemble de pistes affectées à un fichier.

\* Le formatage consiste à écrire la totalité des enregistrements physiques sur disque (Write Count Key & Data) de façon à prédéterminer leur emplacement et leur taille.

Une table d'allocation dynamique est également maintenue en mémoire centrale pour permettre aux processus demandeurs d'allouer les pistes sans conflit, elle est mise à jour à chaque demande d'allocation de piste nécessaire pour un processus utilisateur, ou bien lorsqu'un utilisateur désire libérer un ensemble de pistes affectées à un fichier. Les pistes allouées une à une pour un utilisateur seront libérées globalement s'il abandonne l'enregistrement de son fichier.

Le premier enregistrement de la file d'attente contient une table d'allocation des pistes (image, en cours d'exploitation de la table d'allocation statique), les  $n$  enregistrements suivants constituent les blocs descripteurs de la file d'attente et permettent de retrouver les fichiers enregistrés à un moment donné. Les enregistrements suivants contiennent les fichiers proprement dits qui sont enregistrés sur des pistes chaînées séquentiellement. Cette file d'attente sur disque a été conçue pour faciliter l'accès au serveur de traitement par lots mais a pu aussi être utilisée sans modifications pour un concentrateur de terminaux lourds clients du réseau Cyclades développé sous TELCØM sous ØS.

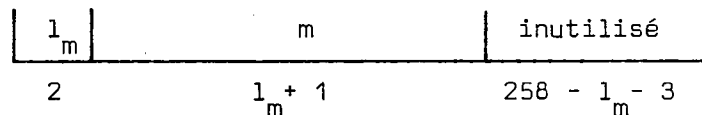
Un schéma de l'organisation de la file d'attente est fourni en 5.3.2.4.

### 5.3.2 Structure des informations contenues dans la file d'attente

#### 5.3.2.1 La table d'allocation des pistes

Elle est contenue dans un enregistrement de 258 octets. Les deux premiers octets contiennent la longueur moins un ( $l_m$ ) en octets de la partie utile de la table d'allocation des pistes.  $0 \leq l_m \leq 255$ .

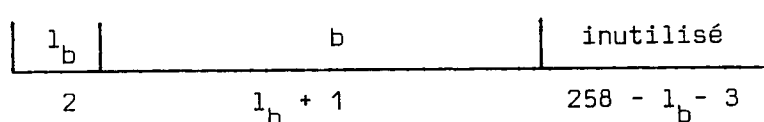
Les octets suivants constituent la partie utile  $m$  de la table d'allocation des pistes.



Le premier bit du champ  $m$  est associé à la première piste du "data set ØS" affectée aux fichiers TELCØM. Un bit à 1 du champ  $m$  indique une piste allouée, un bit à 0 indique une piste libre.

### 5.3.2.2 Les blocs descripteurs de la file d'attente

Ils contiennent sur deux octets la longueur moins un ( $l_b$ ) en octets de la partie b effectivement utilisée en bloc.



L'ensemble des champs b concaténés les uns aux autres dans leur ordre d'occurrence constitue une suite de descripteurs d'utilisateurs définie comme suit :

Suite de descripteurs d'utilisateurs ::= |X'00'|descripteur d'utilisateur/X'00'|  
                                   descripteur d'utilisateur/suite de descripteurs d'utilisateur|

Descripteur d'utilisateur ::= |en-tête de descripteur d'utilisateur/descripteur de  
                                   fichier|descripteur d'utilisateur/descripteur de fichier|

En-tête de descripteur d'utilisateur ::= |X'01'/nombre de fichiers module 256/  
                                   I-USAGER|

Descripteur de fichier ::= |X'02'/MCH|  
                                   X'03'/jobname/ddname/MCH|

Un descripteur d'utilisateur peut occuper plusieurs blocs descripteurs, mais pas un descripteur de fichier ni un en-tête de descripteur d'utilisateur.

On peut considérer que l'entrée qui débute par le code X'00' constitue un descripteur d'utilisateur fictif de fin.

Les champs MCH des descripteurs de fichiers fournissent l'adresse de la première piste du fichier concerné.

#### Exemple d'une suite de descripteurs d'utilisateurs :

X'01'/2/IREP/X'02'/MCH/X'03'/ØSIRGUY/SYSMSG/MCH/

X'01'/1/IRIA/X'03'/JØB2/SYSRINT/MCH/

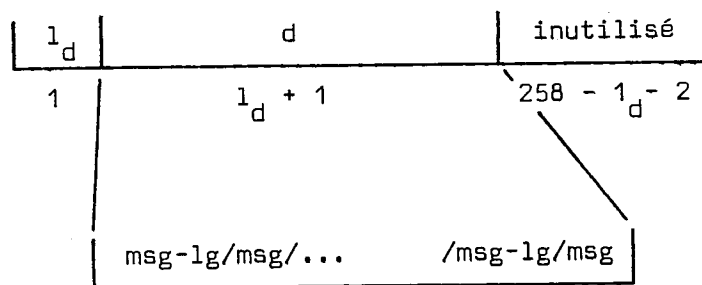
X'01'/2/AR02/X'03'/TP1/SYSMSG/MCH/X'03'/TP1/SYSUDUMP/MCH/X'00'

Dans cet exemple la file d'attente contient :

- deux fichiers pour l'IREP :
  - un flot de travaux
  - la liste SYSMSG pour le "job" ØSIRGUY
- un fichier pour l'IRIA
  - la liste SYSPRINT pour le "job" JØB2
- deux fichiers pour le terminal lourd ARO2 (voir chapitre 6)
  - la liste SYSMSG pour le "job" TP1
  - la liste SYSUDUMP pour le "job" TP1

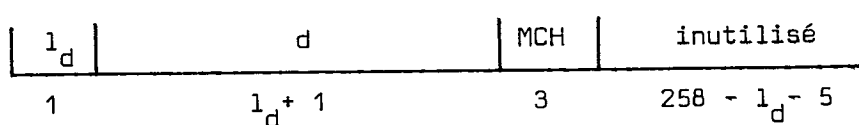
### 5.3.2.3 Les fichiers dans la file d'attente

Les fichiers, de type cartes ou de type lignes, sont rangés dans la file d'attente en respectant le protocole appareil virtuel. Le premier octet de chaque bloc contient la longueur moins un ( $l_d$ ) de l'information utile qui suit.



Seuls les messages de niveau 0 (Ø-TEXT, voir VTP ter 503-1) figurent dans la file d'attente.

Un enregistrement contenant un champ  $l_d$  nul indique la fin de fichier. Le dernier enregistrement d'une piste a la structure suivante :



Si le champ MCH contient 0, cet enregistrement est le dernier du fichier, sinon MCH adresse la piste suivante.

#### 5.3.2.4 Exemple et schéma

On prend pour exemple un cas simple où la file d'attente est constituée d'un fichier ØS de 8 pistes qui débute sur la piste 0 du cylindre 5. La première piste contient la table d'allocation des pistes et les blocs descripteurs d'utilisateurs.

Les 7 pistes suivantes contiennent trois fichiers utilisateurs.

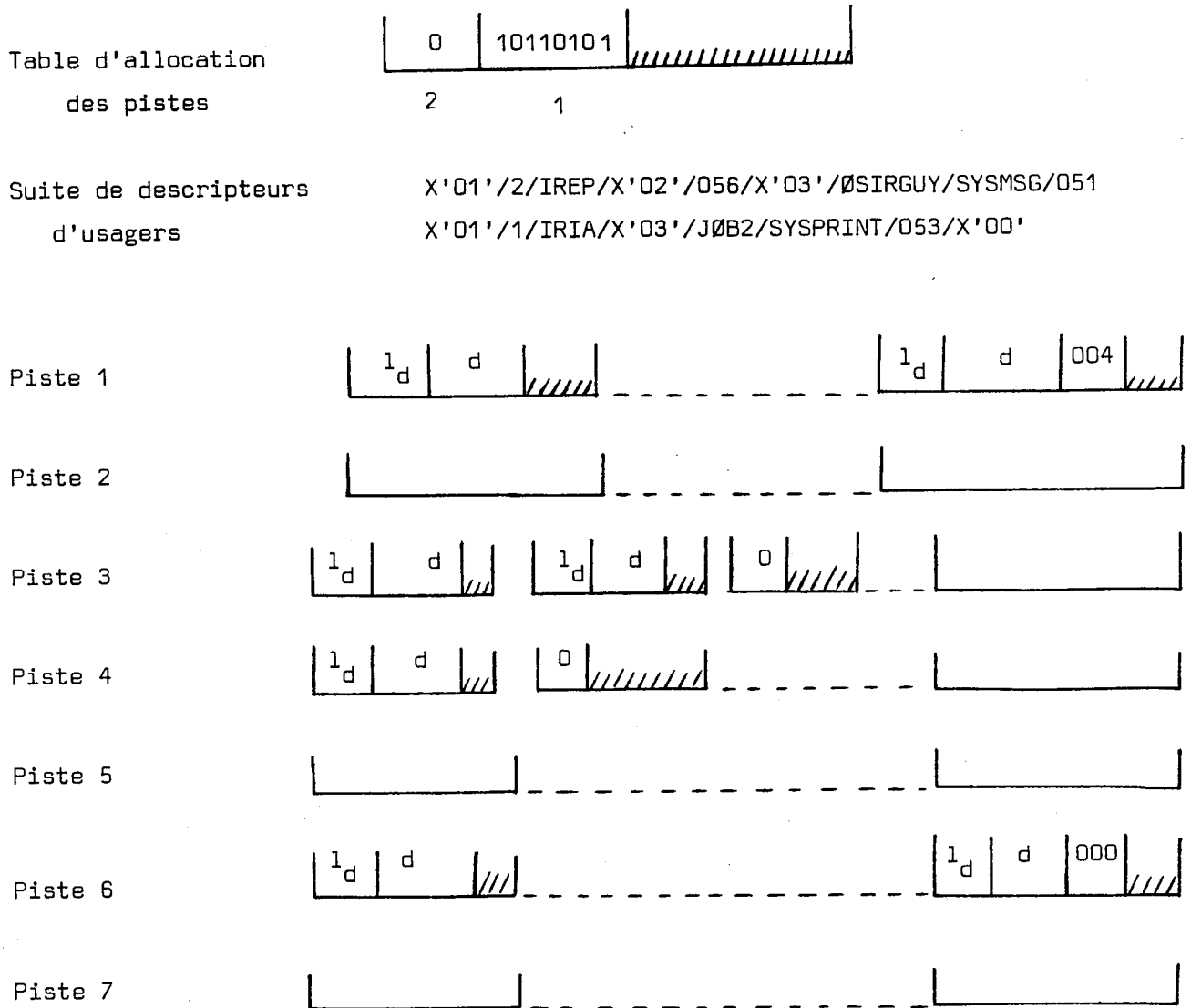
- un flot de travaux pour l'IREP qui occupe la totalité de la piste 6
- la liste SYSMSG du "job" ØSIRGUY destinée à l'IREP qui occupe la piste 1 et le premier enregistrement de la piste 4
- la liste SYSPRINT du "job" JØB2 qui occupe les deux premiers enregistrements de la piste 3.

Les pistes 2,5, et 7 (du cylindre 5) sont libres comme l'indique la table d'allocation des pistes.

Cette situation (pistes libres avant la dernière occupées) n'est possible qu'à la suite de la suppression d'un (ou deux) fichier qui occupait les pistes 2 et 5.

La piste 8 du cylindre 5 qui n'appartient pas à la file d'attente est spécifiée non disponible dans la table d'allocation des pistes dont la longueur utile est spécifiée en octets.





5.3.3 Les requêtes des utilisateurs et leur traitement

Tout processus TELCØM qui veut lire ou écrire dans la file d'attente adresse au processus FA des requêtes qu'il insère dans la liste d'entrée de ce processus.

5.3.3.1 Les requêtes d'ouverture de fichier

D'une façon générale, les requêtes d'ouverture doivent spécifier un identificateur de fichier (I-FICHER) auquel une valeur est affectée par le processus FA et qui sera utilisé dans toute nouvelle requête concernant ce fichier.

Structure générale :

`ØPEN/I-FICHER/@ECB/TYPE...`

Pour chaque ØPEN il faut créer un bloc de contrôle de fichier (BCF). Les BCF des fichiers en cours de traitement sont chaînés entre eux.

a) Ouverture d'un fichier pour lecture de cartes

`ØPEN/I-FICHER/@ECB/ØLECCRD`

Il y a création d'un BCF contenant une table d'allocation de pistes dont tous les bits sont initialisés à 1. Pour chaque piste totalement lue le bit correspondant sera remis à zéro.

Puis il faut rechercher un descripteur de fichier de type carte dans les blocs descripteurs de la file d'attente. Si un tel fichier existe, l'adresse de sa première piste (ainsi que son identificateur d'utilisateur) est rangée dans le BCF créé pour la requête ØPEN et l'utilisateur est réveillé par un ØPOST sur l'ECB associé à sa requête. S'il n'existe pas, l'utilisateur en est informé et il pourra se mettre en attente de disponibilité d'un fichier de type carte dans la file d'attente, c'est-à-dire que son réveil sera demandé par le premier processus effectuant un ØCLOSE pour un fichier ouvert en écriture de cartes.

b) Ouverture d'un fichier pour écriture de cartes

`ØPEN/I-FICHER/@ECB/ØECRCRD/I-USAGER`

Il y a création d'un BCF contenant une table d'allocation de pistes dont tous les bits sont initialisés à 0, puis réveil de l'utilisateur.

c) Ouverture d'un fichier pour lecture de lignes

`ØPEN/I-FICHER/@ECB/ØLECLIN/I-USAGER`

Il y a création d'un BCF contenant une table d'allocation de pistes dont tous les bits sont initialisés à 1, puis recherche d'un descripteur d'utilisateur dans les blocs descripteurs correspondant à l'I-USAGER spécifié et recherche d'un descripteur de fichier de type lignes dans ce descripteur d'utilisateur. Si une telle recherche est satisfaite, l'utilisateur

est réveillé après rangement de l'adresse de la première piste dans le BCF, sinon l'utilisateur pourra se mettre en attente de disponibilité d'un fichier de type ligne pour l'utilisateur le concernant.

#### d) Ouverture d'un fichier pour écriture de lignes

```
ØOPEN/I-FICHER/@ECB/ØECLIN/I-USAGER/JØBNAME/DDNAME
```

Il y a création d'un BCF contenant les caractéristiques spécifiées pour le fichier et une table d'allocation de pistes dont tous les bits sont initialisés à 0, puis réveil de l'utilisateur.

#### 5.3.3.2 Requête de lecture

```
ØREAD/I-FICHER/@ECB/@BUFFER
```

Le premier enregistrement ou l'enregistrement suivant celui précédemment lu (s'il existe) est rangé dans le buffer adressé par @BUFFER. La structure des enregistrements lus sur disque n'est pas modifiée. Si le dernier enregistrement du fichier a déjà été lu, pour toute nouvelle requête de lecture l'utilisateur sera réveillé avec un champ  $l_d$  (premier octet du buffer) nul. Si l'enregistrement lu est le dernier d'une piste, le bit correspondant dans la table d'allocation associée à ce fichier est remis à zéro. Si cette piste est la dernière du fichier, on indique la fin de fichier dans le BCF, sinon on prépare la lecture de la piste suivante.

Les buffers de lecture sont totalement gérés par l'utilisateur. Ils doivent avoir la longueur du bloc physique, c'est-à-dire 258 octets.

L'utilisateur est réveillé par un  $ØPØST$  sur l'ECB adressé par @ECB.

#### 5.3.3.3 Requête d'écriture

```
ØWRITE/I-FICHER/@ECB/@BUFFER
```

A la première requête d'écriture une piste est allouée et l'enregistrement adressé par @BUFFER est écrit en début de piste.

Pour les requêtes suivantes, l'enregistrement fourni par l'utilisateur est écrit à la suite de celui précédemment écrit sur la piste en cours. Avant d'écrire le dernier enregistrement d'une piste la suivante est allouée de façon à ranger son adresse dans le dernier bloc de celle en cours.

L'allocation d'une piste pour un utilisateur consiste à rechercher la première piste libre dans la table d'allocation dynamique, à mettre son adresse dans le BCF de l'utilisateur et à forcer à 1 les bits correspondant à cette piste dans la table d'allocation dynamique et dans la table d'allocation associée au BCF de l'utilisateur.

Le buffer fourni par l'utilisateur doit avoir une longueur de 258 octets même s'il n'est pas totalement rempli (pour permettre l'écriture éventuelle de l'adresse de la piste suivante).

L'utilisateur est réveillé par un `ΣPØST` sur l'ECB adressé par `@ECB`.

#### 5.3.3.4 Les requêtes de fermeture

Il existe deux façons d'arrêter le traitement d'un fichier :

`ΣCLØSE` qui conserve sur disque tout ce qui a été précédemment lu ou écrit.

`ΣABØRT` qui supprime de la file d'attente la partie précédemment lue ou écrite du fichier concerné.

##### a) Requête ΣCLØSE

`ΣCLØSE/I-FICHER/@ECB`

Pour un fichier ouvert en lecture, `ΣCLØSE` n'effectue que la suppression du BCF associé. En effet, la file d'attente et les tables d'allocation de piste doivent rester inchangées.

Par contre, pour un fichier ouvert en écriture, il faut d'abord mettre à jour la table d'allocation statique à partir de la table d'allocation associée au fichier concerné, puis il faut indiquer la fin de fichier

soit en écrivant un enregistrement fictif de fin (ld = 0) sur la piste en cours soit en forçant à zéro l'adresse de la piste suivante dans le dernier enregistrement écrit et en libérant la dernière piste allouée (dans le cas où le dernier enregistrement écrit était le dernier d'une piste), puis il faut écrire la table d'allocation statique à jour sur disque et enfin il faut insérer un nouveau descripteur de fichier dans la file d'attente.

Si un descripteur d'utilisateur existe déjà pour l'utilisateur concerné, on insère un nouveau descripteur de fichier en bout du descripteur d'utilisateur existant, sinon on crée un nouveau descripteur d'utilisateur à la suite des descripteurs d'utilisateur existant. On peut alors libérer le BCF associé au fichier concerné et réveiller l'utilisateur par un `SPØST` sur l'ECB adressé par la requête.

#### b) Requête `§ABØRT`

`§ABØRT/I-FICHER/@ECB`

Pour un fichier ouvert en écriture, `§ABØRT` n'effectue pas la mise à jour de la table d'allocation statique et des blocs descripteurs puisque les informations écrites ne doivent plus être accessibles. Il suffit donc de supprimer le BCF associé.

Pour un fichier ouvert en lecture, il faut supprimer de la table d'allocation statique et de la table d'allocation dynamique les pistes déjà lues (fournies par la table d'allocation du BCF concerné) puis il faut écrire la table d'allocation statique à jour sur disque. Si le fichier a été totalement lu le descripteur de fichier correspondant est supprimé de la suite de descripteurs d'utilisateurs, sinon il est seulement mis à jour pour adresser la nouvelle première piste du fichier (la dernière piste lue). On peut alors libérer le BCF associé au fichier concerné et réveiller l'utilisateur par un `SPØST` sur l'ECB adressé par la requête.

## 5.4 Les abonnés serveurs

Un processus permanent de contact crée les processus chargés de l'exploitation sur demande des utilisateurs.

### 5.4.1 Le processus de contact (ABC)

Le processus de contact est un processus permanent qui reconnaît les demandes de contact émises par les clients du réseau Cyclades qui veulent se connecter au serveur de traitement par lot. Ces demandes de contact diffèrent suivant que l'on travaille en version 1 ou en version 2 de la station de transport.

#### a) ST1

En version 1 le processus de contact est un abonné particularisé (abonné contact), d'adresse hexadécimale C1C3D481 dans notre cas, qui reçoit les demandes de contact sous forme de lettres régulières dont le format est défini par le protocole client serveur de terminaux [I.4]. Une lettre régulière de demande de contact identifie le type de service requis par un ensemble de deux identificateurs : I-ENTREE/I-SERVICE et identifie le demandeur par I-USAGER.

On a choisi d'utiliser l'identificateur de service BATCH360 pour l'ensemble du serveur de traitement par lots et les identificateurs d'entrée :

- CØNS pour la connexion d'une console opérateur
- LECR pour la connexion d'un lecteur de cartes
- IMPR pour la connexion d'une imprimante

Le processus de contact crée un processus spécialisé dans le traitement de l'un de ces 3 types de connexion sur reconnaissance d'une lettre régulière de demande de contact.

Ce processus spécialisé sera chargé d'activer l'abonnement d'un "abonné-exploitation" et d'établir une voie virtuelle avec le client demandeur

pour assurer la connexion. Il restera actif, ainsi que la voie virtuelle qui lui est associée jusqu'au moment où le client décidera de rompre la connexion ou sur une déconnexion accidentelle.

b) ST2

En version 2 le couple d'identificateurs : I-ENTREE/I-SERVICE qui permettait de reconnaître le type de connexion souhaité par le demandeur est remplacé par un numéro de porte.

Ce qui tiendra lieu de demande contact est alors une demande d'établissement de flot sur l'une des 3 portes identifiant les 3 types de connexion autorisés. Sur une telle demande un processus spécialisé dans le type de connexion souhaité est créé, comme en version 1, et restera actif jusqu'à la fermeture du flot qui lui est associé, qu'elle s'effectue accidentellement ou sur demande de l'utilisateur.

#### 5.4.2 Les processus RMCØNS

Un processus RMCØNS est créé sur une demande de connexion d'une console opérateur d'un client Cyclades au serveur de traitement par lots du système ØS. Il assure la connexion à l'alternat d'un appareil virtuel serveur avec un appareil virtuel client qui est mis en correspondance avec l'appareil réel par le concentrateur de terminaux du site client.

Les messages à destination de l'appareil réel sont reçus par un processus RMCØNS dans une liste qui lui est associée. Ce mécanisme permet en particulier au processus chargé de la gestion de la console opérateur du système TELCØM d'émettre des messages à destination d'une console gérée par un processus RMCØNS. Ceci permet à l'opérateur local du système TELCØM d'émettre des messages vers une (ou toute) console du réseau connectée au serveur, pour lui demander sa déconnexion peu avant l'arrêt du système par exemple. Tout processus RMCØNS est mis en correspondance avec une pseudo console opérateur du système ASP suivant le mécanisme présenté précédemment. Cependant, du fait

que les pseudo consoles doivent être déclarées au système ASP et que leur nombre doit ainsi être choisi à priori, alors que le nombre de processus RMCONS n'est limité que par la charge du système, on a prévu le cas où un processus RMCONS ne trouve pas de pseudo console opérateur ASP disponible. Tout message reçu par un processus RMCØNS sur la voie virtuelle (ou le flot) qui le met en correspondance avec la console réelle sera considéré comme une commande opérateur destinée à ASP si son texte commence par le caractère \* (astérisque). Dans ce cas la commande est émise sur la pseudo console opérateur associée (si elle existe) et quand la réponse parviendra au processus RMCØNS elle sera aussitôt émise vers la console réelle. Tout message reçu de la console réelle qui ne commence pas par \* sera considéré comme une commande à analyser localement par le processus RMCØNS. Actuellement la seule commande reconnue est :

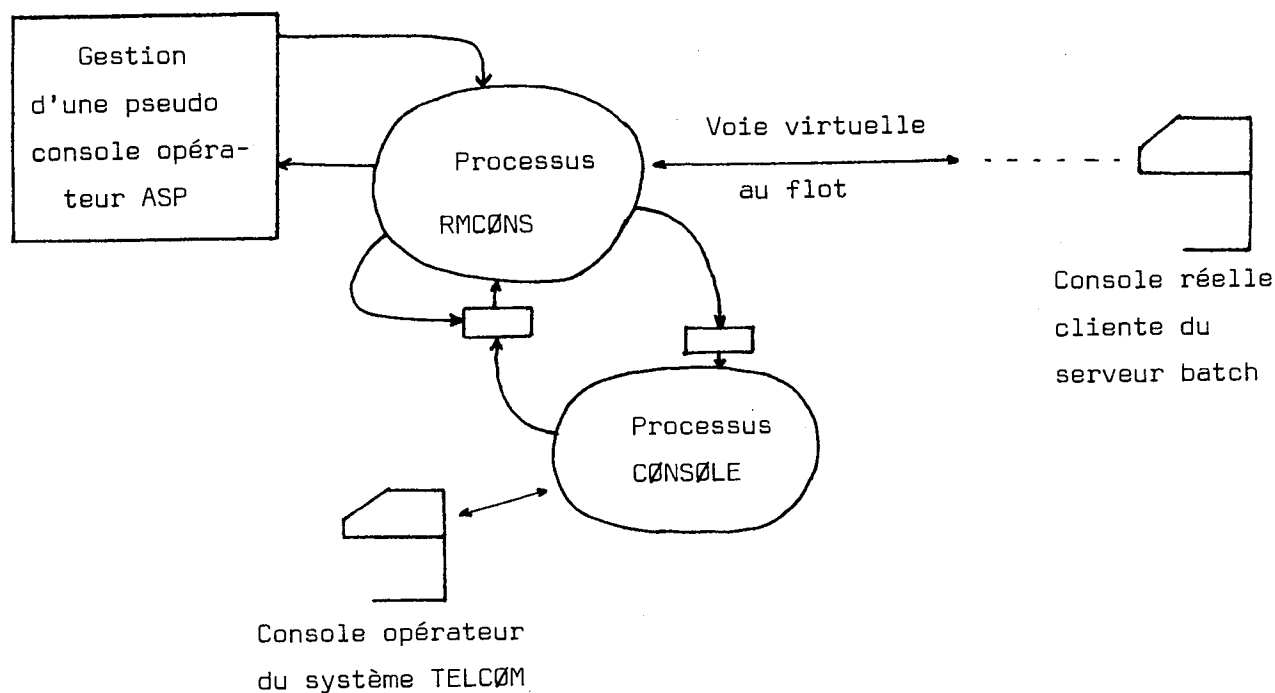
STØP IMPR

à sa réception le processus RMCØNS informe le processus (s'il existe) chargé de la gestion de l'imprimante du client ayant le même nom d'utilisateur que celui de la console réelle associée à RMCØNS, d'arrêter la liste en cours d'impression. Ceci pour permettre à l'opérateur du terminal lourd client d'arrêter des listes en cours d'impression.

Tout message en provenance d'une console client est également émis sur la console opérateur du système TELCØM.

Les processus RMCØNS sont créés dynamiquement et se désactivent après la fermeture de la voie virtuelle (ou du flot) qui leur est associé.





### 5.4.3 Les processus IMPR

Un processus IMPR est créé sur une demande de connexion de l'imprimante d'un client Cyclades.

Après ouverture de la voie virtuelle (ou du flot) chargée de mettre en correspondance l'appareil virtuel serveur avec l'appareil virtuel client, le processus IMPR effectue un `ØPEN` sur la file d'attente pour lecture d'une liste destinée à l'utilisateur (ou le numéro de ST en version 2 des stations de transport) associé au client qui a fait la demande de connexion.

C'est-à-dire que le processus IMPR obtiendra l'identificateur de fichier associé à la liste la plus ancienne (dans l'ordre d'insertion dans la file d'attente) qui lui est associée si un tel fichier existe, sinon il se mettra en attente d'insertion dans la file d'attente d'une liste adressée à l'utilisateur qui le concerne.

Dès que le processus IMPR a obtenu un identificateur de fichier il émet vers l'imprimante réelle du client un message identifiant la liste qui va suivre (ce message est émis suivant le protocole appareil virtuel sur une page séparée) puis il retire dans la file d'attente les enregistrements du fichier qui sont déjà en format PAV et les émet vers le client. A l'obtention de fin de fichier sur disque une requête `ΣABØRT` est émise au processus FA pour supprimer la liste qui vient d'être totalement lue de la file d'attente, puis on émet un message sur la console opérateur du système TELCØM et sur la console opérateur du terminal lourd (si elle est connectée) de même nom d'utilisateur que l'imprimante gérée pour les informer de la fin de la liste en cours et le processus IMPR effectue à nouveau un `ΣØPEN` pour retrait de listes dans la file d'attente du système TELCØM.

Les consoles opérateur de TELCØM et du client reçoivent aussi un message en début d'émission d'une liste pour identifier la liste qui va suivre et conserver une trace des listes.

Si l'opération du terminal lourd a demandé d'abandonner l'impression de la liste en cours, son retrait dans la file d'attente continue jusqu'à l'obtention de fin de fichier, mais les enregistrements lus ne sont plus émis sur la voie virtuelle (ou le flot), ceci pour permettre la suppression complète du fichier de la file d'attente au moment du `ΣABØRT`.

Une amélioration possible de ce mécanisme de retrait de listes consisterait, en modifiant le processus de gestion de la file d'attente, à permettre l'interrogation par l'opérateur des listes en attente pour un usager concerné et la modification éventuelle de leur ordre de sortie sur l'imprimante ainsi que la suppression de certaines d'entre elles. Mais ces opérations nécessitent beaucoup d'opérations d'entrée-sortie sur disque et leur utilisation par certains clients se ferait au détriment des autres.

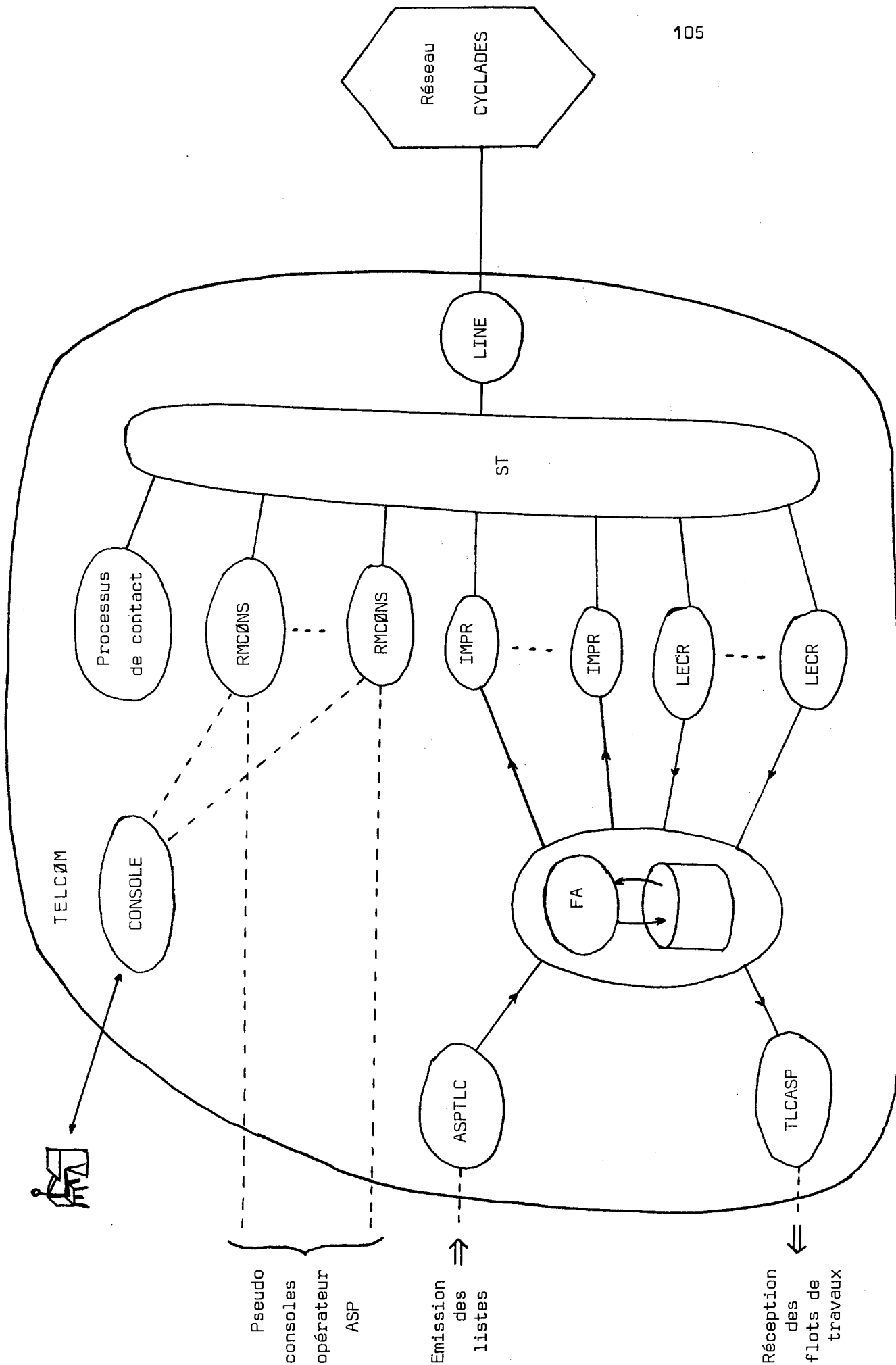
#### 5.4.4 Les processus LECR

Un processus LECR est créé sur une demande de connexion du lecteur de cartes d'un client Cyclades.

Après ouverture d'une voie virtuelle (ou d'un flot), le processus LECR effectue un `ØPEN` sur la file d'attente pour écriture d'un flot de travaux, il obtient en retour un identificateur de fichier qui lui permettra d'adresser des requêtes d'écriture dans la file d'attente. Il se met alors en attente de réception de messages respectant le protocole appareil virtuel en provenance du client, il en extrait les messages de texte qu'il cumule dans un buffer de travail qui est écrit sur disque lorsqu'il n'y a plus de place pour le message suivant. Une requête `ØCLØSE` est transmise au processus FA lorsqu'on reçoit du client une information spécifiant que la lecture d'un flot de travaux est terminée (réception de `Ø-STATUS/IDLE` en version 1 des protocoles et `NEW-PAGE` en version 2) et un nouveau fichier est aussitôt ouvert(`ØPEN`).

Dans toutes les situations impliquant la fermeture de la voie virtuelle (ou du flot) assurant la connexion avec le client (sur initiative du client ou suite à une anomalie de fonctionnement), une requête `ØABORT` est émise pour demander la suppression du flot de travaux partiellement lu ou en attente de réception.

De même que pour les processus `RMCØNS` et `IMPR`, les processus LECR sont créés dynamiquement et se désactivent après la fermeture de la voie virtuelle (ou du flot) qui leur est associé.



## 6. UN CONCENTRATEUR DE TERMINAUX LOURDS SOUS TELCØM

Il s'agit de prendre en charge sous le système TELCØM le support des terminaux lourds et d'en faire des clients du réseau CYCLADES.

Ces terminaux lourds étaient supportés jusqu'à présent par le système ASP en temps que clients de traitement par lots des systèmes sous le contrôle du système ASP, c'est-à-dire OS/MVT dans notre cas.

On utilise donc des liaisons existantes à des machines de divers constructeurs mais possédant toutes un logiciel permettant à ces machines de se comporter comme un terminal lourd suivant le protocole de 'multi-leaving' HASP.

On ne modifie donc ni les liaisons existantes ni le logiciel des machines connectées mais seulement le logiciel du côté du 360 pour permettre de connecter à des serveurs du réseau CYCLADES les différents appareils (lecteurs de cartes, imprimante, console opérateur) d'un terminal lourd et ceci à l'initiative de l'opérateur du terminal lourd. On permet de connecter un appareil réel à plusieurs serveurs; on a autant d'appareils virtuels que de serveurs connectés mais la correspondance appareil réel-appareil virtuel se fait de façon différente selon le type d'appareil :

- Pour la console opérateur on peut mélanger aussi bien en émission qu'en réception les messages des différents serveurs connectés, il suffit de préfixer les messages reçus pour le nom du serveur qui les a émis et de fournir à l'opérateur des possibilités d'adressage des serveurs vers lesquels il veut émettre des attentions ou des messages.
- Pour le lecteur de cartes, l'appareil réel est mis en correspondance à un moment donné avec un appareil virtuel et un seul. Les autres appareils virtuels se trouvent dans l'état d'un lecteur de cartes qui n'a pas de cartes à lire. Il suffit donc de permettre à l'opérateur de choisir parmi les appareils virtuels existants celui qui doit être mis en correspondance avec l'appareil réel, ou, autrement dit, le serveur vers lequel sera émis le prochain flot de cartes lues.

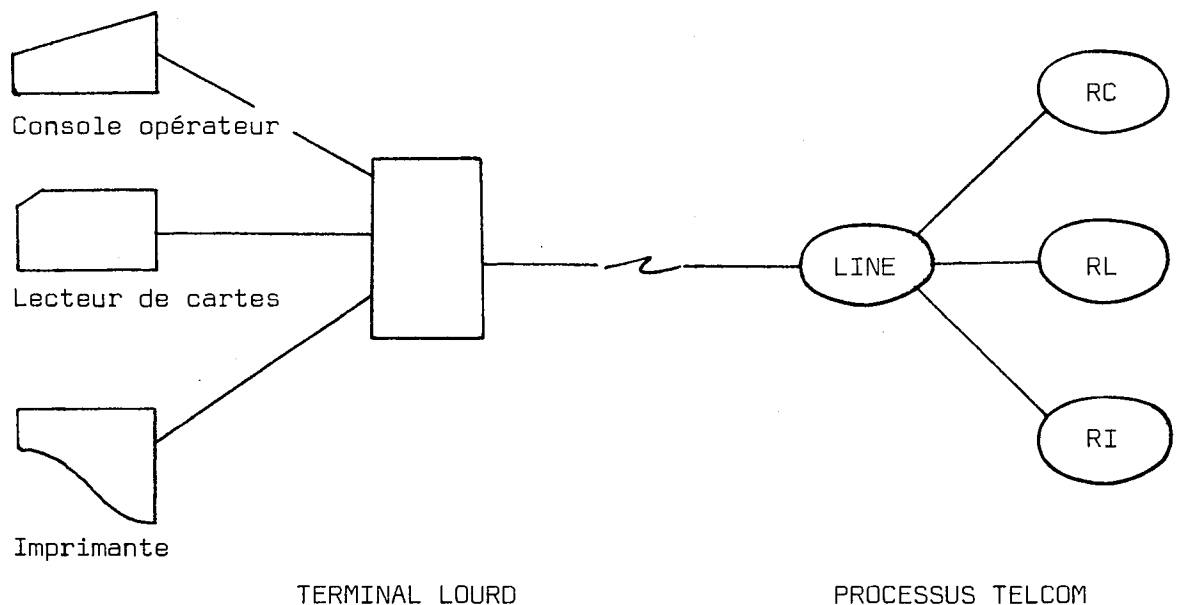
- Pour l'imprimante, les appareils virtuels débitent en parallèle dans une file d'attente les listes reçues et un processus associé à l'imprimante réelle est chargé d'émettre les listes vers le terminal lourd au fur et à mesure de leur disponibilité dans la file d'attente.

On va d'abord présenter l'organisation des processus chargés de la gestion du 'multi-leaving' pour un terminal lourd puis on s'intéressera à la gestion des appareils virtuels associés à chaque connexion.

## 6.1 Les moyens de communication avec le terminal lourd (gestion du 'Multi-leaving')

### 6.1.1 Les processus nécessaires à la gestion du 'multi-leaving'

A chaque terminal lourd supporté sont associés quatre processus dont la durée de vie doit être celle de la connexion de TELCOM au terminal, ou plus précisément la période pendant laquelle l'opérateur du système TELCOM permet cette connexion.

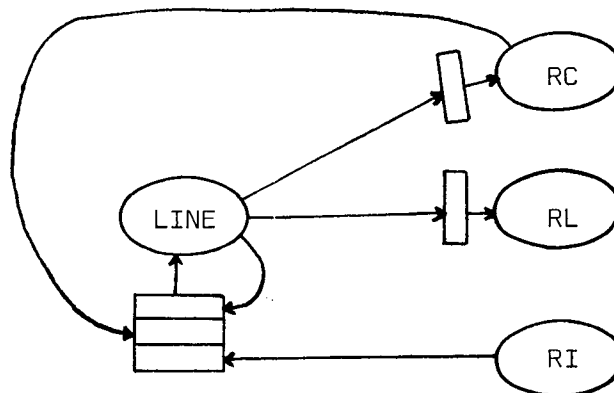


Ces processus sont les suivants :

- LINE : Processus chargé du support de la procédure de ligne BSC.
- RC : Processus représentant la console opérateur du terminal lourd. Il est chargé en particulier de l'analyse et de l'exécution des commandes à la disposition de l'opérateur du terminal lourd.
- RL : Processus représentant le lecteur de cartes du terminal lourd. Il reçoit les cartes en provenance du lecteur réel et les émet sur l'appareil virtuel choisi par l'opérateur.
- RI : Processus représentant l'imprimante d'un terminal lourd. Il lit les listes disponibles émises par les différents serveurs dans la file d'attente de TELCØM et effectue la transformation PAV (protocole appareil virtuel) - 'Multi-leaving' avant de les émettre vers l'imprimante réelle.

Ces différents processus communiquent entre eux par l'intermédiaire de listes.

### 6.1.2 Les listes nécessaires aux échanges d'information entre les processus chargés du 'Multi-leaving'



Une liste est associée au processus RC (la liste RCLST) et contient les messages provenant de la console opérateur du terminal lourd. Ces messages

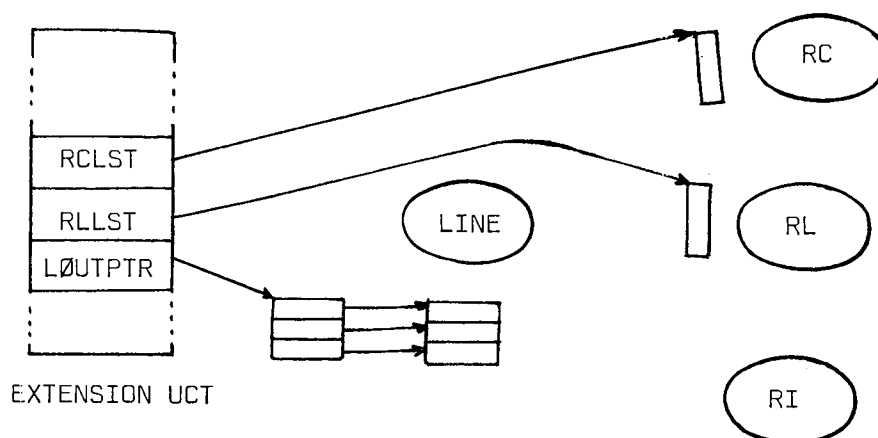
sont soit des commandes opérateur destinées à un serveur de traitement par lots, soit des messages destinés à un serveur conversationnel, selon le mode de connexion, connu par le processus RC.

Une liste est associée au processus RL (la liste RLLST) et contient les cartes provenant du lecteur de cartes du terminal lourd.

Une liste multiple est associée au processus LINE (la liste LØUTPTR) et est constituée de trois listes contenant respectivement :

- Des buffers contenant des enregistrements de contrôle de 'multi-leaving' (par exemple : demande de transmettre une fonction), prêts à émettre sur la ligne.
- Des buffers contenant des messages compactés destinés à l'opérateur du terminal lourd.
- Des buffers contenant des lignes compactées destinées à l'imprimante du terminal lourd.

Ces différentes listes sont adressées par des pointeurs se trouvant dans l'extension de l'UCT associé à la ligne.



La liste multiple est adressée par l'intermédiaire d'une table d'adresses.



On va s'intéresser successivement aux échanges d'information dans le sens 'terminal lourd-TELCOM' puis dans le sens 'TELCØM-terminal lourd'.

### 6.1.3 Traitement des informations reçues sur la ligne

Le processus LINE reçoit des buffers contenant des enregistrements de divers types suivant un format défini par le protocole de 'multi-leaving' HASP. Pour chaque buffer reçu le processus LINE appelle un sous-programme de décompactage (DCPT) qui effectue un traitement spécifique à chaque type d'enregistrement reconnu. Ces types d'enregistrement sont les suivants :

- Demande de transmettre une fonction. Dans ce cas il y a construction et émission immédiate d'un enregistrement de contrôle de permission de transmettre la fonction concernée dans la première liste de la liste multiple LØUTPTR du processus LINE. Prenons un cas précis par exemple : le terminal lourd veut émettre des cartes vers TELCØM, il demande la permission d'émettre des cartes, le processus LINE lui donne aussitôt la permission et les cartes émises sont transmises vers le processus RC qui pourra prendre la décision de suspendre le flot de cartes dès que le nombre limite de cartes en attente de traitement sera atteint.
- Permission de transmettre une fonction. Il s'agit soit d'une permission pour émettre sur l'imprimante, soit d'une permission pour émettre un message à l'opérateur. On effectue simplement un ØPOST sur l'ECB correspondant dans l'extension de l'UCT (RPTRECB ou RCØNECB).
- Un enregistrement compacté provenant du lecteur de cartes ou de la console opérateur. L'enregistrement est décompacté et rangé dans la liste du processus correspondant (liste RLLST pour le processus RL correspondant au lecteur de cartes et liste RCLST pour le processus RC correspondant à la console opérateur). S'il s'agit d'un enregistrement vide indiquant la fin de fichier, un enregistrement vide est créé avec un indicateur dans l'entête spécifiant la fin de fichier. Toutes les fois qu'un élément (un enregistrement décompacté) est inséré dans une liste RLLST ou RCLST le nombre d'éléments de la liste est consulté, s'il dépasse la limite supérieure de

la liste correspondante le FCS émis par la procédure de ligne vers le terminal lourd est mis à jour (FCSOUT dans l'extension de l'UCT) de la façon suivante :

- . S'il indique déjà 'tous flots suspendus' il reste inchangé, sinon le bit associé au flot correspondant est forcé à 'flot suspendu' et les autres bits de fonctions sont consultés : s'ils indiquent tous 'flot suspendu', le bit 'tous flots suspendus' est forcé à 1 pour éviter au correspondant de consulter tous les bits de fonction.
  - . La limite supérieure associée à une liste correspondant à un nombre d'éléments tels qu'on puisse insérer encore dans cette liste le nombre maximum d'enregistrements qu'un buffer de télétraitement pour la ligne correspondante peut contenir.
  - . Une limite inférieure est également associée à chaque liste et correspond à la valeur en dessous de laquelle le processus consommateur (RC ou RL) débloque le flot correspondant.
- A la première occurrence d'un enregistrement ne respectant pas le protocole de "multi-leaving" HASP, le buffer de télétraitement est considéré comme erroné, son analyse est abandonnée et son espace mémoire est libéré.

#### 6.1.4 Emission d'informations vers le terminal lourd

Tout processus qui veut émettre des informations vers le terminal lourd effectue un appel au programme CMPT (compactage) qui compacte l'enregistrement spécifié par l'appelant et le fournit au processus LINE.

L'appelant de CMPT doit fournir en paramètres l'adresse de l'enregistrement à compacter et l'adresse de la liste des buffers de télétraitement dans laquelle l'enregistrement compacté sera rangé. Cette liste doit être l'une des trois listes de la liste multiple LØUTPTR du processus LINE concerné.

L'enregistrement fourni par l'appelleur est compacté dans une zone de travail.

Quand la longueur de l'enregistrement compacté est connue on essaie de le ranger dans un buffer de télétraitement de la liste spécifiée par l'appelleur. On utilise pour cela le service UPDLST qui permet pour une liste choisie de savoir si elle est vide et dans le cas où elle est non vide d'obtenir l'adresse du dernier élément inséré. On sait alors que, si la liste est vide il faut acquérir un nouveau buffer de télétraitement, l'initialiser avec l'enregistrement que l'on vient de compacter et l'insérer dans la liste, si la liste n'est pas vide on essaie d'insérer l'enregistrement compacté à la suite des enregistrements précédemment insérés dans le buffer de télétraitement obtenu par UPDLST dans la mesure où la place est disponible, sinon on laisse ce buffer inchangé, ou on en crée un nouveau qu'on initialise avec l'enregistrement que l'on vient de compacter et on l'insère dans la liste.

Longueur	RCB	SRCB		texte
2	1	1	1	longueur-5

Enregistrement fourni par l'appelleur

RCB	SRCB		SCB	chaîne	...	SCB	chaîne	EØR
-----	------	--	-----	--------	-----	-----	--------	-----

Enregistrement compacté

Dans le cas où l'appelleur fournit un enregistrement sans texte (le champ longueur contient 5), l'enregistrement compacté contient seulement les champs RCB et SRCB de l'appelleur suivis d'un octet nul (EØR), ce qui correspond soit à des fonctions ne nécessitant pas de texte précisées par le RCB (par exemple : demande de transmettre une fonction), soit à l'envoi d'une ligne vide (indication de fin de fichier dans le protocole de 'multi-leaving').

On va décrire en pseudo-algol l'algorithme de compactage utilisé, en désignant par I un index du texte à compacter (I=0 indexe le premier octet de

texte à compacter), par J un index de la zone de travail dans laquelle sera rangé l'enregistrement compacté, par N le nombre de caractères du texte à compacter.

```

I:=0;
J:=0;
tant que I < N faire
  début
    CHERCHE(I);
    si lgt#0 alors
      début
        si index#I alors
          début
            CØPY(I,INDEX-I,J);
            I:=INDEX;
          fin;
        BLØC(I,LGT,J);
        I:=INDEX+LGT;
      fin;
    sinon
      début
        CØPY(I,N-I,J);
        aller à FIN;
      fin;
  fin;
FIN: FIN(J);

```

On utilise pour cette description les procédures CHERCHE(I), CØPY(I,K,J), BLØC(I,K,J) et FIN(J) dont les rôles sont les suivants :

CHERCHE(I) : Son rôle est de déterminer si, à partir du caractère du texte à compacter indexé par I, il existe dans la zone de texte restant à compacter plus de 2 octets successifs identiques. Pour

faire cette recherche on effectue un TRT (Translate and Test) sur une zone de travail obtenue avant de commencer le compactage en effectuant un 'et logique' de la zone de texte à compacter sur elle-même décalée d'un octet. Tout octet à zéro dans cette zone correspond à 2 octets successifs identiques dans la zone à compacter. Une instruction TRT permet d'obtenir le premier octet nul, si l'octet suivant est aussi à zéro on a trouvé 3 caractères successifs identiques, on cherche alors par un nouveau TRT le premier octet non nul pour obtenir le nombre de caractères successifs identiques, si l'octet suivant n'est pas à zéro on reprend la recherche d'un octet nul à partir de ce point.

L'avantage de cette solution est d'exécuter peu d'instructions dans le cas où il n'y a pas de possibilité de compactage.

En retour de la procédure CHERCHE :

LGT = 0 Indique qu'on n'a pas réussi à trouver 3 octets successifs identiques.

LGT ≠ 0 Indique le nombre d'octets successifs identiques trouvés depuis la position repérée par INDEX.

COPY(I,K,J) Permet de ranger dans l'enregistrement compacté, à partir de la position J une chaîne de caractères de longueur K sans répétitions qui débute dans l'enregistrement à compacter à la position I. C'est-à-dire qu'il y a construction de une ou plusieurs chaînes de la forme :

11 jj jjjj | chaîne de jjjjjj caractères (jjjjjj<63)

BLØC(I,K,J) Permet de ranger dans l'enregistrement compacté, à partir de la position J une chaîne de caractères répétitifs de longueur K qui débute dans l'enregistrement à compacter à la position I. Si le caractère répété est un blanc, il y a construction de une

ou plusieurs chaînes réduites à un octet de la forme :

| 1 0 0 j jjjj | avec j jjjj < 31 (indique jjjjj blancs successifs)

Si le caractère répété n'est pas un blanc, il y a construction de une ou plusieurs chaînes de 2 octets de la forme :

| 1 0 1 j jjjj | c | avec jjjjj < 31 (indique jjjjj caractères c successifs)

FIN(J) Il faut insérer à la position J de l'enregistrement compacté un SCB nul (EØR) pour indiquer la fin de l'enregistrement, puis ranger l'enregistrement compacté dans la liste spécifiée par l'appel leur doit dans le dernier buffer de télétraitement inséré dans cette liste soit dans un nouveau.

Les buffers de télétraitement sont retirés par le processus LINE dans les 3 listes auxquelles il a accès suivant l'ordre de priorité décroissant suivant :

- les enregistrements de contrôle
- les messages destinés à la console opérateur
- les enregistrements destinés à l'imprimante.

## 6.2 La gestion des appareils virtuels

A chaque appareil réel du terminal lourd peut être associé un ou plusieurs appareil(s) virtuel(s). Un bloc de contrôle (BCPSU) et une voie virtuelle sont associés à chaque appareil virtuel. On va s'intéresser successivement à la gestion des appareils de type console, de type lecteur de cartes et de type imprimante.

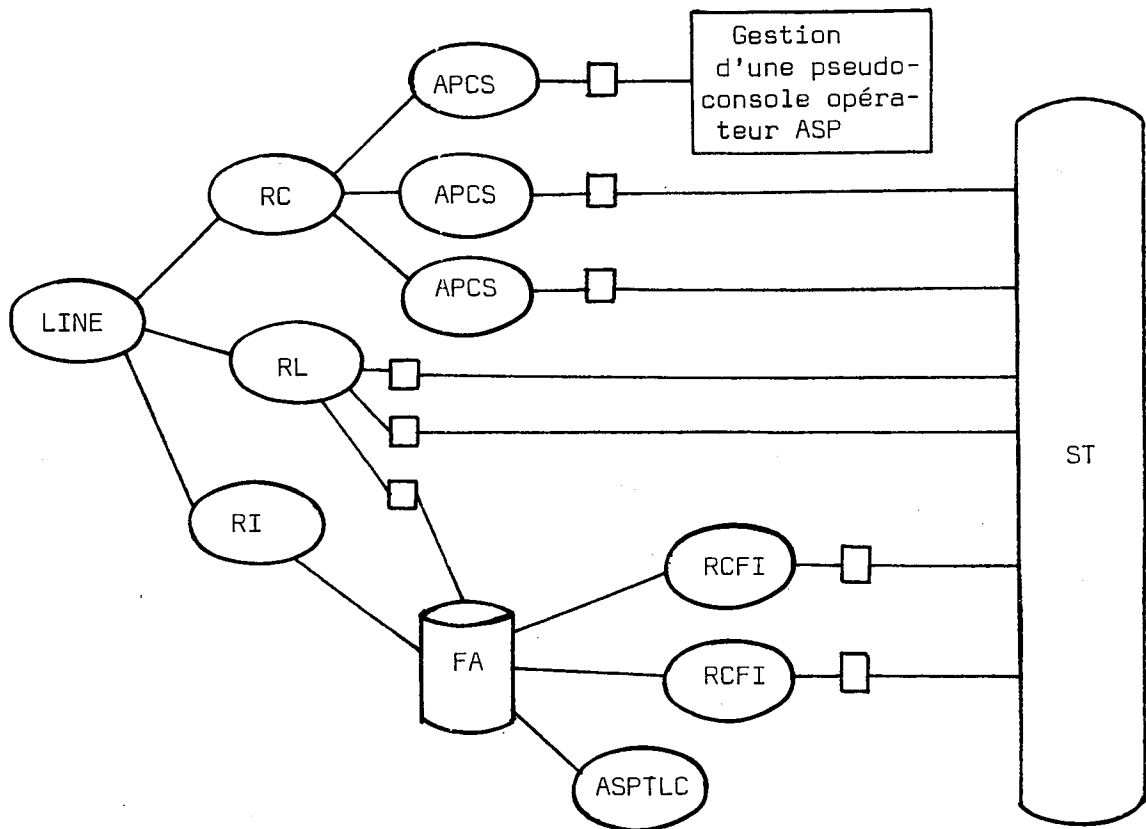
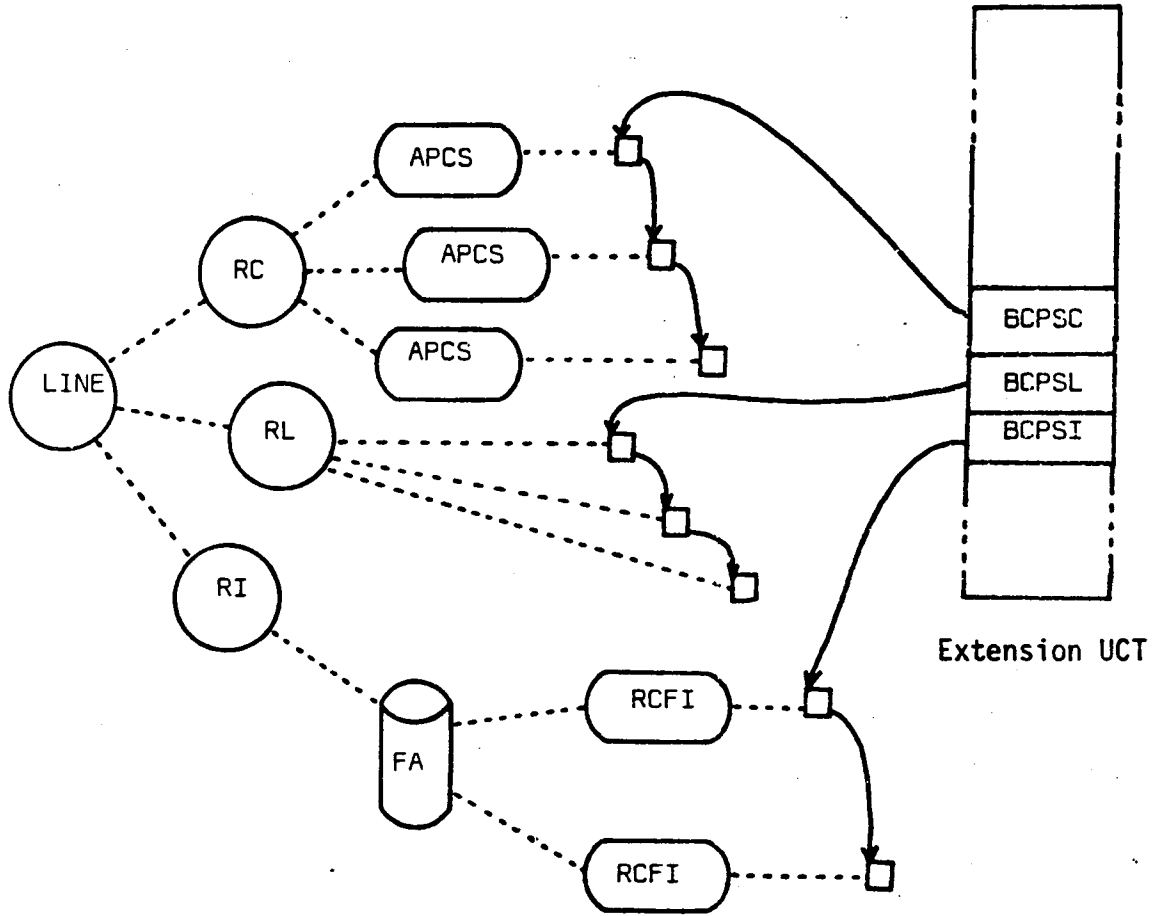


Schéma des processus concernés par les demandes de connexion. Les carrés représentent les blocs de contrôle associés aux appareils virtuels (BCPSU).



Adressage et chainage des BCPSU (blocs de contrôle associés aux appareils virtuels)



### 6.2.1 La gestion des appareils de type console

Un processus RC (représentant console) est associé à la console opérateur d'un terminal lourd, il retire des messages décompactés dans la liste RCLST et analyse les commandes correspondantes.

Chaque code commande peut être réduit à une seule lettre ou doit être utilisé sans abréviation.

La liste des commandes disponibles et leur syntaxe est spécifiée dans : Manuel opérateur d'un terminal lourd connecté au réseau CYCLADES par TELCOM [F.6].

#### 6.2.1.1 Traitement des demandes de connexion

Pour chaque connexion demandée par l'opérateur du terminal lourd, un bloc de contrôle caractérisant un nouvel appareil virtuel est initialisé et un appel au module DCNT (demande de contact) est effectué. Le module DCNT s'assure que l'abonné client associé au terminal lourd concerné est actif et l'active si nécessaire, puis il émet un D-CØNTACT (Cf Protocole client-serveur de terminaux TER 502-2) vers le serveur concerné, informe l'opérateur du système TELCØM qu'une demande de contact a été émise pour un terminal lourd et se met en attente d'une réponse du serveur pendant un délai maximum de 6 secondes. Si aucune réponse ne parvient ou si la réponse reçue est Ø-RUPTURE, on informe l'opérateur du terminal lourd de l'échec et on libère le bloc de contrôle précédemment initialisé avant de rendre le contrôle à l'appelleur de DCNT.

Dans le cas d'une réponse favorable (R-CØNTACT) le module DCNT établit la voie virtuelle et, si succès, informe l'opérateur de TELCOM et l'opérateur du terminal lourd de la fin normale de la prise de contact puis il complète le bloc de contrôle caractérisant le nouvel appareil virtuel et le chaîne aux précédents (3 chaînes distinctes suivant le type d'appareil).

Il faut alors créer, ou simplement réveiller s'il existe déjà, le processus chargé de l'exploitation de la voie virtuelle. Dans le cas d'un appareil de type lecteur de cartes il suffit de réveiller le processus RL (représentant

lecteur) qui prend en charge la gestion de tous les appareils virtuels de type lecteur de cartes. Dans le cas d'un appareil de type console il faut créer un processus APCS, chargé de la gestion d'un appareil virtuel de type console, et lui fournir les informations caractérisant l'appareil virtuel qui vient d'être créé (par l'intermédiaire de l'adresse du BCPSU). Dans le cas d'un appareil de type imprimante il faut créer un processus RCFI, chargé de la réception des fichiers destinés à l'imprimante du terminal lourd (de même que pour APCS on lui fournit l'adresse du BCPSU qui le concerne), et il faut également réveiller le processus RI chargé d'émettre les listes disponibles sur l'imprimante du terminal lourd.

Le module DCNT peut alors rendre le contrôle à son appeleur.

Dans le cas où le serveur est géré par le système TELCØM qui supporte le client, si la demande de connexion concerne un appareil de type imprimante on évite alors tout le mécanisme de prise de contact et d'établissement de voie virtuelle puisque le serveur local effectue déjà un travail analogue aux processus RCFI, c'est-à-dire qu'il range dans la file d'attente du système TELCØM les listes émises pour un client de traitements par lots. Il suffit alors d'informer l'opérateur du terminal lourd de la fin normale de la prise de contact et de réveiller le processus RI (représentant l'imprimante)..

Dans une première phase le mécanisme ainsi implémenté a permis d'effectuer facilement des tests depuis un terminal lourd supporté par le concentrateur en demandant des connexions au serveur de traitements par lots de la même machine. Mais une fois la mise au point suffisamment avancée un gain appréciable dans les temps d'accès a été obtenu en particularisant la connexion de tous les appareils au serveur local.

Dans le cas où la demande de connexion concerne un appareil de type console, pour une connexion au serveur local, on évite d'activer l'abonné client et d'établir une voie virtuelle. On affecte une pseudo console opérateur ASP au client (obtenue en utilisant l'identificateur du terminal lourd comme

index d'une table de pseudo consoles réservées à cet effet) et on range son adresse, ainsi que l'adresse de son bloc descripteur dans le BCPSU créé pour cette connexion, puis on active le processus APCS qui sera chargé de la gestion de cette pseudo console opérateur ASP. En cas de non disponibilité d'une pseudo console opérateur ASP correspondant à l'index associé à ce terminal lourd, un message d'erreur (E17) est émis sur la console opérateur du terminal lourd.

Dans le cas où la demande de connexion concerne un appareil de type lecteur de cartes, pour une connexion au serveur local, on évite aussi d'établir une voie virtuelle, le mot d'adressage du BVV est mis à zéro dans le BCPSU par le module DCNT (demande de contact) pour informer le processus RL qu'il devra écrire les flots de travaux reçus pour cette connexion dans la file d'attente du système TELCØM au lieu de les émettre sur la voie virtuelle du serveur connecté.

#### 6.2.1.2 Exploitation des voies virtuelles associées associées aux consoles.

Une fois créé et activé par un processus RC, un processus APCS associé à un appareil virtuel de type console se met en retrait sur la voie virtuelle correspondante.

Il gère cette voie suivant le protocole appareil virtuel (Cf. Virtual terminal protocol proposed specifications TEL 503.1) pour une connexion à l'alternat.

Les seuls messages de contrôle traités sont : Ø-YØUR-TURN et Ø-STATUS. Sur réception de Ø-STATUS/STATUS-ØUT la voie virtuelle correspondante est détruite par appel du module PVVRB qui est chargé d'effectuer une demande de purge VV à la ST, de supprimer de la chaîne à laquelle il appartient le BCPSU caractérisant l'appareil virtuel, de libérer sa mémoire et d'informer l'opérateur du terminal lourd ainsi que l'opérateur de TELCØM. Puis le processus APCS se détruit lui-même (service \$KILL).

Une destruction analogue de la voie virtuelle, du BCPSU et du processus APCS a lieu en cas d'anomalie détectée en cours d'utilisation de la voie virtuelle (par exemple dans le cas où le correspondant a effectué une déconnexion ou n'est plus accessible).

Les messages de texte (Ø-TEXT/ADDRESS/TEXT) sont émis sur la console opérateur du terminal, précédés du nom du serveur qui les a émis.

Le seul évènement reconnu est D-STATUS auquel on répond par un message de contrôle Ø-STATUS indiquant l'état de l'appareil virtuel client. Les autres évènements sont ignorés.

Les messages de texte sont émis vers un serveur de type console directement par le processus RC. On distingue pour effectuer ces émissions sans ambiguïté deux modes de travail pour la console opérateur du terminal lourd :

- Le mode général dans lequel l'opérateur du terminal lourd émet des commandes qui sont analysées par le processus RC du système TELCØM. Dans ce mode, pour émettre une attention ou un message vers la console d'un serveur connecté l'opérateur doit utiliser des commandes qui nomment le serveur auquel il veut s'adresser, par exemple :

```

a b 2      pour envoyer une attention 2 au serveur b
t a msg    pour envoyer le texte 'msg' au serveur a.

```

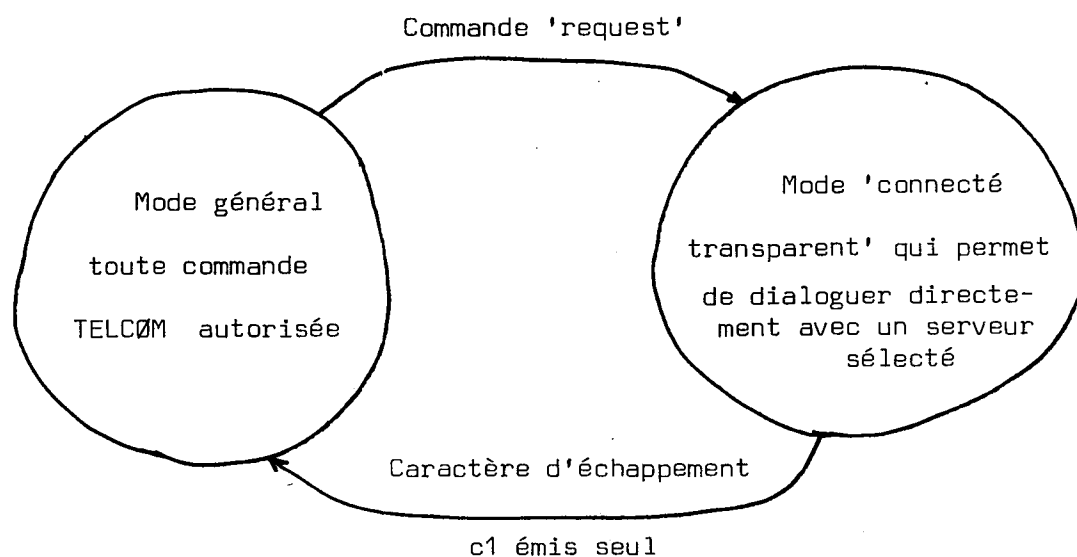
- Le mode 'connecté transparent' dans lequel les messages émis par l'opérateur sont directement dirigés par le processus RC vers le serveur dans ce mode. Une commande opérateur utilisable dans le mode général permet de passer en mode 'connecté transparent' pour un serveur délecté et de définir 2 caractères d'échappement C1 et C2 qui permettent respectivement de quitter le mode 'connecté transparent' et d'émettre une attention 1 vers le serveur connecté dans ce mode. Il s'agit de la commande 'request'.

Par exemple :

```

r a        permet de passer en mode connecté transparent pour le
           serveur a et d'utiliser les caractères d'échappement
           par défaut qui sont % et %.

```



**Schématisation des deux modes de travail sur  
de la console opérateur du terminal lourd  
connecté à TELCØM**

Dans le cas où la console du terminal lourd s'est connectée au serveur ASP du CP (Centre Participant) Cyclades qui supporte le concentrateur de terminaux lourds, le processus APCS est chargé de la gestion d'une pseudo console opérateur ASP et non d'une voie virtuelle. Ce cas est identifié par l'égalité de l'identificateur de l'abonné contact pour la connexion demandée et celui de l'abonné contact du serveur de traitements par lots du 360. Le processus APCS se met alors en attente d'une commande opérateur destinée à ASP ou d'une opération d'entrée-sortie lancée par ASP sur la pseudo console correspondante.

En cas de réception d'une commande ASP émise par l'opérateur du terminal lourd et livrée au processus APCS dans un buffer adressé par le BCPSU et acquis par le processus RC, une attention est émise sur la pseudo console opérateur ASP et le processus APCS se remet à nouveau en attente multiple.

Il sera réveillé dès que le système ASP lancera une lecture sur la pseudo console, suite à l'attention reçue.

Dans le cas où le réveil du processus APCS a lieu parce que le système ASP a lancé une opération d'entrée-sortie sur la pseudo console, l'IØB utilisé par ASP pour définir l'opération d'entrée-sortie est consulté. Seuls les CCW de code commande Read(2) ou Write(1) sont acceptés.

En cas de lecture, si une commande en attente de lecture est adressée par le BCPSU associé à la connexion, cette commande est rangée dans le buffer de lecture utilisé par ASP pour son opération de lecture et une fin normale d'opération d'entrée-sortie est simulée. Le buffer décrivant la commande est alors libéré et un  $\$PØST$  est fait pour réveiller éventuellement le processus producteur de commandes en attente de disponibilité d'émission de commande. S'il n'y a pas de commande en attente de lecture, une commande de longueur nulle est émise à ASP pour terminer la lecture.

En cas d'écriture, le message adressé par le programme canal d'écriture est compacté et livré au processus LINE pour émission sur la console du terminal lourd, à l'aide du service CMPT.

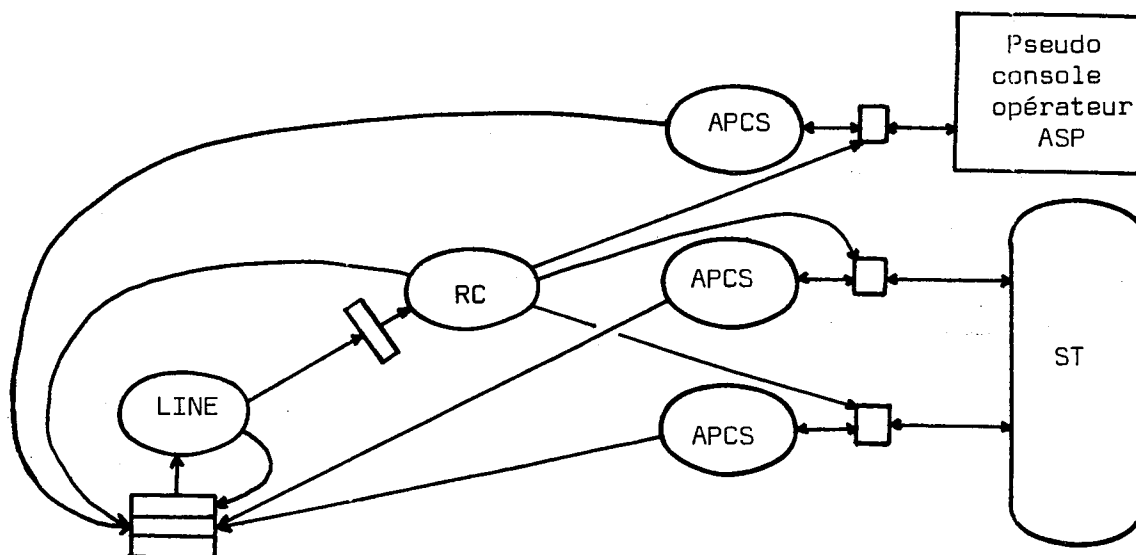
Après traitement de l'opération d'entrée-sortie lancée par ASP le processus APCS se remet en attente multiple comme à son initialisation.

La particularisation de la connexion d'une console au serveur local nécessite aussi un traitement particulier de certaines commandes par le processus RC :

- Dans le cas où une commande "Attention" adresse le serveur qui se trouve sur le même CP que le client, un message d'erreur est émis sur la console du terminal lourd pour l'informer qu'une attention est inutile pour ce type de connexion.
- Dans le cas où une commande "Transmit" adresse le serveur qui se trouve sur le même CP que le client, le texte de la commande à émettre sur la pseudo console du serveur est rangé dans un buffer dont l'adresse est mise

dans le BCPSU associé à la connexion si le pointeur correspondant est libre. S'il n'est pas libre, le processus RC est mis en attente de disponibilité sur l'ECB correspondant. Dès que le BCPSU adresse le buffer construit, le processus consommateur APCS est réveillé.

- Dans le cas où la console du terminal lourd est en "mode connecté réseau" pour le serveur qui se trouve sur le même CP que le client, l'émission d'une attention est refusée de même que pour le traitement de la commande "attention" et les commandes sont livrées au processus APCS de la même façon qu'au cours du traitement de la commande "transmit" en mode général.



Cheminement des informations  
concernant la gestion des appareils virtuels de type console  
(console opérateur terminal d'un système conversationnel)

### 6.2.2 La gestion des appareils de type lecteur de cartes

En ce qui concerne la connexion du lecteur de cartes aux serveurs de traitement par lots le processus RL (Représentant Lecteur) se charge de la gestion des différentes voies virtuelles activées. Il retire des images de cartes décompactées dans la liste RLLST et les expédie sur la voie virtuelle associée au premier BCPSU de la chaîne des blocs de contrôle associés aux lecteurs de cartes virtuels. La commande opérateur 'START' permet de faire passer en tête de chaîne le BCPSU de l'un quelconque des lecteurs de cartes connectés, c'est-à-dire que les cartes lues sur le lecteur réel seront émises vers le dernier lecteur virtuel pour lequel une commande START a été émise par l'opérateur du terminal lourd ou, si aucune commande START n'a été émise, vers le premier lecteur de cartes virtuel qui a été connecté.

Pour les autres voies virtuelles le processus RC se contente de retirer les événements et les lettres reçues et il ne reconnaît parmi ceux-ci que le message de contrôle Ø-STATUS et l'évènement D-STATUS auquel il répond par un message de contrôle Ø-STATUS. Les appareils virtuels correspondants sont toujours considérés comme étant dans l'état non prêt à moins qu'un changement d'état soit spécifié par le correspondant. Le seul lecteur de cartes virtuel qui peut être dans l'état prêt est celui sur lequel sont émises les cartes reçues du lecteur réel.

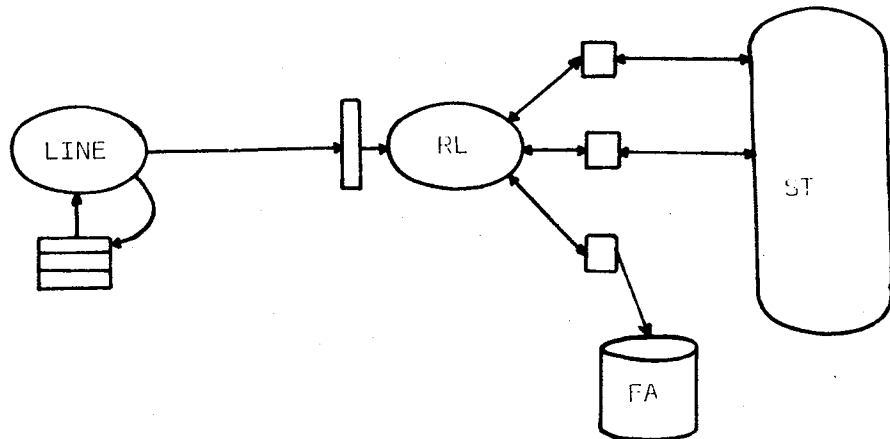
Pour effectuer cette gestion multiple le processus RL se met en attente sur un seul ECB (BCPLECB se trouvant dans l'extension de l'UCT de la ligne) et peut être réveillé pour l'une des raisons suivantes :

- un nouveau BCPSU a été rajouté à la chaîne des BCPSU de type lecteur de cartes associée à cette ligne. Ce rajout, ainsi que l'activation de la voie virtuelle correspondante a été effectué par le processus RC, qui a également réveillé le processus RL, à la suite d'une demande de connexion du lecteur d'un serveur de traitement par lots effectué par l'opérateur du terminal lourd.
- Un évènement ou une lettre est arrivé sur l'une des voies virtuelles associées aux lecteurs des serveurs de traitement par lots connectés.



- La liste RLLST est passée de vide à non vide (une carte a été insérée dans la liste par le processus LINE).

Le cas de la connexion d'un lecteur de cartes au serveur se trouvant sur le même CP que le concentrateur de terminaux lourds est particularisé. Dans ce cas le buffer de travail dans lequel sont rangées les cartes lues en format PAV est écrit dans la file d'attente du système TELCOM au lieu d'être émis sur la voie virtuelle du serveur connecté.



Cheminement des informations concernant la gestion des lecteurs de cartes virtuels

### 6.2.3 La gestion des appareils de type imprimante

Pour permettre les réceptions simultanées des listes provenant de plusieurs serveurs connectés, les fichiers reçus sont rangés dans la file d'attente du système TELCOM avant d'être émis vers l'imprimante du terminal lourd.

### 6.2.3.1 Retrait et impression des listes de la file d'attente des systèmes TELCØM.

Un processus RI (Représentant Imprimante) est associé à l'imprimante d'un terminal lourd distant.

Il est activé par la première connexion d'une imprimante effectuée sur demande de l'opérateur du terminal lourd. Il adresse alors une requête au processus FA pour se mettre en attente de disponibilité dans la file d'attente de TELCØM d'un fichier de type imprimante destiné à l'utilisateur correspondant au terminal lourd. Dès qu'un tel fichier est disponible, l'opérateur de TELCØM ainsi que celui du terminal lourd en sont informés, puis le processus RI adresse des requêtes de lecture au processus FA. Les buffers obtenus contiennent des messages en format PAV (protocole appareil virtuel), qu'il faut transformer en enregistrements respectant le format du 'multi-leaving' HASP, puis émettre dans la troisième liste de la liste multiple du processus LINE correspondant.

La transformation PAV-'multi-leaving' doit se faire en gardant toujours disponible le prochain enregistrement à émettre de façon à pouvoir éventuellement le compléter avant de l'émettre dans le cas où il serait suivi d'un message de texte PAV ayant l'adressage CN (continu). Cette disponibilité du prochain enregistrement est aussi utile dans tous les cas où plusieurs messages PAV peuvent se transformer en un seul enregistrement de 'multi-leaving', par exemple plusieurs lignes vides successives en format PAV peuvent se transformer en un enregistrement spécifiant saut immédiat de plusieurs lignes ou encore une ligne non vide suivie de lignes vides peuvent se transformer en : saut de n lignes après impression.

Dès que la fin du fichier est signalée par le processus FA, le dernier enregistrement est émis, les opérateurs sont informés de la fin de la liste et une requête `ABØRT` est adressée au processus FA pour le fichier qui vient d'être lu, ce qui permet d'effectuer sa suppression de la file d'attente. Le processus RI se remet après cela en attente de disponibilité d'un nouveau fichier destiné à l'imprimante de l'utilisateur correspondant au terminal lourd.

On ne sépare pas les fichiers émis sur l'imprimante du terminal lourd au sens du 'multi-leaving', c'est-à-dire qu'une demande de transmettre et une attente de permission ne sont effectuées qu'une seule fois, après la première demande de connexion de l'imprimante effectuée par l'opérateur du terminal lourd. Aucun enregistrement de fin de fichier n'est émis concernant l'imprimante.

Le comportement d'un processus RI est assez comparable à celui d'un processus IMPR du serveur de traitement par lot chargé de fournir à un client distant les listes qui le concerne au fur et à mesure de leur disponibilité dans la file d'attente. La seule différence est le moyen d'accès au client.

Quelques commandes analysées par le processus RC permettent à l'opérateur d'un terminal lourd de prendre des décisions sur la liste en cours d'impression (arrêt, redémarrage, répétition).

Sur réception d'une commande :

```

    PRINTER RESTART
    PRINTER REPEAT
    PRINTER CANCEL
  
```

un bit d'état est forcé à 1 dans l'extension de l'UCT de la ligne correspondante pour spécifier l'action requise au processus RI.

Sur réception de :

```

    PRINTER START
  
```

un `ΣPØST` est effectué pour réveiller le processus RI et le mettre en attente de disponibilité d'un fichier à imprimer.

Le processus RI teste après chaque lecture dans la file d'attente si une commande 'PRINTER RESTART' ou 'PRINTER CANCEL' a été émise par l'opérateur. En cas de 'RESTART' une requête `ΣCLOSE` pour le fichier imprimante en cours de lecture dans la file d'attente est émise, l'indicateur de 'RESTART' est remis à zéro et le processus RI est mis en attente (réveil par 'PRINTER START' ou 'CONNECT'). En cas de 'CANCEL', l'indicateur de 'CANCEL' est remis à zéro et une requête `ΣABØRT` est émise pour le fichier imprimante en cours de lecture dans la file d'attente, les messages de fin de liste sont émis et le processus RI passe à la liste suivante (`ΣØPEN`).

Le processus RI teste, après émission du dernier enregistrement d'un fichier, si une commande 'PRINTER REPEAT' a été émise. Si oui, un  $\$CLOSE$  est effectué au lieu d'un  $\$ABORT$  pour le fichier imprimante dont on vient de finir la lecture dans la file d'attente. Ceci permet d'obtenir au prochain  $\$OPEN$  pour le même usager une nouvelle fois la liste dont la lecture vient de se terminer.

### 6.2.3.2 Réception des listes des serveurs connectés

Du fait que le processus RI effectue des retraits dans la file d'attente directement pour l'usager qui le concerne, il est absolument nécessaire de particulariser l'accès au serveur de traitement par lots BATCH360 qui tourne sous le même système, car sans cela le processus IMPR créé par le serveur serait en retrait de listes pour le même usager que le processus RI, ce qui n'est pas permis.

#### a) Réception des listes du serveur local

Dans le cas du serveur local de traitement par lots les listes pour l'usager correspondant au terminal lourd sont directement émises par le processus ASPTLC du serveur dans la file d'attente sous le format reconnu par le processus consommateur RI.

#### b) Réception des listes des serveurs distants

Dans le cas de la connexion de l'imprimante à un serveur distant on a vu (paragraphe 2.1.1) qu'il y a création d'un processus RCFI chargé de recevoir une liste sur une voie virtuelle et de la ranger dans la file d'attente. Dès sa création un processus RCFI adresse une requête d'ouverture de fichier de type imprimante au processus FA. Il identifie le fichier en utilisant les champs I-SERVICE et I-ENTREE du serveur connecté et il identifie l'utilisateur par le nom d'usager associé au terminal lourd et créé comme suit : un caractère pour identifier la machine (A dans le cas du 360 du CIOG), 3 caractères de la forme RXX qui identifient le terminal lourd (cet identificateur du terminal lourd se trouve dans l'extension de l'UCI de la ligne) et 4 caractères blancs.

Le processus RCFI se met alors en retrait sur la voie virtuelle qui lui est associée. Sur réception d'un message de texte (Ø-TEXT/ADDRESS/TEXT) il range le message reçu dans le buffer courant associé à la gestion de la file d'attente s'il y reste suffisamment de place, sinon il écrit le buffer courant dans la file d'attente puis l'initialise avec le message reçu. Dans le cas où le message de texte contient 'New Book' pour adresse, le fichier en cours est fermé (requête %CLØSE émise à FA) et un nouveau est ouvert.

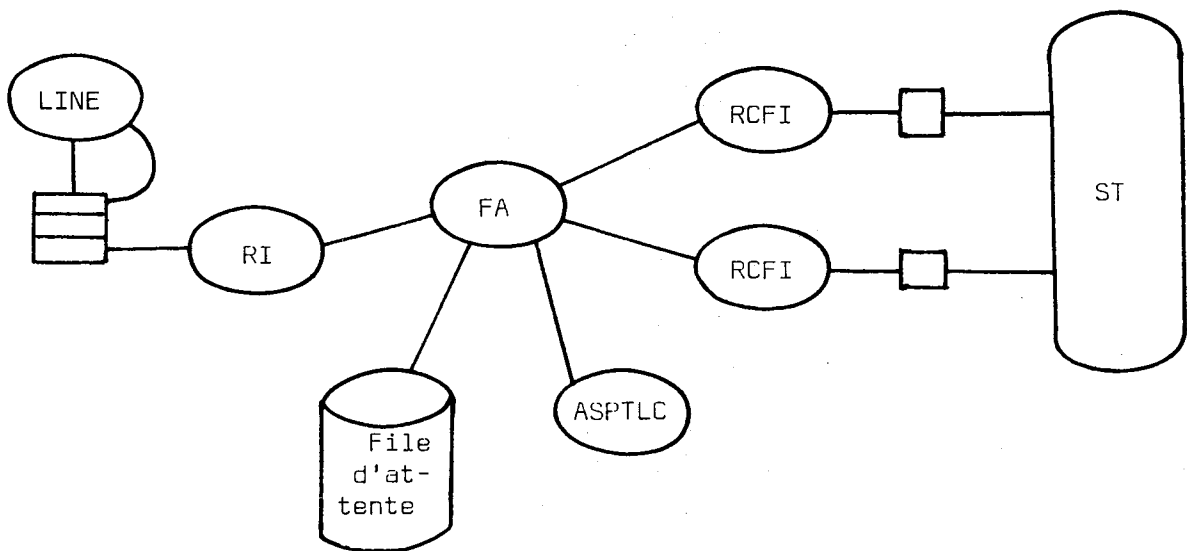
Le seul message de contrôle reconnu est 'Ø-STATUS/STATUS-VALUE', les autres sont ignorés. Si l'état spécifié est ØUT un appel au module PVVRB permet de supprimer la voie virtuelle ainsi que le bloc de contrôle associé à l'imprimante virtuelle puis le processus RCFI émet une requête SCLØSE pour le fichier partiellement écrit et se détruit lui-même (service %KILL).

Une destruction analogue a lieu en cas d'anomalie grave détectée en cours d'utilisation de la voie virtuelle.

Le seul évènement reconnu est D-STATUS auquel on répond par un message de contrôle Ø-STATUS indiquant l'état de l'imprimante virtuelle du client. Les autres évènements sont ignorés.

La fin normale du processus RCFI a lieu après une demande de déconnexion effectuée par l'opérateur du terminal lourd. Dans ce cas le processus RCFI détruit la voie virtuelle, ce qui signale une anomalie grave dans la gestion de la voie virtuelle au processus RCFI, lequel termine proprement son travail.

On rappelle qu'un %CLØSE effectué sur un fichier vide (cas où RCFI vient d'ouvrir un nouveau fichier et ne l'a pas encore reçu) n'effectue pas de modification des blocs descripteurs de la file d'attente (fichier ignoré) tandis que s'il s'agit d'un fichier partiellement reçu, la partie reçue sera correctement enregistrée.



Cheminement des informations  
concernant la gestion des imprimantes

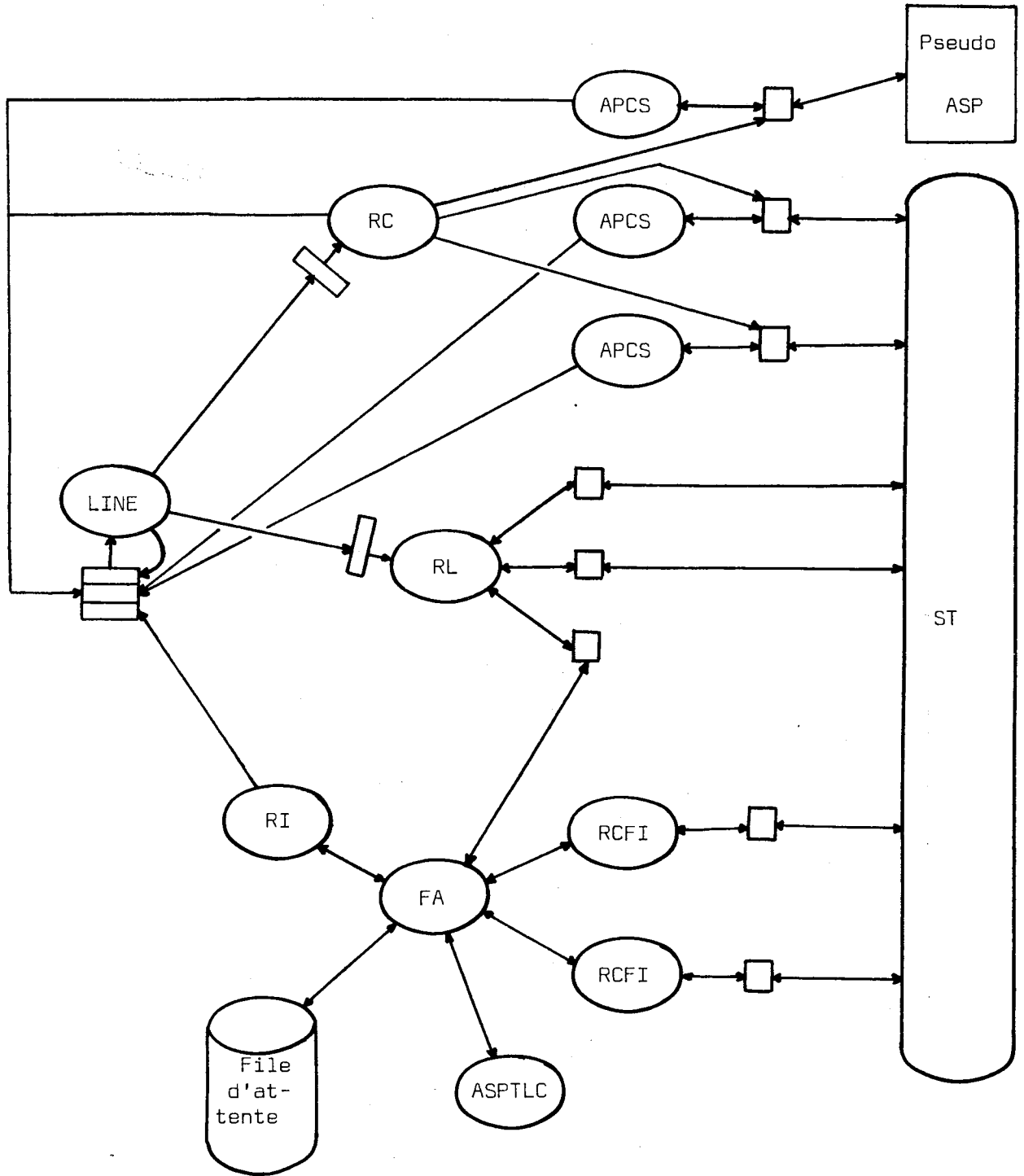
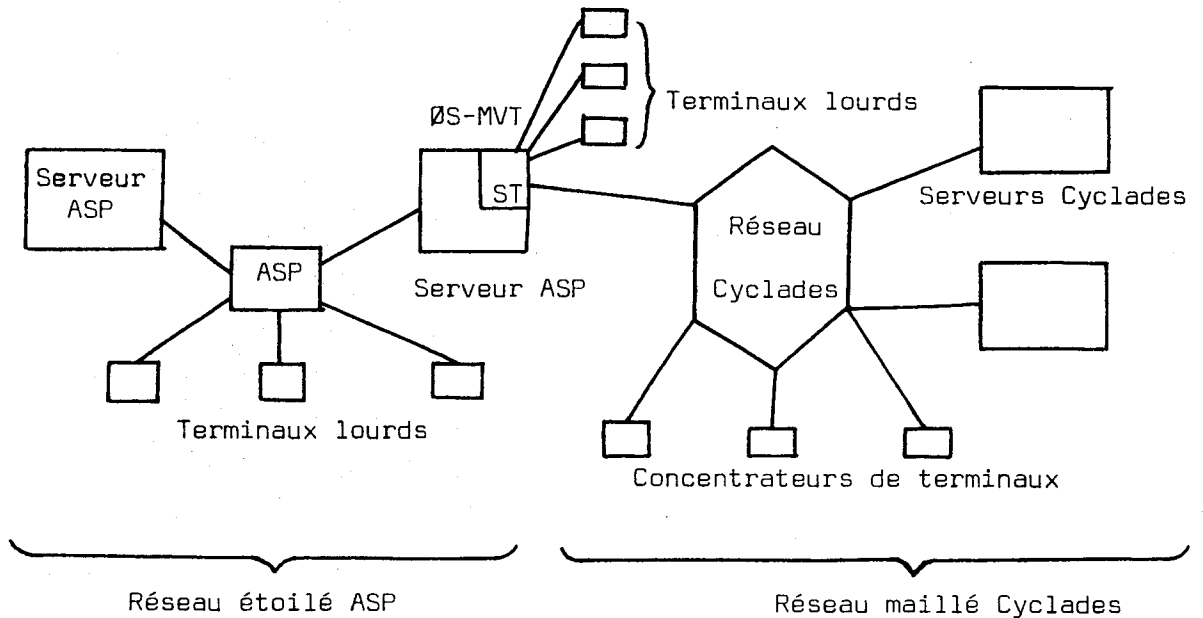


Schéma général du cheminement des informations associées à la gestion d'un terminal lourd client de traitements par lots du réseau CYCLADES

## 7. CONCLUSION

On peut considérer que le système ASP constitue un logiciel de support de réseau étoilé. L'ordinateur central, qui peut être également un serveur du réseau, supporte le système ASP et les ordinateurs qui lui sont connectés sont soit des serveurs du réseau ("Main processor" dans la terminologie ASP) soit des terminaux lourds clients.

Le choix d'implémenter la station de transport sous le contrôle d'une tâche du système  $\text{\O S}$  (justifié dans l'introduction de ce document) signifie alors qu'on a inséré du logiciel sur l'ordinateur serveur du réseau étoilé pour le connecter au réseau maillé Cyclades.



On peut alors se demander, en se plaçant dans ce cadre plus général de connexion de deux réseaux, si la connexion est réalisée au mieux pour le service de traitement par lots auquel on s'est intéressé ici.

Tout terminal lourd client du réseau Cyclades peut accéder à l'un quelconque des serveurs de traitements par lots du réseau ASP, puisqu'il sait émettre



ses travaux vers la file d'attente du système ASP et qu'il peut à l'aide de cartes de contrôle sélectionner le serveur sur lequel il veut travailler. Par contre l'utilisateur d'un terminal lourd du système ASP ne peut pas accéder à un serveur de traitement par lots du réseau Cyclades.

Le fait d'avoir développé un concentrateur de terminaux lourds clients du réseau Cyclades a permis de remédier à cette difficulté puisque ces terminaux lourds, sans modification de logiciel de leur côté (ni de leur liaison dans le cas de notre configuration) deviennent clients à la fois du système ASP et du réseau Cyclades.

Il semble cependant que pour ce type de services un problème reste difficile à résoudre sur un réseau maillé hétérogène à l'heure actuelle, c'est celui de la répartition des travaux. En supposant que tout utilisateur connaisse pour chacun de ses travaux les machines susceptibles de le servir il faudrait encore un mécanisme de choix de la machine la plus disponible à un moment donné, assez complexe à élaborer.

Dans le cas d'un réseau homogène étoilé géré par ASP la répartition de la charge des serveurs se fait sous le contrôle de l'ordinateur central qui est informé de la fin de chaque travail et qui peut demander à chacun des serveurs qui lui sont connectés la taille du plus grand trou de mémoire centrale disponible à l'aide de la commande "SIZE" implémentée sur le système serveur. On peut envisager une solution semblable dans le cas d'un réseau maillé qui consiste à permettre aux clients de questionner chacun des serveurs non pas sur la disponibilité de la mémoire centrale mais plutôt sur une estimation globale des travaux en attente. Cette solution se prête bien au développement de concentrateurs de terminaux lourds auxquels on affecte alors le rôle de répartir la charge des serveurs. Une autre solution consiste à attribuer ce travail aux serveurs eux-mêmes en leur permettant de questionner les autres serveurs pour leur transmettre des travaux si nécessaire. La mise en place de ces deux mécanismes simultanément serait sans doute la solution la plus souple, en laissant une plus grande possibilité d'intervention humaine dans le deuxième mécanisme pour permettre d'affecter la charge en fonction d'impératifs locaux tels que la disponibilité du personnel.

Dans le cas d'un réseau hétérogène maillé une autre difficulté apparaît : celle de l'incompatibilité des langages de commande des travaux. L'étude d'un langage de commande réseau (NJCL : Network Job Control Langage) a été réalisée mais n'a pas encore conduit à une implémentation et une mise en exploitation d'un tel service pour le réseau Cyclades.

Dans le cas du serveur de traitement par lots étudié ici (système ASP-ØS/MVT on n'a permis l'accès à la station de transport qu'à des processus du système TELCOM ou à des tâches ØS bien spécifiques (soumission des travaux, retrait des listes). L'étape suivante consiste à permettre cet accès à toutes les tâches ØS et de la façon la plus transparente possible pour l'utilisateur. En adressant des unités fictives réservées à cet effet, le programmeur pourra utiliser les méthodes d'accès standard du système ØS pour accéder à des fichiers distants. Il faudra pour cela modifier le superviseur d'entrées-sorties du système ØS pour que l'exécution des opérations d'entrée-sortie demandées sur ces unités particulières soit remplacée par des appels aux services de transfert de fichiers accédant à la station de transport. Il faudra également fournir au programmeur des conventions de spécification des fichiers du réseau adressés qui devront être traitées en respectant le protocole de transfert de fichiers. Il est certain qu'un tel développement ne peut être envisagé qu'avec des vitesses de transfert sur le réseau suffisamment élevées puisque la logique générale de développement du service de traitement par lots sur les réseaux tend à transporter les unités lentes sur les ordinateurs périphériques pour laisser aux gros serveurs que des unités rapides.

D'autres extensions sont en cours de développement telles que la synchronisation des processus de serveurs différents, l'accès à des bases de données réparties chez plusieurs serveurs, la mise en place de méthodes d'accès réseau.

Il serait également souhaitable de favoriser le développement d'ordinateurs "clients" (concentrateurs de terminaux et terminaux lourds) plus évolués que ceux utilisés jusqu'à présent : possibilités d'édition locale de petits fichiers, gestion de files d'attente pour les travaux destinés aux serveurs du réseau et les fichiers en attente d'impression ou d'édition locale.

## ANNEXE 1

LISTE DES MACROS INSTRUCTIONS DISPONIBLES SOUS LA VERSION ØS  
DU SYSTÈME SYNCØP, ASSOCIÉES AUX SERVICES DEVANT SE DÉROULER  
EN MODE MAITRE (APPEL DE LA SVC SYNCØP)

ØATTN

ØATTN appelle la SVC SYNCØP avec le code ATTNCØDE. L'appel est permis pour le système SYNCØP seulement.

ØATTN permet de simuler une interruption de type attention sur la pseudo console opérateur ASP dont l'adresse est contenue dans le registre 1 avant l'appel.

[symbole]	ØATTN	
-----------	-------	--

ØCEDE

ØCEDE appelle la SVC SYNCØP avec le code CEDECØDE. L'appel est permis pour le système SYNCØP seulement.

ØCEDE permet de simuler la fin normale d'une opération d'entrée sortie sur une pseudo unité pour la méthode d'accès utilisant cette unité.

L'utilisateur de cette macro instruction doit être un processus SYNCØP qui se comporte comme une unité d'entrée-sortie. Il émet ØCEDE après analyse du programme canal et simulation de son exécution. A l'appel R2 doit contenir l'adresse de l'IØB et R3 cell du TCB de la tâche ØS qui a lancé l'opération d'entrée-sortie sur la pseudo unité.

[symbole]	ØCEDE	
-----------	-------	--

SIØAPPN

SIØAPPN appelle la SVC SYNCØP avec le code APDGCØDE. L'appel est permis pour le système SYNCØP seulement.

SIØAPPN permet de définir le module IØAPPN du système SYNCØP comme 'normal channel end appendage' et comme 'abnormal channel end appendage' pour un data set ØS sur lequel un ØPEN EXCP a été effectué.

[symbole]	SIØAPPN	[DCBAD=adr1][,NØRMAL=adr2] [,ABNRML=adr3]
-----------	---------	--

adr1 : registre contenant l'adresse du DCB du data set concerné (spécifié entre parenthèses), ou adresse du DCB.

Si ce paramètre est omis, sa valeur par défaut est (R1).

adr2 : adresse de l'instruction qui sera exécutée après la fin normale du service requis.

Si ce paramètre est omis, l'instruction qui suit la macro SIØAPPN sera exécutée après la fin normale.

adr3 : adresse de l'instruction qui sera exécutée en cas d'impossibilité de réaliser le service requis.

Si cette adresse est omise, sa valeur par défaut est l'adresse de l'instruction qui suit la macro SIØAPPN, de même que pour adr2.

ØKO

ØKO appelle la SVC SYNCØP avec le code KOEXCØDE.

L'appel est permis pour le système SYNCØP seulement.

ØKO permet de demander l'exécution, sans modification, de l'ins-

SDEND

SDEND appelle la SVC SYNCØP avec le code DENDCØDE.

L'appel est permis pour le système SYNCØP seulement.

SDEND permet de simuler une interruption d'entrée-sortie de type 'channel end' sur la pseudo console opérateur ASP dont l'adresse est contenue dans le registre 1 avant l'appel.

[symbole]	SDEND	
-----------	-------	--

SDMSK

SDMSK appelle la SVC SYNCØP avec le code DMSKCØDE.

L'appel est permis pour le système SYNCØP et pour tout programme se déroulant sous le contrôle d'un PSW de clé de protection 0.

SDMSK permet de démasquer les interruptions d'entrée-sortie, externes, machine check et de revenir en mode programme.

[symbole]	SDMSK	
-----------	-------	--

SEX

SEX appelle la SVC SYNCØP avec le code EXECCØDE. L'appel est permis pour le système SYNCØP seulement.

SEX permet de demander l'exécution de l'instruction adressée par R14 en mode superviseur, clé de protection 0. Son exécution se fait à l'aide de l'instruction EXECUTE en utilisant le masque contenu dans le registre 1.

[symbole]	SEX	
-----------	-----	--

struction adressée par R15 par l'intermédiaire d'une instruction EXECUTE se déroulant en mode superviseur, clé de protection 0.

[symbole]	ΣKO	[EX=adresse]
-----------	-----	--------------

adresse : registre contenant l'adresse (spécifié entre parenthèse) ou adresse de l'instruction à exécuter en mode superviseur, clé 0.

Sa valeur par défaut est EX=(R15)

### ΣMSK

ΣMSK appelle la SVC SYNCØP avec le code MSKCØDE.

L'appel est permis pour un processus du système SYNCØP et pour une tâche ØS de clé 0.

ΣMSK permet de masquer les interruptions d'entrée-sortie, externes, machine check, et de passer en mode superviseur.

[symbole]	ΣMSK	
-----------	------	--

### ΣPEXCP

ΣPEXCP appelle la SVC SYNCØP avec le code PXCPCØDE. L'appel est permis par un programme se déroulant sous le contrôle d'un PSW de clé de protection nulle seulement.

L'appel de ΣPEXCP se fait dans le "SIØ Appendage" d'une opération d'entrée-sortie lancée sur une pseudo unité, avec les conventions d'appel des registres d'un "SIØ Appendage" : R1 adresse le RQE et R2 adresse l'IØB. Il permet de signaler au processus SYNCØP chargé de la gestion de cette pseudo unité qu'une opération d'entrée-sortie a été lancée et il lui fournit les informations nécessaires

pour la traiter (adresse de l'IØB et adresse du TCB de l'appel-  
leur). Le retour au superviseur d'entrée-sortie du système ØS  
doit se faire avec demande de non exécution du SIØ.

[symbole]	ØPEXCP	
-----------	--------	--

### ØSVCSYNC

ØSVCSYNC permet de générer les définitions des codes mnémoniques  
utilisés par toutes les macro instructions effectuant un appel  
à la SVC SYNCØP.

Appelée dans la version CP de SYNCØP cette macro ne génère  
aucune instruction.

	ØSVCSYNC	
--	----------	--

### ØTRA

ØTRA appelle la SVC SYNCØP avec le code TRACCØDE. L'appel est  
permis pour le système SYNCØP et pour tout programme se dérou-  
lant sous le contrôle d'un PSW de clé de protection nulle.  
ØTRA permet de compter le nombre de passages pour chaque appe-  
leur et de mémoriser l'ordre des derniers appels effectués.

[symbole]	ØTRA	
-----------	------	--

## ANNEXE 2

UNE TRACE DE LA CONSOLE OPÉRATEUR DU SYSTÈME TELCOM  
EN COURS D'EXPLOITATION

ENTER YOUR NAME

->telcom -----Nom réservé à l'opérateur du système TELCOM

CYCLADES/IMAG ST1

->start lmtr -----Création et activation du processus chargé  
de la gestion de la procédure de ligne pour  
la liaison avec le noeud CYCLADES.  
\*\*OK\*\*

->local  
\*\*OK\*\*

->start lr02 -----Création et activation des processus chargés  
du support du terminal lourd ARO2 en temps  
que client du réseau CYCLADES.  
\*\*OK\*\*

AR02 - LISTE @SCD0308 SYSMSG

AR02 - FIN LISTE

AR02 - LISTE @SCD0308 SYSPRINT

AR02 - FIN LISTE

AR02 - LISTE @SCD0308 SYSLOUT ----- Début d'impression du fichier SYSLOUT pour  
le job @SCD0308 sur l'imprimante du terminal  
lourd ARO2.

AR02 - FIN LISTE

AR02 - LISTE PART1 SYSMSG

AR02 - FIN LISTE

->store 8e988 0308e7d5 ----- Permet de modifier un mot dans la région  
affectée à TELCOM.  
\*\*OK\*\*

AR02 - FIN LISTE

AR02 - LISTE @SCS1517 SYSMSG

AR02 - FIN LISTE

AR02 - LISTE @SCS1517 SYSLOUT

AR02 - FIN LISTE

AR02 - LISTE @SCS1517 SYSPRINT

AR02 - FIN LISTE

AR02 - LISTE @SCS1317 SYSMSG

AR02 - FIN LISTE

AR02 - LISTE @SCS1317 SYSPRINT

AR02 - FIN LISTE

AR02 - LISTE @SCS1317 SYSLOUT

AR02 - FIN LISTE



D-CONTACT/IREP /BATCH360/CONS/0000/0000 -----Demande de contact pour la console  
de l'IREP.

R-CONTACT/0000/03000480 -----Réponse favorable.

CONS : VV 0001 ETABLIE POUR AB 0300

D-CONTACT/IREP /BATCH360/LECR/0000/0002 -----Demande de contact pour le lecteur  
de cartes de l'IREP.

R-CONTACT/0000/01000480

LECR : VV 0002 ETABLIE POUR AB 0100

IREP - LECTEUR NON PRET

D-CONTACT/IREP /BATCH360/IMPR/0000/0001 -----Demande de contact pour l'imprimante  
de l'IREP.

R-CONTACT/0000/02000480

IMPR : VV 0003 ETABLIE POUR AB 0200

IREP - LECTEUR PRET -----Changement d'état du lecteur de car-  
tes de l'IREP.

IREP : \*I A

IREP - FLOT RECU

IREP - LECTEUR NON PRET

IREP : \*I J=COMPRETR

IREP - LECTEUR PRET

IREP - FLOT RECU ----- Un flot de travaux en provenance  
de l'IREP vient d'être enregistré  
dans la file d'attente du système  
TELCOM et va être émis vers le  
système ASP.

IREP - LECTEUR NON PRET

IREP - LECTEUR PRET

IREP - FLCT RECU

IREP - LECTEUR NON PRET

AR02 - LISTE @SC0109 SYSMSG

AR02 - FIN LISTE

AR02 - LISTE @SC0109 SYSPRINT

AR02 - FIN LISTE

AR02 - LISTE @SC0109 SYSLOGOUT

AR02 - FIN LISTE

->map drcons -----Permet d'obtenir l'adresse de début du module DRCØNS  
DRCONS39 0008E210

->display 8e990 -----Permet d'obtenir le contenu d'un mot quelconque de la  
06E990 = 01000000 mémoire centrale.

```

IREP      : *I J=COMPCTAB
IREP      - LECTEUR PRET
IREP      : *I J=CATT
IREP      - FLOT RECU
IREP      - LECTEUR NON PRET
IREP      : *I A
IREP      : *I A
IREP      : *I J=3172,P=)
IREP      : *I J=3172,P=9 -----Trace d'une commande opérateur émise
IREP      - LISTE COMPGENE SYSMSG          pour ASP sur la console du concentra-
AR02      - LISTE DISKPP  SYSMSG          teur de l'IREP.
AR02      - FIN LISTE
AR02      - LISTE DISKPP  SYSPRINT
AR02      - FIN LISTE
AR02      - LISTE DISKPP  FT06F001
IREP      - FIN LISTE
IREP      - LISTE COMPGENE SYSPRINT
IREP      - FIN LISTE
IREP      - LISTE COMPGENE COMMANDE
IREP      - FIN LISTE
CONS      : SUSP AB 0300 -----Suspension de la voie virtuelle asso-
D-COMTACT/IREP /BATCHE60/CONS/0000/0000 ciée à la console opérateur du concen-
R-COMTACT/0000/03000480          trateur de l'IREP et désactivation du
CONS      : VV 0001 ETABLIE POUR AB 0300 processus correspondant.
IREP      : *I J=COMPELG
IREP      : *I J=3183,R

```

```

AR02      - LISTE BRACUDA  SYSMSG
AR02      - FIN LISTE
AR02      - LISTE BRAUDA  FT06F001
AR02      - FIN LISTE

```

->z all on coupe ... -----On informe tous les clients connectés.Le message  
\*\*OK\*\* "on coupe..." est émis sur les voies virtuelles  
associées aux consoles connectées.

->return -----Arrêt du système TELCØM.

A N N E X E 3

MANUEL OPÉRATEUR D'UN TERMINAL LOURD  
CONNECTÉ AU RÉSEAU CYCLADES PAR TELCOM

## SOMMAIRE

----

1. INTRODUCTION	PAGE 2
2. COMMANDES OPERATEUR DISPONIBLES	PAGE 3
2.1. NOTATIONS UTILISEES	PAGE 3
2.2. LISTE DES SERVEURS DECLARES	PAGE 5
2.2.1. SERVEURS DE TRAITEMENT PAR LOTS	PAGE 5
2.2.2. SERVEURS CONVERSATIONNELS	PAGE 5
2.3. LISTE DES COMMANDES	PAGE 6
2.3.1. CONNECT	PAGE 6
2.3.2. ATTENTION	PAGE 7
2.3.3. TRANSMIT	PAGE 8
2.3.4. DISCONNECT	PAGE 9
2.3.5. START	PAGE 10
2.3.6. REQUEST	PAGE 11
2.3.7. QUERY	PAGE 12
2.3.8. PRINTER	PAGE 13
3. CODES DES MESSAGES D'ERREUR EMIS PAR TELCOM	PAGE 14
4. PARTICULARITES DES SERVEURS DISPONIBLES	PAGE 17
4.1. LE SERVEUR SIRIS8 SUR IRIS80	PAGE 17
4.2. LE SERVEUR ASP/OS-MVT SUR 360	PAGE 18

## 1. INTRODUCTION

----

UN TERMINAL LOURD CONNECTE AU RESEAU CYCLADES PAR L'INTERMEDIAIRE DU SYSTEME TELCOM PEUT ACTIVER SIMULTANEMENT PLUSIEURS CONNEXIONS A DES SERVEURS DE TRAIITEMENT PAR LOTS DU RESEAU CYCLADES , AINSI QU'A DES SERVEURS CONVERSATIONNELS.

QUAND L'OPERATEUR DU TERMINAL LOURD VEUT EMETTRE DES ATTENTIONS OU DES COMMANDES OPERATEUR SUR L'APPAREIL VIRTUEL DE TYPE CONSOLE OPERATEUR OU TERMINAL D'UN SERVEUR AUQUEL IL EST CONNECTE , IL DOIT UTILISER DES COMMANDES DU SYSTEME TELCOM ET SPECIFIER LE SERVEUR AUQUEL IL S'ADRESSE (VOIR PLUS LOIN LES COMMANDES 'ATTENTION' ET 'TRANSMIT').

UN MODE DE CONNEXION DIRECTE A UN SERVEUR ET UN SEUL PERMET D'EVITER L'ADRESSAGE DU SERVEUR AUQUEL LES COMMANDES ET LES ATTENTIONS SONT DESTINEES (VOIR PLUS LOIN LA COMMANDE 'REQUEST').

LES MESSAGES EMIS SUR LES CONSOLES OPERATEUR ET LES TERMINAUX DES SERVEURS CONNECTES APPARAISSENT SUR LA CONSOLE OPERATEUR DU TERMINAL LOURD PREFIXES PAR LE NOM DU SERVEUR QUI LES A EMIS.

LES MESSAGES EMIS PAR LE SYSTEME TELCOM APPARAISSENT PREFIXES PAR LA LETTRE 'T'.

CEUX EMIS PAR LE SYSTEME LOCAL ASP APPARAISSENT PREFIXES PAR LA LETTRE 'A'.

SI LE LECTEUR DE CARTES DU TERMINAL LOURD EST CONNECTE A PLUSIEURS SERVEURS , CELUI VERS LEQUEL SERONT EMISES LES CARTES LUES EST LE DERNIER POUR LEQUEL UNE COMMANDE START A ETE EMISE , OU , SI AUCUNE COMMANDE START N'A ETE EMISE , LE PREMIER CONNECTE.

LES LISTES EMISES PAR LES SERVEURS CONNECTES SONT RANGEES DANS UNE FILE D'ATTENTE ET SONT ENVOYEEES SUR L'IMPRIMANTE DU TERMINAL LOURD DANS LEUR ORDRE D'ARRIVEE.

## 2. COMMANDES OPERATEUR DISPONIBLES.

-----

## 2.1. NOTATIONS UTILISEES :

- \* CHAQUE CODE COMMANDE PEUT ETRE REDUIT A UNE SEULE LETTRE OU UTILISE SANS ABBREVIATION.  
PAR EXEMPLE : 'CONNECT A' ET 'C A' ONT LE MEME EFFET.
- \* LE CODE COMMANDE EST SEPRE DES OPERANDES PAR UN OU PLUSIFURS BLANCS, AINSI QUE LES DIFFERENTS OPERANDES ENTRE EUX.
- \* 'APP' : DESIGNE UN APPAREIL DU TERMINAL LOURD.  
LES VALEURS POSSIBLES SONT :  
  'LECR' : POUR LE LECTEUR DE CARTES 1.  
  'IMPR' : POUR L'IMPRIMANTE 1.  
  'CONS' : POUR LA CONSOLE OPERATEUR.
- \* 'ADRESSE' : DESIGNE L'IDENTIFICATEUR DE L'ARONNE CONTACT DU SERVEUR CONCERNE EN HEXADECIMAL.  
PAR EXEMPLE : 'C1C30481' POUR LE 360/67 DU C.I.C.G.  
              '00010482' POUR SIRIS8 SUR IRIS80 AU C.I.C.G.
- \* 'I-SERVICE' : DESIGNE L'IDENTIFICATEUR DE SERVICE DU SERVEUR CONCERNE.  
PAR EXEMPLE : 'BATCH360' POUR ASP/MVT SUR 360 DU C.I.C.G.  
              'SSRT' POUR SIRIS8 SUR IRIS80.
- \* 'I-ENTREE' : DESIGNE L'ENTREE DU SERVICE 'I-SERVICE'.  
PAR EXEMPLE 'LECR' POUR LE LECTEUR DE CARTES DU SERVICE 'BATCH360'.  
              'CN00' POUR LA CONSOLE OPERATEUR DU SERVICE 'SSRT'.
- \* 'U' : DESIGNE PAR UN CODE REDUIT A UNE SEULE LETTRE UN SERVEUR DE TRAITEMENT PAR LOT DU RESEAU CYCLADES, OU BIEN UN SERVEUR CONVERSATIONNEL POUR LEQUEL SEULE L'ENTREE DE TYPE CONSOLE EST UTILISABLE.  
VOIR PLUS LOIN LA LISTE DES SERVEURS DECLARES.

\* LES COMMANDES 'CONNECT' ET 'DISCONNECT' SONT DISPONIBLES SOUS TROIS MODES DIFFERENTS :

- UN PREMIER MODE QUI PERMET D'EXPLICITER L'APPAREIL A CONNECTER ET LES IDENTIFICATIONS COMPLETES DU SERVEUR CONCERNE.
- UN DEUXIEME MODE QUI PERMET D'EXPLICITER L'APPAREIL A CONNECTER ET D'UTILISER UN CODE REDUIT A UNE SEULE LETTRE POUR SPECIFIER LE SERVEUR CONCERNE.
- UN TROISIEME MODE QUI PERMET DE CONNECTER TOUS LES APPAREILS DU TERMINAL LOURD (CONSOLE, LECTEUR DE CARTES, IMPRIMANTE) AU SERVEUR SPECIFIE PAR UN CODE REDUIT A UNE SEULE LETTRE.

\* CERTAINES COMMANDES ( ATTENTION, TRANSMIT, START, REQUEST ) SONT DISPONIBLES SOUS DEUX MODES DISTINCTS :

- UN PREMIER MODE QUI PERMET D'ADRESSER UN SERVEUR PAR SON IDENTIFICATEUR DE SERVICE.

SI IL EXISTE PLUSIEURS CONNEXIONS A L'ENTREE CONCERNEE DE SERVEURS IDENTIFIES PAR LE MEME I-SERVICE, IL Y A UNE AMBIGUITE ET LE PREMIER TROUVE EST CONSIDERE COMME LE BON. IL VAUT MIEUX DANS CE CAS UTILISER LE DEUXIEME MODE DE LA COMMANDE.

- UN DEUXIEME MODE QUI PERMET D'ADRESSER UN SERVEUR PAR SON CODE 'U'. CETTE FORME EST PLUS AGREABLE A UTILISER QUE LA PREMIERE ( MOINS DE CARACTERES A FRAPPER ) ET SUPPRIME AUSSI TOUTE AMBIGUITE.

LE NOMBRE DE CARACTERES DU DEUXIEME OPERANDE PERMET DE DIFFERENCIER LES DEUX MODES. DANS LE CAS OU IL EST REDUIT A UN SEUL CARACTERE, ON CONSIDERE D'ABORD QU'IL S'AGIT DU CODE 'U' ET, SI CETTE VALEUR DE CODE N'EST PAS DEFINIE ON CONSIDERE QU'IL S'AGIT DE LA PREMIERE LETTRE D'UN IDENTIFICATEUR DE SERVICE. DANS LE CAS OU IL COMPREND PLUSIEURS CARACTERES ON EN DEDUIT QU'IL S'AGIT DU PREMIER MODE ET ON PREND EN CONSIDERATION LE PREMIER SERVEUR CONNECTE DONT L'IDENTIFICATEUR DE SERVICE COMMENCE PAR LES CARACTERES SPECIFIES.

DANS LE CAS OU L'ENTREE CONCERNEE DU SERVEUR SPECIFIE N'EST PAS CONNECTEE QUAND L'OPERATEUR EMET UNE DE CES COMMANDES, IL EST INFORME DE L'IMPOSSIBILITE D'EFFECTUER L'ACTION REQUISE PAR LE MESSAGE :

T CMD INV

## 2.2. LISTE DES SERVEURS DECLARES :

## 2.2.1. SERVEURS DE TRAITEMENT PAR LOT :

I	I	I	I	I	I	I	I	I	I
I SERVEUR DE	I I	I I	I I	I I	I I	I I	I I	I I	I I
I TRAITEMENT	IU	IADRESSE	II-SERVICE	I	I	I	I	I	I
I PAR LOTS	I I	I I	I I	I I	I I	I I	I I	I I	I I
I	I I	I I	I I	I I	I I	I I	I I	I I	I I
I	I	I	I	I	I	I	I	I	I
I 360-C.I.C.G.	IA	IC1C30481	IRATCH360	I	CONS	I	LECR	I	IMPR
I	I	I	I	I	I	I	I	I	I
I IRIS80-C.I.C.G.	IR	I00010482	ISSBT	I	CN00	I	SCR1	I	ISLP1
I	I	I	I	I	I	I	I	I	I
I IRIS80-IRIA	IC	I0001028C	ISSBT	I	CN00	I	SCR1	I	ISLP1
I	I	I	I	I	I	I	I	I	I
I IRIS80-CCILS	ID	I00010487	ISSBT	I	CN00	I	SCR1	I	ISLP1
I	I	I	I	I	I	I	I	I	I

## 2.2.2. SERVEURS CONVERSATIONNELS :

I	I	I	I	I	I
I SERVEUR	I I	I I	I I	I I	I I
I CONVERSATIONNEL	IU	IADRESSE	II-SERVICE	I	I
I	I I	I I	I I	I I	I I
I	I I	I I	I I	I I	I I
I	I	I	I	I	I
I IRIS80-C.I.C.G.	IX	I00010482	ISSSTS	I	ARXX
I	I	I	I	I	I
I IRIS80-IRIA	IY	I0001028C	ISSSTS	I	ARYY
I	I	I	I	I	I
I IRIS80-CCILS	I7	I00010487	ISSSTS	I	ARZZ
I	I	I	I	I	I



## 2.3. LISTE DES COMMANDES :

### 2.3.1. CONNECT

TOUTE DEMANDE DE CONNEXION D'UN APPAREIL DU TERMINAL LOURD A UNE ENTREE D'UN SERVEUR CYCLADES SE TERMINE DE L'UNE DES FACONS SUIVANTES :

- APRES EMISSION D'UNE DEMANDE DE CONTACT PAR TELCOM (CF PROTOCOLE CLIENT-SERVEUR DE TERMINAUX TER 502.2) , AUCUNE REPONSE N'EST PARVENUE A L'ABONNE QUI A EMIS LA DEMANDE APRES 6 SECONDES D'ATTENTE. CE CAS CORRESPOND EN GENERAL AU FAIT QUE LE SERVEUR N'EST PAS ACTIF.

L'OPERATEUR DU TERMINAL LOURD EN EST INFORME PAR LE MESSAGE :  
T E01

- LA PRISE DE CONTACT AVEC LE SERVEUR NE PEUT PAS S'EFFECTUER POUR UNE RAISON QUI EST FOURNIE A L'OPERATEUR PAR L'UN DES MESSAGES D'ERREUR : E02, E03, E04, E05, E06, E07, E08 OU E17. (VOIR PLUS LOIN, PARAGRAPHE 3).
- LE CONTACT AVEC LE SERVEUR A PU S'ETABLIR , C'EST A DIRE QU'UNE VOIE VIRTUELLE A ETE ETABLIE ENTRE TELCOM ET LE SERVEUR ET UN APPAREIL VIRTUEL ASSOCIE A LA GESTION DE CETTE VOIE EST PRIS EN CHARGE PAR TELCOM.

DANS LE CAS DE LA CONNEXION DE L'IMPRIMANTE DU TERMINAL LOURD AU SERVEUR DE TRAITEMENT PAR LOTS DU SYSTEME ASP/OS-MVT DU 360 SUPPORTANT TELCOM , IL N'Y A PAS D'ETABLISSEMENT DE VOIE VIRTUELLE NI CREATION D'UN APPAREIL VIRTUEL . IL SUFFIT EN EFFET DE REVEILLER LE PROCESSUS CHARGE D'EMETTRE SUR L'IMPRIMANTE DU TERMINAL LOURD LES LISTES ENREGISTREES DANS LA FILE D'ATTENTE DE TELCOM.

DANS LE CAS DE LA CONNEXION DU LECTEUR DE CARTES AU SERVEUR LOCAL ON EVITE EGALEMENT D'UTILISER LA STATION DE TRANSPORT CYCLADES EN ECRIVANT DIRECTEMENT LES CARTES LUES DANS LA FILE D'ATTENTE DU SYSTEME TELCOM.

DE MEME POUR LA CONNEXION DE LA CONSOLE OPERATEUR AU SERVEUR LOCAL, LE PROCESSUS CHARGE DE LA CONNEXION NE GERE PAS UNE VOIE VIRTUELLE MAIS IL GERE DIRECTEMENT UNE PSEUDO CONSOLE OPERATEUR ASP ASSOCIEE A CETTE CONNEXION.

DANS LES TROIS CAS DE CONNEXION AU SERVEUR LOCAL L'OPERATEUR DU TERMINAL LOURD UTILISE CE SERVICE DE LA MEME FACON QUE TOUT AUTRE SERVICE CYCLADES BIEN QUE LES MECANISMES DE LA STATION DE TRANSPORT NE SOIENT PAS MIS EN OEUVRE.

DANS TOUS LES CAS OU LE CONTACT A PU S'ETABLIR CORRECTEMENT, L'OPERATEUR EN EST INFORME PAR LE MESSAGE :  
T CONTACT ETABLI AVEC ENTREE 'I-ENTREE' DU SERVICE 'I-SERVICE'

- \* C(ONNECT) APP ADRESSE T-SERVICE I-ENTREE  
PERMET DE CONNECTER L'APPAREIL 'APP' A L'ENTREE 'I-ENTREE' DU SERVICE 'I-SERVICE' EN S'ADRESSANT A L'ABONNE CONTACT IDENTIFIE PAR 'ADRESSE'.  
EXEMPLE : C CONS CIC30481 BATCH360 CONS
- \* C(ONNECT) APP U  
PERMET DE CONNECTER L'APPAREIL 'APP' A L'ENTREE CORRESPONDANTE A CE TYPE D'APPAREIL DU SERVEUR IDENTIFIE PAR LE CODE 'U'.  
EXEMPLE : C CONS A
- \* C(ONNECT) U  
PERMET DE CONNECTER LES 3 APPAREILS (CONSOLE, LECTEUR, IMPRIMANTE) AU SERVEUR IDENTIFIE PAR LE CODE 'U'.  
EXEMPLE : C A

### 2.3.2. ATTENTION

L'EMISSION D'UNE ATTENTION SUR LA CONSOLE OPERATEUR DU SERVEUR LOCAL (BATCH360) EST INUTILE CAR DANS CE CAS LE PROCESSUS CHARGE DU SUPPORT DE LA PSEUDO CONSOLE ASP EMET LUI MEME LES ATTENTIONS SUR LA PSEUDO CONSOLE OPERATEUR QUAND C'EST NECESSAIRE, C'EST A DIRE AVANT L'EMISSION D'UNE COMMANDE OPERATEUR. UNE COMMANDE ATTENTION ADRESSEE AU SERVEUR LOCAL EST DONC REFUSEE PAR LE MESSAGE D'ERREUR E18.

LE SYSTEME TELCOM N'INFORME PAS L'OPERATEUR DU TERMINAL LOURD DE LA FIN NORMALE DU TRAITEMENT DE LA COMMANDE ATTENTION, C'EST AU ROLE DU SERVEUR D'INFORMER SON CLIENT SI NECESSAIRE.

- \* A(TTENTION) I-SERVICE N  
 PERMET D'EMETTRE UNE ATTENTION SUR L'APPAREIL VIRTUEL DE TYPE  
 CONSOLE OPERATEUR DU SERVEUR IDENTIFIE PAR 'I-SERVICE'.  
 'N' : INDIQUE UN NUMERO D'ATTENTION COMPRIS ENTRE 1 ET 9 INCLUS.  
 IL PEUT ETRE OMIS. LA VALEUR PAR DEFAUT EST 1.  
 EXEMPLE : A BATCH360
- \* A(TTENTION) U N  
 PERMET D'EMETTRE UNE ATTENTION SUR L'APPAREIL VIRTUEL DE TYPE  
 CONSOLE OPERATEUR DU SERVEUR IDENTIFIE PAR LE CODE 'U'.  
 'N' : INDIQUE UN NUMERO D'ATTENTION COMPRIS ENTRE 1 ET 9 ICLUS.  
 IL PEUT ETRE OMIS. LA VALEUR PAR DEFAUT EST 1.  
 EXEMPLE : A A

### 2.3.3. TRANSMIT

DE MEME QUE POUR LA COMMANDE 'ATTENTION', LE SYSTEME TELCOM  
 N'INFORME PAS L'OPERATEUR DE LA FIN NORMALE DU TRAITEMENT DE LA  
 COMMANDE.

- \* T(RANSMIT) I-SERVICE CMD  
 PERMET D'EMETTRE LA COMMANDE 'CMD' SUR L'APPAREIL VIRTUEL DE TYPE  
 CONSOLE OPERATEUR DU SERVEUR IDENTIFIE PAR I-SERVICE.  
 EXEMPLE : T BATCH360 \*I A
- \* T(RANSMIT) U CMD  
 PERMET D'EMETTRE LA COMMANDE 'CMD' SUR L'APPAREIL VIRTUEL DE TYPE  
 CONSOLE OPERATEUR DU SERVEUR IDENTIFIE PAR LE CODE 'U'.  
 EXEMPLE : T A \*T A

## 2.3.4. DISCONNECT

POUR DECONNECTER L'ACCES A UNE ENTREE D'UN SERVEUR , ON SUPPRIME LA VOIE VIRTUELLE CORRESPONDANTE DANS LE CAS D'UNE CONNEXION A UN SERVEUR ACCESSIBLE PAR LE RESEAU CYCLADES SEULEMENT . CE SONT LES PROCESSUS CHARGES DE L'EXPLOITATION DE CETTE VOIE VIRTUELLE QUI , A LEUR PROCHAINE TENTATIVE D'UTILISATION S'APERCEVront QU'ELLE N'EXISTE PLUS ET EFFECTUERONT LES ACTIONS COMPLEMENTAIRES POUR UNE DECONNEXION CORRECTE , EN PARTICULIER EMISSION D'UN MESSAGE SUR LA CONSOLE OPERATEUR DU TERMINAL LOURD :

T FIN CONTACT AVEC ENTREE 'I-ENTREE' DU SERVICE 'I-SERVICE'

LE TRAITEMENT DE LA COMMANDE DISCONNECT EST DONC ASSEZ RAPIDE , MAIS L'ACTION REQUISE NE FINI DE S'EXECUTER QU'APRES LA FIN DU TRAITEMENT DE LA COMMANDE .

- \* D (DISCONNECT) APP ADRESSE  
PERMET DE DECONNECTER L'APPAREIL 'APP' DE L'ENTREE CORRESPONDANTE A CE TYPE D'APPAREIL DU SERVEUR IDENTIFIE PAR L'ADRESSE DE SON ARONNE CONTACT.

EXEMPLE : D CONS C1C30481

- \* D (DISCONNECT) APP U  
PERMET DE DECONNECTER L'APPAREIL 'APP' DE L'ENTREE CORRESPONDANTE A CE TYPE D'APPAREIL DU SERVEUR IDENTIFIE PAR LE CODE 'U'.

EXEMPLF : D CONS A

- \* D (DISCONNECT) U  
PERMET DE DECONNECTER LES 3 APPAREILS (CONSOLE , LECTEUR , IMPRIMANTE) DU SERVEUR IDENTIFIE PAR LE CODE 'U'.

EXEMPLF : D A

## 2.3.5. START

PLUSIEURS ENTREES DE TYPE LECTEUR DE CARTES D'UN SERVEUR DE TRAITEMENTS PAR LOTS PEUVENT ETRE ACTIVES A UN MOMENT DONNE . POUR CHOISIR CELLE QUI SERA MISE EN CORRESPONDANCE AVEC LE LECTEUR DE CARTES DU TERMINAL LOURD , L'OPERATEUR DOIT UTILISER LA COMMANDE START.

SI AUCUNE ENTREE DE TYPE LECTEUR DE CARTES N'EST DECONNECTEE IL N'EST NECESSAIRE DE FRAPPER UNE NOUVELLE COMMANDE START QUE LORSQUE L'OPERATEUR DECIDE D'EMETTRE SES FLOTS DE CARTES VERS UN AUTRE SERVEUR CONNECTE.

POUR SAVOIR A UN MOMENT DONNE AVEC QUEL SERVEUR LE LECTEUR DE CARTES DU TERMINAL LOURD EST MIS EN CORRESPONDANCE , IL EST POSSIBLE D'UTILISER LA COMMANDE 'QUERY' ( VOIR PLUS LOIN ) . LA PREMIERE CONNEXION LISTEE CONCERNANT LE LECTEUR DE CARTES EST CELLE UTILISEE POUR EMETTRE LES FLOTS DE CARTES LUS.

SI LE LECTEUR DU TERMINAL LOURD LIT DES CARTES ALORS QU' AUCUN SERVEUR N'EST CONNECTE AU LECTEUR DE CARTES , LES CARTES LUES SONT IGNOREES.

- \* S(TART) I-SERVICE  
PERMET DE SPECIFIER LE SERVEUR VERS LEQUEL SERA FMIS LE PROCHAIN FLOT DE CARTES LUES.  
UNE CONNEXION DU LECTEUR DE CARTES POUR CE SERVEUR DOIT ETRE EN COURS.

EXEMPLE : S SSRT

- \* S(TART) U  
PERMET DE SPECIFIER LE SERVEUR VERS LEQUEL SERA FMIS LE PROCHAIN FLOT DE CARTES LUES.  
UNE CONNEXION DU LECTEUR DE CARTES POUR CE SERVEUR DOIT ETRE EN COURS.

EXEMPLE : S B

## 2.3.6. REQUEST

LE FAIT DE NOMMER LE SERVEUR AUQUEL L'OPERATEUR DU TERMINAL LOURD S'ADRESSE CHAQUE FOIS QU'IL VEUT EMETTRE UNE ATTENTION OU UNE COMMANDE VERS LA CONSOLE OPERATEUR D'UN SERVEUR CONNEXE PEUT PARAITRE ASSEZ DESAGREABLE. IL SEMBLE ENCORE PLUS NECESSAIRE DE SUPPRIMER CET ADRESSAGE PAR L'UTILISATEUR AU COURS D'UNE CONNEXION AVEC UN SERVEUR CONVERSATIONNEL.

POUR OBTENIR CETTE FACILITE D'UTILISATION, ON PERMET A UNE CONNEXION A LA CONSOLE OPERATEUR DU TERMINAL LOURD ET A UNE SEULE A LA FOIS DE PASSER EN MODE 'CONNECTE TRANSPARENT', A L'AIDE DE LA COMMANDE REQUEST.

DES QU'UNE CONNEXION EST PASSEE DANS CE MODE, TOUT TEXTE EMIS SUR LA CONSOLE OPERATEUR DU TERMINAL LOURD EST DIRIGE, SI POSSIBLE, VERS LE SERVEUR EN MODE 'CONNECTE TRANSPARENT' (SI CE SERVEUR A LA MAIN ET S'IL EST ENCORE ACTIF DANS LE CAS D'UNE CONNEXION UTILISANT LE MECANISME DE VOIE VIRTUELLE).

IL EXISTE DANS CE MODE DEUX CARACTERES RESERVES APPELES C1 ET C2 QUI PERMETTENT RESPECTIVEMENT DE QUITTER LE MODE 'CONNECTE TRANSPARENT' ET D'EMETTRE UNE ATTENTION 1 VERS LE SERVEUR EN MODE 'CONNECTE TRANSPARENT' SI CE SERVEUR EST ACCEDÉ PAR CYCLADES. CES CARACTERES RESERVES DOIVENT ETRE EMIS SEULS POUR REALISER LA FONCTION QUI LEUR EST ASSOCIEE; S'ILS SONT SUIVIS D'AUTRES CARACTERES (MEME DES BLANCS) ON CONSIDERE QU'ILS CONSTITUENT DU TEXTE A EMETTRE VERS LE SERVEUR EN MODE 'CONNECTE TRANSPARENT'.

LA VALEUR DES CARACTERES C1 ET C2 EST CHOISIE A CHAQUE PASSAGE DANS LE MODE 'CONNECTE TRANSPARENT' A L'AIDE DE LA COMMANDE REQUEST. SI CES PARAMETRES SONT OMIS, LA VALEUR PAR DEFAUT EST \* POUR C1 ET # POUR C2.

LE CARACTERE C1 NE PEUT PAS ETRE OMIS SEUL.

\* R(EQUEST) I-SERVICE C1 C2  
PERMET DE FAIRE PASSER LA CONNEXION DE L'APPAREIL DE TYPE CONSOLE OPERATEUR OU TERMINAL AU SERVEUR IDENTIFIE PAR I-SERVICE EN MODE 'CONNECTE TRANSPARENT'.

UNE CONNEXION DE LA CONSOLE POUR CE SERVEUR DOIT ETRE EN COURS.

EXEMPLE : R SSTS

\* P(EQUEST) U C1 C2  
 PERMET DE FAIRE PASSER LA CONNEXION DE L'APPAREIL DE TYPE CONSOLE  
 OPERATEUR OU TERMINAL AU SERVEUR IDENTIFIE PAR LE CODE 'U' EN MODE  
 'CONNECTE TRANSPARENT'.  
 UNE CONNEXION DE LA CONSOLE POUR CE SERVEUR DOIT ETRE EN COURS.  
 EXEMPLE : P X \* J

### 2.3.7. QUERY

\* Q(QUERY)  
 PERMET D'OBTENIR LA LISTE DES SERVEURS CONNECTES.  
 UN MESSAGE DE LA FORME SUIVANTE EST EMIS PAR TELCOM POUR CHAQUE  
 CONNEXION A UNE ENTREE D'UN SERVEUR :

T PERIPHERIQUE U I-AR-CNT I-SERVICE I-ENTREE ETAT SENS

- PERIPHERIQUE PEUT PRENDRE LES VALEURS : CONSOLE  
 READER  
 PRINTER

SELON LE TYPE DE L'UNITE DU TERMINAL LOURD CONNECTEE.

- U EST LE CODE 'U' DU SERVEUR S'IL APPARTIENT A LA TABLE DES  
 SERVEURS DECLARES, SINON C'EST UN CARACTERE BLANC.
- I-AR-CNT EST L'ADRESSE DE L'ABONNE CONTACT DU SERVEUR SUR HUIT  
 CARACTERES EN HEXADECIMAL.
- I-SERVICE EST L'IDENTIFICATEUR DE SERVICE SUR HUIT CARACTERES  
 DU SERVEUR CONNECTE.
- I-ENTREE EST L'IDENTIFICATEUR DE L'ENTREE SUR QUATRE CARACTERES
- ETAT FOURNI L'ETAT DE L'APPAREIL VIRTUEL CLIENT SUR UN CARACTERE  
 LES VALEURS POSSIBLES SONT LES SUIVANTES :

N POUR NOT-READY  
 I POUR IDLE  
 O POUR OUT  
 R POUR READY

- SENS INDIQUE POUR UNE CONNEXION A L'ALTERNAT LEQUEL DES DEUX  
 APPAREILS A LA MAIN (CLIENT OU SERVEUR).

LES VALEURS POSSIBLES SONT LES SUIVANTES :

-> POUR PRET A EMETTRE (LE CLIENT A LA MAIN)  
 <- POUR PRET A RECEVOIR (LE SERVEUR A LA MAIN)

DANS LE CAS D'UNE CONNEXION DIRECTE A UNE PSEUDO CONSOLE  
 OPERATEUR ASP, LE CLIENT A TOUJOURS LA MAIN (->).

LE DERNIER MESSAGE EMIS PAR TELCOM EN REPONSE A LA COMMANDE QUERY  
 EST SUIVI DE :

T CMD ACK

## 2.3.8. PRINTER

LA COMMANDE 'PRINTER' PERMET D'EFFECTUER UNE ACTION SPECIFIEE PAR SON PARAMETRE CONCERNANT LA GESTION DE L'IMPRIMANTE DU TERMINAL LOURD.

LE PARAMETRE DE LA COMMANDE 'PRINTER' EST OBLIGATOIRE ET EST IDENTIFIE PAR SES QUATRE PREMIERS CARACTERES.

- \* P(RINTER) REST(ART)  
PERMET D'ARRETER L'IMPRESSION DE LA LISTE EN COURS . L'IMPRIMANTE NE SERA RELANCEE QUE PAR SA CONNEXION A UN SERVEUR OU A L'AIDE DE LA COMMANDE 'PRINTER START' . LA LISTE INTERROMPUE SERA ALORS REPRISE AU DEBUT.
- \* P(RINTER) STAR(T)  
PERMET DE DEMARRER L'IMPRESSION DES LISTES EN ATTENTE S'IL EN EXISTE . UNE ACTION ANALOGUE EST REALISEE POUR TOUTE CONNEXION DE L'IMPRIMANTE A UN SERVEUR DE TRAITEMENT PAR LOTS.
- \* P(RINTER) REPE(AT)  
PERMET DE RELANCER UNE FOIS L'IMPRESSION DE LA LISTE EN COURS APRES SA FIN NORMALE D'IMPRESSION . CETTE COMMANDE PEUT ETRE A NOUVEAU UTILISEE PENDANT L'IMPRESSION D'UNE LISTE DEJA REPETEE POUR OBTENIR PLUS DE DEUX EXEMPLAIRES.
- \* P(RINTER) CANC(EL)  
PERMET D'ARRETER LA LISTE EN COURS D'IMPRESSION ET DE PASSER AUSSITOT AU FICHIER SUIVANT.



## 3. CODES DES MESSAGES D'ERREUR EMIS PAR TELCOM

-----

- E01 : T.O. DE 6 SECONDES ECOULE SANS REPONSE APRES EMISSION DU D-CONTACT.
- E02 : ACTIVATION DE L'ABONNE CLIENT ASSOCIE A CE TERMINAL LOURD IMPOSSIBLE.
- E03 : EXPEDITION DE LA LETTRE 'D-CONTACT' IMPOSSIBLE PAR L'ABONNE CLIENT.
- E04 : RETRAIT DE LETTRE REGULIERE PAR L'ABONNE CLIENT IMPOSSIBLE.
- E05 RXX: LE SERVEUR A REPONDU 'O-RUPTURE' A UN 'D-CONTACT'.  
 XX : INDIQUE LA RAISON DU REFUS FOURNIE PAR LE SERVEUR.  
 01 : SATURATION DU SERVEUR.  
 02 : COUPLE (AB-CLIENT,IVV) DEJA UTILISE.  
 03 : TYPE DE PROTOCOLE NON COMPATIBLE.  
 04 : 'I-ENTREE/I-SERVICE' INVALIDES.
- E06 : LE 'IVV' FOURNI PAR LE SERVEUR, DANS LE 'R-CONTACT' NE CORRESPOND PAS A CELUI DEMANDE PAR LE CLIENT DANS LE 'D-CONTACT'.
- E07 RXX : REFUS DE LA 'ST' A UNE DEMANDE D'ETABLISSEMENT DE 'VV'.  
 XX : INDIQUE LA RAISON DU REFUS FOURNI PAR LA 'ST'.  
 01 : 'AB' FAUX.  
 02 : SATURATION.  
 03 : CARACTERISTIQUES ERRONNEES.  
 04 : ERREUR CODE DIRECTIVE .  
 09 : UNE MEME 'VV' EXISTE DEJA.

- E08 : L'ETABLISSEMENT D'UNE 'VV' S'EST TERMINE ANORMALEMENT.  
L'ACCORD ENTRE LE CLIENT ET LE SERVEUR EST IMPOSSIBLE.
- E09 RXX : UN RETRAIT SUR 'VV' S'EST TERMINE ANORMALEMENT POUR UNE  
ENTREE DE TYPE CONSOLE.  
XX : INDIQUE LE CODE D'ERREUR FOURNI PAR LA 'ST'.  
01 : 'AB' FAUX.  
04 : ERREUR CODE DIRECTIVE.  
05 : ETAT 'VV' INCORRECT.  
09 : 'VV' INEXISTANTE.
- E10 XXXXXXXX Y : UN MESSAGE EN PROVENANCE D'UNE ENTREE DE TYPE CONSOLE  
DU SERVEUR 'XXXXXXX' NE RESPECTE PAS LE PROTOCOLE APPAREIL  
VIRTUEL (TER 503.1). IL EST IGNORE.  
Y : EST UN CODE D'ERREUR.  
1 : CHAMP-ADRESSE INVALIDE DANS 'O-TEXT/ADRESS/TEXT'  
2 : CHAMP 'MSG-LG' INVALIDE.  
3 : MESSAGE 'PAV' TRONQUE.  
4 : MESSAGE DE CONTROLE INVALIDE.
- E11 : IL N'EST PAS POSSIBLE D'EMETTRE DES COMMANDES SUR LA CONSOLE  
DU SERVEUR SPECIFIE. ON N'A PAS LA MAIN.
- E12 RXX : UN RETRAIT SUR 'VV' S'EST TERMINE ANORMALEMENT POUR UNE ENTREE  
DE TYPE LECTEUR DE CARTES.  
XX : INDIQUE LE CODE D'ERREUR FOURNI PAR LA 'ST' DE MEME QUE  
POUR LE MESSAGE E09.
- E13 XXXXXXXX Y : UN MESSAGE EN PROVENANCE D'UNE ENTREE DE TYPE LECTEUR  
DE CARTES DU SERVEUR 'XXXXXXX' NE RESPECTE PAS LE PROTOCOLE  
APPAREIL VIRTUEL (TER 503.1). IL EST IGNORE.  
Y : EST UN CODE D'ERREUR (VOIR E10).
- E14 Y : UN MESSAGE EN PROVENANCE DE LA FILE D'ATTENTE NE RESPECTE PAS  
PROTOCOLE APPAREIL VIRTUEL (TER 503.1) . IL EST IGNORE AINSI  
QUE LES MESSAGES QUI LE SUIVENT DANS LE BUFFER QUI LE CONTIENT  
Y : EST UN CODE D'ERREUR (VOIR E10).

- F15 RXX : UN RETRAIT SUR 'VV' S'EST TERMINE ANORMALEMENT POUR UNE ENTREE DE TYPE IMPRIMANTE.  
XX : INDIQUE LE CODE D'ERREUR FOURNI PAR LA 'ST' DE MEME QUE POUR LE MESSAGE E09.
- F16 XXXXXXXX Y : UN MESSAGE EN PROVENANCE D'UNE ENTREE DE TYPE IMPRIMANTE DU SERVEUR 'XXXXXXXX' NE RESPECTE PAS LE PROTOCOLE APPAREIL VIRTUEL (TER 503.1). IL EST IGNORE.  
Y : EST UN CODE D'ERREUR (VOIR E10).
- F17 : UNE DEMANDE DE CONNEXION DE LA CONSOLE D'UN TERMINAL LOURD AU SERVEUR ASP LOCAL N'A PAS PU SE REALISER CAR IL N'Y A PAS DE PSEUDO CONSOLE OPERATEUR ASP DISPONIBLE POUR CE TERMINAL LOURD
- F18 : UNE ATTENTION A ETE EMISE AU SERVEUR ASP QUI SE TROUVE SUR LE MEME CALCULATEUR QUE LE CONCENTRATEUR DE TERMINAUX LOURDS. ELLE EST IGNOREE CAR ELLE EST INUTILE DANS CE CAS.

#### 4. PARTICULARITES DES SERVEURS DISPONIBLES.

-----

##### 4.1. LE SERVEUR SIRIS8 SUR IRIS80

- IL N'EST POSSIBLE DE CONNECTER LE LECTEUR OU L'IMPRIMANTE QUE SI LA CONSOLE OPERATEUR EST DEJA CONNECTEE.
- IL EXISTE DES POSTES FIXES DE SERVEURS QUI DOIVENT ETRE PREDECLARES A TOUT UTILISATEUR QUI N'A PAS DE POSTE FIXE DECLARE EST AFFECTE UN POSTE DE TRAVAIL QU'IL N'EST PAS SUR DE RETROUVER A LA SESSION SUIVANTE.
- LES VOIES VIRTUELLES ASSOCIEES AU LECTEUR DE CARTES ET A L'IMPRIMANTE ONT UNE DUREE DE VIE SANS ACTIVITE LIMITEE A UNE MINUTE.
- LORSQU'ON A CONNECTE LA CONSOLE AU SERVEUR SIRIS8,
  - \* POUR ENVOYER UNE COMMANDE A SIRIS8, IL FAUT EMETTRE UNE ATTENTION 1. ON RECOIT ALORS : ?? ET ON PASSE EN LECTURE.
  - \* LES ORDRES QUE L'ON PEUT ENVOYER SONT DECRITS DANS LE MANUEL OPERATEUR SIRIS8/C09 (REF 4500 E/FR). LES PLUS COURANTS SONT :
 

D RS	POUR CONNAITRE L'ETAT DES RESSOURCES
D AQ	POUR AVOIR LA LISTE DES JOBS ACTIFS SOUMIS PAR LA STATION
D WQ	POUR AVOIR LA LISTE DES JOBS EN ATTENTE SOUMIS PAR LA STATION
D ST.XXXX	POUR CONNAITRE L'ETAT DU JOB XXXX
D SY	POUR CONNAITRE LE NOMBRE DE PAGES A SORTIR
G TY01 MSG	POUR ENVOYER LE MESSAGE 'MSG' A L'OPERATEUR CENTRAL
I SY0X.D	POUR TUFER UNE LISTE (SY0X IDENTIFIE LE SYMBIONT)
J SY0X.L	POUR ARRETER UN SYMBIONT APRES LA FIN DE LA LISTE EN COURS
W	REORIENTATION DES LISTES (VOIR MANUEL)

## 4.2. LE SERVEUR ASP/OS-MVT SUR 360

- LES APPAREILS PEUVENT ETRE CONNECTES DANS UN ORDRE QUELCONQUE.
- ACTUELLEMENT IL EST NECESSAIRE DE SPECIFIER PAR DES CARTES CONTROLF LES FICHIERS A SORTIR SUR L'IMPRIMANTE DE LA STATION EMETTRICE. CES CARTES DOIVENT ETRE PLACEES APRES LA CARTE JOB.  
IL FAUT UNE CARTE /\*MAIN :  
/\*MAIN CRMAIN=M  
ET UNE CARTE /\*FORMAT POUR CHAQUE FICHIER A LISTER SUR L'IMPRIMANTE DU TERMINAL LOURD CONNECTE :  
/\*FORMAT CR,DDNAME=SYSMSG,USER=AR02  
DDNAME SPECIFIE LE NOM DU FICHIER A LISTER.  
USER IDENTIFIE LE TERMINAL LOURD SUR LEQUEL LA LISTE DOIT SORTIR.  
TOUT FICHIER DE TYPE IMPRIMANTE POUR LEQUEL UNE TELLE CARTE N'EST PAS UTILISEE SERA LISTE SUR UNE IMPRIMANTE LOCALE DU 360.
- LES VOIES VIRTUELLES NE SONT DESACTIVEES QUE PAR L'INITIATIVE DU CLIENT (COMMANDE DISCONNECT) OU SUR UNE ANOMALIE DE FONCTIONNEMENT.
- LORSQU'ON A CONNECTE LA CONSOLE AU SERVEUR DE TRAITEMENT PAR LOTS ASP/OS-MVT :
  - \* POUR ENVOYER UNE COMMANDE IL FAUT EMETTRE UNE ATTENTION 1 , ON RECOIT ALORS : CIG -  
ET ON PASSE EN LECTURE.
  - \* TOUTE COMMANDE DEBUTANT PAR LE CARACTERE \* EST CONSIDEREE COMME DESTINEE A ASP, SINON C'EST UN MESSAGE POUR LA CONSOLE DU SYSTEME TELCOM.
  - \* LES COMMANDES QUE L'ON PEUT ENVOYER A ASP SONT DECRITES DANS LE MANUEL OPERATEUR ASP (CF ASP CONSOLE OPERATOR'S MANUAL) . LES PLUS COURANTES SONT :
    - \*I J=XXXX FOURNIT L'ETAT DU JOB XXXX.  
XXXX EST SOIT LE NOM DU JOB , SOIT SON NUMERO AFFECTE PAR ASP.
    - \*I A FOURNIT LA LISTE DES JOBS ACTIFS.
    - \*I J=XXXX,H FAIT PASSER LE JOB XXXX DANS L'ETAT 'HOLD'.
    - \*I J=XXXX,R FAIT SORTIR LE JOB XXXX DE L'ETAT 'HOLD'.

- DANS LE CAS D'UN TERMINAL LOURD CONNECTE DIRECTEMENT AU 360 IL N'EST PAS NECESSAIRE D'EMETTRE UNE ATTENTION POUR PRENDRE LA MAIN. LA COMMANDE OPERATEUR ASP 'CMD' PEUT ALORS ETRE EMISE SOIT DANS LE MODE GENERAL A L'AIDE DE LA COMMANDE TELCOM 'TRANSMIT' :

T A CMD

SOIT DANS LE MODE 'CONNECTE TRANSPARENT' AU SERVEUR HATCH360 PAR EMISSION DIRECTE DE LA COMMANDE ASP :

CMD

---

CE MANUEL PEUT ETRE LISTE SUR L'IMPRIMANTE D'UN TERMINAL LOURD  
CONNECTE A TELCOM SUR DEMANDE A :

FOURNIER ROBERT . C.I.C.G. TELEPHONE : 54 61 45 POSTE 204

## BIBLIOGRAPHIE

- [A.1]           Système interactif dans un environnement réseau.  
Connexion d'une machine virtuelle "IBM 360-67" au  
réseau CYCLADES.  
J.P.ANSART. Thèse de Docteur-Ingénieur INPG. 6 Février 76.
- [A.2]           Services Cyclades sous CP/67.  
J.P.ANSART  
C.I.C.G. Note technique n°11. Mars 1975.
- [A.3]           Concentrateurs de terminaux conversationnels CP-67.  
(Version 2).  
Manuel de l'utilisateur.  
J.P.ANSART  
C.I.C.G. Note technique n°12. Mars 1975.
- [A.4]           Utilisation de la ST CP67.  
Réseau Cyclades. IMAG. J.P.ANSART. Mai 1974.
- [D.1]           Implémentation de la station de transport ST2 sous le  
système CP/67.  
NG.X.DANG. C.I.C.G. A20. Janvier 76. Réseau Cyclades.
- [F.1]           Serveurs et clients CYCLADES sous ØS/MVT.  
C.I.C.G. Note technique n°17. R.FOURNIER. Mai 1975.
- [F.2]           Un concentrateur de terminaux lourds sous TELCØM.  
C.I.C.G. Note technique n°32. R.FOURNIER. Décembre 1975.
- [F.3]           Gestion d'une file d'attente réseau sur une unité à accès  
direct.  
C.I.C.G. Note technique n°33. R.FOURNIER. Décembre 1975.
- [F.4]           Gestion de pseudo consoles opérateur ASP.  
C.I.C.G. Note technique n°34. R.FOURNIER. Janvier 1976.
- [F.5]           Gestion d'horloges au niveau des processus d'un sous-système  
de multiprogrammation d'ØS/MVT.  
C.I.C.G. Note technique n°35. R.FOURNIER. Janvier 1976.

- [F.6] Manuel opérateur d'un terminal lourd connecté au réseau CYCLADES par TELCØM. Version 2.  
CICG. 18 Décembre 1975. Robert FOURNIER.
- [I.1] Spécifications fonctionnelles des stations de transport du réseau Cyclades. Protocoles ST-ST.  
SCH 502-3. Mai 1973. IRIA.  
CHAMBON-ELIE-LE BIHAN-LE LANN-SIMMERMANN.
- [I.2] Transport protocol. Standard en-to-end protocol for heterogeneous networks.  
SCH 519-2. May 1975.  
H.ZIMMERMANN. M.ELIE.
- [I.3] Virtual terminal protocol (VTP) proposed specifications réseau Cyclades. TER 503-1. Novembre 1974.  
A.VIVIER. H.ZIMMERMANN.
- [I.4] Protocole client -serveur de terminaux.  
Réseau Cyclades. TER 502.2. Novembre 1974.  
H.ZIMMERMANN.
- [I.5] Procédures d'exploitation d'une liaison entre un noeud Cyclades et un participant avec la procédure IBM-BSC.  
Réseau Cyclades. TRA 512-3. Septembre 1973.  
F.DENJEAN.
- [I.6] Spécification du programme de gestion de la procédure IBM-BSC dans CIGALE.  
Réseau Cyclades. MIT 522. Mai 1973.  
F.DENJEAN.
- [IBM.1] Component Description : IBM 2701 Data Adapter Unit  
SRL File No.2701-09.  
Order No.GA22-6864-5.
- [IBM.2] IBM System/360 Model 67.  
Functional Characteristics.  
SRL. File No.S360-01.  
Form GA27-2719-2.
- [IBM.3] IBM System/360 Component Descriptions.  
2314 Direct Access Storage Facility and  
2844 Auxiliary Storage Control  
SRL File No. S360-07  
Order GA26-3599-6.



- [IBM.4] 2740/2741 Communication Terminal  
Original Equipement Manufacturer's Information.  
SRL S360/TP-19.  
A27-3002-0
- [IBM.5] IBM System/360 Operating System  
MVT Supervisor.  
Program Logic. File No.S360-36.  
GY28-6659-5
- [IBM.6] IBM System/360 Operating System :  
MVT Job Management,  
Program Logic Manual  
File No.S360-36  
Order No.GY28-6660-9
- [IBM-7] General Information. Binary Synchronous Communications.  
SRL. File No.TP-09.  
Order No. GA27-S004-1
- [M.1] One Year of Experience with CYCLADES.  
J.du MASLE. SEAS 75. DUBLIN.
- [M.2] Support for the CYCLADES network on the IBM 360/67 at the  
University of Grenoble.  
J.du MASLE, J.P.ANSART, R.FOURNIER. SEAS 75. DUBLIN.
- [M.3] Computer Network Activities at the University of Grenoble.  
J.du MASLE. International Symposium : 'Computer at the  
University'. Zagreb. Oct.1974.
- [P.1] Le système TELCØM.  
Thèse présentée en 1974 par Monsieur Z.PAPACHRISTODOULOU à  
l'USMG.
- [S.1] SYNCØP. Système normalisé de commutation de processus. Im-  
plémentation sous CP/67.  
Equipe réseaux ENSIMAG, N.X.DANG, R.FOURNIER, V.QUINT.  
Septembre 1975.

- [S.2] SYNCØP. Système normalisé de commutation de processus.  
Macro-instructions sous CP/67.  
Equipe réseaux ENSIMAG. Juin 1975.
- [S.3] SYNCØP. Système normalisé de commutation de processus.  
Manuel d'utilisation. Spécifications réalisation  
IRIS 80/10070.  
Jean SEGUIN. Equipe réseaux CII.  
Guy SERGEANT. Equipe réseaux ENSIMAG.  
GRENOBLE. Septembre 1975.
- [U.1] Multi-leaving. SOC-IT-ML-1  
Pierre ULLMANN  
EMP.FONTAINEBLEAU.  
2 novembre 1970.
- [W.1] An NCP for the ARPA Network.  
James E.WHITE.  
Computer Research Laboratory. University of California.  
Santa Barbara, California. 21 December 1970.
- [W.2] NCP Practices.  
James E.WHITE.  
Computer Research Laboratory. 26 May 1971.
- [W.3] Dynamic Extension of ØS 360 for a network environment.  
James E.WHITE.  
Computer Research Laboratory. University of California.  
Santa Barbara. June 72.

	Page
<b>ANNEXE 2</b>	141
Une trace de la console opérateur du système TELCOM en cours d'exploitation.	
<b>ANNEXE 3</b>	144
Manuel opérateur d'un terminal lourd connecté au réseau Cyclades par TELCOM.	
<b>BIBLIOGRAPHIE.</b>	164