



HAL
open science

Définition et mise en œuvre d'un service de traitements par lots dans un réseau hétérogène d'ordinateurs

Silvia Lumelsky

► **To cite this version:**

Silvia Lumelsky. Définition et mise en œuvre d'un service de traitements par lots dans un réseau hétérogène d'ordinateurs. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1979. Français. NNT : . tel-00290427

HAL Id: tel-00290427

<https://theses.hal.science/tel-00290427>

Submitted on 25 Jun 2008

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

Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR DE 3^{ème} CYCLE

"Génie Informatique"

par

Silvia LUMELSKY



**DEFINITION ET MISE EN ŒUVRE
D'UN SERVICE DE TRAITEMENTS PAR LOTS
DANS UN RESEAU HETEROGENE D'ORDINATEURS**



Thèse soutenue le 6 décembre 1979 devant la Commission d'Examen :

Président : S. KRAKOWIAK
Examineurs : C. KAISER
J. MOSSIERE
Invités : N. NAFFAH
V. QUINT

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET

M. Georges LESPINARD

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Electronique - automatique
BESSON Jean	Chimie minérale
BLOCH Daniel	Physique du solide - cristallographie
BONNETAIN Lucien	Génie chimique
BONNIER Etienne	Métallurgie
* BOUDOURIS Georges	Electronique - automatique
BRISSONNEAU Pierre	Physique du solide - cristallographie
BUYLE-BODIN Maurice	Electronique - automatique
COUMES André	Electronique - automatique
DURAND Francis	Métallurgie
FELICI Noël	Electronique - automatique
FOULARD Claude	Electronique - automatique
LANCIA Roland	Electronique - automatique
LONGEQUEUE Jean-Pierre	Physique nucléaire corpusculaire
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie - physique
PAUTHENET René	Electronique - automatique
PERRET René	Electronique - automatique
POLOUJADOFF Michel	Electronique - automatique
TRAYNARD Philippe	Chimie - physique
VEILLON Gérard	Informatique fondamentale et appliquée
* en congé pour études	

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuël	Electronique - automatique
BOUVARD Maurice	Génie mécanique
COHEN Joseph	Electronique - automatique
GUYOT Pierre	Métallurgie physique
LACOUME Jean-Louis	Electronique - automatique
JOUBERT Jean-Claude	Physique du solide - cristallographie

MM.	ROBERT André	Chimie appliquée et des matériaux
	ROBERT François	Analyse numérique
	ZADWORNY François	Electronique - automatique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	CHARTIER Germain	Electronique - automatique
	CHIAVERINA Jean	Biologie, biochimie, agronomie
	IVANES Marcel	Electronique - automatique
	LESIEUR Marcel	Mécanique
	MORET Roger	Physique nucléaire - corpusculaire
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie Physique

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

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
	BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
	DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

MM. KOBYLANSKI André	Ecole des Mines St. Etienne (Métallurgie)
LE COZE Jean	Ecole des Mines St. Etienne (Métallurgie)
LESBATS Pierre	Ecole des Mines St. Etienne (Métallurgie)
LEVY Jacques	Ecole des Mines St. Etienne (Métallurgie)
RIEU Jean	Ecole des Mines St. Etienne (Métallurgie)
SAINFORT	C.E.N. Grenoble (Métallurgie)
SOUQUET	U.S.M.G.
CAILLET Marcel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
COULON Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
GUILHOT Bernard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LALAUZE René	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LANCELOT Francis	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SARRAZIN Pierre	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SOUSTELLE Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THEVENOT François	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THOMAS Gérard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TOUZAIN Philippe	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TRAN MINH Canh	Ecole des Mines St. Etienne (Chim. Min. Ph.)

E.N.S.E.R.G.

MM. BOREL	Centre d'études nucléaires de Grenoble
KAMARINOS	Centre national recherche scientifique

E.N.S.E.G.P.

M. BORNARD	Centre national recherche scientifique
Mme CHERUY	Centre national recherche scientifique
MM. DAVID	Centre national recherche scientifique
DESCHIZEAUX	Centre national recherche scientifique

Je tiens à remercier ici :

Monsieur S. KRAKOWIAK, Professeur à l'Université I de Grenoble, qui m'a fait l'honneur de présider le jury de cette thèse et a manifesté de l'intérêt pour mon travail.

Monsieur C. KAISER, Professeur au C.N.A.M., qui a fait une lecture critique détaillée du manuscrit de cet ouvrage, et a contribué par ses remarques à son amélioration.

Monsieur J. MOSSIERE, Maître-Assistant à l'I.N.P.G., pour les fructueuses discussions que nous avons eues et pour les encouragements qu'il m'a prodigués.

Monsieur N. NAFFAH, Ingénieur de recherche à l'IRIA, qui a participé à la définition initiale de mon projet, et qui a toujours témoigné sa confiance dans les travaux réalisés par l'équipe réseau de l'ENSIMAG.

J'ai eu la chance de réaliser une partie de ce travail en collaboration avec Monsieur V. QUINT, Ingénieur de recherche IRIA, affecté à Grenoble, qui m'a fait profiter de sa large expérience dans le domaine réseau. La réussite de ce travail doit beaucoup à son aide constante.

Je tiens également à remercier tous les membres de l'équipe réseau de l'ENSIMAG avec qui j'ai pu travailler et échanger des idées, particulièrement à Monsieur G. SERGEANT, responsable de l'équipe, qui ne m'a pas ménagé son temps pour résoudre aussi bien les problèmes administratifs me concernant que pour corriger mon manuscrit tant dans son contenu scientifique que dans sa forme "syntaxique".

J'exprime enfin ma gratitude à Madame M.J. DOREL de tout le mal qu'elle s'est donné dans la frappe de cette thèse.

S.L.

TABLE DES MATIÈRES

INTRODUCTION	1
<u>PARTIE I - DEFINITION DU PROJET</u>	4
<u>Chapitre I - "CYCLADES" : un réseau d'ordinateurs</u>	5
I.0. Introduction	6
I.1. "CIGALE" : un sous-réseau de communication	8
I.1.1. Services fournis	8
I.1.2. Particularités des services	8
I.1.3. Interface externe	9
I.2. Le modèle de référence "ISO"	9
I.2.1. Présentation	9
I.2.2. Les "7 niveaux" du modèle ISO	13
I.3. L'architecture du réseau CYCLADES	15
I.4. L'adressage dans le réseau CYCLADES	22
<u>Chapitre II - Le traitement par lots dans un réseau hété-</u> <u>rogène</u>	24
II.0. Introduction	25
II.1. Le traitement par lots	25
II.2. Le télétraitement par lots	25
II.3. Le traitement par lots dans un réseau d'ordinateurs hétérogène	27
II.4. Le traitement par lots dans un réseau CYCLADES	30
II.4.1. Fonctions des entités-applications	32
II.4.2. L'appareil virtuel et le protocole d'appareil virtuel	33
II.4.3. Identification d'un serveur-réseau	36
II.4.4. Identification d'un terminal lourd-réseau ...	37
II.4.5. Connexion d'un terminal lourd-réseau	37
II.4.6. Possibilités d'extension	37

<u>Chapitre III</u> - Définition d'un serveur de traitement par lots pour le réseau CYCLADES	40
III.0. Introduction	41
III.1. Examen des solutions existantes	41
III.1.1. Méthode d'adaptation interne	41
III.1.2. Méthode d'adaptation externe	46
III.1.3. Méthode d'adaptation mixte	48
III.2. La solution adoptée : un FRONTAL	49
III.2.1. Fonctions du FRONTAL	51
III.2.2. Options envisagées pour l'adaptation	51
III.2.3. Configuration et mode de connexion d'un terminal lourd-réseau	56
III.2.4. Architecture du FRONTAL	56
III.2.5. La conversion des protocoles	58
III.3. Définition du serveur-réseau HASP selon le modèle de référence ISO	60
<u>Chapitre IV</u> - Définition d'un poste d'accès banalisé	63
IV.0. Introduction	64
IV.1. Choix d'une solution	64
IV.2. Configuration disponible	65
IV.3. Commandes opérateur	66
IV.4. Architecture du poste d'accès banalisé	66
IV.5. La couche application selon le modèle de référence ISO	68

<u>PARTIE II</u> - DIFFERENTS ELEMENTS DE LA REALISATION	70
<u>Chapitre I</u> - Etude de la mise en oeuvre	71
I.0. Introduction	72
I.1. Caractéristiques de l'application	72
I.1.1. Système constitué de sous-systèmes	73
I.2. Caractéristiques du matériel choisi	77
I.3. Caractéristiques du moniteur utilisé	78
I.4. Modifications apportées au moniteur	79
I.4.1. Les horloges de garde	79
I.4.2. La synchronisation	79
I.5. Techniques de mise en oeuvre	79
I.5.1. Organisation modulaire	79
I.5.2. Interfaces des différents sous-systèmes	82
I.5.3. Contrôle de flux	85
<u>Chapitre II</u> - Outils pour le développement et la mise au point	91
II.0. Introduction	92
II.1. Les outils disponibles	92
II.2. LPORDDO : un langage de programmation	94
II.2.1. Objectifs	94
II.2.2. Particularités du langage	94
II.2.3. Définition du traducteur	95
II.2.4. Le traducteur réalisé	96
II.2.5. Evaluation du traducteur	97
II.2.6. Evaluation du langage	99
II.3. Les outils développés pour la mise au point	100

<u>PARTIE III</u> - EVALUATION DE LA SOLUTION ADOPTÉE	103
<u>Chapitre I</u> - Evaluation des performances	104
I.0. Introduction	105
I.1. Outils et techniques pour les mesures	106
I.2. Taux d'occupation de l'unité centrale	106
I.3. Taux d'occupation des files d'attente	109
I.4. Etude du contrôle de flux	114
I.5. Quelques remarques	116
<u>Chapitre II</u> - Evaluation du coût du projet en ressources humaines et matérielles	117
CONCLUSIONS	122
BIBLIOGRAPHIE	126
<u>ANNEXE A</u> - Le FRONTAL et le Modèle de Référence ISO	133
<u>ANNEXE B</u> - Le Poste d'Accès et le Modèle de Référence ISO	140

INTRODUCTION

Sujet

Le présent document porte en partie sur les différents aspects de la *conception* et la *définition* d'un service de traitement par lots dans un *réseau hétérogène* d'ordinateurs. On étudie également les problèmes soulevés par la mise en place sur un réseau, de systèmes de télétraitement par lots, non conçus initialement pour cet environnement et les techniques envisageables pour leur adaptation.

Origine du projet

Cette étude est issue de la décision de deux Centres de Calcul Universitaires (CICG - Grenoble et CIRCE - Orsay) d'offrir leur service de télétraitements par lots, à la communauté des utilisateurs du Réseau CYCLADES tout en élargissant les possibilités des usagers locaux en leur fournissant un moyen d'accès à d'autres systèmes du même réseau.

Lieu du développement

Ce travail a été effectué au sein de l'Equipe Réseaux d'Ordinateurs de l'Institut National Polytechnique de Grenoble et la plupart du logiciel développé pour ce projet a été réalisé avec les moyens de traitement du CICG.

Portée et intérêt du projet

L'apparition et le développement des réseaux d'ordinateurs, favorisés par l'évolution des moyens de communication et l'abaissement du coût des mini-ordinateurs et des terminaux en général, ont donné lieu à de nombreux travaux de recherche.

Ces travaux tournent autour de deux grands axes :

- adaptation des applications préexistantes pour les rendre accessible aux utilisateurs éloignés au travers d'un réseau ;
- la définition et la réalisation de nouvelles applications spécifique d'un environnement réparti [CHU77], [ADI78], et l'étude des problème de contrôle et de cohérence qu'elles supposent [RIC78], [SEG78].

Notre projet se situe dans la première catégorie citée.

La définition d'un service de traitement par lots proposée dans ce travail est formulée suivant les recommandations de travaux récents de normalisation [ISO78]. Nous pensons ainsi faciliter la tâche des lecteurs familiarisés avec ces travaux et permettre une éventuelle confrontation de notre proposition avec d'autres, décrites dans des termes analogues.

Outre ces aspects, nous décrivons et analysons quelques problèmes concernant les spécifications et la mise en oeuvre, abordés lors de l'adaptation d'un système de télétraitement par lots.

Le plan de la thèse

Ce document est structuré en trois *parties*. Dans la *première*, nous définissons notre projet, en présentant d'abord le réseau CYCLADES (Chapitre I), ensuite les caractéristiques de son service de traitement par lots (Chapitre II), et finalement les spécifications pour la mise en oeuvre d'un système de traitement par lots (Chapitre III) et celles d'un poste d'accès à ce service-réseau (Chapitre IV).

Nous consacrons la *deuxième* partie à l'analyse de divers aspects concernant la mise en oeuvre (Chapitre I) et les outils utilisés pour le développement et la mise au point (Chapitre II).

Dans la *troisième* partie, nous expliquons les difficultés rencontrées pour effectuer une étude approfondie des performances de notre réalisation (Chapitre I) ; nous fournissons néanmoins quelques résultats d'ordre qualitatif, et nous y donnons aussi une évaluation sommaire du coût des différentes étapes de notre projet (Chapitre II).

Dans la *conclusion*, nous décrivons les caractéristiques des outils susceptibles de faciliter les tâches concernant le développement et la mise au point des applications *temps réel* et *réparties*.

Remarque : La lecture de ce travail suppose une connaissance générale du domaine d'architecture des réseaux d'ordinateurs et des principes de base d'un système de télétraitement par lots.

PARTIE I

DEFINITION DU PROJET

CHAPITRE I

"CYCLADES" : UN RÉSEAU D'ORDINATEURS

I.O. INTRODUCTION

Depuis quelques années, le réseau CYCLADES (*) a fait l'objet de multiples publications et a servi de support à de nombreux projets de recherche.

Notre propos n'est pas ici de renouveler ces exposés, mais plus simplement de présenter le réseau afin d'introduire la *terminologie* utilisée dans la suite de ce document.

On constate dans la littérature spécialisée l'adoption d'une terminologie et d'un vocabulaire différenciés pour définir et désigner les composants d'un réseau d'ordinateurs en fonction de deux grandes catégories d'activités : le *traitement* et la *communication*.

Un *réseau de communication* est constitué d'un ensemble de calculateurs appelés *noeuds*, interconnectés par des moyens de transmission.

Ces noeuds (généralement des mini-ordinateurs) assurent une fonction de transmission et d'aiguillage de messages (considérés comme des blocs d'information).

Un *réseau informatique* est un ensemble d'équipements de traitement (gros ordinateur, mini-ordinateur, microprocesseur, terminal programmable ou non ...), appelés *hôtes*, reliés entre eux par des *moyens de transmission* (lignes téléphoniques spécialisées ou du réseau commuté, liaisons par satellite, faisceau hertzien, ...) ou par des *réseaux de communication* (à commutation de paquets, de messages ou de circuits virtuels, ...).

Les réseaux informatiques construits sur un réseau de communication sont dits à *deux niveaux* et ce dernier est généralement nommé *sous-réseau*, pour renforcer l'idée, qu'il s'agit d'un réseau à l'intérieur d'un autre.

La différence essentielle entre les *hôtes* d'un réseau informatique et les *noeuds* d'un réseau de communication réside dans leur utilisation de l'information.

Les *hôtes* peuvent être source ou puits de l'information échangée au travers du réseau et ils sont, a priori, capables de la traiter, la stocker ou la produire.

(*) Le réseau CYCLADES résulte d'une suite d'études entreprises depuis 1970 dans le cadre du Comité de Recherche en Informatique (CRI).

Par contre, les *noeuds* ne sont utilisés que pour stocker des informations temporaires, pour réaliser des procédures de contrôle et pour déterminer le routage de l'information acheminée.

Un réseau informatique peut être caractérisé par des *critères* divers [POU73] définissant un certain nombre de classes.

On citera ici quelques-uns de ces critères s'appliquant au réseau CYCLADES.

1) Critère de composition

Un réseau peut être défini par rapport à la composition de ses hôtes en *homogène* ou *hétérogène*.

Il est dit *homogène* si ses hôtes comportent des équipements informatiques présentant des propriétés ou caractéristiques identiques.

Le réseau CYCLADES est fortement *hétérogène* puisque sont interconnectés des ordinateurs de types et de constructeurs très divers (MITRA15, IRIS 80, IRIS 50, IBM 360, IBM 370, PDP10, PHILIPS P1200, ...).

2) Critère d'utilisation

Un réseau est dit *spécialisé* ou à *usage général* selon qu'il est conçu ou non pour résoudre une classe de problèmes bien déterminée.

Le réseau CYCLADES est à vocation *générale* puisqu'il supporte un nombre assez varié d'applications (applications conversationnelles, gestion de base de données, ...).

3) Critère topologique

Un réseau peut être classé aussi en fonction de la *topologie* définie par l'interconnexion de ses hôtes (maillé, en arbre, étoilé, ...), par rapport aux *moyens de transmission* ou de *communications*.

Dans ce sens, on peut dire que le réseau CYCLADES est structuré à deux niveaux, car ses hôtes ne sont pas directement reliés entre eux, mais par l'intermédiaire d'un réseau de communication, connu sous le nom de CIGALE.

I.1. "CIGALE" : UN SOUS-RESEAU DE COMMUNICATION

CIGALE est un réseau spécialisé dont les noeuds sont des mini-ordinateurs (MITRA 15) raccordés entre eux par des lignes téléphoniques spécialisées.

Sa topologie est de type *maillé*, c'est-à-dire qu'il existe en général plus d'un chemin pour la transmission de données entre deux points du réseau.

I.1.1. Services fournis

Le réseau CYCLADES permet aux *hôtes* de CYCLADES d'échanger des messages entre eux et avec les ordinateurs connectés à d'autres réseaux tel que EIN, TRANSPAC [DEN79].

L'information destinée à être transportée par CIGALE lui est délivrée dans des blocs appelés *paquets* dont la taille maximale est fixée (256 octets).

Chaque paquet contient une partie utile d'information et une partie de service qui identifie notamment les adresses de l'émetteur et du destinataire (voir Fig. I.4).

La technique d'acheminement utilisée est la *commutation*, c'est-à-dire que chaque paquet est aiguillé de noeud en noeud jusqu'à atteindre le noeud auquel le destinataire est raccordé.

L'existence d'un maillage, allié à la capacité de traitement des noeuds, permet de modifier le chemin suivi par les paquets pour contourner, le cas échéant, les lignes ou les noeuds encombrés ou en panne. Pour cette raison, on parle d'*acheminement adaptatif*.

I.1.2. Particularités des services

. Les paquets d'un message sont acheminés de façon autonome vers son destinataire et peuvent donc emprunter des chemins différents.

Cela peut conduire à les délivrer au destinataire dans un *ordre différent* de l'ordre d'émission.

. Pour maîtriser les situations de congestion (ex : capacité de stockage saturée), les noeuds peuvent être amenés à détruire des paquets. Ceci est

à l'origine de la *perte* éventuelle de paquets émis.

. Dans les cas d'anomalies, les mécanismes de reprise d'erreur peuvent provoquer la *duplication* d'un paquet, c'est-à-dire que le destinataire le recevra en double exemplaire.

I.1.3. Interface externe

Les hôtes de CYCLADES sont raccordés à CIGALE par une ou plusieurs lignes de transmission synchrones, contrôlées par des procédures (BSC, TMM-UC, ...).

I.2. LE MODELE DE REFERENCE "ISO"

I.2.1. Présentation

Bien que la définition du réseau CYCLADES ait précédé l'élaboration d'un modèle de référence pour l'*Interconnexion de Systèmes Ouverts* [ISO78]^(*), [ZIM78] et qu'elle ne s'y conforme pas strictement, on essaiera de présenter son architecture dans les termes les plus proches du modèle cité.

Une description *fonctionnelle* et *relationnelle* est proposée pour définir l'Architecture d'Interconnexion de Systèmes, destinée à permettre le dialogue au travers de moyens de communications, des applications résidant sur des calculateurs différents.

L'architecture proposée est structurée par *niveaux*. Chaque *niveau* i peut être vu, comme un système distribué, constitué par des *entités* $E(i)$ qui fournissent un *ensemble de services* $S(i)$ au niveau supérieur $i+1$, utilisant les services du niveau (voir Fig. I.1) inférieur $i-1$.

Pour réaliser un service, les entités d'un même niveau coopèrent entre elles selon les normes d'un *protocole* spécifique à leur niveau.

La fonction d'un "protocole" est :

(*) La sous-commission SC16 a été créée en Mars 1977 à partir du groupe de travail TC97 de l'Organisation Internationale de Normalisation (ISO) afin d'étudier les problèmes de normalisation concernant l'interconnexion de calculateurs et celles de calculateurs-terminaux.

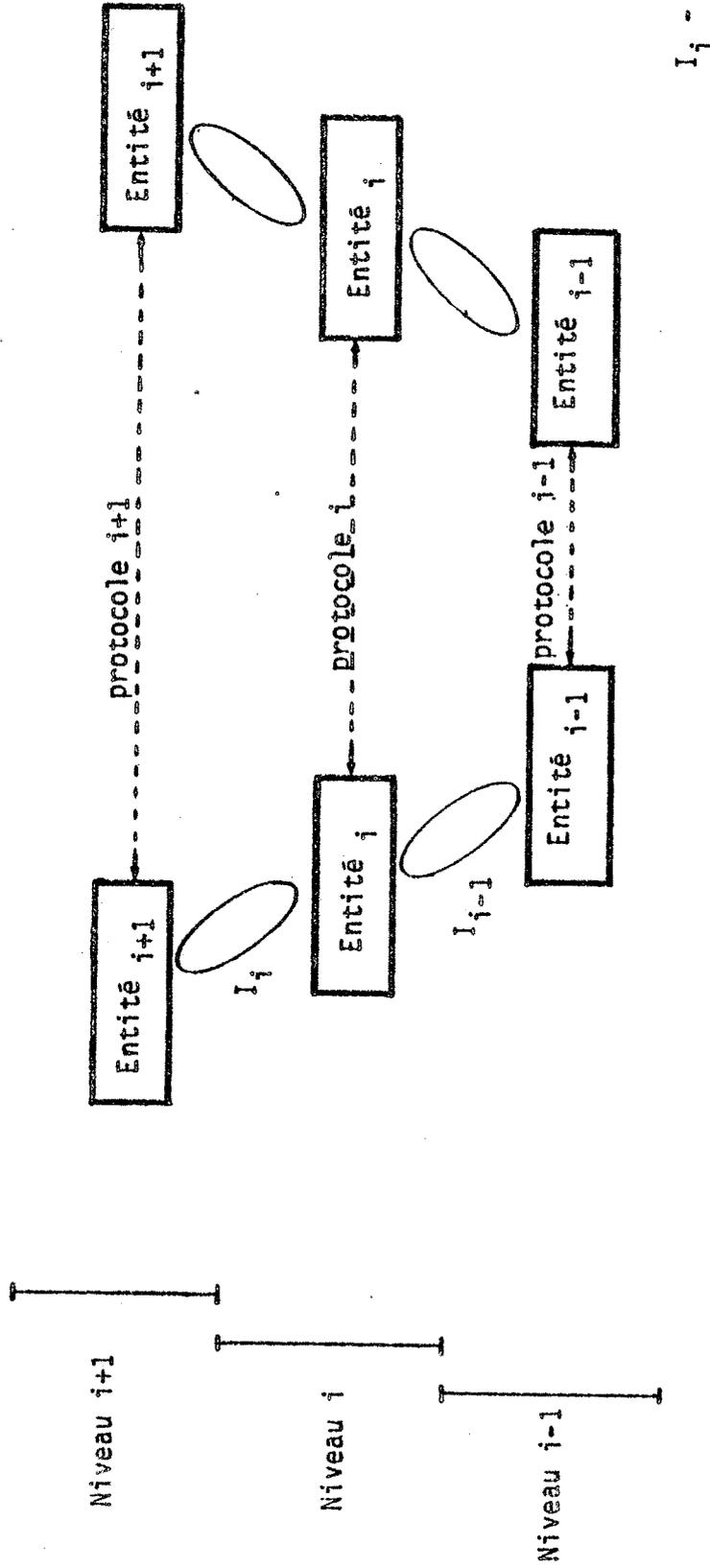


Figure I.1 - Architecture structurée par niveaux proposée dans le Modèle de Référence ISO

- de définir la syntaxe et la sémantique des messages échangés entre les entités d'un même niveau ;
- d'établir les règles qui régissent leur dialogue ;
- de préciser les services fournis aux entités du niveau supérieur.

Pour tous les niveaux, à l'exception du niveau inférieur, le dialogue entre deux entités d'un même niveau se réalise au travers d'une *connexion logique* établie par le niveau immédiatement inférieur, soit implicitement, soit explicitement à la demande de l'une d'entre elles.

L'*établissement*, le *maintien* et la *libération* sont les opérations associées à une connexion.

Les services rendus par les entités d'un niveau destinés à celles du niveau supérieur sont sollicités et fournis au travers d'une *interface* définie entre ces deux niveaux adjacents.

Afin qu'une connexion puisse être établie entre deux entités, chacune possède une ou plusieurs adresses (*points finaux de connexion*) visibles par les entités du niveau inférieur à travers leur interface respective.

Un *identificateur local* est un nom qui permet l'identification univoque des entités ou des points finaux de connexion d'un niveau i seulement à partir d'une entité de niveau $i-1$ donnée.

La notion d'identificateur local est intéressante car elle permet de désigner par le même nom une classe d'objets appartenant à un même niveau et ayant les mêmes fonctions. Un exemple d'utilisation peut être trouvé dans la définition du Service de Traitement par Lots faite pour le réseau CYCLADES (cf. II.4.3).

Un *identificateur global* est composé d'un identificateur local préfixé par la suite des identificateurs désignant les entités des niveaux inférieurs qui servent à définir les environnements englobants.

Dans le modèle décrit, les différents niveaux sont groupés en trois catégories :

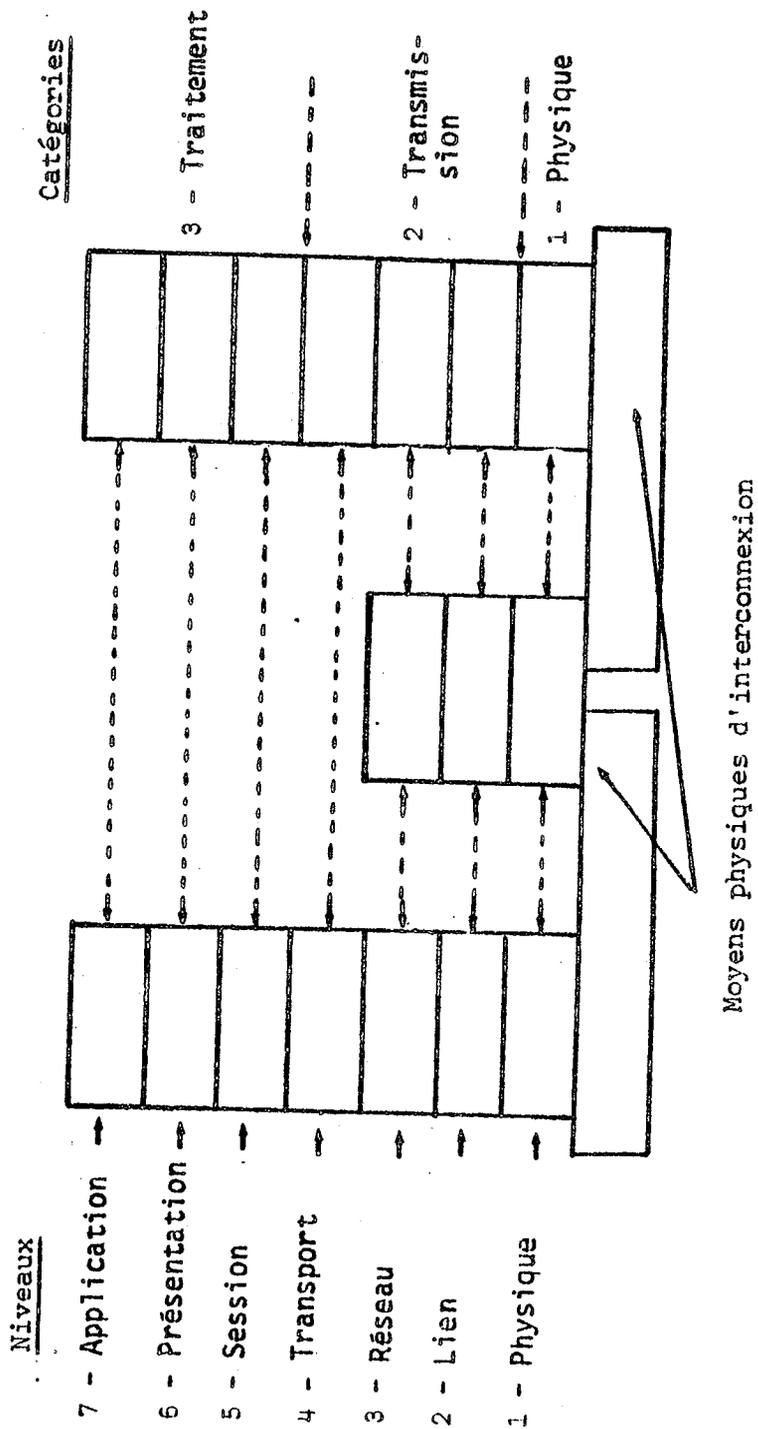


Figure 1.2 - Les "7 Niveaux ISO"

1ère catégorie : pour l'*interconnexion physique* des différents systèmes.

2ème catégorie : pour la *transmission*.

3ème catégorie : pour le *traitement* des données.

Des sous-divisions sont suggérées pour les deux dernières catégories.

On présentera maintenant l'ensemble des niveaux proposés dans le modèle en décrivant sommairement leurs fonctions.

I.2.2. Les "7 niveaux" du modèle ISO

7. Niveau Application

C'est le niveau supérieur. Les protocoles de ce niveau sont associés directement à l'utilisateur en lui fournissant un *service distribué*, c'est-à-dire réalisé par la coopération des entités-applications réparties sur différents systèmes. Le but des niveaux inférieurs est de supporter ce dernier.

6. Niveau Présentation

L'objet de ce niveau est de fournir un ensemble de services choisis par l'application pour lui permettre d'interpréter la signification des données échangées.

C'est à travers l'utilisation de ces services que les applications dans un environnement réparti peuvent communiquer entre elles sans avoir besoin de développer des interfaces adaptées au dialogue pour chaque partenaire différent.

5. Niveau Session

Le but de ce niveau est de gérer la *coopération entre les applications*.

Pour cela deux types de services sont fournis :

- l'établissement d'une connexion logique entre deux entités-applications selon un protocole déterminé par l'application ;
- le contrôle et la synchronisation des échanges entre les entités-applications.

4. Niveau Transport

Ce niveau fournit un moyen pour le *transfert transparent des données* sur une connexion établie entre deux entités de la couche supérieure.

3. Niveau Réseau

Ce niveau met en oeuvre les moyens nécessaires à deux entités du niveau transport pour l'*échange des messages*.

2. Niveau Lien

Ce niveau procure les moyens fonctionnels et procéduraux pour établir et gérer une *liaison de données* (*).

1. Niveau physique

Ce niveau offre les moyens mécaniques, électriques et procéduraux pour établir, maintenir un *circuit de données* (**).

(*) - Définition 53.052 "data link" - Livre Vert du CCITT, Tome 8, page 34.
- ISO DIS 4335, Annexe A.

(**) Ensemble des équipements physiques pour la transmission de signaux destinés au transfert de données (Chap. III [GVI76]).

I.3. L'ARCHITECTURE DU RESEAU CYCLADES

L'architecture du réseau CYCLADES est habituellement présentée comme une structure à deux niveaux. Le premier déjà décrit correspond au réseau CIGALE, le deuxième est décomposé en trois couches :

- COMMUNICATION : qui réalise l'interface d'accès au réseau CIGALE
- TRANSPORT : qui fournit un service plus élaboré que CIGALE
- APPLICATION : qui offre des services aux utilisateurs.

Bien qu'il n'y ait de correspondance apparente entre les *niveaux* du modèle de référence ISO et les *couches* du réseau CYCLADES, les fonctions qu'ils décrivent se recouvrent.

On décrira par la suite les deux premières couches du réseau CYCLADES selon le découpage et la méthodologie proposés dans le modèle de référence ISO (voir Fig. I.3).

La présentation d'une Application suivant les mêmes critères sera donnée dans les Annexes A et B.

1ère couche : Service de Communication

1 - Niveau Physique

Fonctions :

Etablir un circuit-de-données entre un hôte du réseau CYCLADES et un noeud de CIGALE.

Entités :

Le *circuit-de-données* est une liaison synchrone, point à point, matérialisée par deux modems reliés par une ligne de transmission

Protocole :

Les modems et la ligne utilisés sont conformes aux avis CCITT (V24, V35, ...). Ces avis ont pour but de définir des caractéristiques telles que :

- la vitesse de transmission (de 4,8 à 48 Kb/s)
- le type de transmission (synchrone)
- le support de transmission (nombre de fils de la ligne spécialisée)
- le mode de transmission (bidirectionnel simultané ou à l'alternat)

Réseau CYCLADES

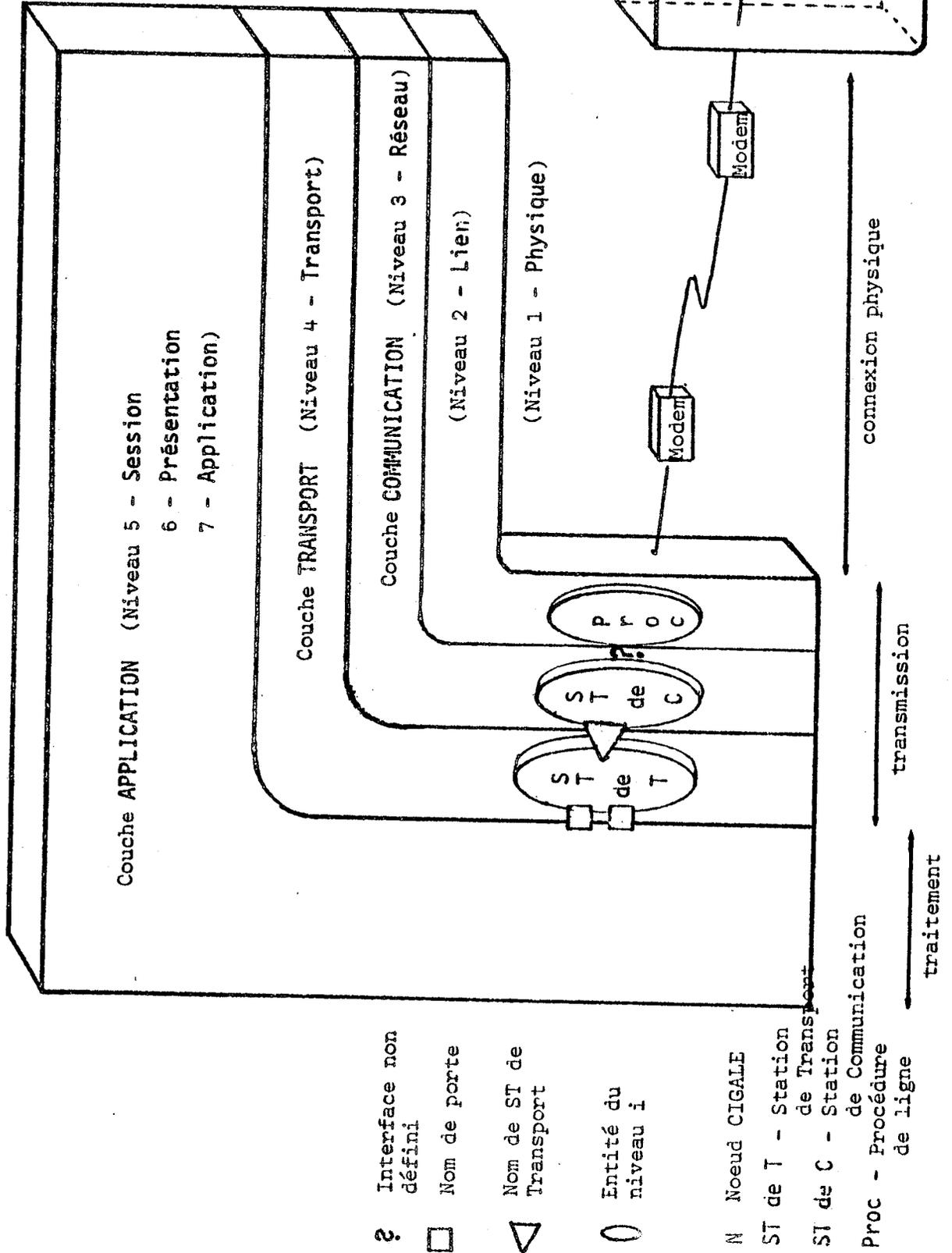


Figure 1.3 - Correspondance entre les couches CYCLADES et les "7 niveaux ISO"

- le principe de transmission (modulation de phase, ...)
- le codage (nombre de données binaires représentées par chaque signal)
- l'interface logique entre le Modem et le Calculateur (jonction ...).

Services fournis au niveau supérieur :

- Etablissement, maintien et libération du circuit de données (synchronisation des horloges des modems ...).
- Transmission et réception d'une suite de valeurs binaires.

Interface :

Le coupleur du calculateur hôte connecté à CIGALE matérialise l'interface à ce niveau.

2 - Niveau Lien

Fonctions :

Réaliser la gestion du circuit de données afin d'offrir au niveau supérieur une liaison de données.

Entités :

Procédure de ligne (logicielle ou câblée).

Protocole :

Les fonctions réalisées et les services fournis sont établis par une *procédure de ligne* (TMM-UC), BSC, ...) qui définit :

- le mode d'exploitation d'une ligne (bidirectionnel simultané ou à l'alternat)
- les principes pour la détection d'erreur (redondance verticale, cyclique, ...)
- les procédures de reprise en cas d'erreur (réémission spontanée ou sollicitée, ...)
- le contrôle et la coordination du transfert (numérotation des messages, acquittement, ...)
- le code ou alphabet utilisé (EBCDIC, ...)
- la structure des messages (début, fin, ...).

Services fournis au niveau supérieur :

- Activer, maintenir et libérer la liaison de données.
- Emettre et recevoir des messages à travers cette liaison.

Interface :

Il s'agit d'une interface locale propre à chaque mise en oeuvre.

3 - Niveau Réseau

Fonctions :

Réaliser une méthode d'accès au réseau CIGALE. Si nécessaire, ce niveau démultiplexe et multiplexe les paquets à destination ou en provenance des différentes Stations de Transport (cf. Interface).

Entités :

Elles ne possèdent pas de nom conventionnel, baptisons-les "Station de Communication".

Protocole :

Le protocole du réseau CIGALE [POU74] est utilisé comme référence pour définir :

- le format et la taille des paquets.

Services fournis au niveau supérieur :

- L'émission et la réception de paquets à destination ou en provenance d'un hôte de CYCLADES ou d'un autre réseau accessible à travers CIGALE (EIN, TRANSPAC, ...)

Interface :

Les Stations de Transport sont les seules entités du niveau supérieur adressables à travers cette interface.

Les Stations de Transport de CYCLADES peuvent échanger des paquets sans établir un lien explicite ; il existe de fait par les services de CIGALE.

2^{ème} couche : Service de Transport

Les services fournis par ce niveau sont utilisés par les Applications pour dialoguer au travers du réseau. Afin de faciliter la présentation du Chapitre II, du Service Réseau de Traitements par Lots, nous les décrivons ici avec plus de détails.

Ces services sont réalisés par la coopération des logiciels appelés *Stations de Transport*, résidentes sur chaque calculateur hôte du réseau CYCLADES.

Les entités-application qui utilisent le Service de Transport s'appellent *abonnés*. Pour rendre effectif leur abonnement, elles déclarent une ou plusieurs *portes* auprès de la Station de Transport locale à leur site (unité logique d'adressage).

L'information est véhiculée par le Service de Transport sous forme de *lettre* ou de *télégramme*.

Les télégrammes ne contiennent qu'une information très brève (2 octets) et sont expédiés prioritairement.

Les *Applications* peuvent échanger des lettres ou des télégrammes en adressant leurs portes respectives. Mais si deux applications désirent que les lettres échangées soient remises à la réception dans l'ordre de leur émission, elles sollicitent auprès du Service de Transport l'établissement d'une *liaison* entre deux de leurs portes respectives, valable pour la durée du dialogue.

Des services additionnels peuvent être demandés au moment de l'établissement d'une liaison permettant de contrôler les pertes de lettres (*contrôle d'erreurs*) d'asservir le rythme des échanges à la capacité du récepteur (*contrôle de flux*).

Les télégrammes échangés sur une liaison ne sont soumis à aucun contrôle.

Les lettres supérieures à un paquet sont découpées en *fragments* de taille fixe (sauf éventuellement le dernier), numérotés et expédiés chacun dans un paquet et rassemblés à l'arrivée pour recomposer la lettre.

Les Stations de Transport s'adressent des *commandes* (rangées dans des paquets). Selon le code fonction de la commande, elles déterminent si la commande contient de l'information destinée aux applications (lettres ou télégrammes) ou des renseignements de service (accusé de réception, ...) (Voir Fig. I.4).

Par la suite, on résume les particularités du niveau Transport selon le schéma suivi pour la présentation des niveaux précédents.

4 - Niveau Transport

Fonctions :

Eliminer les éventuelles erreurs du service fourni par CIGALE (duplication, perte, ...) et offrir un service plus élaboré aux entités du niveau supérieur (adressage, contrôle de flux, fragmentation, assemblage, ...).

Entités :

Stations de Transport (ST).

Protocole :

Le protocole de bout en bout du réseau CYCLADES [ZIM75a] définit le format des messages échangés par les Stations de Transport et les normes pour leur dialogue.

Les messages s'appellent *commandes* et peuvent contenir :

- de l'information utile destinée aux applications sous forme de *lettres* ou sous forme de *télégramme* ;
- des demandes pour l'établissement ou la libération d'une *liaison* (association de deux portes) ; de renseignements pour le contrôle ou le maintien de ces liaisons (accusé de réception, ...) ;
- si la taille des lettres échangées par les applications est supérieure à celle d'un paquet véhiculé par CIGALE, la "ST" émettrice *fragmente* les lettres en paquets et la "ST" réceptrice *assemble* les paquets pour reconstituer les lettres.

(Ces opérations ainsi que d'autres contrôles effectués par les ST sont transparents aux applications).

Services fournis au niveau supérieur :

- L'établissement, le maintien et la libération d'une *liaison*.
- L'émission et la réception de *lettres* et de *télégrammes* adressés soit à une porte, soit sur une liaison.

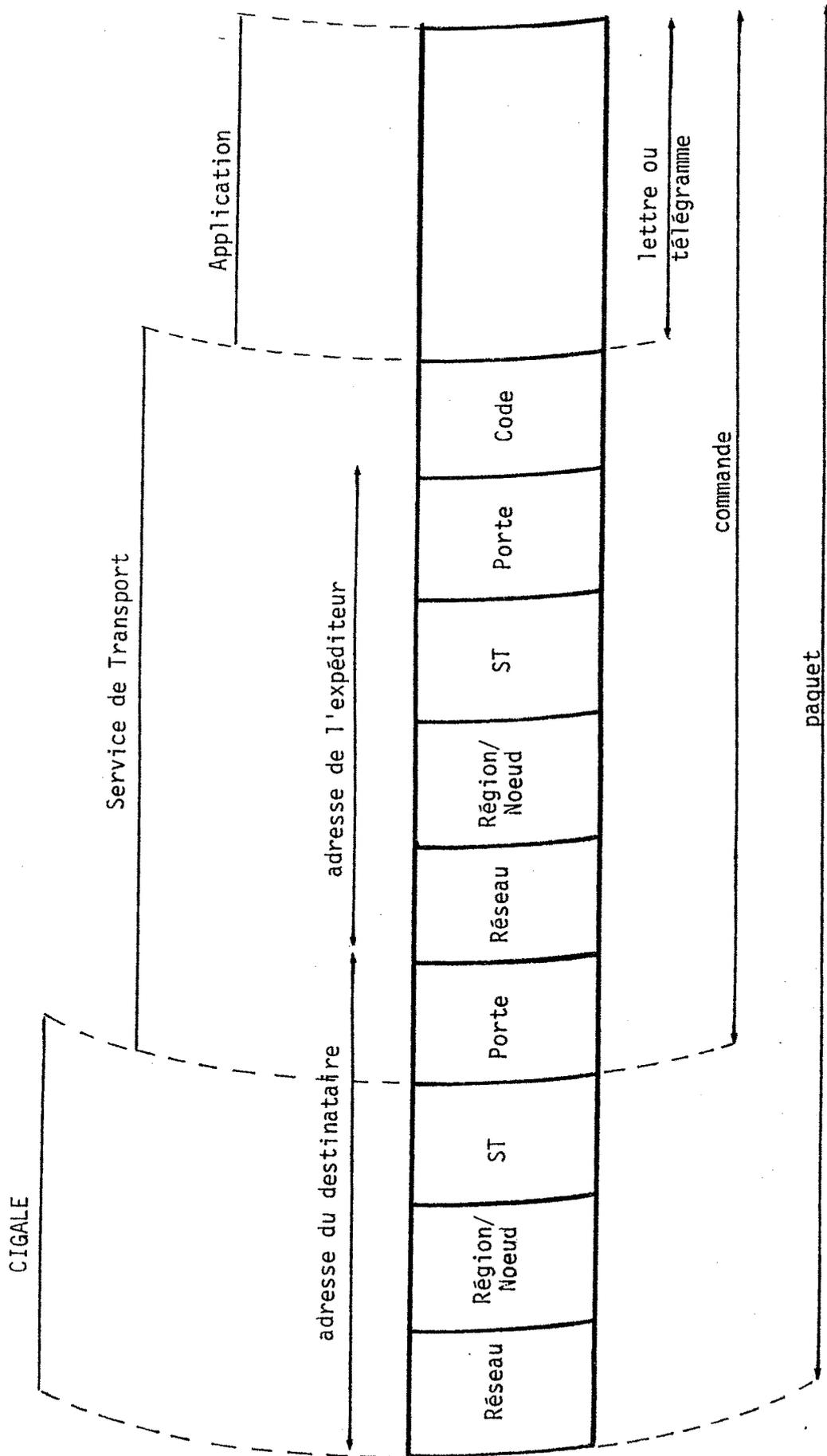


Figure I.4 - Structure d'un paquet

- Des services additionnels sont réalisés à la demande des applications pour garantir que la totalité des messages émis sont délivrés au récepteur (*contrôle d'erreur*) et la régulation du rythme des échanges (*contrôle de flux*).

Interface :

Les *portes* et les *liaisons* qui les associent constituent les uniques entités adressables à travers cette interface.

Une application peut *ouvrir* et *fermer* une porte auprès de sa ST locale pour signaler si elle est prête ou non à recevoir de l'information ou des demandes de connexion sur cette porte.

I.4. L'ADRESSAGE DANS LE RESEAU CYCLADES

Les *adresses globales* du destinataire et de l'expéditeur (voir Fig. I.4) accompagnent un paquet tout au long de son acheminement. Le paquet est aiguillé vers sa destination selon un *mécanisme d'adressage hiérarchique* où chaque niveau ne traite que la partie d'adresse le concernant.

L'adresse globale d'un abonné du Service de Transport contient :

- l'identificateur du réseau CIGALE ;
- le numéro d'une des régions^(*) de CIGALE ;
- le numéro d'un noeud dans la région ;
- le numéro d'une Station de Transport raccordée à un noeud ;
- le numéro d'une porte à l'intérieur d'une Station de Transport.

Dans l'adressage hiérarchique, on considère que les entités à traverser pour atteindre un destinataire sont emboîtées.

On peut représenter la structure d'adressage du réseau CYCLADES par des courbes de niveaux dont les *portes* des Applications constituent les sommets (voir Fig. I.5).

Par exemple, l'information émise par une Application à travers une *porte* à destination d'une autre *porte* est transférée à partir d'un sommet en descendant niveau par niveau jusqu'à atteindre la première courbe de niveau commune à ces deux *portes*, ensuite elle est transmise en remontant niveau par niveau jusqu'à atteindre le sommet de la porte réceptrice.

(*) Pour faciliter l'adressage, dans CIGALE, les noeuds sont groupés par région.

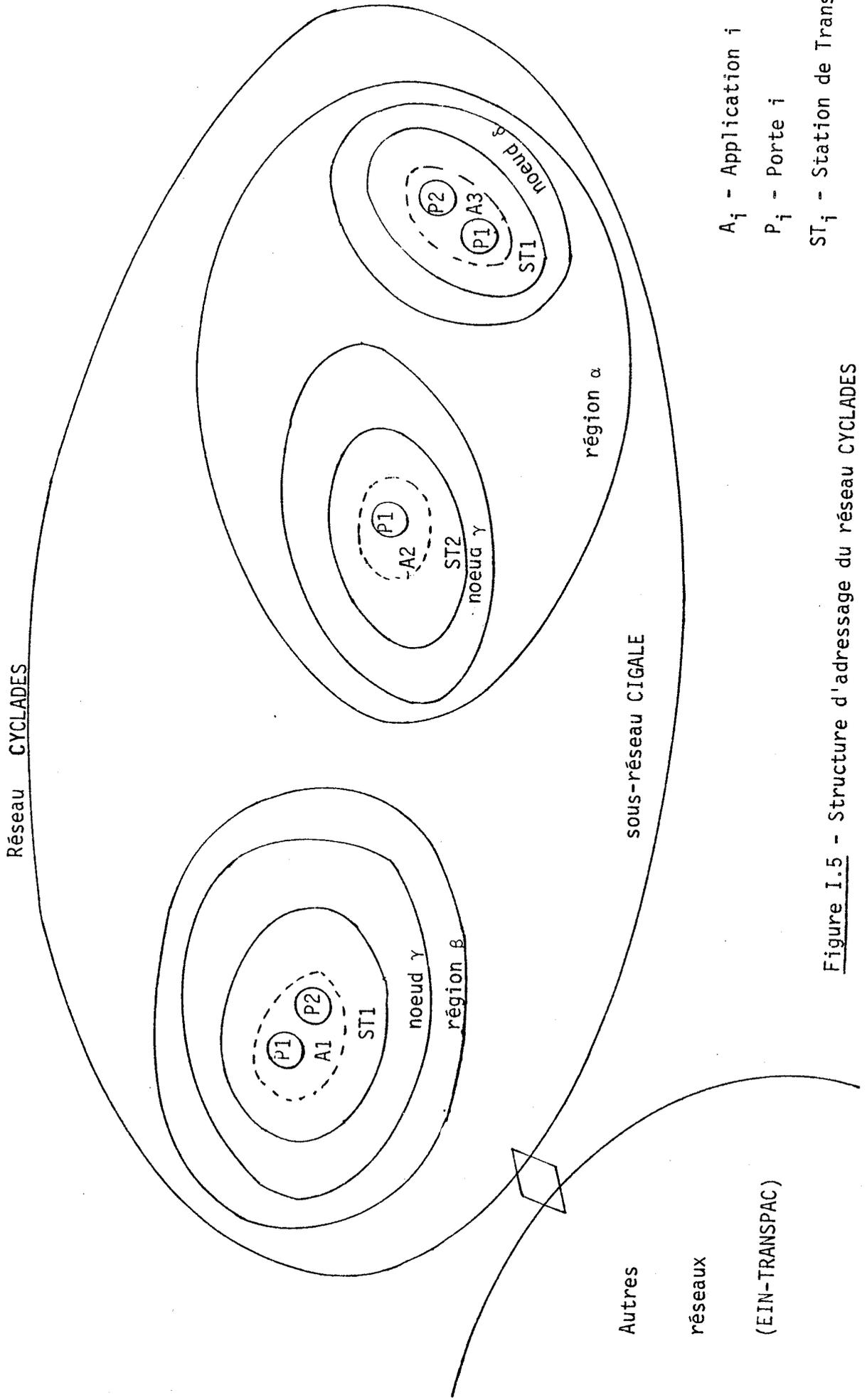


Figure I.5 - Structure d'adressage du réseau CYCLADES

CHAPITRE II

LE TRAITEMENT PAR LOTS DANS UN RÉSEAU HÉTÉROGÈNE

II.0. INTRODUCTION

Dans le présent chapitre, nous décrivons différentes approches pour la mise en place d'un service de traitement par lots dans un réseau hétérogène d'ordinateurs.

Une estimation rapide des avantages et inconvénients de chaque approche est proposée. Comme illustration de l'une d'entre elles, nous présentons la définition du Service de Traitements par Lots dans le réseau CYCLADES.

II.1. LE TRAITEMENT PAR LOTS

Ce type de traitement, vue de l'utilisateur, se caractérise par :

- la soumission au système de traitements par lots d'un ensemble de programmes et de données (STEP) groupés dans un travail (JOB) accompagnés d'une série de *commandes ou informations de contrôle* (JCL) spécifiant l'ordre d'exécution de ces différents programmes et l'ensemble des ressources nécessaires à leur exécution ;
- l'exécution de ce travail et la récupération de ses résultats en *différé* par rapport au moment de son entrée dans le système.

Note : Rappelons que pour des raisons d'efficacité, le système range les travaux rentrés et leurs résultats dans une mémoire auxiliaire, afin de remplacer, lors de l'exécution d'un programme, les opérations d'entrée-sortie sur périphérique lent (lecteur de cartes, imprimante) par des opérations d'entrée-sortie sur supports à accès plus rapide.

II.2. LE TELETRAITEMENT PAR LOTS

Par rapport au fonctionnement précédent, le télétraitement par lots ("Remote Batch") apporte des facilités aux utilisateurs, en permettant la soumission des travaux et la récupération des résultats à partir de "postes éloignés" de l'ordinateur central.

Ces postes d'accès appelés *terminaux lourds* sont connectés à l'ordinateur hôte du système de télétraitement, par un support de transmission de données

Un *terminal lourd* (TL) est constitué d'un ensemble de périphériques conventionnels (lecteurs de cartes, perforateurs de cartes, console, imprimante, ...) et d'une logique de contrôle.

Ses fonctions sont d'assurer la gestion des organes d'entrée-sortie qui lui sont attachés et des échanges avec l'ordinateur (dit "site central") à travers une ligne de transmission.

Protocole

L'adoption d'un protocole pour normaliser le format et l'interprétation des messages échangés entre le "site central" et les terminaux lourds s'avère nécessaire si l'on veut rendre possible l'utilisation des périphériques hétérogènes sans alourdir leur gestion par le système de télétraitement.

Liaison physique

En général, les terminaux lourds sont raccordés à l'ordinateur central :

- soit par une *liaison spécialisée*,
- soit par une *ligne téléphonique du réseau commuté*.

Cette dernière possibilité permet l'utilisation en alternance d'un même point d'accès à l'ordinateur par différents terminaux lourds. D'où le besoin pour le système d'*identifier* le terminal lourd actif à l'extrémité d'une ligne de transmission.

La liaison entre un terminal lourd et l'ordinateur peut être interrompue pour des raisons diverses, involontaires (pannes), ou volontaires (mise en service du terminal lourd pendant une tranche horaire donnée).

Lors du rétablissement de cette connexion, le système de gestion de terminaux lourds doit pouvoir retourner les résultats des travaux non encore transmis au terminal d'origine.

En conséquence, pour éliminer les possibilités d'erreur d'adressage, chaque terminal doit posséder un *nom unique* dans le système.

Dans certains systèmes de gestion de terminaux lourds [IBM a], [CII], chaque terminal lourd client est déclaré à l'aide d'un identificateur unique, accompagné d'une description de ses caractéristiques (configuration, type de liaison

physique, taille de tampons, valeur des différentes horloges de garde, etc).

Les différents systèmes de télétraitement fournissant des services et des facilités analogues, peuvent néanmoins utiliser diverses techniques ou méthodes pour résoudre les aspects concernant

- le raccordement physique,
- le contrôle de la ligne de transmission,
- la gestion des périphériques,
- le langage de commandes,
-

II.3. LE TRAITEMENT PAR LOTS DANS UN RESEAU D'ORDINATEURS HETEROGENE

Dans la présente section, nous présentons différentes approches pour la mise en place d'un service de traitement par lots dans un réseau d'ordinateurs et nous les commentons brièvement.

- 1 - Un réseau peut fournir ses supports de transmission et ses services de communication pour remplacer les liaisons directes entre l'ordinateur du système de télétraitement par lots et ses terminaux lourds.
- 2 - Dans un réseau, différents systèmes de télétraitement peuvent offrir leurs services d'une façon normalisée sans que les utilisateurs aient besoin de connaître leurs particularités.
(Ex : le langage de commandes, etc) [ISO78 - Annexe C].
- 3 - Dans un réseau, différents systèmes de télétraitement peuvent être adressés et accédés d'une façon banalisée, mais les caractéristiques de chaque système sont reflétées aux utilisateurs.

1 * Dans la première approche, le réseau vient *remplacer un fil*. Les avantages sont liés à une meilleure utilisation des supports de transmission, mis en commun, et à la possibilité offerte aux terminaux lourds d'accéder au travers du réseau à tous les systèmes de traitements par lots compatibles.

Par contre, malgré l'accès au service de communication du réseau, cela n'élargit pas les possibilités offertes aux utilisateurs de ces terminaux de soumettre leurs travaux à une plus large variété de systèmes de traitement par lots ayant des caractéristiques dissemblables.

Cette approche est bien adaptée par exemple dans le cadre d'un réseau homogène où l'on veut pratiquer une politique de distribution de la charge des travaux [SOM75].

Un exemple de réalisation inspirée de ces principes figure dans l'adaptation faite à un système IBM HASP MULTI-LEAVING et à ses terminaux lourds pour dialoguer au travers d'un réseau de commutation de paquets du type SL-10 [MEL79].

- 2 * La deuxième approche présente un grand intérêt pour les utilisateurs car elle apporte des avantages touchant aussi bien à la *souplesse* qu'aux *commodités* d'utilisation. En effet, elle permet aux utilisateurs d'un même terminal lourd de soumettre des travaux à des systèmes hétérogènes de traitement par lots, selon une *méthode d'accès normalisée* sans que leurs différences leur soient visibles.

Si cette approche présente des avantages du point de vue de l'utilisateur, elle offre néanmoins des difficultés pour sa définition et sa mise en oeuvre.

Dans le cadre du présent travail, nous nous contenterons de donner seulement un aperçu de la complexité d'une telle réalisation.

Pour éviter aux utilisateurs de connaître les particularités d'un système donné, il faut leur permettre de spécifier les ressources et directives nécessaires à l'exécution de leurs travaux sous une forme unique pour les différents systèmes.

On est donc conduit à définir un *Langage de Commande Réseau - LCR* ("Network Job Control Language - NJCL) traduisible dans les langages des différents systèmes mis sur le réseau, ou à mettre en place dans l'avenir.

Cette idée n'est pas nouvelle [RAY73], [SOM75], mais elle n'a été résolue d'une façon satisfaisante que pour des réseaux homogènes ou de petits réseaux hétérogènes [DEC73], où le LCR reflète la nature des systèmes des calculateurs participants.

Les difficultés à surmonter pour définir un LCR convenable deviennent évidentes même pour les non-initiés quand on constate qu'il y a au moins autant de langages de commande différents que de constructeurs de calculateurs.

Malgré les progrès effectués par des groupes de travail tels que le *British Computer Society (BCS) JCL group* pour la définition d'un langage de commande normalisé [RAY77], aucune solution universelle n'a encore été proposée.

D'autres problèmes restent aussi à résoudre, par exemple, la traduction de ce *langage réseau* dans les langages objets des différents systèmes. Certains travaux en cours de développement concourent à résoudre différents aspects du même problème [DAN78].

- 3 * La troisième approche offre la possibilité de *connecter une grande variété de terminaux lourds* au travers du réseau, à une large gamme de systèmes de traitement par lots tout en répercutant leurs caractéristiques aux utilisateurs.

Cette approche possède les principaux avantages de la deuxième, en ce qui concerne la souplesse vue :

- soit par les systèmes de traitement par lots qui peuvent satisfaire des demandes de connexion provenant de terminaux lourds initialement non programmés pour ces systèmes ;
- soit par l'utilisateur ayant à sa disposition un terminal lourd connecté au réseau avec la possibilité de choisir un "système" parmi plusieurs pour soumettre ses travaux en fonction de son application, de son budget, etc.

Cette troisième approche, en raison de ses avantages et de la faisabilité d'une mise en oeuvre relativement rapide, a été choisie pour la mise en place d'un service de traitement par lots dans le réseau CYCLADES.

II.4. LE TRAITEMENT PAR LOTS DANS LE RESEAU CYCLADES

Au début de cette section, on introduit la terminologie utilisée pour définir l'application-réseau dont il est question dans ce travail.

Nous entendons par *service réseau de traitements par lots* les facilités mises à la disposition des utilisateurs pour accéder de façon normalisée à différents systèmes de traitement par lots.

On appelle *serveur réseau de traitement par lots* (SRTL) tout système de traitement par lots accessible au travers du réseau.

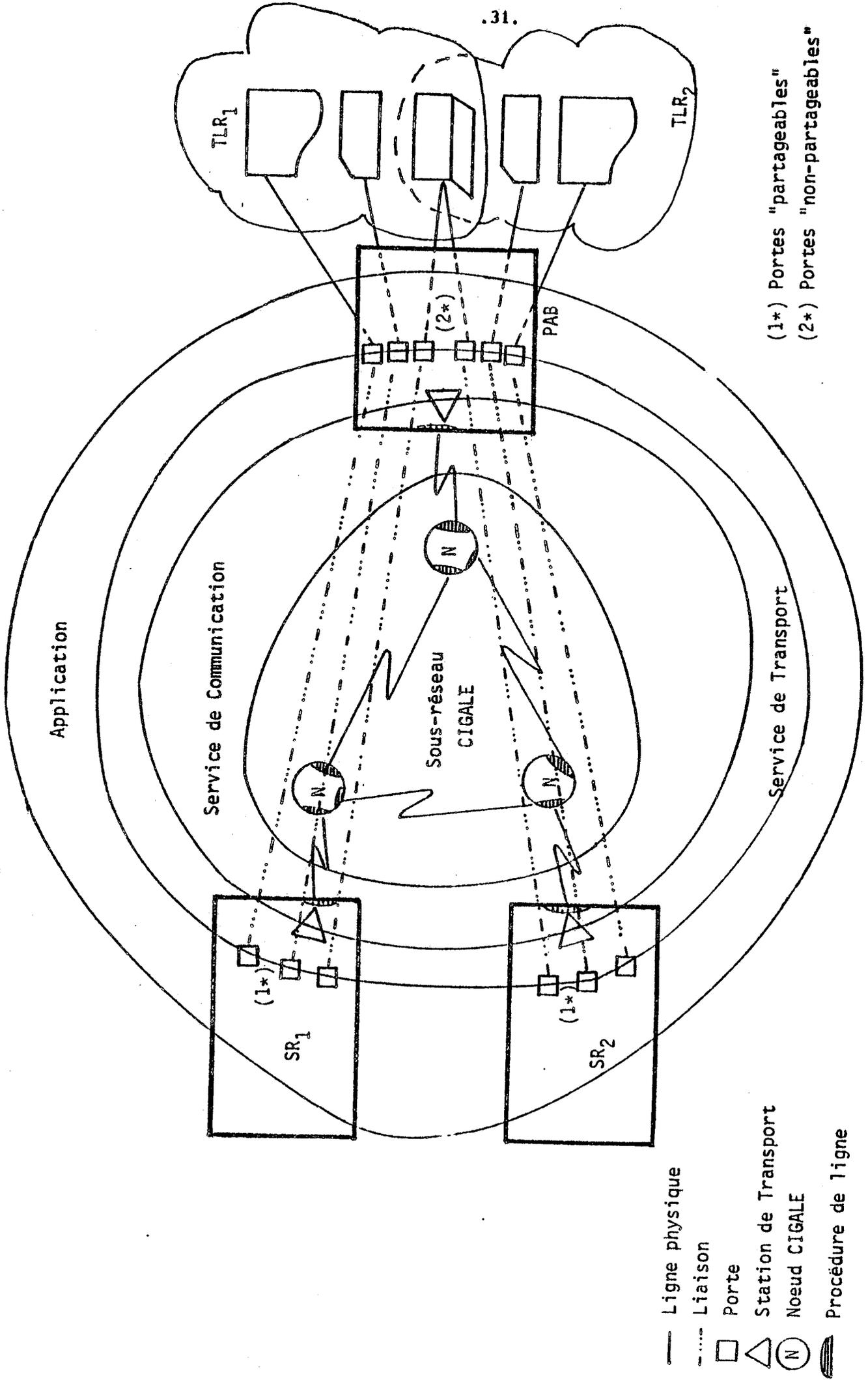
Le *terminal lourd réseau* (TLR) a vis-à-vis d'un Serveur-Réseau des fonctions analogues à celles d'un terminal lourd dans un système de télétraitement classique (cf. II.2). Il est vu par un serveur-réseau comme la *source* des travaux et le *puits* des résultats, mais ce dernier ignore la façon dont le terminal lourd réseau est réalisé, s'il est réel, émulé, câblé ou programmé.

Lorsqu'il n'y a pas d'ambiguïté, le serveur-réseau est nommé simplement *serveur* et le terminal-lourd-réseau *terminal*.

On désigne par *poste d'accès banalisé* (PAB) au service-réseau de traitements par lots l'entité (mini-ordinateur, émulateur, processus d'application, ...) qui fournit un moyen d'accès normalisé aux serveurs.

Il est du ressort du poste d'accès banalisé de mettre en oeuvre un langage de commande local pour que les utilisateurs puissent demander la connexion, déconnexion et configuration d'un terminal en spécifiant le serveur choisi.

Selon le mode de réalisation et les ressources disponibles (mémoire, périphériques, ...), un poste d'accès banalisé peut émuler ou gérer simultanément plusieurs terminaux vis-à-vis différents serveurs.



(1*) Portes "partageables"
 (2*) Portes "non-partageables"

Figure II.1 - Le Service de Traitements par Lots dans l'Architecture du réseau CYCLADES

II.4.1. Fonctions des entités-applications

Un Serveur-Réseau est une application résidant sur un ordinateur hôte du réseau CYCLADES capable (voir Fig. II.1) :

- d'accepter (ou de refuser) des demandes de connexion d'un terminal,
- de recevoir des travaux et de les délivrer au sous-système adéquat, pour leur traitement,
- de retourner les résultats des travaux au terminal d'origine.

Un terminal est aussi mis en oeuvre sur un hôte du réseau. Il possède l'initiative pour entamer les négociations avec un serveur, conduisant à une connexion.

Un serveur et un terminal peuvent être vus comme des "partenaires" d'une *session* de travail.

En général, les serveurs possèdent des renseignements sur chaque terminal utilisant leurs services (nom, adresse, configuration, ...).

Les terminaux, de leur côté, sont supposés connaître les normes de connexion et les modalités de fonctionnement établies par les serveurs accédés.

Dans ce sens, on peut dire qu'un terminal est *orienté* vers un serveur quand il s'adapte aux conventions de ce dernier pour se connecter et travailler avec lui.

Les normes fixées par un serveur peuvent comporter entre autres indications : l'ordre de connexion des différents appareils d'un terminal et certaines particularités de leur gestion.

Un serveur connaît l'entité terminal, mais l'existence du poste d'accès banalisé et sa fonction d'adaptateur lui sont transparentes.

Un utilisateur adresse au poste d'accès banalisé des directives pour préciser à quel serveur un terminal doit être connecté. Il peut spécifier le nom ou l'adresse du serveur choisi, signaler les périphériques utilisés au cours de cette séance de travail, etc.

Les postes d'accès banalisés peuvent mettre en oeuvre une gamme d'outils assez variée, plus ou moins sophistiquée pour faciliter le dialogue avec les utilisateurs ainsi que des techniques très diverses pour émuler les terminaux [BOL79], [ROS78], [ANS76], [FOU76].

II.4.2. L'appareil virtuel et le protocole d'appareil virtuel

Dans le but de garantir la compatibilité entre les périphériques associés aux différents terminaux et les conventions utilisées par les serveurs pour leur gestion, un *modèle abstrait* a été défini pour les classes d'appareils suivants :

- console,
- lecteur de cartes,
- imprimante,
- perforateur de cartes.

Ce modèle, appelé *appareil virtuel (AV)* a pour but de fournir une représentation normalisée des fonctions et caractéristiques de chaque classe.

Il définit :

- implicitement la notion de *ligne virtuelle* par les fonctions de "fin de ligne" ou "nouvelle ligne" ;
- les *fonctions d'adressage* spécifiques à l'appareil.
(ex : pour l'imprimante : "saut canal" ; pour la console : "prochaine tabulation") ;
- les *interruptions* ou *fonctions de contrôle*, c'est-à-dire les mécanismes de l'appareil qui facilite sa gestion ou contrôle
(ex : pour la console : touches fonctions, sonnette, voyant lumineux).

Un ensemble de règles désigné par *protocole d'appareil virtuel (PAV)* est défini pour chaque classe d'appareil virtuel, afin de spécifier les conventions que les deux parties de l'application (le serveur et le terminal) doivent respecter pour effectuer les échanges au travers du réseau [ZIM75a].

Les règles d'un protocole portent sur :

- l'organisation des données à transférer,
- la syntaxe et la sémantique des messages à échanger,
- le mode et le contrôle du dialogue.

Un *code* est défini pour chaque *fonction d'adressage* d'un appareil virtuel, ainsi que pour chaque *critère de compression de texte* utilisé.

a - Structure des messages

Les données à transférer sont organisées en séquence de *directives d'adressage* et de *fragments de texte*, compressé ou non.

Les différents protocoles d'appareil virtuel définissent l'*item* comme l'unité élémentaire de transfert qui peut contenir une fonction d'adressage et/ou un fragment de texte.

Les *items* sont rangés dans des lettres et celles-ci envoyées sur la liaison établie pour l'appareil virtuel concerné.

b - Mode de fonctionnement

Chaque classe d'appareil virtuel possède un mode de fonctionnement qui lui est propre.

Par exemple, il est établi que :

- les *consoles* permettent des échanges bi-directionnels à l'alternat,
- les *lecteurs* et les *perforateurs de cartes*, les *imprimantes* ont un fonctionnement uni-directionnel :
 - . en émission du terminal au serveur pour les lecteurs,
 - . et en émission du serveur vers le terminal pour les imprimantes et les perforateurs.

c - Contrôle du dialogue

Les protocoles reconnaissent trois phases distinctes dans le dialogue :

- l'initialisation,
- le transfert de données,
- la terminaison

et établissent des règles pour chacune d'entre elles.

c1 - L'initialisation

Il est du ressort du terminal de demander au Service de Transport l'établissement d'une *liaison* entre lui et le serveur concerné, avec les services supplémentaires de *contrôle d'erreur* et *contrôle de flux*, pour chaque appareil virtuel utilisé (cf. I.3).

c2 - Le transfert

Le serveur possède la prérogative d'initialiser les échanges sur une *liaison* après son établissement :

- si la *liaison* concerne un appareil virtuel de type *lecteur de cartes*, c'est au serveur d'autoriser le terminal à commencer les transferts à partir de cet appareil ;
- si, par contre, la *liaison* est associée à un appareil virtuel de type *console*, *imprimante* ou *perforateur de cartes*, le serveur a le droit d'émettre sur cette *liaison*.

Remarque : Le serveur et le terminal échangent alternativement le droit d'émission sur une *liaison* associée à la console. Le terminal réclame ce droit au serveur chaque fois qu'il lui est nécessaire et il le rend après l'émission d'un message.

c3 - La terminaison

Le serveur ou le terminal peuvent indistinctement demander au Service de Transport la *libération d'une liaison* partagée.

Note : Un "appareil virtuel" peut être émulé par des appareils réels appartenant à différentes classes. Par exemple, un lecteur de cartes virtuel peut être émulé par un dérouleur de bande magnétique ou une disquette souple.

II.4.3. Identification d'un serveur-réseau

Un serveur est identifié par l'adresse de sa Station de Transport et un ensemble de portes.

Il possède une porte par classe d'appareil virtuel utilisée pour accomplir son service.

On rappelle que les *liaisons* établies pour chaque appareil virtuel sont des associations logiques entre deux *portes*.

Pour que les terminaux puissent formuler au Service de Transport les demandes d'établissement de liaison, il faut qu'ils connaissent d'une part l'adresse de la Station de Transport du serveur, et d'autre part les noms de ses portes.

Afin de simplifier la mise en oeuvre des postes d'accès banalisé et réduire la taille de leurs tables d'adressage, il a été décidé que les portes des différents serveurs correspondant à une même classe d'appareil virtuel auront le même nom (voir tableau II.1) sur tout le réseau.

Note : L'adressage hiérarchique utilisé dans CYCLADES permet l'utilisation du même identificateur local à une Station de Transport pour désigner l'ensemble des portes définissant une classe (cf. I.2.1).

Quand un serveur est prêt à recevoir des demandes de connexion, il *active* ses portes auprès de sa Station de Transport.

TYPE D'APPAREIL VIRTUEL	NOM DE PORTE
console	"BT" (BATCH)
lecteur de cartes	"CR" (CARD READER)
perforateur de cartes	"CP" (CARD PUNCH)
imprimante	"LP" (LINE PRINTER)

Tableau II.1 - Nom des portes associées aux différentes classes d'appareil virtuel

II.4.4. Identification d'un terminal-lourd-réseau

Les serveurs actuellement mis en place sur le réseau identifient un terminal par une adresse de Station de Transport (ST) en supposant que toutes les liaisons concernant les différents appareils virtuels sont gérés du côté du terminal au travers de la même ST.

Le terminal (ou PAB), pour sa part, possède une porte pour chaque appareil virtuel supporté. Par exemple, si deux appareils virtuels sont émulés simultanément sur le même appareil réel, une porte est associée à chacun.

Les portes des serveurs sont dites *partageables* car plusieurs liaisons simultanées peuvent utiliser la même porte.

A l'opposé, les portes d'un terminal sont *non partageables* car chacune est associée à un appareil virtuel différent.

D'autre part, les noms des portes d'un terminal ne se plient à aucune convention, du fait que les terminaux ont l'initiative des demandes de connexion.

Remarque : Un serveur, en fonction de la porte adressée par un terminal dans une demande de connexion, sélectionne le protocole adéquat pour gérer l'appareil virtuel associé à la liaison.

II.4.5. Connexion d'un terminal-lourd-réseau

La connexion d'un terminal à un serveur se réalise par la connexion successive de ses appareils virtuels, c'est-à-dire par l'établissement d'une liaison pour chaque appareil virtuel. L'ordre dans lequel ces connexions doivent être effectuées, ainsi que la configuration admise du terminal (nombre et type des appareils virtuels) sont définis par chaque serveur.

II.4.6. Possibilités d'extension

Les serveurs actuellement mis en place sur le réseau utilisent au plus quatre classes d'appareil virtuel pour réaliser le traitement par lots.

Rien n'empêche, s'ils désirent élargir les possibilités de configuration pour leurs terminaux, d'inclure une nouvelle classe d'appareil tel que :

- lecteur de ruban perforé,
- disquette souple,
- cassette,
-

Il suffirait de définir :

- un appareil virtuel associé à la classe,
- le protocole correspondant

et de convenir

- du nom de porte à utiliser par les serveurs.

D'autre part, comme il a été déjà dit, les serveurs actuels, par simplicité de mise en oeuvre, déterminent l'identité d'un terminal par une adresse de Station de Transport. En conséquence, il n'est pas possible, à l'heure actuelle, d'utiliser un terminal géographiquement dispersé, c'est-à-dire émulé par la coopération de plusieurs postes d'accès banalisés (voir Fig. II.2).

De légères modifications, qui ne mettent pas en cause la philosophie générale du service réseau, sont envisageables pour adapter les serveurs afin qu'ils admettent la connexion de terminaux distribués.

Leur définition pourrait se faire statiquement ou dynamiquement lors de la phase de négociation.

L'intérêt et les avantages d'un terminal géographiquement distribué peuvent être multiples. Cela peut répondre à une politique de meilleure utilisation des périphériques par leur mise en commun, ou bien à des besoins temporaires, en cas de pannes.

Par exemple, on peut soumettre un travail à un serveur à partir d'un lecteur de cartes associé à un poste d'accès banalisé et récupérer les résultats par l'imprimante d'un autre, ou vice-versa.

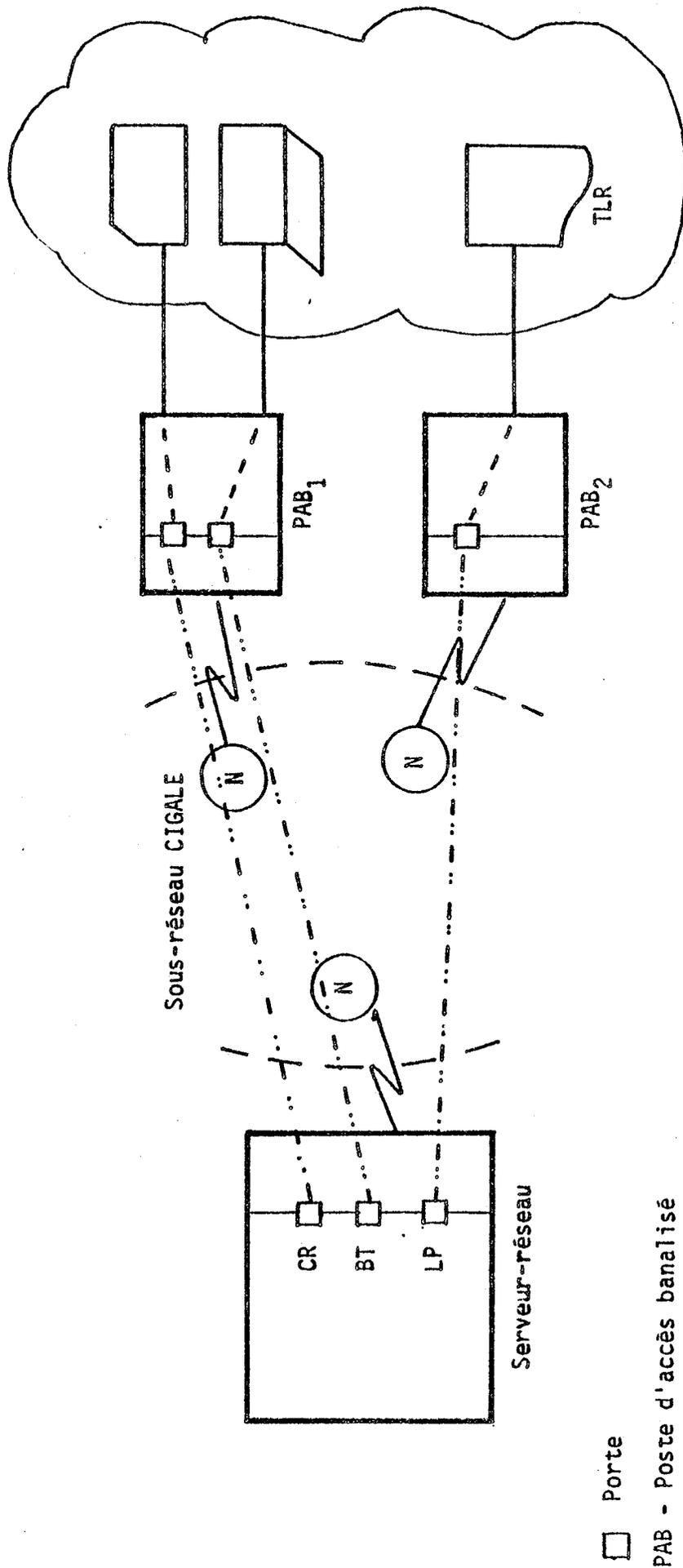


Figure II.2 - Exemple d'un TLR géographiquement distribué
émulé par deux postes d'accès banalisés

CHAPITRE III

DÉFINITION D'UN SERVEUR DE TRAITEMENT PAR LOTS POUR LE RÉSEAU CYCLADES

III.0. INTRODUCTION

Comme indiqué dans la présentation, nous avons réalisé l'adaptation de deux systèmes compatibles de télétraitements par lots HASP et JES [IBM a] s'exécutant respectivement sur des calculateurs IBM 360 et 370, afin d'offrir deux nouveaux serveurs CYCLADES.

Une étude préliminaire sur les avantages et les inconvénients des différentes solutions possibles a déterminé notre choix. Celui-ci s'est porté sur la mise en place, devant chaque nouvel hôte CYCLADES, d'un calculateur FRONTAL effectuant la *conversion d'interface*.

III.1. EXAMEN DES SOLUTIONS EXISTANTES

Les solutions étudiées, utilisées lors des réalisations précédentes, peuvent être groupées en trois classes suivant la méthode employée pour l'*adaptation* ou l'*ouverture* d'un système sur un réseau d'ordinateurs :

- Méthode d'adaptation interne :
Des modifications logicielles et/ou matérielles sont effectuées au niveau du nouvel hôte-réseau.
- Méthode d'adaptation externe :
L'adaptation est réalisée dans un processeur frontal.
- Méthode d'adaptation mixte :
C'est une combinaison des deux méthodes précédentes.

III.1.1. Méthode d'adaptation interne

a) Adaptation d'un système ASP/OS-MVT

Un des rôles essentiels du système ASP^(*) est d'effectuer la gestion des travaux parvenant au système OS-MVT^(**) par l'intermédiaire des *terminaux lourds* connectés à l'ordinateur hôte du système OS-MVT.

(*) ASP - Attached Support System.

(**) OS-MVT Operating System - Multiprogramming with a variable number of Tasks.

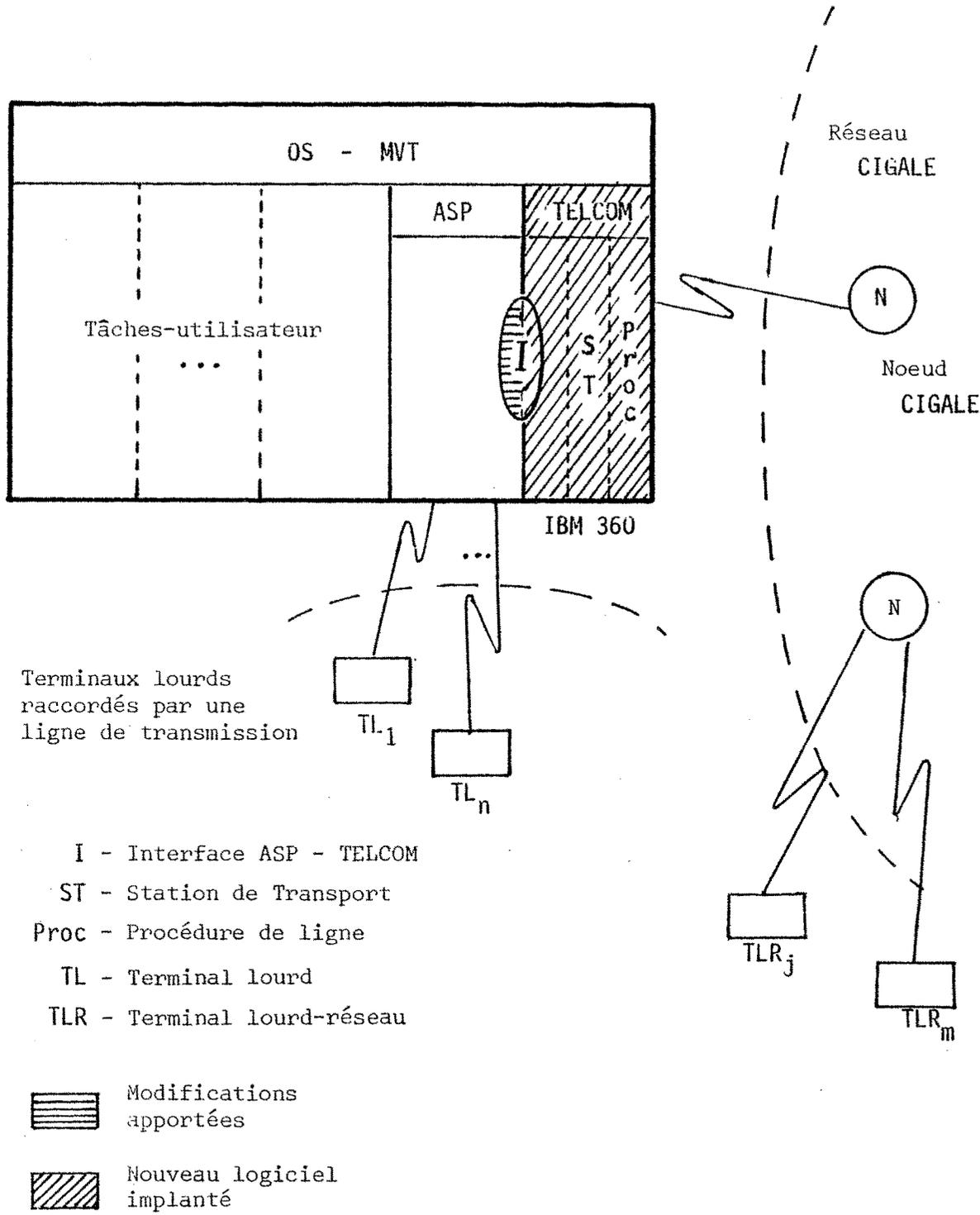


Figure III.1 - "Ouverture" d'un système ASP/OS-MVT au réseau CYCLADES

Le sous-système ASP est implanté comme une tâche utilisateur dans une région du système OS-MVT (voir Fig. III.1).

L'"ouverture" du système de télétraitement par lots ASP/OS-MVT sur le réseau CYCLADES a été effectuée [FOU76] par :

- la mise en place d'une nouvelle *tâche résidente* du système OS-MVT, constituée d'un sous-système de multiprogrammation, appelé TELCOM [PAP74] et des programmes nécessaires pour le raccordement au réseau (procédure de ligne, Station de Transport, ...) ;
- la modification de l'interface du sous-système ASP gérant les terminaux lourds afin que ce dernier puisse communiquer avec les processus du sous-système TELCOM et gérer ainsi les "terminaux lourds-réseau".

L'échange d'information entre les sous-systèmes ASP et TELCOM utilisait les mécanismes de communication fournis par le système OS-MVT pour les tâches-utilisateurs.

Permettant la connexion des terminaux lourds-réseau sans ajout de matériel supplémentaire (autre qu'une ligne de transmission entre l'ordinateur hôte et un noeud CIGALE), cette solution présente néanmoins de sérieux inconvénients tels que :

- la modification du sous-système ASP délivré par le constructeur ;
- la dégradation des performances du système OS-MVT par la présence permanente d'une tâche supplémentaire assez volumineuse (environ 24.800 octets).

b) Adaptation du système interactif CP/CMS

Le système CP s'exécutant sur un ordinateur IBM 360/67 est un générateur de machines virtuelles mettant à la disposition de chaque utilisateur une machine virtuelle simulant un ordinateur IBM 360.

Pour activer une machine virtuelle, un utilisateur doit accéder au système CP au moyen d'une console.

Le travail d'adaptation [ANS76] a été réalisé afin de permettre l'activation d'une machine virtuelle à partir d'une console connectée, via le réseau CYCLADES, à l'ordinateur IBM 360.

Pour activer une machine virtuelle, un utilisateur doit accéder au système CP, au moyen d'une console.

Le travail d'adaptation [ANS76] a été réalisé afin de permettre l'activation d'une machine virtuelle à partir d'une console connectée, via le réseau CYCLADES, à l'ordinateur IBM 360.

Le logiciel nécessaire au raccordement du nouvel hôte au réseau CYCLADES et à la gestion des machines virtuelles des utilisateurs-réseau a été logé dans une machine virtuelle (voir Fig. III.2).

Le mécanisme de communication entre deux machines virtuelles fourni par le système CP a été utilisé pour réaliser le dialogue entre la machine virtuelle-réseau et les différentes machines virtuelles activées par les utilisateurs éloignés.

Puisque le système CP possède un mécanisme de pagination, la mémoire ne constitue pas ici une ressource critique.

Dans ce cas, la source principale des difficultés rencontrées résidait dans le critère utilisé par le distributeur (*dispatcher*) du système CP, pour allouer l'unité centrale aux différentes machines virtuelles supportées.

Quand une machine virtuelle déroule beaucoup de code et épuise sa "tranche de temps", le distributeur CP pénalise cette machine en la classant "non interactive".

Par ailleurs, certains protocoles du réseau CYCLADES prévoient qu'une "connexion logique" n'est maintenue que lorsque l'intervalle entre deux échanges sur cette connexion est inférieur à un délai préétabli.

En conséquence, quand le système CP est surchargé (c'est-à-dire quand le nombre de machines virtuelles émulées simultanément dépasse un certain seuil), la tranche de temps réel allouée à la machine-virtuelle-réseau n'est pas suffisante pour garantir l'entretien correct de ces connexions.

Dans ces cas, par exemple les Stations de Transport distantes, gérant les "connexions" des consoles raccordées au système CP à travers le réseau, peuvent décider la libération de ces "connexions" en raison du long intervalle de silence détecté.

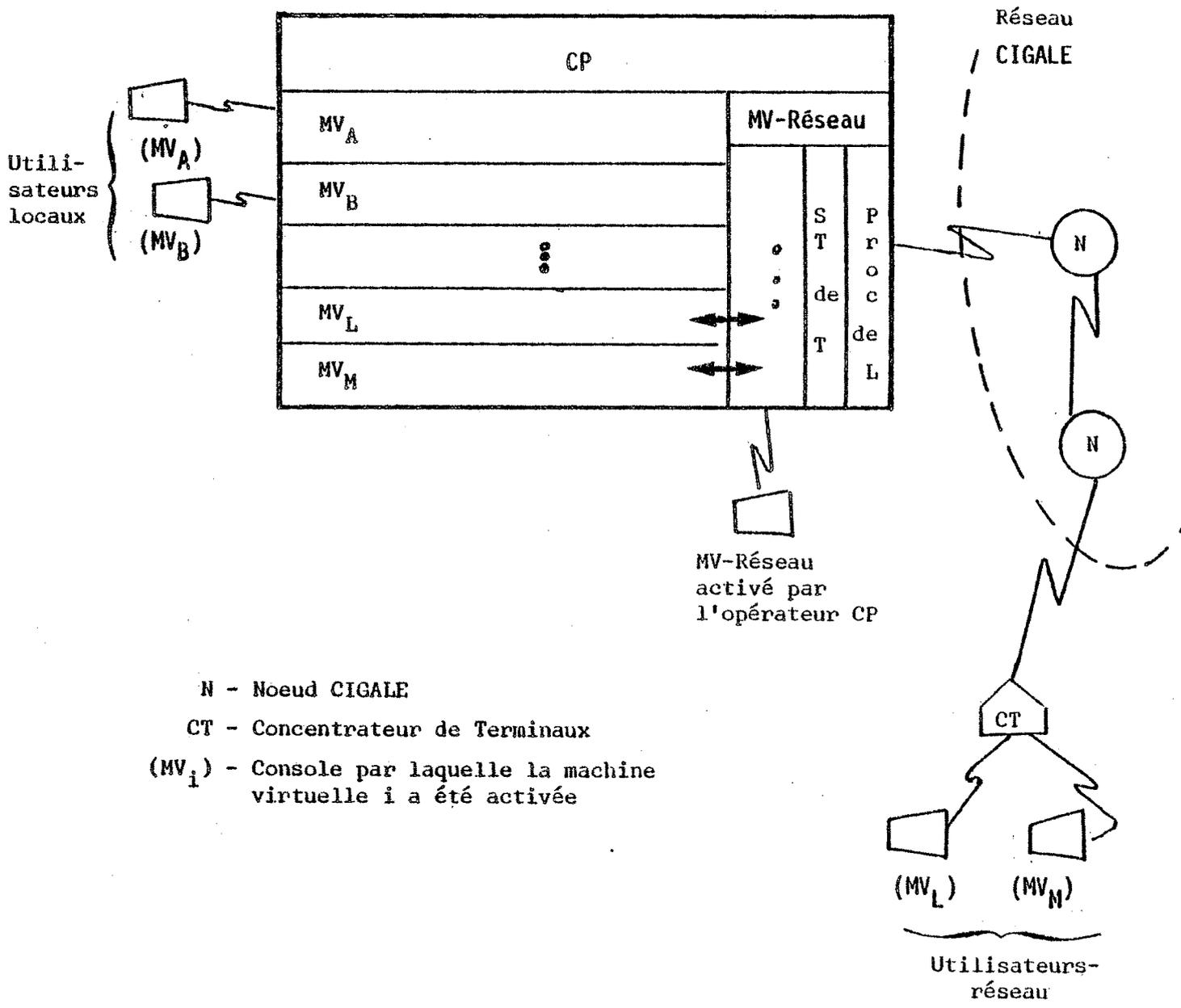


Figure III.2 - Schéma suivi pour l'adaptation du système CP au réseau CYCLADES

Pour résoudre ce problème, la modification du "distributeur" avait été envisagée, mais finalement rejetée, en raison des risques qu'elle présentait de perturber les performances du système, cette fois, vis-à-vis des utilisateurs locaux.

La solution adoptée a consisté à réduire le nombre de machines virtuelles émulables simultanément.

III.1.2. Méthode d'adaptation externe

a) Connexion d'un Centre de Calcul Réparti

Dans le cadre d'un Centre de Calcul réparti, réalisé à l'aide d'un réseau de mini-ordinateurs à fonctions spécialisées [GUI75], un "frontal" a été prévu pour réaliser l'ouverture vers l'extérieur [CHA76].

Les rôles assignés au frontal (voir Fig. III.3) sont principalement de :

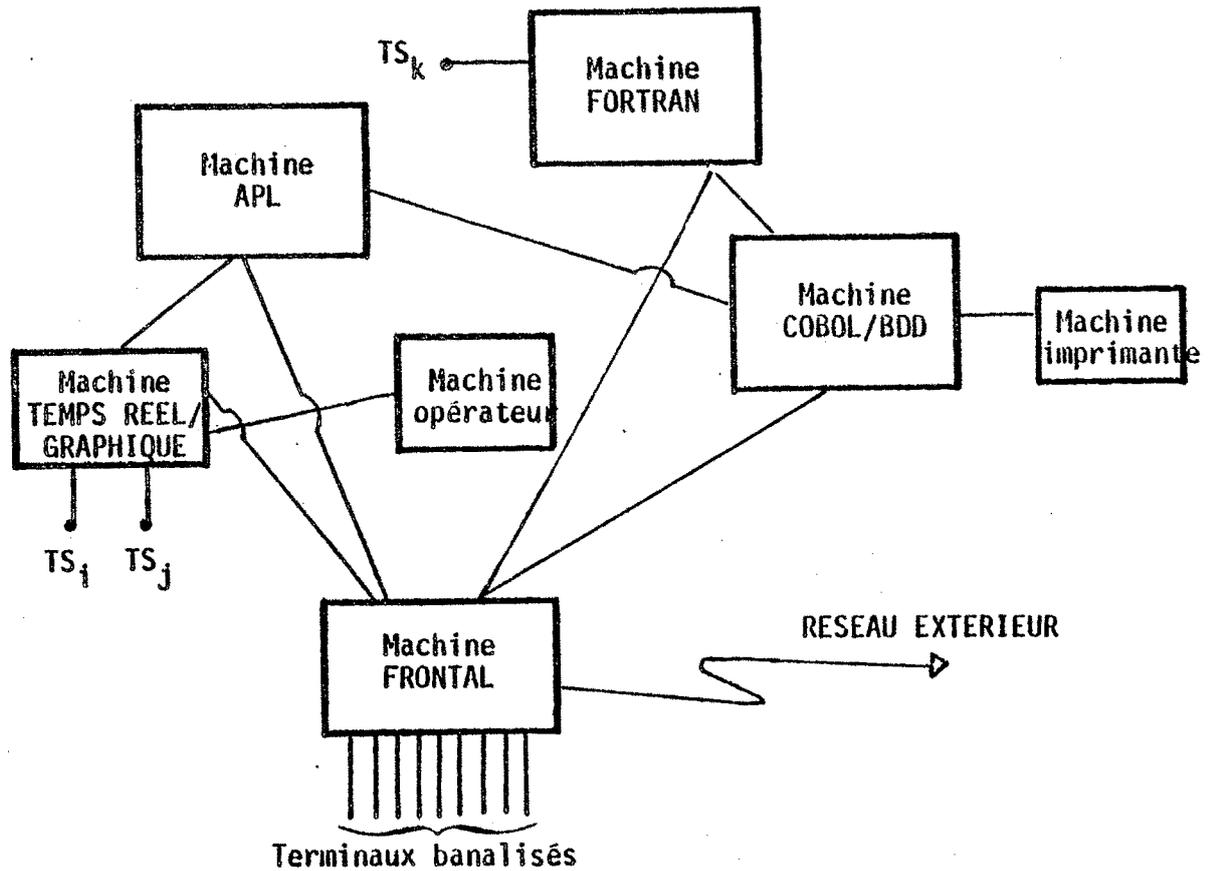
- réaliser la méthode d'accès aux différents "sites" du réseau local aussi bien pour les terminaux locaux du Centre que pour les terminaux distants connectés au travers d'un réseau de communication ;
- permettre aux terminaux locaux d'accéder les services d'un réseau extérieur.

Les avantages de cette solution sont multiples. En effet, elle permet de décharger les différents calculateurs du Centre de la gestion des terminaux. De plus, le regroupement de ce logiciel sur une seule machine facilite son développement et simplifie sa maintenance.

Néanmoins, si le FRONTAL n'est pas correctement dimensionné, l'activité simultanée de plusieurs lignes de transmission risque de saturer la capacité de traitement et entraîner des phénomènes de désynchronisation.

b) Adaptateurs câblés ou programmés

L'équipe CYCLADES de l'IRIA a été à l'initiative de la conception et de la réalisation de divers *prototypes d'adaptateur*, pour connecter au réseau CYCLADES tout type de terminal et de calculateur sans en modifier ni le matériel ni le logiciel [NAF77].



TS_i, TS_j, TS_k : terminaux spécialisés

Figure III.3 - Centre de Calcul réparti

La méthode générale proposée est l'interposition d'un *adaptateur câblé ou programmé* entre le calculateur ou le terminal à raccorder et le réseau, pour y émuler l'interface du nouvel hôte d'un côté, et la méthode d'accès au réseau de l'autre.

Cette *adaptation externe* est toujours réalisable dans la mesure où les services offerts par le réseau peuvent se traduire à travers l'interface externe du nouvel hôte.

L'intérêt de cette méthode réside dans le fait qu'elle évite toute modification logicielle et matérielle du nouvel hôte. En contre partie, elle nécessite l'utilisation de matériel supplémentaire (*adaptateur externe*) et dans la plupart des cas, les contrôles concernant la bonne transmission de l'information sont doublés. Par exemple, ils sont

effectués au niveau de la liaison entre le noeud CIGALE et l'adaptateur externe, et aussi au niveau de la liaison entre l'adaptateur externe et le nouvel hôte-réseau.

Dans la Figure III.5, les schémas de quelques réalisations faites selon la méthode d'adaptation décrite ci-dessus sont données en exemple.

III.1.3. Méthode d'adaptation mixte

Un périphérique intelligent

A l'époque où débutait notre étude, un projet de connexion d'un ordinateur IBM 360 au réseau CYCLADES à travers un calculateur d'architecture multiprocesseur était en cours de définition [SAE79].

Le calculateur hôte devait être raccordé au *multiprocesseur* par un canal d'entrée-sortie permettant de voir le réseau comme un périphérique, le *multiprocesseur* et le réseau étant eux-mêmes reliés par une ligne de transmission synchrone (voir Fig. III.4).

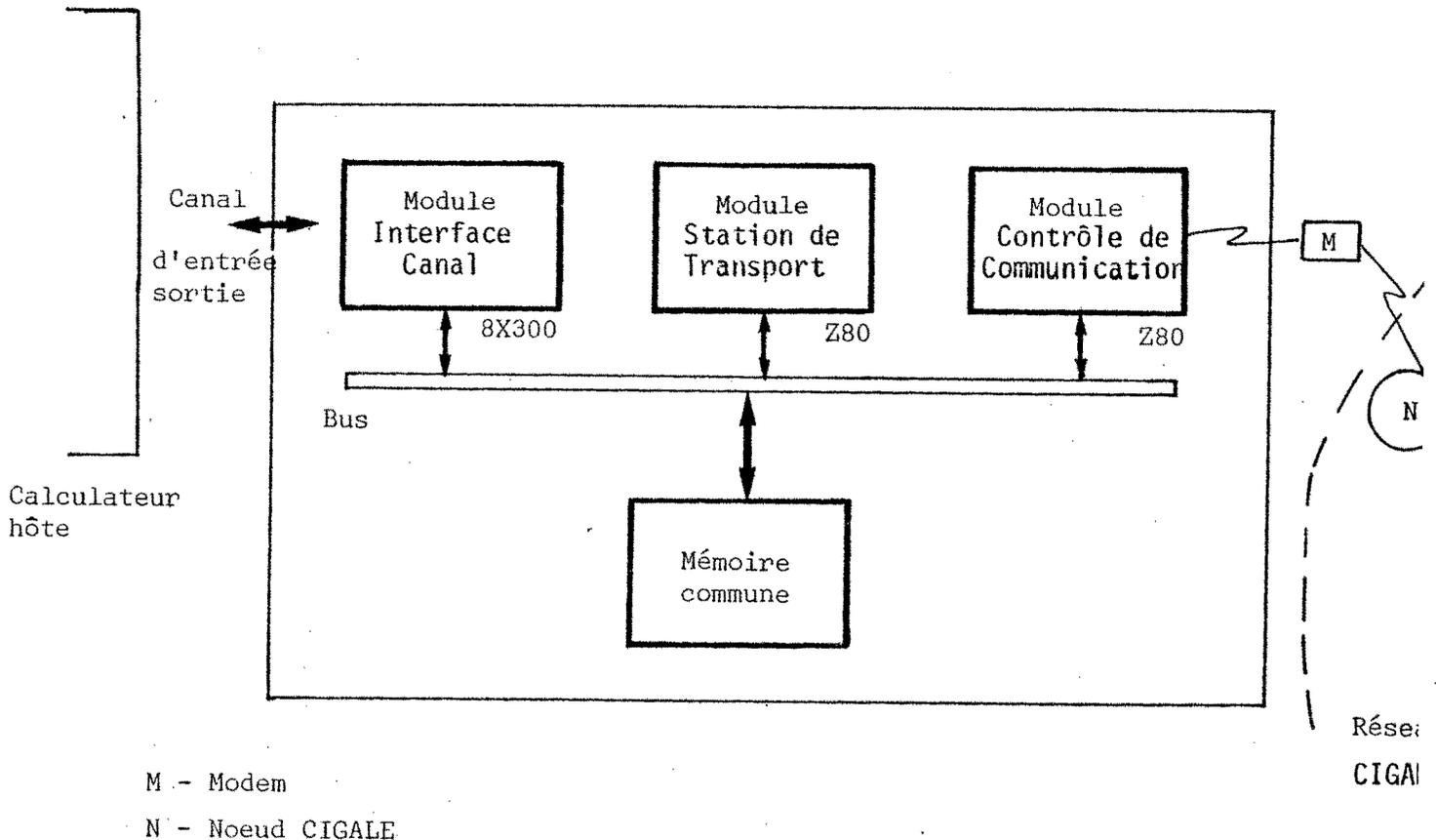


Figure III.4 - Architecture du "multiprocesseur"

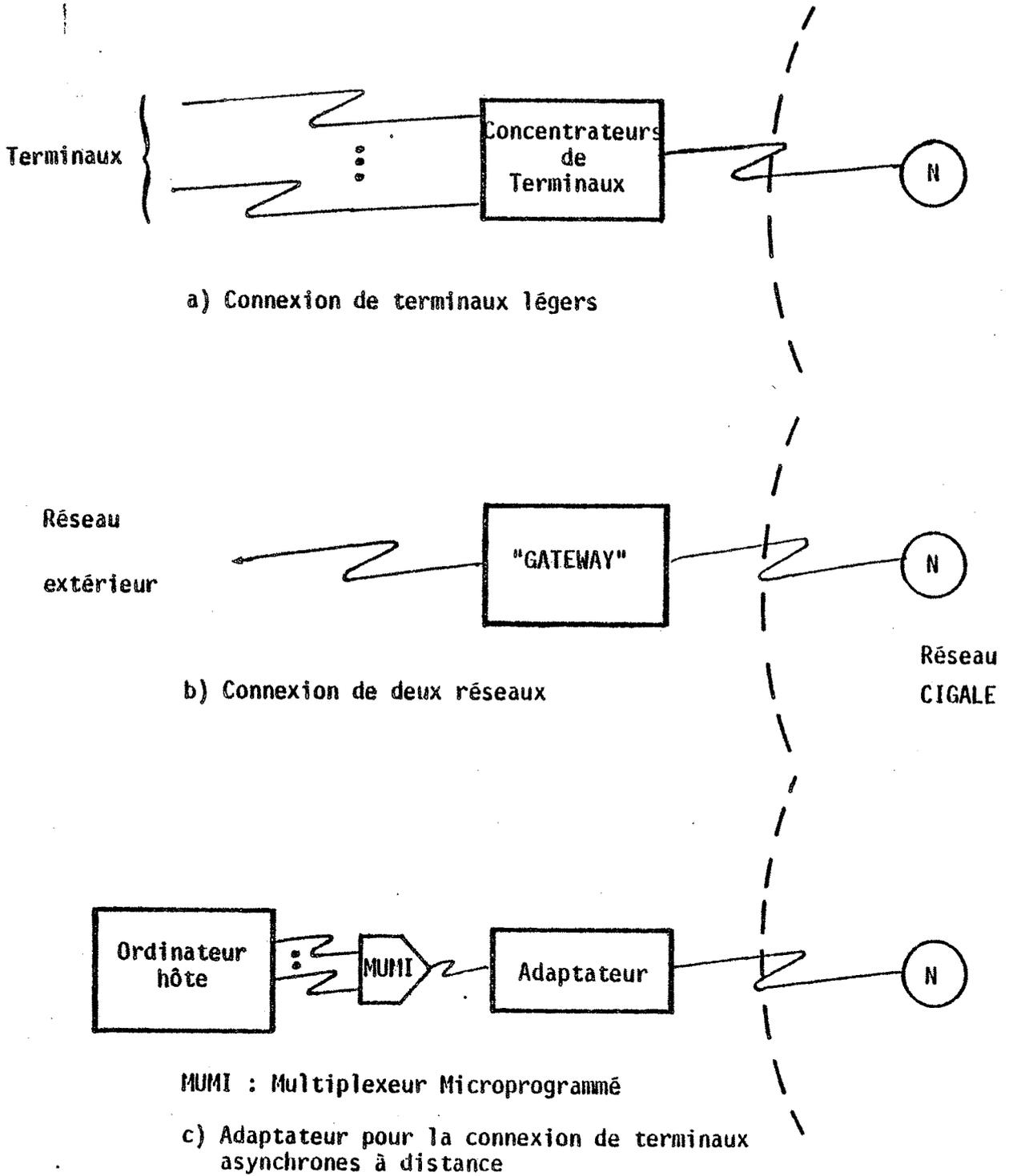


Figure III.5 - Exemples de connexions de nouveaux hôtes CYCLADES par un adaptateur externe

Le "multi-processeur" devait accomplir la plupart des fonctions requises pour l'accès au réseau, une faible partie (*Station de Transport résiduelle*) restant à la charge du calculateur hôte.

Bien que cette solution ne libère pas complètement l'ordinateur de tout traitement concernant sa connexion au réseau, la liaison à haute vitesse et très fiable qui le raccorde à un adaptateur externe réduit considérablement le temps nécessaire pour ses échanges avec le réseau.

III.2. LA SOLUTION ADOPTÉE : UN FRONTAL

A la lumière de l'expérience acquise par les réalisations précédentes, nous avons décidé de développer une solution présentant les caractéristiques suivantes :

- a - aucune modification n'est faite sur le système HASP lui-même.
- b - aucune dégradation des performances du système n'est induite par cette solution.

La première caractéristique répond non seulement au désir de faciliter la mise en oeuvre, en évitant toute modification du système délivré par le constructeur, mais aussi à celui de pouvoir bénéficier de l'évolution du système sans avoir besoin de reporter les modifications à chaque nouvelle version.

La deuxième est réalisée en excluant du calculateur hôte le logiciel nécessaire à l'adaptation afin d'éviter une utilisation concurrentielle des ressources (mémoire, unité centrale, ...) qui puisse perturber les performances du système HASP.

Donc nous avons opté pour la réalisation d'une adaptation externe à l'aide d'un calculateur *frontal* interposé entre le nouvel hôte et le réseau.

III.2.1. Fonctions du FRONTAL

Compte tenu de l'objectif initial de ne pas modifier le logiciel standard de l'ordinateur-serveur, le FRONTAL doit assurer les fonctions suivantes :

- 1) Emission et réception de l'information de et vers l'ordinateur du système serveur selon la procédure de ligne que ce dernier utilise de manière conventionnelle pour ses terminaux lourds.
- 2) Accès au réseau CYCLADES, c'est-à-dire gestion d'une "Station de Transport", d'une "Station de Communication" et d'une procédure de ligne ouvrant la communication entre le FRONTAL et un noeud CIGALE.
- 3) Identification d'un terminal lourd réseau en acceptant ou en refusant sa demande de connexion, puis présentation au système de télétraitement par lots comme un terminal lourd classique.
- 4) Conversion des messages reçus du système de télétraitement codés selon son protocole en messages du protocole d'appareil virtuel du réseau CYCLADES et vice-versa.

III.2.2. Options envisagées pour l'adaptation

Dans cette section, nous présentons certaines des options examinées lors de la réalisation du FRONTAL, dans le but d'utiliser au mieux la puissance de calcul et de stockage du processeur FRONTAL. Notre examen visait à définir une solution viable permettant de maximiser :

- 1 - le nombre de terminaux lourds HASP supportés simultanément par le FRONTAL.
- 2 - le nombre de terminaux lourds-réseau émulables par un seul terminal lourd HASP.

En raison de la compatibilité des systèmes HASP et JES, on citera dans le texte uniquement le premier.

- 1 * Rappelons qu'à travers une même ligne physique, il est possible de connecter successivement au système HASP différents terminaux lourds, car le système associe dynamiquement une ligne physique à un terminal lourd donné, en fonction d'un message de présentation (SIGNON) émis par ce dernier, lors de sa connexion.

Pour augmenter le nombre de terminaux lourds émulables simultanément, on a donc envisager de raccorder le FRONTAL au calculateur hôte du système HASP par plusieurs lignes de transmission^(*).

2 * Dans le système HASP, la configuration maximale d'un terminal lourd comporte :

- une console opérateur
- sept lecteurs de cartes
- sept imprimantes ou perforateurs de cartes.

De ce fait, on a été tenté de distribuer parmi sept terminaux lourds-réseau les différents couples de lecteur/imprimante d'un terminal lourd HASP et de leur faire partager l'unique console opérateur (voir Fig. III.6), c'est-à-dire de présenter au système HASP sept terminaux lourds-réseau, comme un seul terminal lourd de configuration maximale.

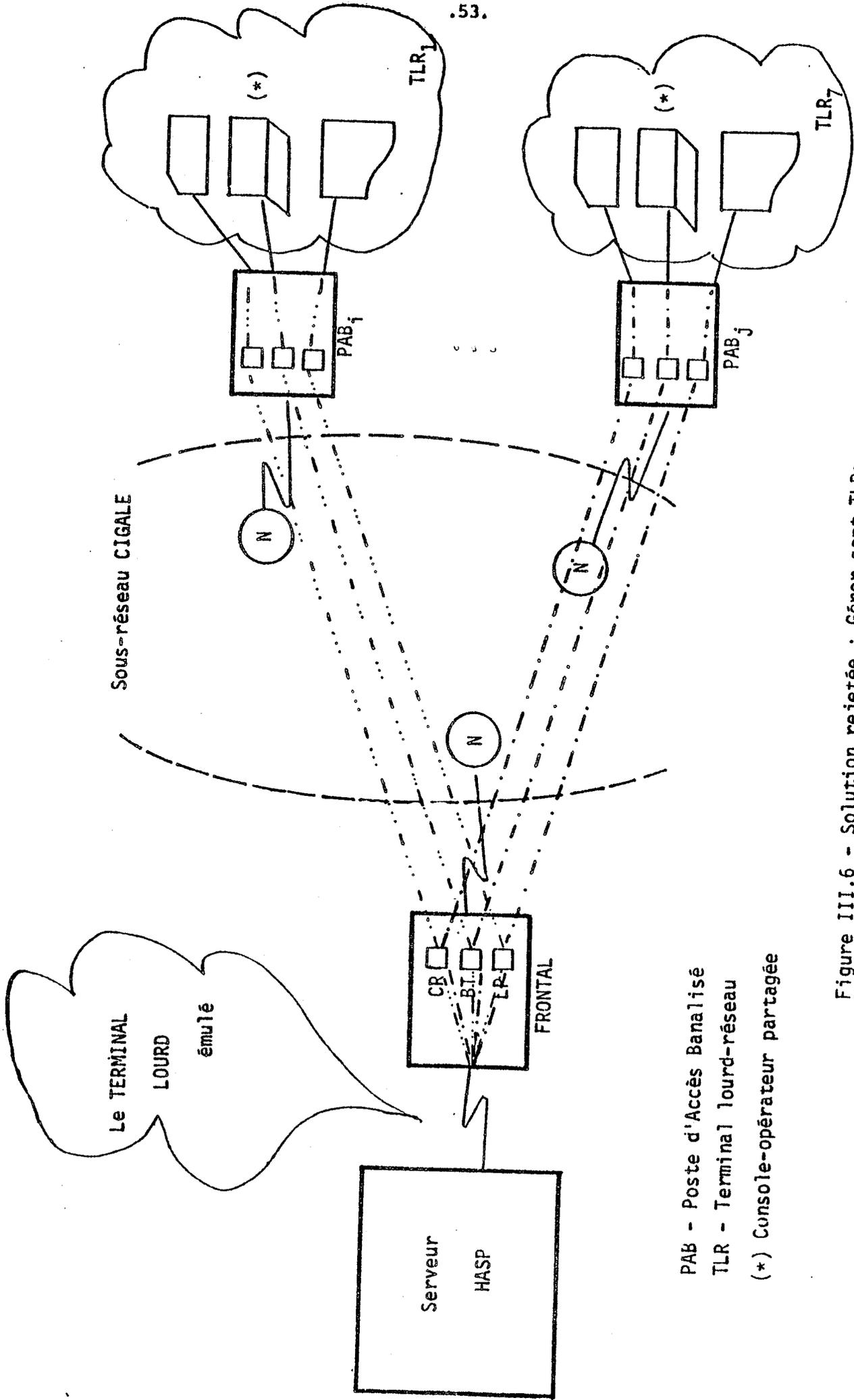
Inconvénients :

Mais la proposition de partage de la console présente des inconvénients. En effet, à partir de la console d'un terminal lourd, on peut agir et interroger le système HASP sur l'ensemble de travaux soumis à partir d'un lecteur quelconque du terminal.

De plus, des commandes émises à partir d'une console permettent de modifier la priorité, d'annuler ou de suspendre l'exécution d'un travail etc.

En conséquence, le partage d'une console opérateur entre sept terminaux lourds-réseau peut être à l'origine d'erreurs ou d'actions malveillantes. Si plusieurs travaux rentrés à partir d'un terminal ont le même nom, une commande opérateur pour annuler l'exécution d'un travail de ce nom portera sur le premier soumis au système. Supposons que des travaux ayant des noms identiques soient rentrés à partir des différents terminaux lourds-réseau émulés par le même terminal lourd, un ordre pour annuler l'un d'entre eux peut ne pas porter sur le travail effectivement visé.

(*) Cette solution a été adoptée pour le FRONTAL du système JES mis en oeuvre au CIRCE.



PAB - Poste d'Accès Banalisé
 TLR - Terminal lourd-réseau
 (*) Console-opérateur partagée

Figure III.6 - Solution rejetée : Gérer sept TLRs émulant un seul TL-HASP

On a étudié aussi la possibilité de fournir à chaque terminal lourd-réseau une copie de la console du Terminal HASP avec des droits réduits. Mais il est nécessaire pour filtrer les consultations et aiguiller les réponses de modifier le système HASP ou de réaliser au niveau du FRONTAL un traitement complexe d'analyse et d'enregistrement de l'information qui nécessiterait obligatoirement des supports de mémoire secondaire.

La proposition d'éliminer la console est évidemment sans intérêt.

Les inconvénients précédents ne sont pas les seuls, car le système HASP retourne les résultats des travaux soumis par un terminal à l'une quelconque de ses imprimantes, sauf si l'on explicite le contraire pour un travail particulier.

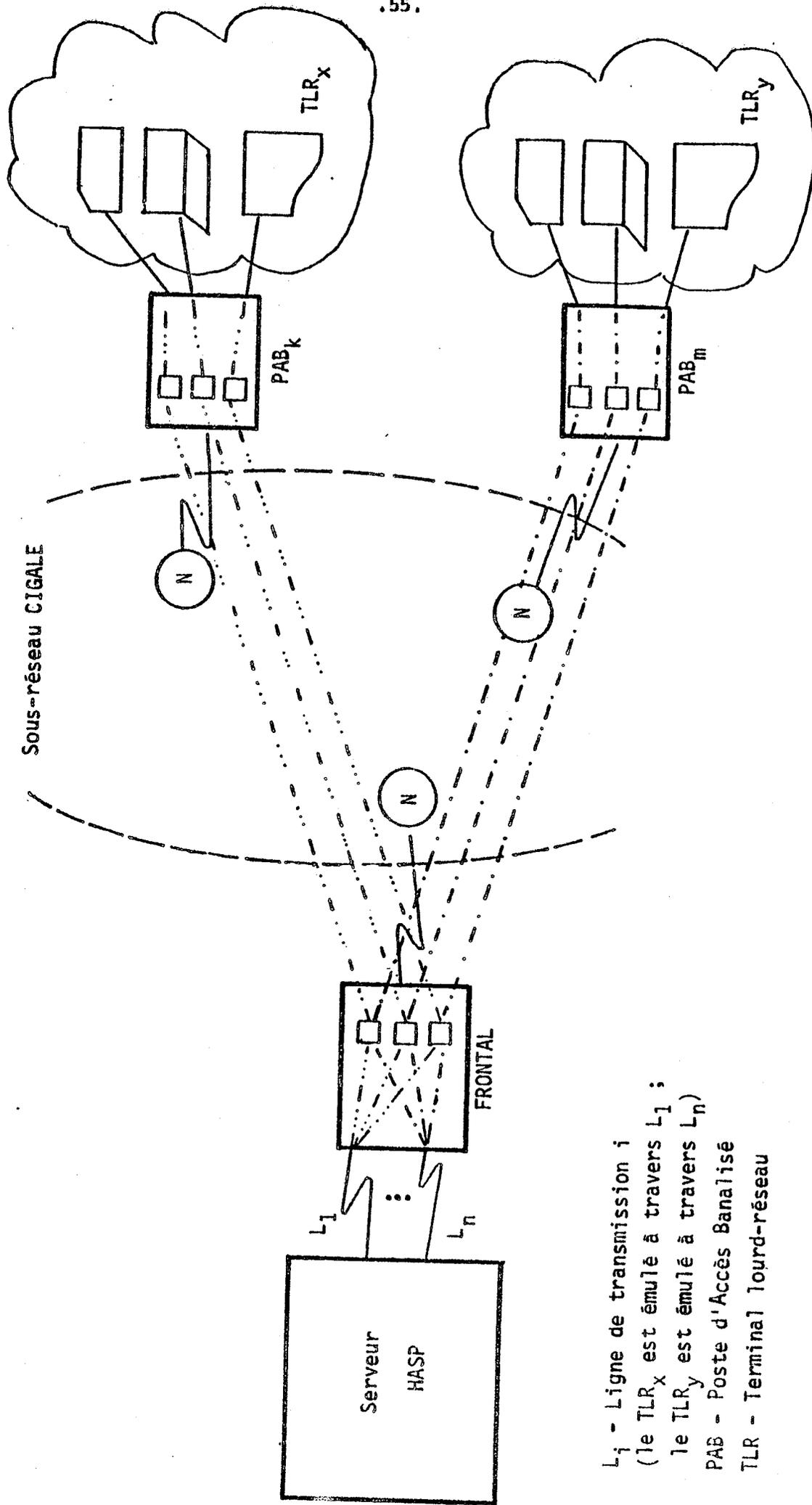
Pour forcer l'envoi d'une liste sur une imprimante donnée, il faut inclure une commande spéciale parmi les cartes de contrôle du travail correspondant.

On a donc décidé d'abandonner cette solution car le partage de la console présentait des risques et l'emploi de l'imprimante imposait des contraintes aux utilisateurs.

Solution adoptée :

On a préféré de gérer un terminal lourd-réseau pour chaque terminal émulé. En conséquence, on doit déclarer dans le système HASP autant de nouveaux terminaux que de terminaux lourds-réseau autorisés à utiliser ses services.

La solution adoptée permet la connexion d'un seul terminal lourd-réseau à la fois, par ligne de transmission (LT) reliant le FRONTAL à l'ordinateur (voir Fig. III.7), mais elle élimine toutes les restrictions introduites par la solution précédente.



L_i - Ligne de transmission i
 (le TLR_x est émulé à travers L_1 ;
 le TLR_y est émulé à travers L_n)
 PAB - Poste d'Accès Banalisé
 TLR - Terminal lourd-réseau

Figure III.7 - Solution adoptée : un terminal lourd-réseau est émulé par un terminal lourd HASP

III.2.3. Configuration et mode de connexion d'un terminal lourd-réseau

Une même configuration a été retenue pour les différents terminaux lourds-réseau, car elle semblait satisfaire les besoins des clients potentiels et contribuait à simplifier la mise en oeuvre du FRONTAL.

Chaque terminal peut être constitué au plus par une console opérateur, un lecteur de cartes et une imprimante.

La connexion d'un terminal au serveur-réseau HASP se matérialise par la connexion de sa console. L'ordre de connexion des deux autres appareils est sans importance et facultative.

La déconnexion de la console-opérateur entraîne la déconnexion automatique de tous les appareils connectés du terminal.

III.2.4. Architecture du FRONTAL

L'architecture proposée peut être schématisée en cinq couches (voir Fig. III.8). Elle traite successivement l'information selon son sens de transmission (serveur-réseau ou réseau-serveur).

Première couche : *Procédure de ligne TMM-UC*

Elle gère la liaison synchrone entre le FRONTAL et un noeud CIGALE, selon la procédure de ligne TMM-UC "full duplex".

Deuxième couche : *Service de Communication*

Elle réalise l'interface d'accès au réseau de commutation de paquets CIGALE.

Troisième couche : *Service de Transport*

Elle est chargée de réaliser une "Station de Transport" selon le protocole du réseau CYCLADES.

Quatrième couche : *Conversion de protocoles*

Les fonctions essentielles concernent la connexion, la déconnexion des terminaux et la transcription des messages d'un protocole dans un autre.

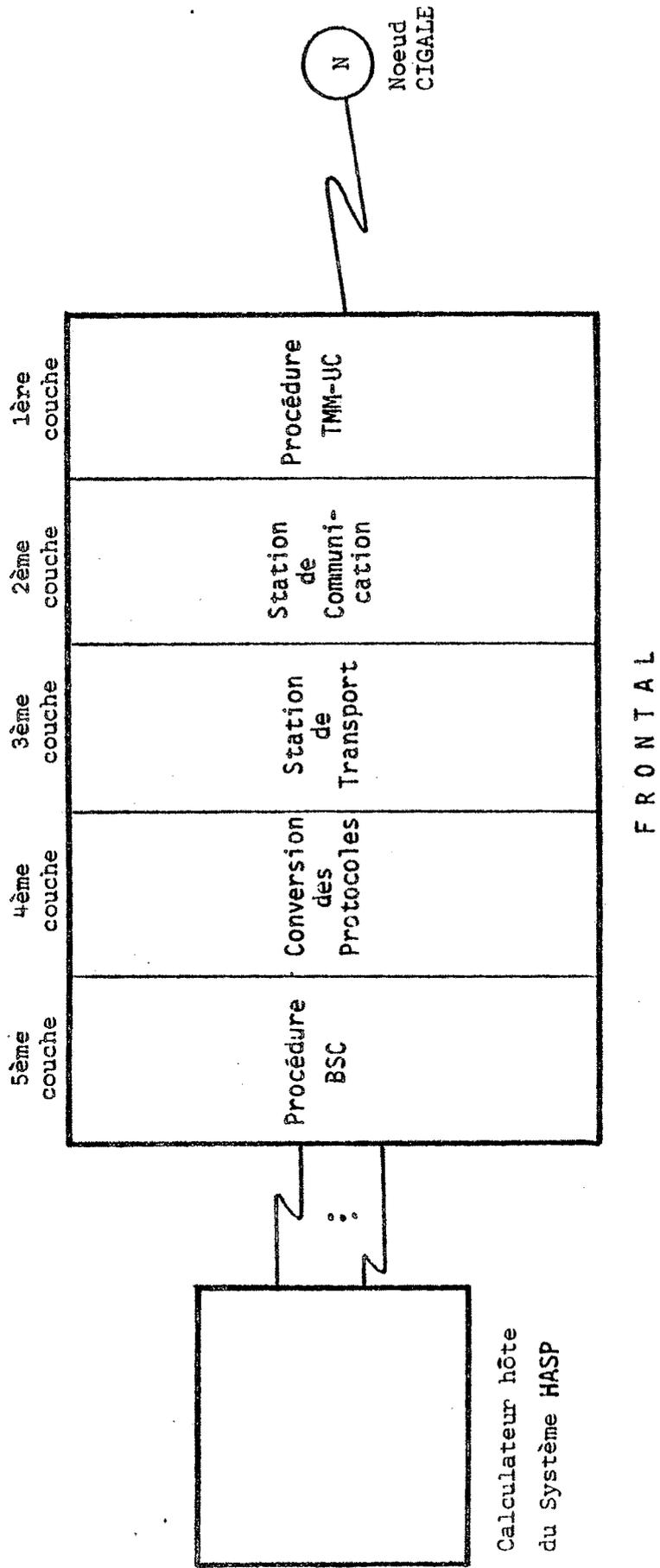


Figure III.8 - Architecture fonctionnelle du FRONTAL

Cinquième couche : Procédure de ligne BSC

Elle gère la liaison synchrone point-à-point entre l'ordinateur-serveur et le FRONTAL selon la procédure BSC "half duplex" [IBM b].

III.2.5. La conversion des protocoles

Afin de pouvoir décrire la méthode suivie pour la traduction des messages d'un protocole vers l'autre, on présentera sommairement le protocole MULTI-LEAVING utilisé par le système HASP pour dialoguer avec ses terminaux.

Le système HASP et un terminal lourd échangent des *blocs* à travers une ligne de transmission. Ces *blocs* contiennent des *enregistrements* dont chacun peut concerner un appareil périphérique différent.

Une liaison logique, désignée par *flot* ("stream") est établie implicitement entre le système HASP et un terminal lourd, pour chaque périphérique géré.

Lors de la réception d'un *bloc*, les *enregistrements* qu'il contient sont démultiplexés selon l'appareil adressé, sur les différents *flots*.

Un *enregistrement* peut contenir de l'information de contrôle (surprendre les transferts, ...) et/ou de l'information utile.

Les données rangées dans un *enregistrement* correspondent à l'information contenue dans une carte ou affichable sur une ligne et elles sont structurées en *chaînes*.

Les *chaînes* dans le protocole MULTI-LEAVING ont des fonctions analogues à celles des *items* dans le "protocole d'appareil virtuel" (PAV) du réseau CYCLADES (cf. II.5.2).

Les fonctions qui peuvent être décrites par l'ensemble des *items* sont entièrement recouvertes par celles de l'ensemble de "chaînes" et vice-versa.

Les conversions se réalisent donc selon les critères de correspondance décrits dans le schéma suivant :

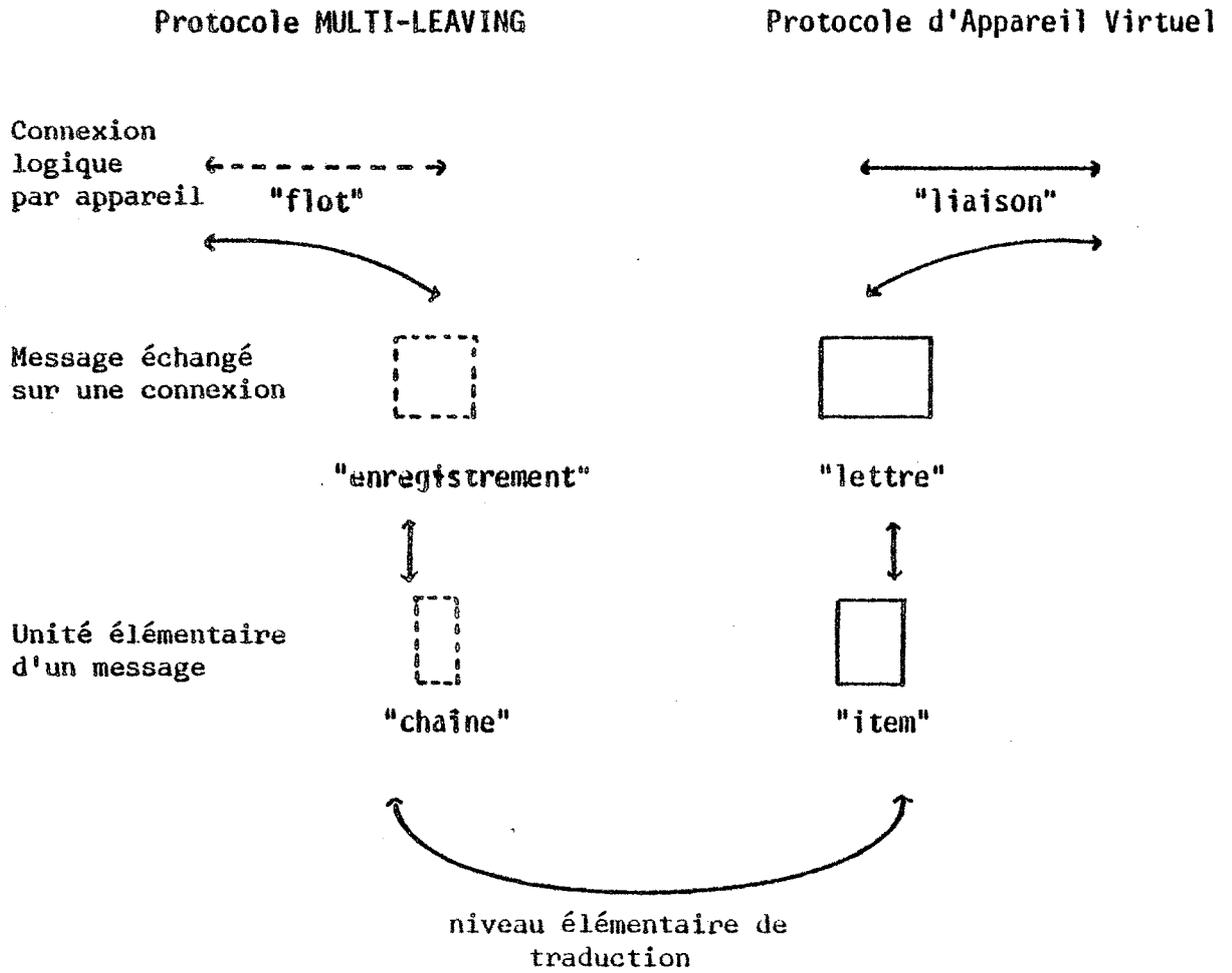


Figure III.9 - Schéma suivi pour la traduction des protocoles PAV et MULTI-LEAVING

Le protocole MULTI-LEAVING prévoit des mécanismes pour réguler le débit

- soit sur l'ensemble des flots correspondant aux différents appareils périphériques d'un terminal lourd ;
- soit sur chaque flot, considéré individuellement.

Quand le système HASP ou le terminal désirent suspendre la réception de données sur l'ensemble des flots, ils préviennent leur partenaire en insérant un message de contrôle adéquat dans l'en-tête d'un bloc MULTI-LEAVING.

Si, par contre, ils désirent inhiber la réception de données sur un flot particulier, un enregistrement contenant le message de contrôle pertinent est émis au partenaire sur le flot concerné.

Ces mécanismes possèdent un rôle analogue au *contrôle de flux* défini dans le Service de Transport CYCLADES.

III.3. DEFINITION DU SERVEUR-RESEAU HASP SELON LE MODELE DE REFERENCE ISO

La couche qui caractérise le mieux les fonctions d'adaptation attribuées au *FRONTAL* est celle où se réalise la conversion de protocoles.

On va restreindre la notion d'*application*, définie dans le modèle de référence ISO (cf. I.2) aux fonctions et aux services réalisés par cette couche.

Rappelons que dans le modèle cité, les trois niveaux proposés pour la description d'une application sont les *Niveaux Session, Présentation et Application* dont les fonctions respectives ont été sommairement décrites lors de la présentation du modèle.

Pour préciser la place attribuée à ces niveaux dans l'architecture définie pour le *FRONTAL*, nous dirons que le Niveau Session réalise l'interface avec la Station de Transport et le Niveau Application, celle de la procédure de ligne (voir Fig. III.10).

Notons que, contrairement à ce qui a été établi pour les Niveau appartenant aux deux premières catégories du modèle (destinées respectivement à l'interconnexion physique des hôtes et à la transmission de données), même des niveaux non adjacents effectuant le traitement de données peuvent

avoir une interface commune. C'est le cas du Niveau Application et du Niveau Session qui échangent à travers leur interface des messages de contrôle. Par contre, les données émises par le serveur HASP à destination des terminaux lourds-réseau ou vice-versa, traversent les trois niveaux.

Dans l'Annexe A, nous décrivons en détail les fonctions réservées à chacun de ces trois niveaux. Ici, nous n'en fournissons qu'un résumé.

- *Niveau Session* : ce niveau réalise les contrôles concernant le dialogue entre le serveur HASP et les terminaux lourds-réseau.
- *Niveau Présentation* : dans ce niveau, les données échangées entre le serveur HASP et les terminaux lourds-réseau sont transcodées du protocole de l'émetteur au protocole du récepteur.
- *Niveau Application* : ce niveau traite essentiellement les demandes de connexion et déconnexion effectuées par les terminaux lourds-réseau, et il les transmet au système HASP.

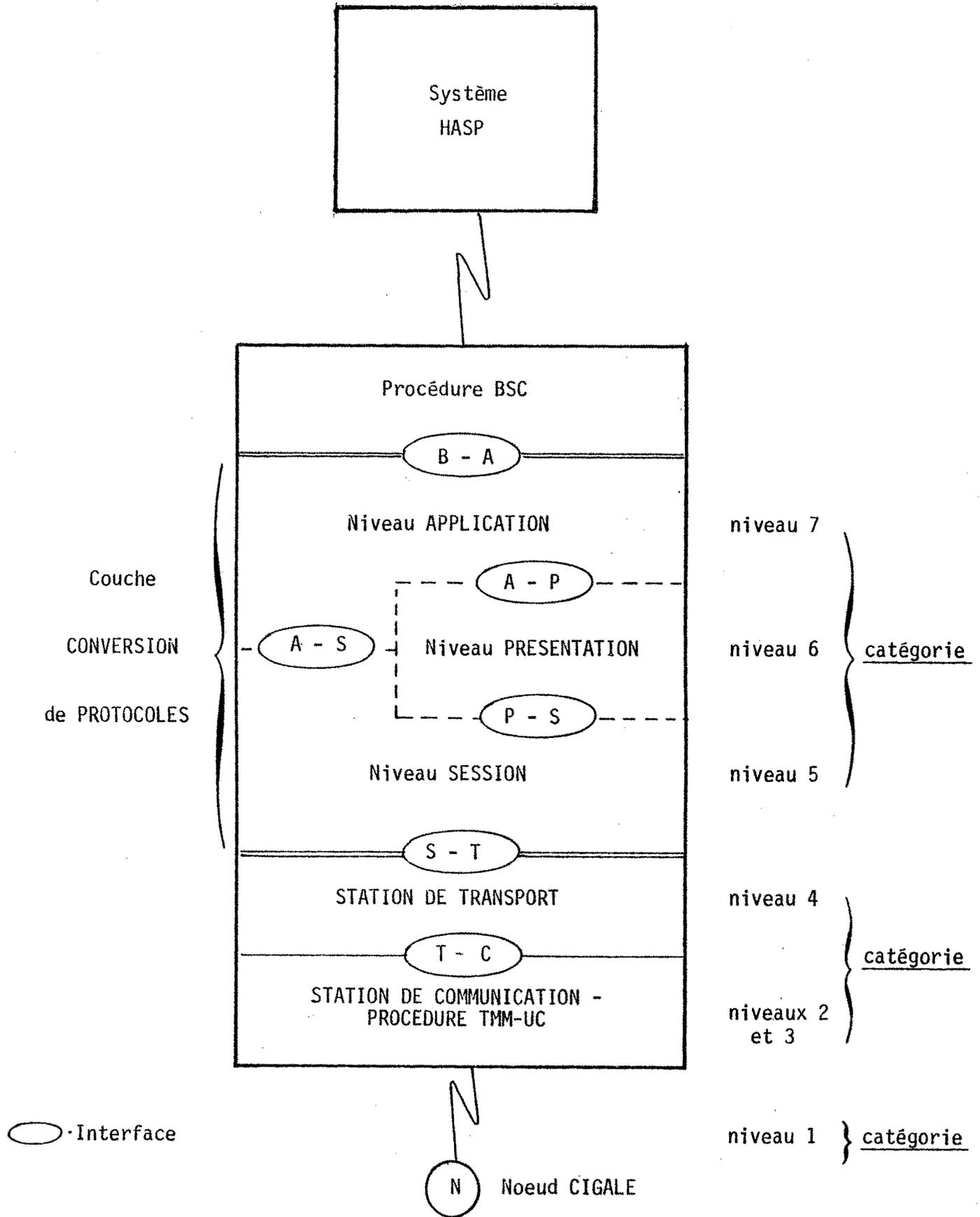


Figure III.10 - Correspondance entre la couche "Conversion de Protocoles" du FRONTAL et les niveaux du modèle de référence ISO

CHAPITRE IV

DÉFINITION D'UN POSTE D'ACCÈS BANALISÉ

IV.0. INTRODUCTION

La définition et la réalisation d'un Poste d'Accès Banalisé (PAB) au CIGG de Grenoble et au CIRCE d'Orsay constituent une suite logique du projet d'adaptation au réseau de leur système de télétraitement, décrit dans le chapitre précédent.

La mise en place de ces nouveaux serveurs élargit les possibilités de choix des utilisateurs-réseau éloignés de ces Centres. Dans le même esprit, on a voulu fournir aux utilisateurs locaux le moyen d'accéder à travers le réseau à d'autres systèmes connectés.

IV.1. CHOIX D'UNE SOLUTION

Chaque Centre de Calcul, concerné par le projet, disposait d'un mini-calculateur jouant le rôle de terminal lourd vis-à-vis du système à mettre en place sur le réseau.

En particulier au CIGG (Grenoble), le même mini-calculateur pouvait devenir alternativement terminal lourd du système HASP (IBM 360) ou du système SIRIS8 (IRIS 80) selon le logiciel actif sur le mini-calculateur.

Ce mini-calculateur (ORDO 400) était relié à l'ordinateur IBM 360 par une ligne spécialisée via un modem et à l'IRIS 80 par une ligne directe.

Le souci d'utiliser au mieux le matériel disponible (mini-ordinateur, périphériques, lignes de transmission, ...) nous a conduit à choisir le mini-calculateur ayant le rôle de terminal lourd, pour réaliser à la fois le **FRONTAL** du système HASP et le **POSTE D'ACCES BANALISE**.

Cette décision ne pénalise en rien les utilisateurs. Tout au contraire puisqu'ils allaient disposer d'un moyen plus souple pour accéder à différents systèmes du réseau et notamment à leurs serveurs habituels (IBM 360 et IRIS 80^(*)).

De plus, le logiciel à développer pour l'accès réseau pouvait être mis en commun pour les deux réalisations.

(*) Le système SIRIS8 (IRIS 80) peut être accessible par des liaisons directes aussi bien qu'au travers du réseau CYCLADES.

Donc les bénéfices de cette solution étaient multiples et concernaient aussi bien l'utilisation du matériel, l'équipe logiciel que les utilisateurs.

IV.2. CONFIGURATION DISPONIBLE

La configuration du terminal lourd qu'on allait remplacer par un poste d'accès banalisé comportait une console opérateur, un lecteur de cartes et une imprimante, et éventuellement un lecteur de cartes et une imprimante supplémentaires.

Rappelons que la plupart des serveurs de traitement par lots mis en place sur le réseau imposent la connexion d'une console opérateur pour débiter une séance de travail avec un terminal lourd-réseau et que, de plus, les échanges sur ce périphérique se caractérisent par de longs intervalles de silence. En conséquence, nous avons décidé d'émuler deux consoles virtuelles sur le même appareil réel et de permettre à l'opérateur du POSTE D'ACCES de configurer à sa guise un terminal en connectant individuellement chaque appareil virtuel. De cette façon, nous sommes en mesure d'émuler simultanément, si nécessaire, deux terminaux lourds-réseau vis-à-vis de deux serveurs différents. Par exemple, l'opérateur peut connecter une console virtuelle et un lecteur de cartes à un serveur donné et l'autre console virtuelle et l'imprimante à un deuxième serveur.

Pour résoudre le problème du partage de la console réelle entre deux consoles virtuelles, le clavier est attaché à tour de rôle à chaque console virtuelle, les messages affichés sur l'écran étant préfixés par le nom du serveur émetteur.

Par contre, à chaque lecteur de cartes et imprimante, on associe un seul appareil virtuel, car on ne dispose pas de mémoire auxiliaire, pour stocker temporairement les paquets de cartes à émettre sur le réseau ou les listes éventuellement envoyés en parallèle par différents serveurs.

IV.3. COMMANDES OPERATEUR

On a défini un jeu de commandes pour permettre le dialogue entre l'opérateur et le POSTE D'ACCES [QUI78a].

Ces commandes servent à demander la connexion ou la déconnexion d'un appareil virtuel à un serveur donné à attacher le clavier à une console virtuelle, ainsi qu'à interroger le POSTE D'ACCES sur l'état des diverses connexions.

La même console réelle sur laquelle sont émulées les consoles virtuelles attachées au terminaux lourds-réseau, est utilisée par l'opérateur pour rentrer ses commandes et par le POSTE D'ACCES pour afficher les messages de contrôle.

IV.4. ARCHITECTURE DU POSTE D'ACCES BANALISE

L'architecture définie pour le POSTE D'ACCES BANALISE est structurée en quatre couches dont les trois premières sont communes à l'architecture du FRONTAL (voir Fig. IV.1).

1ère couche : *Procédure de ligne TMM-UC*

2ème couche : *Service de Communication*

3ème couche : *Service de Transport*

4ème couche : *Application.*

La fonction de cette dernière couche est de réaliser le moniteur du POSTE D'ACCES, c'est-à-dire d'interpréter les commandes-opérateur, d'administrer les ressources nécessaires pour l'émulation des différents terminaux lourds-réseau, de gérer les périphériques et d'effectuer la conversion des protocoles d'appareils virtuels dans ceux des appareils réels et inversement.

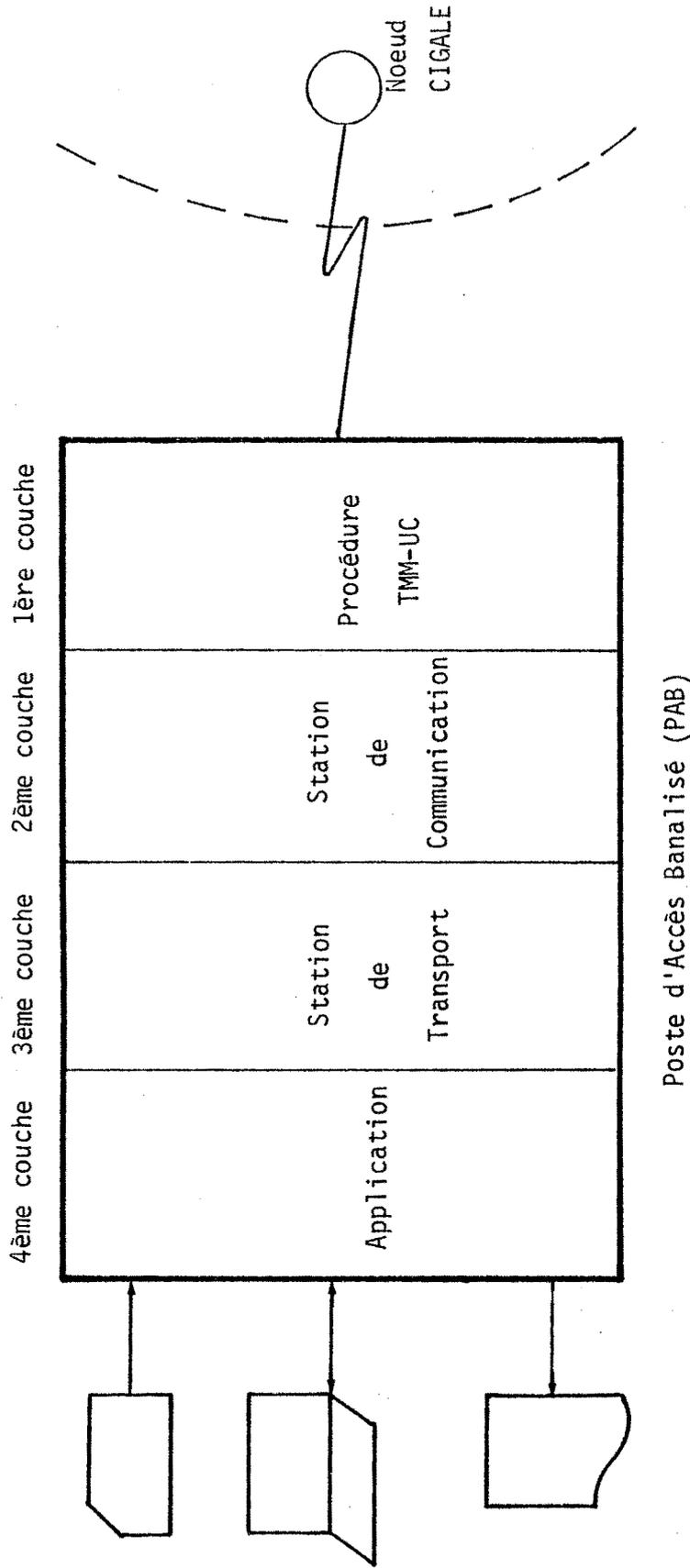


Figure IV.1 - Architecture fonctionnelle d'un PAB

IV.5. LA COUCHE APPLICATION SELON LE MODELE DE REFERENCE ISO

Ainsi comme l'on a fait pour la couche "Conversion de Protocoles" du **FRONTAL**, nous décrivons les fonctions assurées par la couche "Application" du **POSTE D'ACCES** (voir Fig. IV.2).

Les fonctions de chacun de ces trois niveaux sont développées en détail dans l'Annexe B. Voici un résumé :

- Niveau *Application* : ce niveau est chargé de gérer le dialogue entre l'opérateur et le moniteur du **POSTE D'ACCES**. Il assure aussi la gestion des périphériques réels et l'émulation des appareils virtuels associés.
- Niveau *Présentation* : dans ce niveau, les données rentrées à partir du lecteur de cartes et du clavier sont traduites dans le protocole d'appareil virtuel du réseau **CYCLADES** et celles destinées à être affichées sur l'écran ou l'imprimante sont transcrites aux conventions des appareils réels respectifs.
- Niveau *Session* : ce niveau met en oeuvre la partie du protocole du service réseau de traitement par lots relative aux clients (ex : initier les négociations pour l'établissement d'une liaison).

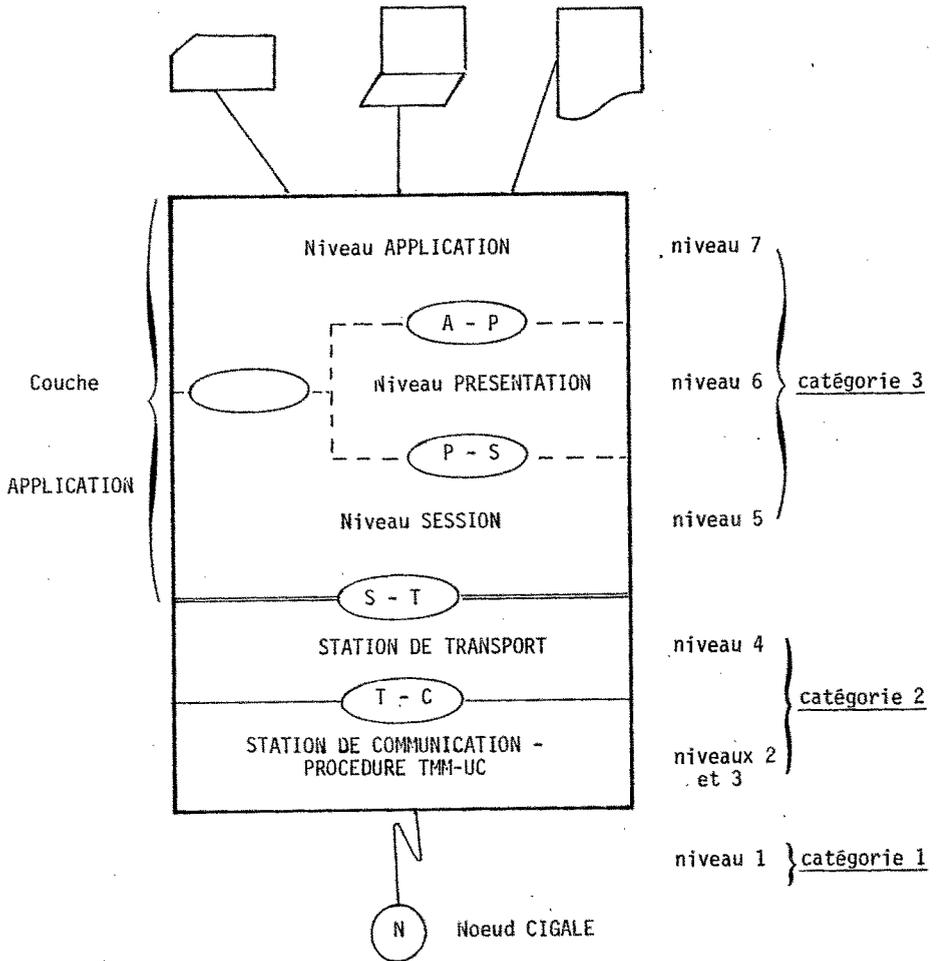


Figure IV.2 - Correspondance entre la couche "Application" du POSTE D'ACCES BANALISE et les niveaux du modèle de référence ISO

PARTIE II

DIFFERENTS ELEMENTS DE LA REALISATION

CHAPITRE I

ÉTUDE DE LA MISE EN OEUVRE

I.0. INTRODUCTION

Nous avons décidé de centrer le présent chapitre sur les aspects de la mise en oeuvre qui découlent des propriétés du **FRONTAL** et du **POSTE D'ACCES** ainsi que du matériel choisi et du logiciel utilisé pour le développement.

Nous analysons en premier les particularités du **FRONTAL** et du **POSTE D'ACCES** en termes de *parallélisme, objets manipulés, interfaces, ...*, indépendamment d'une implémentation donnée.

Nous présentons par la suite les caractéristiques du mini-calculateur et du moniteur utilisés pour la réalisation d'une première version de ces applications au CICG (Grenoble).

Enfin, nous exposons les idées directrices qui ont guidé le choix de certaines techniques de réalisation, les critères retenus pour le découpage modulaire et pour l'articulation des modules. Pour les détails techniques du développement, nous renvoyons le lecteur aux références bibliographiques [LUM78], [QUI78b].

Nous nous sommes inspirés de plusieurs travaux consacrés à l'étude de réalisations système [KRA73], ..., [DAN76] aux problèmes liés à la conception [CRO75], [MOS77], ainsi qu'à la modélisation [SIF79].

I.1. CARACTERISTIQUES DE L'APPLICATION

Puisque les caractéristiques du **FRONTAL** et du **POSTE D'ACCES** sont très voisines, nous analysons seulement celles du **FRONTAL**.

Rappelons que l'architecture proposée pour la mise en oeuvre du **FRONTAL** est structurée en *couches* (cf. III.2.4 - Partie I). Cette architecture a été retenue essentiellement en considérant celle du réseau **CYCLADES** où les règles de fonctionnement de chaque couche sont fixées par un protocole spécifique.

Ce type d'architecture favorise la mise en oeuvre de manière indépendante de chaque couche facilitant ainsi son évolution postérieure, par amélioration des algorithmes internes, ou par emploi de technologies nouvelles.

Notons, toutefois, que ces modifications sont tolérées dans la mesure où elles ne remettent pas en cause les interfaces définies pour les couches adjacentes.

I.1.1. Système constitué de sous-systèmes

Considérons que les fonctions et les services de chaque "couche" sont réalisés par un "sous-système". Le FRONTAL peut alors être décrit comme un *système* constitué d'un ensemble de *sous-systèmes* interagissant et évoluant en parallèle.

I.1.1.1. Le parallélisme des sous-systèmes

Bien que le FRONTAL soit mis en oeuvre sur un ordinateur mono-processeur, l'évolution en parallèle des différents sous-systèmes est *logiquement* concevable puisque les transformations qu'ils réalisent peuvent être considérées comme indépendantes car elles portent sur des objets disjoints (paquets, lettres, blocs, ...).

I.1.1.2. L'interaction des sous-systèmes

Les interactions des différents sous-systèmes correspondent à des facteurs de nature diverse. Certaines interactions proviennent des particularités technologiques de la mise en oeuvre, c'est-à-dire qu'elles sont dues *implicitement* à la réalisation, alors que d'autres sont intrinsèques à l'application, c'est-à-dire *explicitement* définies par celles-ci.

- * Les interactions *implicites* sont purement contingentes et résultent de l'utilisation concurrentielle de certaines ressources matérielles (mémoire, UC, ...) et logicielles (procédures, appels au moniteur, ...).
- * Par contre, les interactions *explicites* des différents sous-systèmes correspondent aux services échangés au travers des interfaces des couches qu'ils réalisent respectivement.

I.1.1.3. L'asynchronisme des sous-systèmes

Les protocoles définissant les actions de chaque sous-système (ou couche) sont dits de bout en bout. Autrement dit, chaque sous-système possède des processus qui dialoguent et échangent des informations selon les normes de son protocole, avec d'autres processus s'exécutant sur des sites distants (voir Fig. I.1).

Processus intermédiaires

Les processus "partenaires", c'est-à-dire les processus locaux au FRONTAL et les processus s'exécutant sur des sites distants qui dialoguent selon un même protocole, réalisent leurs échanges au travers d'autres processus (ou lignes de transmission) dont le rôle est d'acheminer l'information.

La fréquence des échanges entre les processus partenaires est une fonction combinée

- des propriétés *statiques* (mode de réalisation, priorité, ...)
- et des propriétés *dynamiques* (charge ponctuelle d'un système, ressources disponibles, ...)

à la fois du site hôte de chacun de ces processus et des sites des processus intermédiaires. Cela détermine le comportement *asynchrone* du FRONTAL aussi bien en ce qui concerne l'activité de chaque sous-système qu'au niveau de leurs échanges.

I.1.1.4. Les interfaces des sous-systèmes

L'interaction *explicite* entre les différents sous-systèmes se caractérise par une profusion d'échanges :

- de messages,
- et de signaux

au travers de leurs interfaces respectives.

a) Messages

- * La production des messages nécessite un mécanisme pour la gestion de la mémoire.
- * Et leur échange suppose l'existence d'un mécanisme de communication adapté au mode producteur-consommateur.

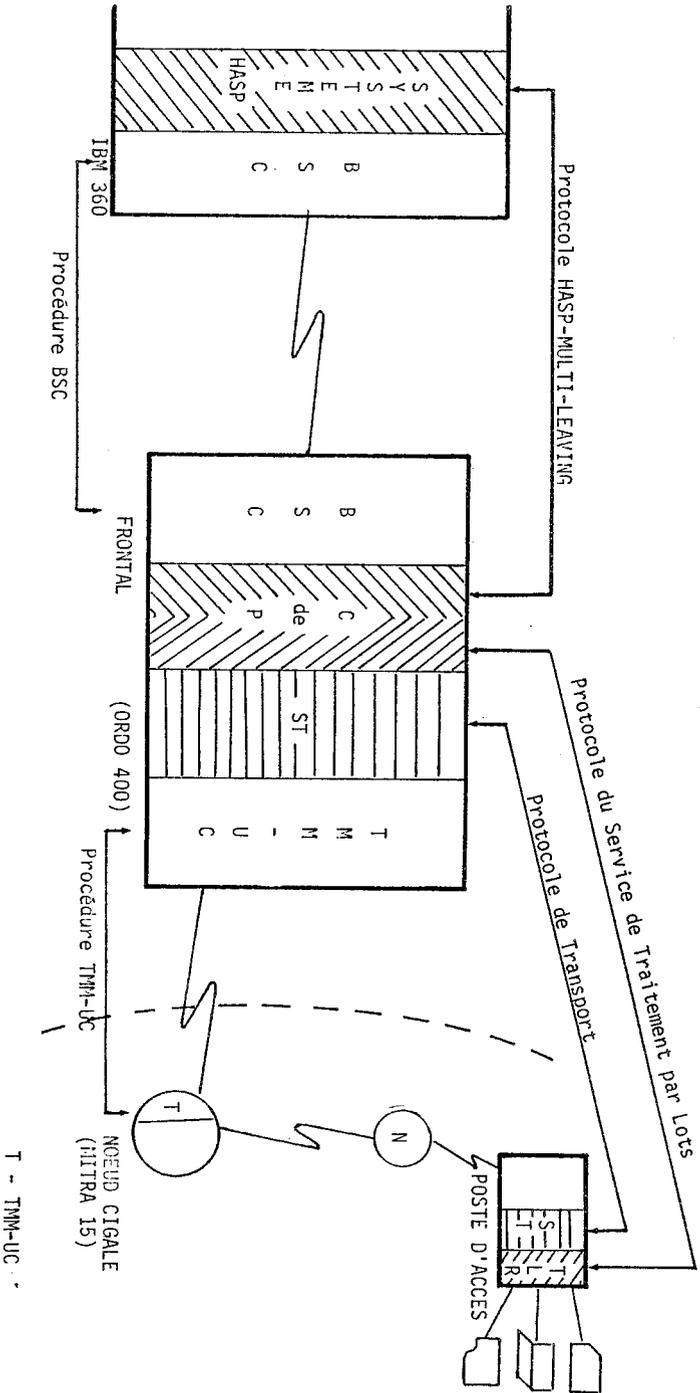


Figure I.1 - Différents sites participant à une session

b) Signaux

L'attente de l'arrivée d'un signal peut bloquer l'activité d'un processus. C'est en fonction du mode d'attente d'un signal (c'est-à-dire attente *bloquante* ou non) que l'on peut déterminer le mécanisme le plus adapté à son envoi.

Par exemple, dans le cas d'une attente *non bloquante*, l'arrivée d'un signal peut être signifié par le positionnement d'une mémoire commune aux processus concernés par l'échange.

Par contre, dans le cas d'une attente *bloquante*, l'envoi du signal par l'intermédiaire du "moniteur-système" est plus intéressant. Car de cette façon, l'allocateur de l'Unité Centrale est en mesure de déterminer quand un processus est bloqué par une attente et quand il ne l'est plus.

1.1.1.5. La réentrance des programmes

Les services fournis par les divers sous-systèmes, en particulier ceux correspondant aux couches "Station de Transport" et "Conversion de Protocoles", sont de par leur définition destinés à satisfaire simultanément plusieurs utilisateurs.

Les actions effectuées pour le compte de chaque utilisateur résultent de l'exécution des programmes en fonction d'un *contexte* propre à chaque utilisateur.

De ce fait, on peut dire que ces programmes sont de nature réentrante. Le nombre de processus exécutant le même programme est fonction du nombre de clients (liaisons, flots, appareils, ...) qui demandent simultanément les mêmes services.

1.1.1.6. Les horloges de garde

Certaines actions d'*entretien* et de *surveillance* réalisées par les procédures de ligne BSC et TMM-UC, et la Station de Transport doivent être déclenchées par la sonnerie d'une horloge de garde associée à chacune de ces actions.

I.2. CARACTERISTIQUES DU MATERIEL CHOISI

Comme nous l'avons déjà dit (cf. IV.1 - Partie I), un mini-ordinateur ORDO 400 [ORDO a], jouant au CICG le rôle de terminal lourd, a été choisi pour la mise en oeuvre du FRONTAL et du POSTE D'ACCES.

Ce mini-ordinateur se caractérise par :

- un processeur unique,
- une mémoire d'application organisée en mots de 16 bits dont le cycle d'accès est de 250 ns,
- 15 registres de travail dont certains sont spécialisés (index, accumulateur, interruption, ...),
- 8 niveaux d'interruptions dont celui correspondant aux entrées-sorties multiplexé en 16 sous-niveaux,
- un jeu d'environ 100 instructions-machine (une instruction peut occuper un ou deux mots-mémoire et son exécution durer de 1 à 11 cycles),
- une horloge temps réel produisant une interruption toutes les 10 millisecondes.

Configuration disponible

La configuration du mini-ordinateur disponible pour la mise en oeuvre du FRONTAL et du POSTE D'ACCES est constituée par :

- 32 K mots mémoire,
- un lecteur de cartes,
- une imprimante,
- une console,
- deux lignes de transmission synchrone avec leur modem respectif.

En plus, on disposait de :

- deux dérouleurs de bandes magnétiques pour le développement et la mise au point.

I.3. CARACTERISTIQUES DU MONITEUR UTILISE

Un moniteur "multi-tâches", délivré par le constructeur, est disponible sur l'ORDO 400. Ce moniteur, nommé METEOR I (Moniteur d'Exploitation en Temps Partagé) [ORDO b], est un système résident qui permet l'exécution de *tâches* en temps partagé.

Il assure :

- l'allocation de l'Unité Centrale,
- la gestion des périphériques d'E/S,
- le traitement des différents niveaux d'interruption et des événements associés,
- la gestion des horloges logicielles,
- la gestion des files d'attente,
- le traitement des appels au moniteur.

Tâche est le nom assigné à un programme binaire chargé en mémoire. Le moniteur peut gérer un nombre limité de tâches et celles-ci ne sont pas rééchantillonnées.

a - Stratégie d'allocation de l'Unité Centrale

L'Unité Centrale est allouée aux différentes tâches candidates à l'exécution en donnant la même priorité à chacune d'elles, selon une stratégie dite du *touriquet*.

b - Horloges logicielles

Le moniteur METEOR I gère au plus une horloge logicielle par tâche. Chaque tâche arme sa propre horloge en effectuant un appel au moniteur. La valeur d'initialisation est un nombre de "cycles de base" de l'horloge temps réel et elle est décrétementée d'une unité lors de chaque "top horloge". Son passage à zéro provoque l'envoi d'un signal à la tâche concernée.

c - Files d'attente

Le moniteur METEOR fournit les files d'attente comme mécanisme pour la communication de données entre les tâches et les primitives correspondantes pour les accéder.

I.4. MODIFICATIONS APPORTEES AU MONITEUR

En règle générale, on a essayé de limiter au minimum les modifications au moniteur.

I.4.1. Les horloges de garde

Puisque la gestion d'une seule horloge logicielle par tâche est insuffisante, notamment pour les processus de la Station de Transport, on a modifié le moniteur pour qu'une tâche puisse consulter et armer l'horloge logicielle d'une autre tâche. Ceci a été fait dans le but de spécialiser une tâche dans la gestion de l'ensemble des "horloges de garde" d'un sous-système.

I.4.2. La synchronisation

Les uniques signaux gérés par le moniteur METEOR concernent :

- la fin d'une opération d'E/S,
- la sonnerie de l'horloge logicielle associée à une tâche.

Des modifications lui ont été apportées pour que les tâches puissent échanger des *signaux* par son intermédiaire dont la sémantique n'est connue que par les tâches concernées.

I.5. TECHNIQUES DE MISE EN OEUVRE

Dans le cadre de cet exposé, nous examinons seulement certains aspects de la mise en oeuvre tels que l'organisation modulaire, les mécanismes pour la réalisation des interfaces.

I.5.1. Organisation modulaire

I.5.1.1. Organisation au niveau du moniteur

Du point de vue de la réalisation effectuée sur le mini-ordinateur ORDO 400, le **FRONTAL** et le **POSTE D'ACCES** peuvent être décrits comme un ensemble de tâches (au sens METEOR) se déroulant sous le contrôle du moniteur du calculateur hôte (voir Fig. I.2).

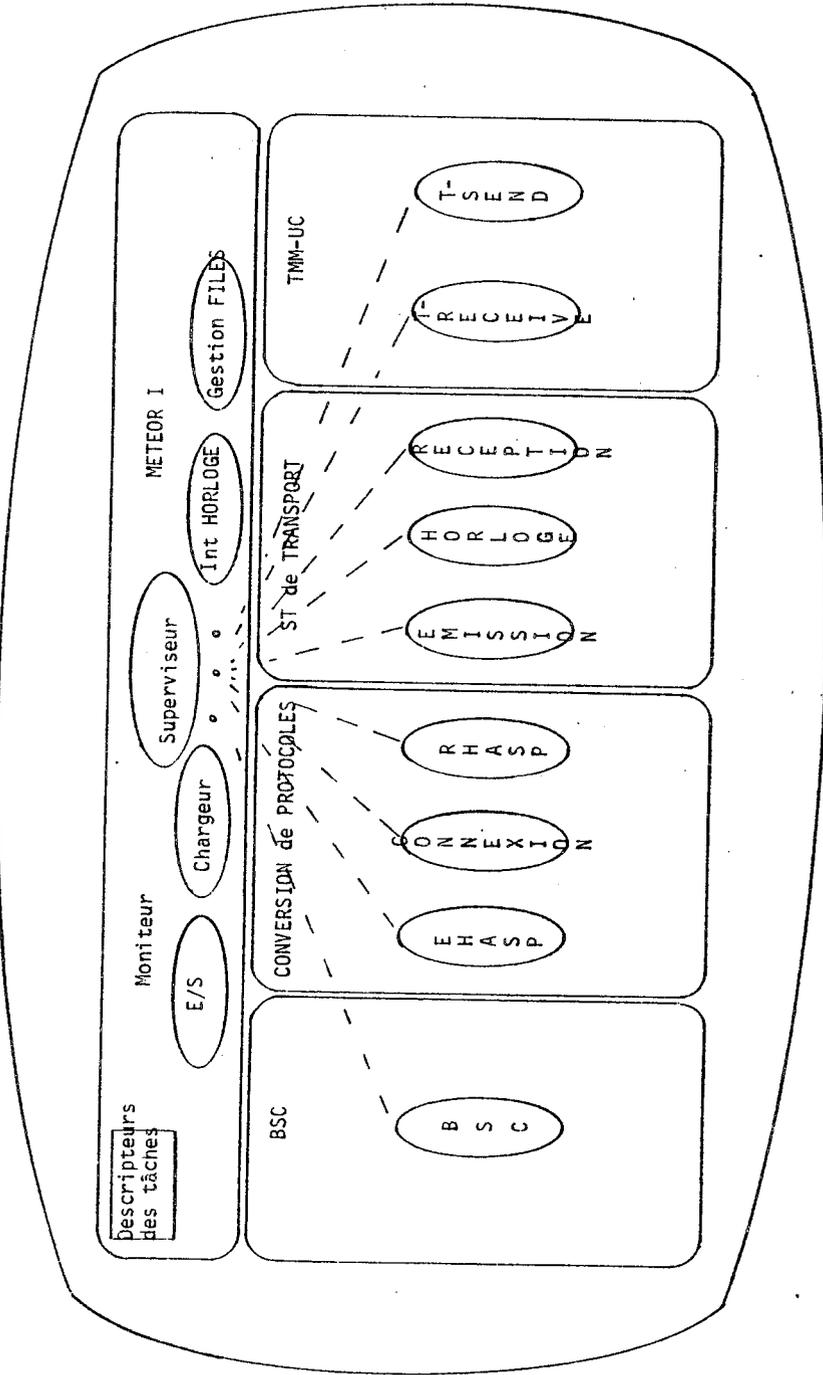


Figure I.2 - Organisation interne du FRONTAL

Le moniteur gère l'ensemble des ressources matérielles utilisées par ces deux applications (U.C., périphériques, lignes de transmission, ...) et quelques-unes de leurs ressources logicielles (files d'attente, horloge, ...).

I.5.1.2. Organisation à l'intérieur d'une tâche

Afin de pallier les carences du moniteur METEOR, en ce qui concerne la création dynamique de processus (nombre maximal de tâches, non réentrance de tâches), nous avons réalisé

- le développement à l'intérieur d'une tâche, en tant que programme principal, des "fonctions de contrôle" destinées à la gestion des processus s'exécutant sur des programmes réentrants,
- la réservation de zones mémoire pour l'enregistrement des contextes associés à ces processus,
- le codage des programmes réentrants comme de sous-routines de la tâche qui active les processus exécutant leur code.

I.5.1.3. Critères pour le découpage en tâches

- * Afin de garantir l'indépendance de mise en oeuvre de chaque couche, chaque sous-système aussi bien du *FRONTAL* que du *POSTE D'ACCES* a été implémenté par au moins une tâche au sens *METEOR*.
- * Par ailleurs, le nombre maximal de tâches supportées par le moniteur a limité le découpage de chaque sous-système.
- * En général, toutes les fonctions d'une couche liées au traitement de l'information relative à un même sens de transmission (par rapport aux couches voisines) ont été groupées dans une même tâche du sous-système.

Remarques : Signalons que la "Station de Communication" du *FRONTAL* devait assurer un nombre très réduit de services, et que ceux-ci ont été implémentés dans le sous-système correspondant à la procédure *TMM-UC*.

La simplification des fonctions de la Station de Communication obéissait au fait qu'une seule Station de Transport devait être servie. En conséquence, le multiplexage et démultiplexage des paquets en provenance ou à destination de plusieurs Stations de Transport n'avaient pas raison d'être réalisés.

Par ailleurs, cette décision restait conforme à la définition en couches du Réseau CYCLADES car l'interface entre une Station de Communication et une procédure de ligne CIGALE (BSC, TMM-UC, ...) n'est pas formellement définie, elle est propre à chaque mise en oeuvre (cf. I.3 - Partie I).

I.5.2. Interfaces des différents sous-systèmes

I.5.2.1. Généralités

On a restreint la communication entre les tâches appartenant à des sous-systèmes différents aux seuls échanges définis par les interfaces des couches respectives.

Chaque sous-système du FRONTAL et du POSTE D'ACCES joue vis-à-vis des sous-systèmes adjacents (ou voisins) le double rôle de producteur et de consommateur.

a) Le mécanisme adopté pour l'échange de messages

Les files d'attente offertes par le moniteur METEOR conviennent fort bien pour l'échange des messages traités dans l'ordre de production. C'est pour cela que ce mécanisme a été adopté pour réaliser l'interface des différents sous-systèmes, à l'exception de l'interface d'accès des abonnés locaux à la Station de Transport (cf. I.5.2.2).

b) Gestion de la mémoire libre

Les messages échangés entre les sous-systèmes sont de taille variable mais limitée.

Puisque le moniteur METEOR ne possède pas un mécanisme pour la gestion dynamique de la mémoire, on a décidé de :

- réserver une *zone tampon* par classe de message (lettre, paquet, bloc) et de la diviser en *cases* de taille fixe,
- ranger les adresses de ces cases dans une *file d'attente* associée à une classe de message.

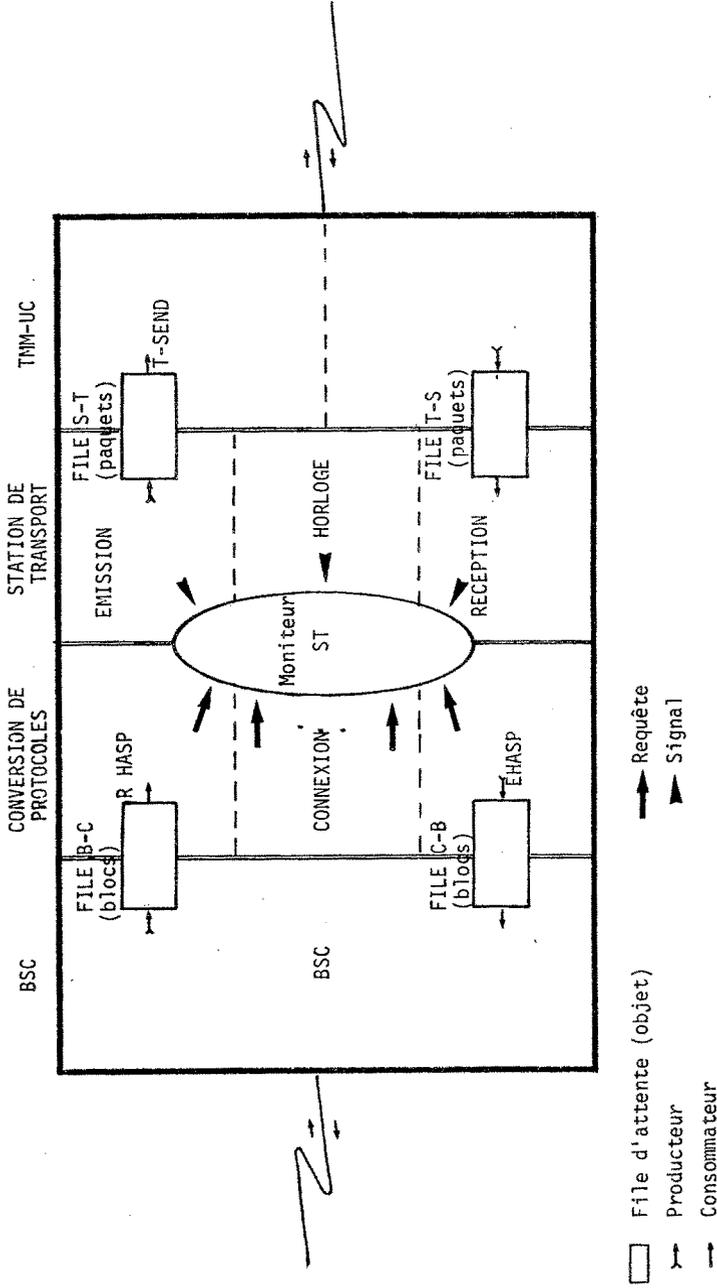


Figure I.3 - Interface des différents sous-systèmes du FRONTAL

1.5.2.2. Interface d'accès locale au Service de Transport

L'utilisation du mécanisme de file d'attente pour réaliser l'interface d'accès à la Station de Transport constitue au premier abord l'idée la plus naturelle. Mais après une analyse plus approfondie, cette solution s'avère mal adaptée aux caractéristiques des services rendus à ce niveau.

Présentons quelques inconvénients de cette méthode d'échanges :

- L'utilisation d'une file d'attente commune pour les messages adressés à la Station de Transport présente le risque que quelques abonnés saturent la capacité de stockage de la file d'attente au détriment de tous les autres abonnés.
- Les messages déposés dans une file d'attente METEOR ne peuvent être prélevés que dans l'ordre d'arrivée. En conséquence, si la Station de Transport utilise une seule file d'attente pour déposer tous les messages destinés aux abonnés locaux, cette organisation se révèle inadéquate pour que les différents abonnés puissent retirer uniquement les messages qui leur sont adressés.
- Une autre solution considérerait à attribuer à chaque abonné une file d'attente pour l'émission de messages et une autre pour leur réception.

Mais cette solution a le désavantage de multiplier considérablement le nombre de files d'attente à gérer et présente les mêmes inconvénients pour les abonnés gérant plusieurs liaisons ou portes que la solution précédente d'une file commune pour tous.

On a donc rejeté l'utilisation des files d'attente pour réaliser cette interface, en faveur d'un autre mécanisme qui permet aussi bien à la Station de Transport qu'aux abonnés de gérer indépendamment l'activité de chaque *liaison* et de chaque *porte*.

Dans la solution adoptée, on a défini une *requête* pour chaque type de service que les abonnés peuvent solliciter (Ex : ouvrir une porte, envoyer une lettre) [LUM77].

Ces requêtes peuvent être assimilées à un appel au "moniteur de la Station de Transport". Toutes les requêtes reçoivent une réponse immédiate, dans laquelle le moniteur de la station signale à l'abonné si

celle-ci a été acceptée ou non (par positionnement d'une code de condition). En cas d'erreur, la requête est refusée.

Quant à la satisfaction de la demande contenue dans une requête, il y a plusieurs variantes :

- la demande peut être servie immédiatement (par exemple, dans le cas où l'abonné désire retirer une lettre qui est disponible) ;
- au moment de la requête, la demande est seulement enregistrée par le moniteur de la Station de Transport dans le contexte de la liaison et sera traitée postérieurement (Ex : c'est le cas des requêtes pour l'ouverture d'une liaison, l'émission d'une lettre, ...)
- la Station de Transport ne peut pas satisfaire immédiatement la demande et ne possède pas les ressources nécessaires pour son enregistrement et traitement postérieur. Dans ces circonstances, l'abonné peut :
 - . soit refaire périodiquement la requête jusqu'à ce qu'elle soit acceptée,
 - . soit solliciter la station pour qu'elle lui envoie un *signal* dès qu'elle sera en mesure d'accepter sa requête.

I.5.3. Contrôle de flux

Dans la présente section, on décrit la façon dont le mécanisme de contrôle de flux a été implémenté dans la Station de Transport, et chez les abonnés de transport locaux, c'est-à-dire dans le sous-système "Conversion de Protocoles" du *FRONTAL* et "Application" du *POSTE D'ACCES*.

Le protocole de Transport *CYCLADES* (cf. I.3 - Partie I) prévoit un mécanisme pour réguler le débit d'émission de chaque liaison, à la capacité de l'abonné récepteur. A ce propos, les Stations de Transport échangent périodiquement pour chacune des liaisons gérées en commun, une valeur appelée *crédit*. Ce *crédit* (qui peut varier entre 0 et 15) sert à préciser à l'abonné émetteur la taille de la fenêtre de réception de l'abonné récepteur, c'est-à-dire le nombre de lettres que l'abonné récepteur est prêt à recevoir.

En conséquence, lorsque le *crédit* d'une liaison donnée est nul, l'émission sur cette liaison est suspendue.

a) Rôle d'une Station de Transport

Chaque Station de Transport doit pouvoir fournir à ses abonnés locaux des moyens pour qu'ils

- apprennent les variations du crédit des liaisons sur lesquelles ils émettent
- et puissent à leur tour réguler la réception de lettres.

b) Rôle des abonnés du POSTE D'ACCES

Il est du ressort des abonnés du POSTE D'ACCES :

- de réguler la réception de lettres sur la liaison associée à l'imprimante
- et d'adapter l'activité du lecteur de cartes à l'évolution du crédit de la liaison associée.

c) Rôle des abonnés du FRONTAL

Rappelons que le FRONTAL joue le rôle de "Convertisseur d'Interface" entre le système HASP et les terminaux lourds-réseau.

En conséquence, le FRONTAL ne doit pas être la *source* d'une demande pour ralentir ou suspendre la réception des données en provenance du lecteur de cartes d'un terminal lourd-réseau, mais simplement il doit réfléchir la demande qui a pu être formulée dans ce sens par le système HASP, véritable destinataire de ces données.

Il est aussi dans le rôle du FRONTAL de répercuter au système HASP, source des données envoyées à l'imprimante d'un terminal, les fluctuations du crédit relatives à la liaison de cet appareil.

Remarque : Il est important que les caractéristiques de mise en oeuvre de la ST et des abonnés du FRONTAL n'ajoutent pas de restrictions supplémentaires au contrôle de flux à celles imposées par les vrais partenaires des échanges, c'est-à-dire le système HASP d'un côté et les terminaux lourds-réseau de l'autre.

1.5.3.1. Mécanisme implémenté dans la Station de Transport

On analysera séparément les mécanismes mis en oeuvre pour réguler l'émission et la réception.

a) En réception

A l'ouverture d'une liaison, une file d'attente est créée et initialisée avec un nombre de cases vides égal à la capacité maximale en réception, estimée par l'abonné local lui-même.

Les cases de cette file servent à ranger les lettres adressées à la liaison associée.

Le crédit accordé pour réguler la réception est donc fonction du nombre de cases vides présentes dans cette file d'attente.

Les fluctuations de cette file se traduisent dans les variations du crédit octroyé; en conséquence quand cette file est vide, le crédit attribué à l'émetteur est nul.

Quand l'abonné local désire ralentir ou suspendre la réception sur une liaison, il ralentit la cadence de prélèvement des lettres et celle de restitution des cases vides.

b) En émission

Un abonné lorsqu'il essaie d'émettre une lettre sur une liaison dont le crédit est nul, voit sa demande refusée.

Dans la mise en oeuvre de la Station de Transport, un autre paramètre peut jouer aussi le rôle de régulateur en émission, il s'agit de l'anticipation.

On appelle *anticipation* le nombre maximal de lettres émises qui peuvent être simultanément en attente d'acquiescement.

Dans le cas de la Station de Transport ORDO 400, l'anticipation est déterminée par la taille de la *table de lettres émises non acquittées* associée à chaque liaison.

Cette table est de taille fixe et inférieure au crédit maximal établi par le protocole de Transport. Elle est destinée à contenir les renseignements des lettres émises jusqu'à leur acquiescement en prévention d'une éventuelle réémission.

I.5.3.2. Description d'un exemple

Pour présenter une vue d'ensemble des différentes étapes nécessaires à la régulation des échanges, et particulièrement pour situer l'insertion du mécanisme présenté dans la section précédente, nous décrivons le circuit parcouru (voir Fig. I.4) par les messages de contrôle pour réguler le débit du lecteur de cartes d'un terminal lourd réseau connecté au serveur HASP.

Note : La numérotation des étapes correspond au cas où un terminal lourd réseau émulé par le poste d'accès banalisé, s'exécutant sur l'ORDO 400, est connecté au serveur HASP à travers le FRONTAL.

Régulation du débit du lecteur de cartes (Ex : Suspension des envois).

a) Système HASP

- (1) - Le système HASP émet un message de contrôle (cf. III.2.5 - Partie I) pour suspendre la réception des données provenant du lecteur de cartes d'un terminal lourd HASP.

b) FRONTAL

- (2) - Ce message est reçu et interprété par la tâche du sous-système "Conversion de Protocoles" chargée de la réception de tous les messages en provenance du système HASP (tâche R HASP).
- (3) - Cette tâche retransmet la demande de suspension des envois à la tâche du même sous-système, chargée de l'émission de messages vers le système HASP (tâche E HASP).
- (4) - Cette dernière tâche suspend les prélèvements des lettres auprès de la Station de Transport, concernant la liaison du lecteur de cartes.

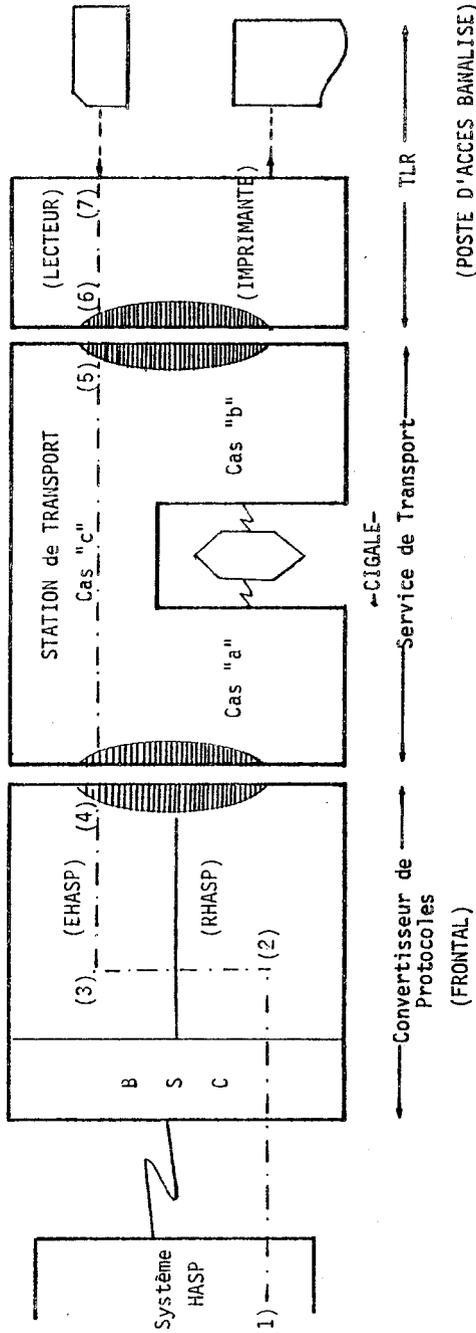
c) Service de Transport

- (5) - Le mécanisme de régulation "abonné Station de Transport" étant mis en jeu, lorsque la file d'attente de "cases libres" est vide, la Station de Transport du FRONTAL enverra un crédit nul à la Station de Transport du terminal lourd-réseau connecté.

d) Poste d'Accès

- (6) - Quand la tâche du POSTE D'ACCES gérant le lecteur de cartes essaie d'émettre une lettre sur la liaison associée à cet appareil, cette demande est refusée par la Station de Transport (crédit nul).
- (7) - En conséquence, l'activité du lecteur de cartes est suspendue jusqu'à réception d'un crédit positif.

Note : Le message de contrôle relançant l'activité du lecteur de cartes suit le même parcours que celui qui l'inhibe.



TLR - Terminal lourd-réseau
 PAB - Poste d'accès banalisé
 Cas "a" : TLR quelconque connecté au serveur HASP (à travers le réseau)
 Cas "b" : TLR émué par le PAB DRDO 400 connecté (à travers le réseau) à un serveur distant
 Cas "c" : Le serveur HASP et le TLR s'exécutent sur le même ordinateur (et partagent une Station de Transport)

Figure I.4 - Circuit des messages de contrôle pour la régulation du débit du lecteur de cartes

CHAPITRE II

OUTILS POUR LE DÉVELOPPEMENT ET LA MISE AU POINT

II.0. INTRODUCTION

On ne peut pas faire le tour des différents aspects d'une réalisation système sans évoquer les outils utilisés pour son développement et pour sa mise au point.

Dans ce chapitre, nous répertorions les caractéristiques des outils disponibles aussi bien pour le développement que pour la mise au point. Nous expliquons les raisons qui nous ont déterminés à définir un langage de programmation et à réaliser un traducteur prototype dans le cadre de ce projet.

Nous présentons les outils développés pour la mise au point au fur et à mesure que nous décrivons les étapes successives du travail de validation.

Après un bref examen des inconvénients rencontrés lors de la mise au point, en raison de l'inadéquation des outils disponibles aux particularités des applications réalisées, nous décrivons le profil des outils qui conviendraient le mieux aux applications du même type (temps réel, évolution en parallèle, ...).

II.1. LES OUTILS DISPONIBLES

L'ensemble des outils disponibles sur l'ORDO 400 pour le développement et la mise au point était fourni par le constructeur [ORD c].

Pour le développement, on disposait d'un *assembleur* et d'un *éditeur de texte*. Dans le chapitre précédent, nous avons présenté quelques particularités de l'ORDO 400, nous décrivons ici davantage les caractéristiques qui se reflètent dans le langage d'assemblage.

Le mini-ordinateur ORDO 400 est une machine à *accumulateur* ; elle possède trois modes d'adressage : direct, indirect et indexé. Ce dernier permet l'accès et la gestion des éléments de tableau. Les instructions qui traitent les chaînes de longueur variable imposent qu'elles soient alignées sur une frontière de mot et que leur longueur soit un nombre entier de mots. Comme opérations arithmétiques, l'ORDO 400 ne possède que l'addition et la soustraction.

Il n'a aucune instruction particulière pour le contrôle de boucles ni la synchronisation, ni de mécanisme de protection mémoire.

Pour réaliser l'exclusion mutuelle, il est nécessaire de masquer toutes les interruptions.

Parmi les outils de mise au point disponibles, il y avait naturellement le "pupitre opérateur" et deux autres outils logiciels : un "metteur au point" et un "traceur de ligne".

Le *pupitre* permet de réaliser la plupart des fonctions habituelles de ce genre d'outil : lancer, arrêter l'exécution, exécuter en mode pas à pas, visualiser et modifier le contenu d'un registre ou d'un mot mémoire.

Signalons que la fonction pupitre pour arrêter l'exécution sur référence adresse n'inclut pas l'arrêt sur accès en lecture ou en écriture à cette adresse, c'est-à-dire que l'arrêt se produit seulement quand l'adresse est référencée par le compteur ordinal.

Le *metteur au point* est un système d'aide à la vérification des programmes. Les commandes de ce système permettent de référencer les instructions et les données d'un programme par leur adresse en mémoire. Il offre les mêmes fonctions que le pupitre pour une seule tâche, à deux différences près : on peut insérer plusieurs points d'arrêt mais l'exécution en mode pas à pas est impossible.

Le *traceur de ligne* est un programme destiné à la mise au point des procédures de lignes, il permet de tracer sur un support magnétique l'activité d'une ligne de transmission.

L'analyse des caractéristiques des outils disponibles, leurs insuffisances en fonction de la complexité des applications à mettre en oeuvre, la perspective d'avoir à traduire ce logiciel pour son implantation sur une autre machine, nous ont déterminés à développer comme nouvel outil un langage de programmation dont les particularités sont décrites dans la section suivante.

II.2. LPORDO : UN LANGAGE DE PROGRAMMATION

II.2.1. Objectifs

Le nouveau langage de programmation devait essentiellement :

- faciliter l'écriture du logiciel et sa mise au point ;
- permettre l'intégration des programmes écrits dans ce langage avec ceux développés en langage d'assemblage ;
- permettre le contrôle du code généré afin de pouvoir utiliser pour les tests de ces programmes le metteur au point déjà existant.

Il était aussi important pour nous que ce langage soit facilement "compilable" afin de pouvoir disposer d'un traducteur prototype à brève échéance.

On a donc conclu que le langage à définir devait posséder certaines facilités des langages de haut niveau tout en restant relativement simple.

II.2.2. Particularités du langage

Le langage défini est du type dit *semi-évolué*, il ressemble au langage PL360 [WIR68] mais adapté aux caractéristiques de la machine ORDO 16^(*).

La définition du langage a fait l'objet d'un document séparé [LUM76].

On résume ici ses traits essentiels.

Ce langage, dénommé LPORDO, offre des facilités aussi bien pour la structuration des données et des programmes que pour l'accès aux primitives de la machine.

Les données peuvent être organisées en vecteurs de mots ou d'octets, et les programmes peuvent être structurés à l'aide des constructions suivantes :

- blocs,
- procédures,
- macro-définitions.

Les macro-définitions ont été introduites dans le langage afin de permettre l'insertion des instructions-machine dans le corps d'un programme LPORDO.

(*) ORDO 16 caractérise l'U.C. et le jeu d'instructions du mini-ordinateur utilisé.

Le langage est enrichi par des constructions de base telles que :

- les affectations,
- les expressions arithmétiques et logiques.

Un certain nombre d'instructions de contrôle ont été définies dans le langage afin de faciliter l'écriture des algorithmes :

- instruction conditionnelle,
- instruction itérative,
- instruction choix,
- instruction de branchement,
- appel et retour de procédure,
- appel de macro-définition.

II.2.3. Définition du traducteur

Dans ce paragraphe, on présente les critères suivis et les choix adoptés pour définir la philosophie générale de construction du traducteur.

a) Vitesse d'écriture

Le trait essentiel exigé du traducteur prototype est de permettre une réalisation rapide du traducteur même au détriment de ses performances.

b) Transportabilité du traducteur

On désirait que le traducteur soit facilement portable afin de générer le code d'une nouvelle machine prévue par le constructeur.

c) Site du développement

Les choix précédents de vitesse, de portabilité excluaient l'ORDO 400 en fonction de ses outils, comme site pour le développement du traducteur. Il fallait choisir un calculateur ayant des facilités telles qu'un système conversationnel, un éditeur de texte puissant, des outils adéquats pour le développement et la mise au point.

d) Méthode de développement du compilateur

Les méthodes envisagées pour la construction du compilateur se groupaient dans les deux catégories suivantes :

- construction dirigée par la syntaxe à l'aide des outils programmés (Ex : [GRI68]) ;
- traduction par macro-génération.

On a préféré cette dernière méthode car elle nous permettait à la fois une implémentation rapide et elle s'avérait plus souple pour l'adaptation du traducteur à la génération du code objet destiné à une autre machine.

e) Nature des optimisations

Dans la version du traducteur prototype, on a décidé de réaliser des optimisations locales, c'est-à-dire au niveau de chaque instruction, sans tenir compte de l'environnement.

f) Code généré

Pour accélérer et simplifier la réalisation du traducteur, nous avons choisi de produire comme code objet le langage d'assemblage.

II.2.4. Le traducteur réalisé

Le macro-générateur STAGE 2 [POO70] a été retenu comme outil pour le développement du traducteur, car on considérait qu'il répondait convenablement aux différents choix énoncés dans le paragraphe précédent.

Et l'on a choisi le calculateur IBM 360 du CICG, comme site pour le développement en raison des facilités offertes par son système temps partagé CP/CMS [IBM c], de la disponibilité de plusieurs versions du même macro-générateur STAGE 2.

STAGE 2 est un macro-générateur *portable* car son noyau (la machine "FLUB") est écrit en FORTRAN. Son mécanisme de reconnaissance d'appel macro, à l'aide d'un modèle (*pattern matching*) permet l'analyse syntaxique des phrases d'un programme LPORDO, ses fonctions de traitement de chaînes facilitent l'analyse lexicographique et son système d'entrées-sorties, la construction de deux fichiers de sortie, l'un avec la "sortie imprimée", l'autre avec le "code objet".

Nous avons soumis la définition syntaxique du langage LPORDO à un transformateur de grammaire [GRI68] afin de vérifier sa cohérence et d'obtenir une définition équivalente en forme de grammaire LL(1) pour faciliter la réalisation de son traducteur.

Structure interne du traducteur

Le jeu de macro-définitions STAGE 2 décrivant le traducteur de LPORDO (lot 1 - voir Fig. II.1) comporte quatre sous-ensembles ayant les fonctions suivantes :

- analyse syntaxique,
- analyse lexicographique,
- création et gestion de la table des symboles,
- génération du code.

Le traducteur génère du code en un seul passage et son fonctionnement est contrôlé par les macro-définitions réalisant l'analyse syntaxique.

II.2.5. Evaluation du traducteur

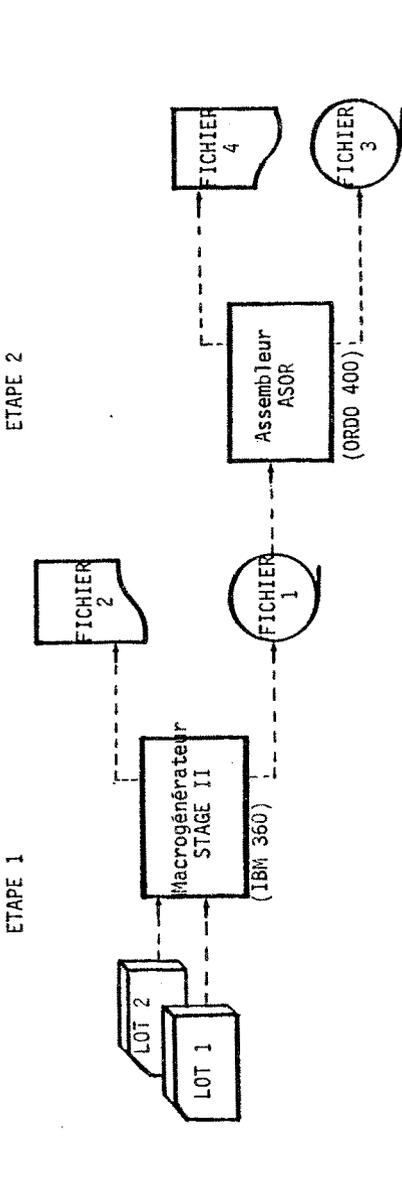
Bien que STAGE 2 dans sa brochure de présentation soit décrit comme un outil très adapté pour la traduction de langages de type ALGOL, notre expérience nous a montré le contraire.

Par exemple, la création et la gestion de la table des symboles d'un programme s'avère très laborieuse et coûteuse en espace mémoire et en temps d'exécution, car STAGE 2 n'offre pas le moyen d'utiliser des données structurées.

Les inconvénients majeurs perçus lors de l'utilisation de STAGE 2 concernent l'occupation mémoire et le temps d'exécution.

Le jeu de macros définissant LPORDO est assez volumineux (environ 4000 lignes d'expansion), en conséquence le temps nécessaire pour reconnaître et traduire une instruction LPORDO est important.

C'est pour cela que nous estimons que l'utilisation d'un outil comme STAGE 2 doit être réservée à la dernière étape d'une traduction, c'est-à-dire pour la génération du code objet, à partir d'un code intermédiaire.



Production d'un programme en langage d'assemblage

Production d'un programme binaire

LOT 1 : Jeu de macro-définition décrivant le traducteur LPORDO

LOT 2 : Programme source LPORDO

FICHIER 1 : Programme "objet"

FICHIER 2 : Liste du programme traduit

FICHIER 3 : Programme "binaire" chargeable

FICHIER 4 : Liste du programme source LPORDO

Figure II.1 - Etapes de la traduction d'un programme LPORDO

II.2.6. Evaluation du langage

a) Utilisation

Dans le cadre du projet qui nous occupe, nous avons utilisé LPORDO pour développer la "Station de Transport". Pour faciliter une estimation du volume du code traduit, on précise que les programmes binaires relatifs à la Station de Transport occupent approximativement 13K octets.

L'écriture du logiciel correspondant au FRONTAL et au POSTE D'ACCES avait commencé avant l'achèvement du traducteur ; c'est la raison pour laquelle ce logiciel n'a pas été entièrement réalisé à l'aide du langage LPORDO.

Le but recherché concernant la facilité d'écriture a été largement atteint. La mise au point de la Station de Transport a été rapide en comparaison du temps passé pour celle des programmes plus simples écrits en langage d'assemblage.

b) Carences du langage

Les caractéristiques de l'application développée en LPORDO nous a montré qu'il serait souhaitable de disposer dans la définition du langage

- d'une option pour déclarer les données avec un qualificateur précisant si elles doivent être accédées dans le corps du programme en section critique.

Cela permettrait au compilateur, lorsqu'une donnée déclarée avec cette option est référencée dans le corps du programme, de vérifier qu'il y ait une section critique ouverte.

Cette option a l'intérêt de libérer le programmeur du travail fastidieux de contrôler les sections critiques.

Notre travail aurait été facilité par la disponibilité dans le langage des structures de données plus riches et des outils d'aide à la mise au point.

II.3. LES OUTILS DEVELOPPES POUR LA MISE AU POINT

La validation du logiciel développé a été réalisée d'une façon expérimentale et progressive. Dans un premier temps, on a vérifié le comportement de chaque sous-système isolé, puis on les a intégrés graduellement.

En règle générale, on a utilisé les outils de mise au point disponibles précédemment présentés (cf. II.1) et par ailleurs, on a développé des *simulateurs* des sous-systèmes voisins pour chaque sous-système testé.

Dans le tableau suivant, on décrit l'ordre dans lequel les différents sous-systèmes ont été vérifiés et on mentionne les outils utilisés.

ETAPES	SOUS-SYSTEMES	OUTILS
1ère	BSC TMM-UC	SNAP ^(*) , Simulateur de Conv. de Protoc. SNAP, Simulateur de la ST (SIMST1)
2ème	ST	. Simulateur de TMM-UC (LINE) . Abonnés de tests
	APPLICATION (Poste d'accès)	. Simulateur de la ST (SIMST2)
3ème	POSTE D'ACCES	. Intégration des sous-systèmes TMM-UC, ST, APPLICATION
4ème	CONVERSION DE PROTOCOLES	. Poste d'Accès, BSC . Analyseur série

Tableau II.1 - Etapes de la mise au point

La fonction des simulateurs est de présenter la même interface que celle du sous-système simulé et d'offrir optionnellement une trace des échanges effectués à travers cette interface. C'est pour cela que deux simulateurs de la "Station de Transport" ont été développés, l'un relatif à l'interface avec le sous-système TMM-UC (SIMST1), l'autre pour l'interface correspondant aux abonnés locaux (SIMST2).

(*) SNAP, traceur de ligne

La fonction de l'émulateur LINE est plus complexe que celles des autres simulateurs, car non seulement il devait présenter la même interface que le sous-système TMM-UC, mais aussi il devait pouvoir simuler certains phénomènes ou anomalies du service CIGALE, tels que la perte, la duplication des paquets, l'altération de l'ordre de livraison des paquets par rapport à leur émission.

Pour la mise au point du POSTE D'ACCES, nous nous sommes servis de certaines facilités offertes à cette fin par le réseau CYCLADES (par exemple ECHO) et nous avons adressé différents serveurs du réseau (serveur de "traitement par lots" de l'IRIS 80 (CICG), serveur "temps partagé" de l'IRIS 80 (IRIA), de l'IBM 360 (CICG), ...).

Pour la vérification du sous-système "Conversion de Protocoles", nous n'avons développé aucun simulateur particulier, nous avons utilisé le POSTE D'ACCES et la procédure de ligne BSC déjà mis au point. Dans cette étape de la vérification, quand il a fallu tracer l'activité d'une ligne de transmission, il n'était plus possible d'utiliser le programme SNAP car il n'y avait pas assez de place mémoire pour son chargement. On a donc été forcé d'utiliser un analyseur série branché en bretelle au MODEM de la ligne de transmission correspondante.

Les difficultés rencontrées lors de la mise au point du FRONTAL et du POSTE D'ACCES provenaient des insuffisances des outils disponibles et particulièrement de leur inadéquation pour la vérification des applications *temps réel* et *multi-tâche*.

- * Le metteur au point dont on disposait ne permet pas d'espionner simultanément plusieurs tâches (ou processus), c'est-à-dire d'insérer des points d'arrêt dans plusieurs tâches à la fois.
- * L'inexistence d'un mécanisme pour réaliser la protection mémoire combinée à l'impossibilité d'obtenir l'arrêt de l'exécution sur référence adresse s'est révélée un handicap très important.

On peut citer quelques traits des outils qui conviendraient mieux pour la mise au point de ce type d'application.

- * Un "Journal de Bord" où sont enregistrés les échanges au niveau des interfaces des différents sous-systèmes faciliterait la détection des erreurs provenant de la synchronisation.
- * Pour une application en temps réel, il serait utile de pouvoir préparer, à l'avance, des routines ou des fichiers de test constitués par des commandes du metteur au point et de les mettre en oeuvre par un seul appel. Ceci permettrait de lancer rapidement une trace du système lorsque les premières anomalies sont détectées, avant qu'il ne soit trop dégradé et que les données saisies ne soient plus significatives.

PARTIE III

EVALUATION DE LA SOLUTION ADOPTÉE

CHAPITRE I

EVALUATION DES PERFORMANCES

I.0. INTRODUCTION

L'observation et l'analyse du comportement du **FRONTAL** et du **POSTE D'ACCES** dans différentes situations de charge s'imposaient pour enrichir et compléter l'expérience acquise dans leur définition et leur développement.

Les seules connaissances qu'on possédait sur ces deux systèmes découlaient de leur architecture, c'est-à-dire des relations fonctionnelles entre les différentes tâches d'un même sous-système et des sous-systèmes adjacents. On désirait alors réaliser une étude de leurs *propriétés dynamiques* afin de déterminer certaines limites d'utilisation.

Par exemple, on ne savait pas si l'utilisation de l'Unité Centrale avoisinait la saturation, si les files d'attente étaient convenablement dimensionnées. On se demandait s'il était possible d'élargir la configuration du **POSTE D'ACCES** en incluant d'autres appareils périphériques, de connecter le **FRONTAL** au calculateur hôte par des lignes de transmission supplémentaires, etc.

Pour répondre à quelques-unes de ces questions, nous avons décidé d'effectuer des mesures et d'évaluer l'utilisation de certaines ressources telles que l'Unité Centrale, la mémoire. Les objectifs de cette étude n'impliquaient pas de mesures *exhaustives* (c'est-à-dire recouvrant toutes les conditions d'observations) ni *précises* (faisant apparaître toutes les nuances). On cherchait plutôt une caractérisation du comportement du **FRONTAL** et du **POSTE D'ACCES** dans des situations particulières mais significatives à notre avis.

On escomptait que les renseignements fournis par ce travail de collection et d'analyse de mesures nous aideraient non seulement à déterminer la valeur optimale de certains paramètres susceptibles d'améliorer les performances des systèmes étudiés, mais aussi nous être utiles pour la définition et la réalisation d'autres applications de caractéristiques analogues.

I.1. OUTILS ET TECHNIQUES POUR LES MESURES

Les caractéristiques de l'horloge temps réel de l'ORDO 400 ont conditionné les modalités de nos mesures. En effet, cette horloge ne produit des interruptions que toutes les 10 millisecondes et cette fréquence ne permet pas de réaliser des observations très fines. Rappelons que l'exécution d'une instruction dure de 1 à 11 cycles de 250 nanosecondes et que entre deux "topshorloges" environ 7000 instructions peuvent s'exécuter.

Comme l'on n'avait pas d'accès à une horloge externe, on a donc renoncé à mesurer la fréquence d'arrivée des événements, la durée des actions, etc.

Par contre, on a décidé de comptabiliser les événements à étudier et de prélever périodiquement l'état de leur compteur respectif. On illustrera cette méthode en décrivant des exemples.

I.2. TAUX D'OCCUPATION DE L'UNITE CENTRALE

L'étude réalisée ne nous a pas permis de faire une évaluation précise du taux d'occupation de l'Unité Centrale par chaque tâche du FRONTAL et du POSTE D'ACCES, mais d'en faire une estimation grossière.

Rappelons que le moniteur METEOR I alloue à tour de rôle l'Unité Centrale aux tâches "activables" et que les tâches peuvent rendre par elles-mêmes le contrôle de l'Unité Centrale ou être suspendues par un "top horloge". En conséquence, l'allocation de l'Unité Centrale est réalisée :

- soit à chaque interruption de l'horloge temps réel
- soit lorsqu'une tâche rend le contrôle pour se mettre
 - . en attente d'un ou plusieurs événements
 - . ou en repos jusqu'au prochain tour de l'allocateur.

Mesures effectuées

Puisqu'on ne pouvait pas mesurer ponctuellement le temps qu'une tâche garde le contrôle de l'Unité Centrale, on a comptabilisé régulièrement le nombre de fois qu'elle rendait le contrôle et le nombre de fois qu'elle

était suspendue à la suite d'un top horloge (la somme de ces deux compteurs nous indique le nombre total d'activation d'une tâche par intervalle).

Parallèlement à ces prélèvements, on enregistrerait aussi certains paramètres dont la comparaison avec les valeurs de ces mêmes paramètres dans l'observation précédente permet de caractériser la charge du système et son évolution (liaisons ouvertes, nombre de lettres émises, ...).

Remarques : Il faut signaler que les mesures n'ont été réalisées que pendant une courte période d'observation, en raison de l'indisponibilité du calculateur ORDO 400 pour nos expériences.

Nous précisons aussi que les mesures ont été faites lors de la connexion d'un terminal lourd-réseau émulé par le POSTE D'ACCES ORDO 400, connecté au serveur HASP à travers le FRONTAL.

Analyse des mesures

1 * On a formulé l'hypothèse que le processus de mesures est indépendant des tâches observées qui se partagent l'Unité Centrale et l'on a estimé que le taux d'occupation d'une tâche est proportionnel au nombre de fois que cette tâche est interrompue à la suite d'un top horloge sur le nombre total de "tops horloges" observés (cf. Tableau I.1).

SYSTEME	COUCHE	% UC
FRONTAL	BSC	0,05
	Conversion de Protocoles	0,10
	Station de Transport	0,33
POSTE D'ACCES	TMM-UC	0,11
	Application	0,08

Tableau I.1 - Estimation du taux d'occupation de l'Unité Centrale par les tâches du FRONTAL et du POSTE D'ACCES

- 2 * Les tâches du FRONTAL et du POSTE D'ACCES sont au nombre de douze. Elles constituent deux groupes en fonction de leur comportement vis-à-vis du moniteur.

Groupe "a" : Ce groupe est constitué de toutes les tâches activées à chaque tour de l'allocateur de l'Unité Centrale (9 tâches).

Groupe "b" : A ce groupe appartiennent les tâches qui ne sont activées que lorsqu'un signal attendu est posté (3 tâches).

Une fois que les tâches du groupe "a" sont activées, elles contrôlent par elles-mêmes leurs conditions d'activité, si leur analyse révèle qu'elles n'ont pas de traitement à réaliser (Ex : la file d'attente de messages est vide), elles rendent au moniteur le contrôle de l'Unité Centrale en se mettant en repos jusqu'au prochain tour.

La stratégie utilisée par l'allocateur de l'Unité Centrale fait que les tâches du groupe "a" sont activées le même nombre de fois par intervalle.

Par contre, les tâches du groupe "b" sont réveillées seulement quand elles ont un traitement à accomplir et leur nombre d'activation par intervalle est fonction du type de traitement qu'elles assurent.

On a analysé aussi ces mesures en fonction d'une charge croissante du FRONTAL et du POSTE D'ACCES.

Quand ces systèmes *tourment à vide*, c'est-à-dire quand toutes les tâches sont lancées, mais qu'il n'y a aucune "connexion" à gérer soit du côté du FRONTAL, soit du côté du POSTE D'ACCES, seules les tâches du groupe "a" sont activées. Du fait que ces tâches n'ont pas de travail effectif, la relation entre la durée d'un intervalle d'observation et le nombre total de tâches activées nous a fourni une estimation du temps moyen nécessaire pour la commutation entre tâches.

On a déduit que l'Unité Centrale n'est pas saturée, car même quand les applications atteignent leur charge maximale, dans un intervalle d'observations, nombreuses sont les tâches qui rendent le contrôle par elles-mêmes (c'est-à-dire qu'elles n'épuisent pas leur quantum de temps).

L'important écart observé dans toutes les mesures entre le nombre d'activation des tâches du groupe "a" et celui d'une tâche quelconque du groupe "b" (voir tableau I.2) a confirmé nos hypothèses qu'une meilleure utilisation de l'Unité Centrale exigerait que les conditions déterminant l'activité d'une tâche puissent être exprimées à l'aide d'événements ou de signaux reconnus par le moniteur. De cette façon, on pourrait garantir qu'une tâche ne sera activée que lorsqu'elle aura un traitement à réaliser.

Activité du FRONTAL et du POSTE D'ACCES	Tâches du groupe "a"	Tâches du groupe "b"
Aucune liaison ouverte	112	0
Plusieurs liaisons ouvertes (6) sans activité	88	1
Plusieurs liaisons ouvertes et un seul appareil virtuel émet ou reçoit de données	74	1

Tableau I.2 - Nombre moyen d'activations d'une tâche par intervalle d'observation (durée de l'intervalle : 0,5 seconde)

I.3. TAUX D'OCCUPATION DES FILES D'ATTENTE

Dans un double souci de prévenir les risques de blocage et d'améliorer le débit des échanges du FRONTAL et du POSTE D'ACCES avec leurs correspondants respectifs, nous avons étudié le taux d'occupation de différentes files d'attente.

Nous présentons dans la figure I.1 les files qui ont fait l'objet de mesures. Ces files peuvent être rangées en deux classes en fonction des phénomènes particuliers que leur sous-dimensionnement provoque.

Nous appelons *classe "a"* l'ensemble des files d'attente de "cases libres" et *classe "b"* l'ensemble des files d'attente de messages.

Le but de nos mesures était d'étudier l'évolution de l'occupation des files d'attente en fonction de la charge du FRONTAL et/ou du POSTE D'ACCES ainsi que d'évaluer la fréquence de certaines situations critiques.

Les situations critiques auxquelles ont fait allusion correspondent aux cas où

- les files d'attente de classe "a" sont vides
- et celles de classe "b" sont pleines.

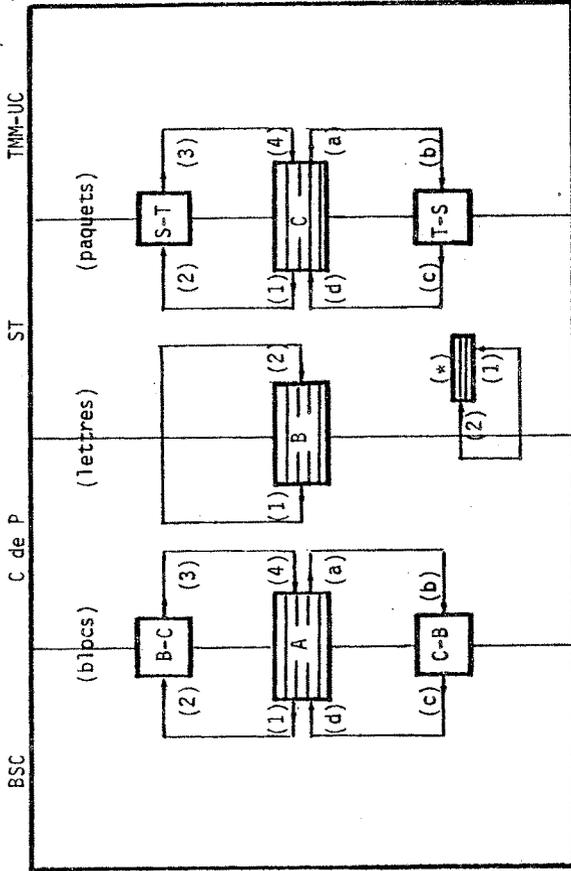
Mesures effectuées

Pour cela, comme dans l'étude précédente, on a enregistré pour chaque file d'attente i l'état atteint par les compteurs suivants dans chaque inter-
valle de temps j observé :

- L_{ij} : le nombre de cases disponibles
- Min_{ij} : les nombres minima et maxima de cases observées
- Max_{ij} : les nombres minima et maxima de cases observées
- DEF_{ij} : le nombre d'opérations de prélèvements et dépôt d'une case
- ENP_{ij} : le nombre d'opérations de prélèvements et dépôt d'une case
- FV_{ij} : le nombre de fois qu'un consommateur de cases libres ne trouve pas de case disponible (file vide) et le nombre de fois qu'un producteur ne peut pas déposer un message (file pleine).
- FP_{ij} : le nombre de fois qu'un consommateur de cases libres ne trouve pas de case disponible (file vide) et le nombre de fois qu'un producteur ne peut pas déposer un message (file pleine).

L'analyse de ces mesures nous a montré que les files étaient surdimensionnées, aucun événement critique n'a été enregistré même lorsque le système tournait à pleine charge.

Le dimensionnement des files d'attente avait été le résultat d'un calcul théorique où l'on surestimait la durée du circuit parcouru par une case.



(*) File d'attente par liaison (son niveau détermine le crédit)

▨ File d'attente de cases fibres (classe "a")

▭ File d'attente de messages (classe "b")

Figure I.1 - Les files d'attente dont le taux d'occupation a été mesuré

Analyse des mesures

Les mesures prises nous ont permis de faire les calculs et les estimations selon les formules suivantes :

L_i - Longueur moyenne de la file i

$$L_i = \sum_{j=1}^N \frac{L_{ij}}{N} \quad N = \text{nombre d'observations}$$

A_i - Débit de cases arrivant dans la file i par seconde

$$A_i = \sum_{j=1}^N \frac{ENF_{ij}}{T} \quad T = N \times m$$

$m = \text{durée de l'intervalle d'observation}$

D_i - Débit de cases sortant de la file i par seconde

$$D_i = \sum_{j=1}^N \frac{DEF_{ij}}{T}$$

Remarque : Pour un nombre suffisamment grand d'observations, A_i doit être approximativement égal à D_i (en "régime permanent").

TA_i - Temps d'attente moyen par case prélevée de la file i

$$TA_i = \frac{L_i}{D_i} \quad (*)$$

TT_i - Temps de traitement moyen d'une case de la file i (calculable seulement lorsque le nombre de cases (M_i) de la file i en attente et en traitement est constante)

$$TT_i = \frac{M_i}{D_i} - TA_i \quad M_i = \text{nombre de cases attachées à la file } i$$

Min_i - Nombre minimum de cases observées dans la file i sur l'ensemble des mesures effectuées.

$$Min_i = \min \{Min_{ij} / \forall 1 \leq j \leq N\}$$

(*) Note : Application de la formule de Little "Opérationnelle" [BUZEN, J.P. "Fundamental Laws of Computer System Performance", Proceedings IFIP-ACM SIGMETRICS, Cambridge, Mass., March 1976].

File d'attente i	Type i	M _i	A _i	D _i	L _i	TA _i	TT _i	Min _i
A	cases libres	29	1,88	2,32	24,06	10,37	2,13	20
B	cases libres	19	4,35	4,15	8,02	1,93	(*)	3
C	cases libres	13	5,16	5,36	10,32	1,92	0,5	6

Tableau I.2 - Quelques résultats des mesures relatives aux files d'attente

(*) Le calcul de TT_i pour la file d'attente B n'est pas représentatif puisqu'un certain nombre de cases (égal au crédit maximal) sont prélevées de cette file lors de l'ouverture d'une liaison et y sont restituées seulement lorsque cette liaison est libérée.

I.4. ETUDE DU CONTROLE DE FLUX

Simultanément aux mesures prises pour étudier le taux d'occupation des files d'attente, on a enregistré des données afin d'analyser le comportement des appareils périphériques du POSTE D'ACCES en particulier et le débit sur les différentes "liaisons" ouvertes en général.

Les données prélevées régulièrement correspondaient au nombre de cartes lues et lignes imprimées par intervalle, les nombres de lettres traitées dans la même période et le nombre de fois que le crédit d'une liaison devenait nul.

A la suite de ces observations, on a introduit une modification au niveau de l'interface "abonné local - Station de Transport" concernant la requête par laquelle l'abonné restitue une "case libre" à sa file d'attente. Quand cette file d'attente est vide (c'est-à-dire le crédit octroyé était nul) lors de la restitution d'une case, on force l'envoi immédiat d'un crédit positif pour accélérer le déblocage de la liaison correspondante. Cela permet d'améliorer le débit de l'imprimante pour qu'il approche la vitesse nominale de l'appareil.

Mais le débit des échanges sur les liaisons de transport n'est pas un problème simple à étudier, car tous les éléments qui peuvent influencer l'évolution du "crédit" ne sont pas observables au niveau du FRONTAL ou du POSTE D'ACCES. Dans le cas où les appareils du POSTE sont connectés au serveur HASP à travers le FRONTAL, on est dans une situation privilégiée, mais là encore on n'a pas tous les renseignements qui caractérisent la charge du système HASP.

A vue, on a constaté que la vitesse du lecteur de cartes et de l'imprimante varie sensiblement selon le serveur auquel les appareils sont connectés et aussi selon l'heure de connexion.

A propos du crédit, on n'a pas pu établir expérimentalement une valeur optimale ; pour cela, il faudrait réaliser des mesures dans les différents sites concernés pendant une longue période afin de faire apparaître toutes les variations possibles.

Une autre façon de résoudre cet aspect est d'incorporer un mécanisme "régulateur" ou "auto-adaptatif" au niveau du protocole de transport ou de l'application.

I.5. QUELQUES REMARQUES

Dans le cas de la présente étude, on ne s'est pas soucié de la dégradation des performances introduites par les mesures elles-mêmes et cela dû au fait que l'on ne mesurait pas la durée ponctuelle de certaines actions ou événements, mais on est conscient qu'il s'agit d'un phénomène important dont il faut tenir compte.

Dans le cas d'un mini-ordinateur ayant une horloge interne peu adéquate pour le travail de mesures, l'utilisation des appareils tels que le "Basic Counter Unit" [IBM e]^(*) peut constituer une solution à condition que les événements à mesurer soient détectables au niveau de signaux physiques (positionnement d'un bus, détection d'une interruption, etc).

(*) "IBM 2989 Basic Counter Unit" (BCU) est un dispositif qui permet de prélever directement des informations sur une machine à l'aide de sondes, sans en gêner le fonctionnement, de transformer ces informations à l'aide de fonctions logiques et de les transmettre à 16 compteurs. Ces informations peuvent être visualisées ou bien transmises à une perforatrice de cartes.

CHAPITRE II

EVALUATION DU COÛT DU PROJET EN RESSOURCES HUMAINES ET MATÉRIELLES

Le but de ce chapitre est de donner une estimation du temps passé dans les différentes étapes du projet et de préciser le volume du code de chaque application.

Nous tenons à souligner que dans l'interprétation des valeurs fournies pour le temps consacré au développement, il faut tenir compte de facteurs difficilement mesurables tels que la connaissance préalable qu'on avait du calculateur ORDO 400 et de son moniteur, ou l'assistant et la collaboration active de chercheurs ayant participé à d'autres projets-réseau.

Nous devons signaler aussi que le constructeur de l'ORDO 400 a mis à notre disposition le code correspondant aux procédures de ligne BSC et TMM-UC, ainsi que celui d'un Terminal Louré HASP. Une partie importante de ce dernier système a été adaptée pour la réalisation du sous-système "Conversion de Protocoles" (voir Tableau I.1).

Deux facteurs importants ont marqué le rythme et l'évolution de la mise au point :

- le temps limité pendant lequel on disposait du mini-calculateur ORDO 400 (car il s'agissait d'une machine mise en exploitation)
- le nombre important de calculateurs (ORDO 400, IBM 360, MITRA 15, IRIS 80, ...), de périphériques et de lignes de transmission simultanément en activité pour réaliser nos tests et mesures (il suffisait qu'un élément soit en panne ou en maintenance pour que le travail soit reporté).

Nous voulons ajouter que pour la traduction du FRONTAL et du POSTE D'ACCES nous avons bénéficié de l'assistance des ingénieurs-systèmes connaissant le mini-calculateur CO/ORDINATEUR 500 [ORD d] et son moniteur METEOR III [ORD e].

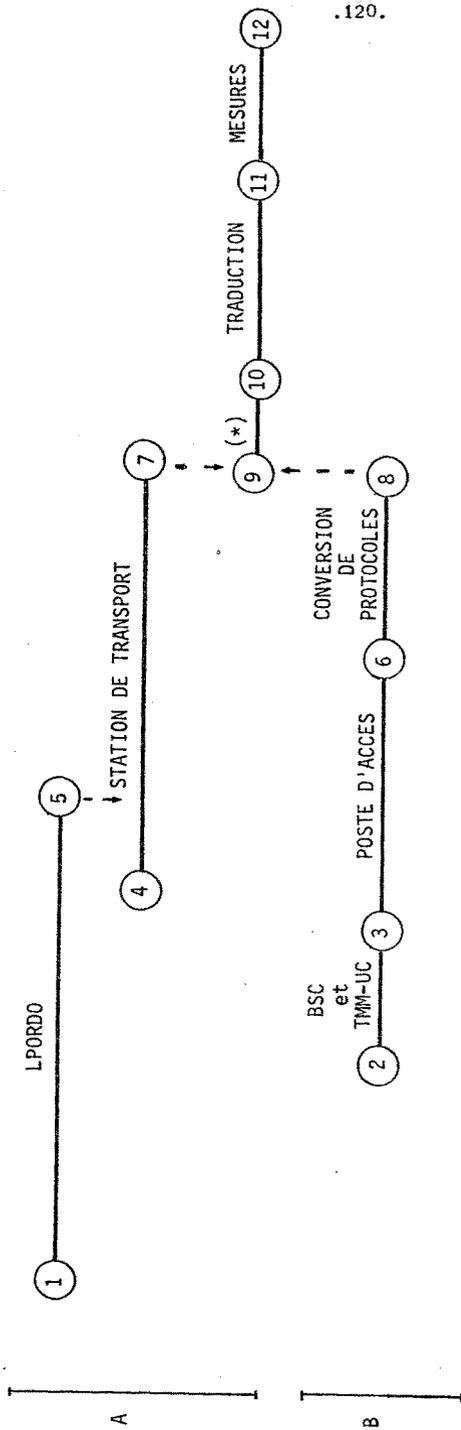
Pour que le volume du code détaillé dans le tableau suivant puisse évoquer d'une certaine façon la complexité ou la longueur des algorithmes codés, on rappelle qu'une instruction machine ORDO 400 occupe deux ou quatre octets-mémoire.

Les renseignements fournis dans le tableau suivant sont complétés par le graphe de la Figure II.1 où l'on précise l'ordre dans lequel les différentes étapes concernant la réalisation du projet ont été réalisées.

LOGICIEL	TEMPS (HOMME-MOIS)			VOLUME (K octets)		LANGAGE
	DEFINITION	DEVELOPPEMENT	MISE AU POINT	du code	des tampons	
OUTILS	3	3	3	4000 lignes	STAGE 2	Stage 2
	1	1,5	1	5		Ass. ORDO 400
APPLICATIONS	2	3	3	13,5	1,5	LPORDO
	1	2	2	4,75	1,03	Ass. ORDO 400
	1	1	1,5	4,2	0,28	Ass. ORDO 400
	—	—	2,5	3,3	3,06	Ass. ORDO 400
MESURES	1 mois	0,5	1,5	3,94		Ass. ORDO 400
TRADUCTION		4				Ass. CO/500

(*) Note : Une partie importante de ce logiciel a été développée par le constructeur

Tableau II.1 - Evaluation du coût du projet



- A - Travail réalisé par un ingénieur approximativement en 23 mois
- B - Travail réalisé par un ingénieur approximativement en 11 mois
- (*) Intégration de l'ensemble du logiciel
Début ou fin d'une étape

Figure II.1 - Séquencement des différentes étapes de la réalisation du FRONTAL et du POSTE D'ACCES BANALISE

CONCLUSIONS

Nous avons présenté la définition d'un Service Réseau de Traitement par Lots basée sur la structure d'adressage hiérarchique d'un réseau général et hétérogène. Nous proposons également quelques possibilités d'extension.

L'adaptation d'un Système de Traitement par Lots que nous avons effectuée à l'aide d'un ordinateur FRONTAL, pour réaliser un serveur réseau, nous a permis d'illustrer une méthode d'adaptation externe qui ne nécessite ni la modification du logiciel ni celle du matériel du ordinateur à raccorder au réseau. Nous avons eu l'occasion de signaler les avantages de cette solution (libérer le ordinateur hôte de certains traitements, garantir le temps de réponse imposé par une application temps réel, ...) et de préciser aussi les conditions de son applicabilité (les services des deux systèmes à interconnecter doivent être traduisibles l'un dans l'autre à travers leurs interfaces externes).

Lors de la présentation de chaque étape de la réalisation, nous avons décrit les solutions adoptées et nous en avons fait un examen critique.

Nous voulons conclure cette thèse en ébauchant pour chacun des trois volets de ce travail (définition, développement et évaluation) les axes de recherche à poursuivre ou à entreprendre, susceptibles de nous apporter des solutions plus satisfaisantes pour le développement d'applications réparties, particulièrement pour celles destinées à s'exécuter sur des mini ou des micro-processeurs.

a) Définition

Les protocoles des applications-réseau sont voués de par leur fonction à être implantés sur différents ordinateurs.

Aussi, pour que les protocoles puissent remplir correctement leur rôle, il est essentiel que leur définition soit formelle, claire et concise.

Nous sommes conscients des efforts entrepris pour établir un formalisme adéquat pour la définition de protocoles-réseau (Ex : description par automate d'états finis [BOC78], description par une grammaire hors contexte [HAR78]). Si aucun de ces formalismes ne s'adapte pour la

description d'un protocole donné, nous suggérons l'utilisation combinée de différents formalismes existants.

Par exemple, un réseau de Petri Temporisé [MER76], [SIF79] peut convenir pour la description des relations dynamiques ou interactions, alors qu'un langage de haut niveau est mieux adapté pour définir les actions associées à chaque "transition".

b) Développement

Les difficultés rencontrées lors du développement du *FRONTAL* en raison de l'insuffisance des outils disponibles nous font proposer que les efforts consacrés pour la réalisation de systèmes intégrés d'outils [SAM71], [KRA75], qui incluent des langages de programmation ainsi que des outils pour l'édition, documentation, gestion et mise au point de programmes, soient orientés pour que ces systèmes puissent fournir :

- d'une part des outils pour la production de logiciel portable (ces outils peuvent être implantés sur de gros calculateurs)
- et d'autre part des outils portables pour la mise au point, adaptés aux particularités des applications réparties (ces outils doivent nécessairement s'exécuter sur le calculateur d'implantation du logiciel à mettre au point).

c) Mesures

Nous considérons qu'il est important pour des raisons d'ordre théorique et d'ordre pratique d'inclure, dans un projet de développement de logiciel d'envergure, un travail de mesures lors de sa conception.

Les raisons d'ordre théorique sont fondées sur le fait que la préparation et la définition d'un tel travail (objets à observer, critères de mesures, ...) facilitent la mise en évidence des ressources critiques, des paramètres dont le dimensionnement est susceptible d'influencer fortement les performances de l'application, et cette connaissance se reflètera sans doute dans les choix pour la réalisation du projet.

Les raisons d'ordre pratique sont fondées sur le fait que la prévision du code nécessaire pour les mesures et/ou leurs points d'insertion dans le corps du projet épargne les difficultés de modification du code a posteriori.

Par exemple, on pourrait prévoir que certains paramètres dont le dimensionnement optimal ne peut être défini que d'une façon expérimentale, soient "facilement modifiables" lors de l'activation du système. On éviterait ainsi que de longues et fastidieuses opérations de génération système s'ensuivent toutes les fois qu'on désire étudier les propriétés dynamiques du système, en attribuant de nouvelles valeurs aux paramètres en question.

Il serait souhaitable que les langages de programmation incluent des instructions ou des directives compilables optionnellement, pour la prise de mesures et la collection de données statistiques (durée d'une action, fréquence d'accès à une procédure, oscillation maximale du niveau d'une pile, ...) afin d'encourager l'évaluation systématique des produits logiciels. Car nous pensons aussi que "l'observation des phénomènes et l'enregistrement systématique des observations faites est un facteur essentiel ... pour la progression des sciences expérimentales et ... des techniques" [BOK75].

BIBLIOGRAPHIE

- [ADI78] ADIBA M.
"Un modèle relationnel et une architecture pour les systèmes de bases de données réparties".
Thèse d'Etat, USMG-INPG, Grenoble, Septembre 1978.
- [ANS76] ANSART J.P.
"Système interactif dans un environnement réseau".
Thèse Docteur-Ingénieur, USMG-INPG, Grenoble, Février 1976.
- [BOC78] BOCHMANN G.V.
"Finite State Description of Communication Protocols".
Proceedings of Computer Network Protocols, Liège, Belgique, Février 1978.
- [BOK75] BOKSENBAUM C.
"A propos de la simulation d'un système informatique".
Thèse d'Etat, INPG-USMG, Grenoble, Septembre 1975.
- [BOL74] BOLT, BERANEK & NEWMAN INC.
"The Remote Job Entry mini-host".
Quarterly Technical Report N°6. Report N°2852, Juillet 1974, pp. 25.
- [CHA76] CHAMBON J.F., LE BIHAN B.
"Architecture d'un Frontal en environnement télé-informatique".
Thèse de Docteur-Ingénieur, Ecole des Mines de St-Etienne, Octobre 1976.
- [CHU77] CHUPIN J.C.
"Répartition d'applications et de bases de données sur un réseau général d'ordinateurs".
Thèse d'Etat, USMG-INPG, Grenoble.
- [CII] CII
"Générateurs du Système SIRIS8".
Version C10 - Manuel d'utilisation et d'opérateur.
56F2172 - Rev2.
- [CRO75] CROCUS
"Systèmes d'exploitation des ordinateurs. Principe de conception".
(Ouvrage collectif). Dunod 1975.

- [DAN76] DANG Ng.X., QUINT V., SEGUIN J., SERGEANT G.
"Présentation et Définition de SYNCOP".
Rapport de Recherche n° 64, INPG-USMG, Grenoble, Novembre 1976
- [DAN78] DANG NG.X.
"Système et Langage portable pour le Traitement des Application Réparties".
Thèse de Docteur de 3ème Cycle, INPG, Grenoble, Mars 1978.
- [DEC73] DE CALUWE R.
"Etude du Langage de Commande et de Contrôle pour le Réseau d'Ordinateurs SOC".
Thèse de Docteur de 3ème Cycle, USMG, Grenoble, Septembre 1973.
- [DEN79] DENIS J.J., GIBERGUES O.
"Principes de conversion entre les protocoles CYCLADES et X25".
Proceedings of the International Symposium on Flow Control in Computer Networks, Versailles, France, Février 1979, pp. 379-38
- [FOU76] FOURNIER R.
"Le Traitement par lots dans un Réseau Hétérogène - Implémentation du Serveur OS-MVT sur IBM 360/67 pour le Réseau CYCLADES".
Thèse de Docteur-Ingénieur, USMG-INPG, Grenoble, Décembre 1976.
- [GUI75] GUIBOUD-RIBAUD S.
"Système de réseaux de mini-ordinateurs spécialisés".
Note interne - Ecole des Mines de St-Etienne, Décembre 1975.
- [GUI76] GUILBERT, MACCHI Editeurs (Ouvrage collectif)
"Séminaire de télé-informatique".
IRIA-AFCET-CNET-IP, Bonas, Gers, Septembre 1976.
- [GRI68] GRIFFITHS M., PELTIER M.
"Grammar Transformation as an aid to compiler production".
Université de Grenoble, Octobre 1968.
- [HAR78] HARANGOZO J.
"Protocol Definition with Formal Grammars".
Proceedings of Computer Network Protocols, Liège, Belgique, Février 1978.

- [IBM a] IBM
HASP II Program Logic.
GY 27-7255.
- [IBM b] IBM
BSC - Binary Synchronous Communication.
GA 27-3007.
- [IBM c] IBM
Control Program-67/Cambridge Monitor System
360 D-05.2.005.
- [IBM d] IBM
2989 Basic Counter Unit
Y27-2236-0.
- [ISO78] ISO
"Reference Model of Open Systems Architecture" - Version 3.
ISO/TC97/SC16/WG1, Novembre 1978.
- [KRA73] KRAKOWIAK S.
*"Conception et réalisation de systèmes à accès multiples :
allocation de ressources".*
Thèse d'Etat, Paris, 1973.
- [KRA75] KRAKOWIAK S., MOSSIERE J.
"Systèmes d'aide à la programmation de systèmes".
Journée sur les systèmes d'aide à la programmation, IRIA,
Paris, 1975.
- [LUM76] LUMELSKY S.
"Définition du langage LPORDO".
Note interne, INPG, Grenoble, Juin 1976.
- [LUM77] LUMELSKY S.
*"Requêtes d'accès à la Station de Transport CYCLADES d'un
ORDO 400".*
Note interne, INPG, Grenoble, Novembre 1977.

- [LUM78] LUMELSKY S.
"Spécifications de la mise en oeuvre d'une Station de Transport CYCLADES sur un ORDO 400".
Note interne, INPG, Grenoble, Mai 1978.
- [MEL79] MELLOR F., MATSUBARA M.M.
"Flow control of access services in SL-10 networks".
Proceedings of the International Symposium on Flow Control in Computer Networks, Versailles, France, Février 1979, pp. 161-171.
- [MER76] MERLIN P.M.
"A Methodology for the Design and Implementation of Communication Protocols".
IEEE Transactions on Communications, Vol. COM-24, N°6, Juin 1976.
- [MOS77] MOSSIÈRE J.
"Méthode pour l'écriture des systèmes d'exploitation".
Thèse d'Etat, INPG-USMG, Grenoble, Septembre 1977.
- [NAF77] NAFFAH N.
"Protocol converters and user Interface in the CYCLADES Network".
Document IRIA : SCH 606, Novembre 1977.
- [ORD a] ORDO
Manuel de Présentation Générale.
ORDO 16, SFENA D.S.I - ORDOPROCESSEURS, Réf.: 145/A0/F.
- [ORD b] ORDO
Logiciel METEOR 1 Volume I.
Réf.: 114/T1/F.
- [ORD c] ORDO
Logiciel ORDOPROCESSEURS Volume II.
Aides à l'exploitation. Réf.: 115/T0/F.
- [ORD d] ORDO
Manuel de Présentation Générale du Système CO/ORDINATEUR 500.
Réf.: 220/T0/F.

- [ORD e] ORDO
Logiciel : Manuel de Référence METEOR 3. Réf.: 213/T1/F.
- [PAP74] PAPACRISTODOULOU Z.
"TECOM : Un support de terminaux lourds sous un système à temps partagé".
Thèse de Docteur-Ingénieur, INPG, Grenoble, Octobre 1974.
- [PO070] POOLE P.C., WAITE W.M.
"The STAGE 2 Macroprocesseur, User Reference Manual".
CLM-PDN 6/70 Culham Laboratory, Juillet 1970.
- [POU73] POUZIN L.
"Architecture et variétés de réseaux".
Automatisme - Tome XVIII - N°12, Décembre 1973.
- [POU74] POUZIN L.
"CIGALE, the packet switching machine of the CYCLADES computer network".
IFIP Congress, Stockholm, Août 1974, pp. 154-159.
- [QUI78a] QUINT V.
"Manuel opérateur du POSTE D'ACCES".
Centre Interuniversitaire de Calcul de Grenoble, Avril 1978.
- [QUI78b] QUINT V.
"Convertisseur d'Interface CYCLADES-HASP. Spécifications de réalisation".
Note Interne, INPG, Avril 1978.
- [RAY73] RAYMOND J.
"NJCL, Un langage de commandes pour réseau d'ordinateurs".
Contrat CRI, Grenoble, Juillet 1973.
- [RAY77] RAYNER D.
"An introduction to network job control language".
NPL report COM 88. NPL, Teddington, Mars 1977.

- [RIC78] RICHY H.
"Confiabilité, Bases de Données et Réseaux d'Ordinateurs".
Thèse de Docteur-Ingénieur, INPG, Grenoble, Février 1978.
- [ROS78] ROSENTHAL R., LUCAS B.D.
"The Design and Implementation of the National Bureau of Standards' Network Access Machine (NAM)".
Computer Science & Technologie, U.S. Department of Commerce,
Juin 1978.
- [SAE79] SAETTONE R.
"Conception d'un périphérique intelligent d'accès à un réseau télé-informatique".
Thèse de Docteur-Ingénieur, INPG, Grenoble, Juillet 1979.
- [SAM71] SAMMET J.E.
"A brief survey of languages used in systems implementation".
ACM, Volume 6, Number 9, Octobre 1971.
- [SEG78] SEGUIN J.
"Traitement distribué d'informations réparties dans les réseaux d'ordinateurs".
Thèse d'Etat, USMG-INPG, Grenoble, Mars 1978.
- [SIF79] SIFAKIS J.
"Le contrôle des systèmes asynchrones : Concepts, Propriétés, Analyse Statique".
Thèse d'Etat, INPG-USMG, Grenoble, Juin 1979.
- [SOM75] SOMIA BRECHET M.
"Etude du système de contrôle d'un réseau d'ordinateurs distribué et de ses relations avec les systèmes de contrôle locaux".
Thèse de Docteur ès-Sciences, Paris IV, Juin 1975.
- [WIR68] WIRTH N.
"PL360, A Programming Language for the 360 Computers".
JACM, Vol.15, n°1, Janvier 1968.

- [ZIM75a] ZIMMERMANN H.
"Protocol-Appareil Virtuel dans CYCLADES".
Journées AIM Mini-ordinateurs, Liège, Janvier 1975.
- [ZIM75b] ZIMMERMANN H.
"The CYCLADES End-to-End Protocol".
Proceeding of ACM/IEE Fourth Data Communication Symposium,
Québec, Octobre 1975, pp. 721-726.
- [ZIM78] ZIMMERMANN H., NAFFAH N.
"On Open Systems Architecture".
ICC 78, Kyoto, Japon, Novembre 1977.

ANNEXE A

LE FRONTAL ET LE MODÈLE DE RÉFÉRENCE ISO

Dans cette annexe, nous décrivons les fonctions assignées à chacun des niveaux du Modèle de Référence ISO associés à la couche "Conversion de Protocoles" du FRONTAL, nous présentons aussi un schéma résumant les échanges effectués au niveau de leurs interfaces pour mettre en évidence le parcours suivi par les différents types de messages (voir Tableau A.2).

On rappelle que, lors de la définition du FRONTAL, un résumé des fonctions de chacun de ces niveaux a été présenté (cf. III.3 - Partie I).

A.1 - Niveau Application

Fonctions : Le Niveau Application met en oeuvre le *protocole privé* du serveur HASP (cf. III.2.3 - Partie I). Ses fonctions concernent essentiellement les traitements des demandes de connexion effectuées par les terminaux lourds-réseau et leur répercussion au système HASP.

a - Ouverture et fermeture des portes (Interface A → S)

Une session HASP est déclarée *ouverte* au réseau dès que les trois portes du serveur-réseau ("BT", "CR" et "LP") sont activées auprès de la Station de Transport.

Lorsque ces portes sont désactivées, toute nouvelle demande de connexion est refusée au Niveau Transport.

Le Niveau Application est à l'initiative de ces opérations d'ouverture et de fermeture.

b - Traitement des demandes de connexion (Interface A ← S)

b.1 - Connexion d'un terminal lourd-réseau

Toute demande pour l'établissement d'une liaison, adressée à la porte "BT" (associée à la console), est assimilée à une demande pour connecter un terminal.

Lors de l'arrivée de cette demande, le Niveau Application réalise une série de contrôles visant à déterminer, par exemple :

- si le terminal n'est pas déjà connecté,
- s'il s'agit d'un "client autorisé",
- si une ligne de transmission BSC est libre
-

A la suite de ces vérifications, la demande peut être acceptée ou refusée.

Dans le cas où elle est acceptée, une ligne de transmission BSC est réservée^(*), au travers de laquelle un message initial est émis vers le Système HASP, présentant le terminal lourd-réseau, par son nom HASP.

b.2 - Connexion d'un lecteur de cartes ou d'une imprimante

Les demandes pour l'établissement des liaisons adressées au portes "CR" et "LP" (associées respectivement au lecteur de cartes et à l'imprimante virtuels) sont l'objet d'autre type de vérifications. Notamment, on contrôle que le terminal, adressant cette demande, possède déjà sa console connectée.

c - Traitement des messages

c.1 - Messages reçus par l'interface BSC (Interface B → A)

Les messages délivrés par la procédure BSC sont triés :

- les *données* sont transmises au Niveau Présentation pour leur transcription (Interface A → P)
- les messages de *contrôle* sont interprétés et répercutés si nécessaire au niveau Session (contrôle de flux) (Interface A → S).

c.2 - Messages reçus par l'interface Présentation (Interface A ← P)

Le Niveau Présentation remet au niveau Application des enregistrements MULTI-LEAVING, destinés au système HASP.

Plusieurs "enregistrements" concernant différents "flots" d'un même terminal lourd-réseau peuvent être groupés, pour leur émission, dans un seul "bloc-MULTI-LEAVING".

(*) Note : Le *FRONTAL* prévu pour le *CIRCE* devait être raccordé à l'ordinateur hôte du système *JES* par trois lignes de transmission "BSC".

c.3 - Messages reçus par l'interface Session (Interface A + S)

Les messages transmis par le Niveau Session au Niveau Application concernent les demandes de connexion adressées aux différentes portes (cf. paragraphe b) ou la gestion des liaisons relatives aux appareils périphériques des terminaux connectés (contrôle de flux, ...).

Dans la plupart des cas, le système HASP est aussi averti de l'arrivée de ces messages.

A.2 - Niveau Présentation

Fonctions : Le Niveau Présentation transcrit le contenu des "lettres" codées selon le protocole d'appareil virtuel (PAV) du réseau CYCLADES dans le protocole MULTI-LEAVING du serveur HASP (Interface P + S) et effectue le traitement inverse pour les "enregistrements MULTI-LEAVING (Interface A + P).

Sens de la traduction PAV - MULTI-LEAVING	Selon la classe d'appareil virtuel
→	clavier } console écran } lecteur de cartes imprimante
→	
→	
←	

Tableau A.1 - Traduction des protocoles réalisés
au niveau du serveur-réseau HASP

A.3 - Niveau Session

Fonctions : Le rôle de ce niveau est de faciliter le dialogue entre le serveur et les terminaux lourds-réseau. Pour ceci, il met en oeuvre le protocole du Service de Traitement par Lots défini pour le réseau CYCLADES (cf. II.5 - Partie I).

a - Messages reçus par l'interface Transport (Interface S ← T)

a.1 - Demandes de connexion

Le niveau Session, avant de transmettre au Niveau Application les demandes de connexion reçues à travers la Station de Transport, vérifie leur conformité aux normes établies dans le protocole cité dans le paragraphe précédent (Ex : longueur maximale des lettres en fonction de la porte adressée).

a.2 - Contrôle du dialogue

Le Niveau Session gère et contrôle le dialogue sur les différentes liaisons, selon le mode de fonctionnement propre à chaque classe d'appareil virtuel (cf. II.5.2 - Partie I).

a.3 - Messages de contrôle

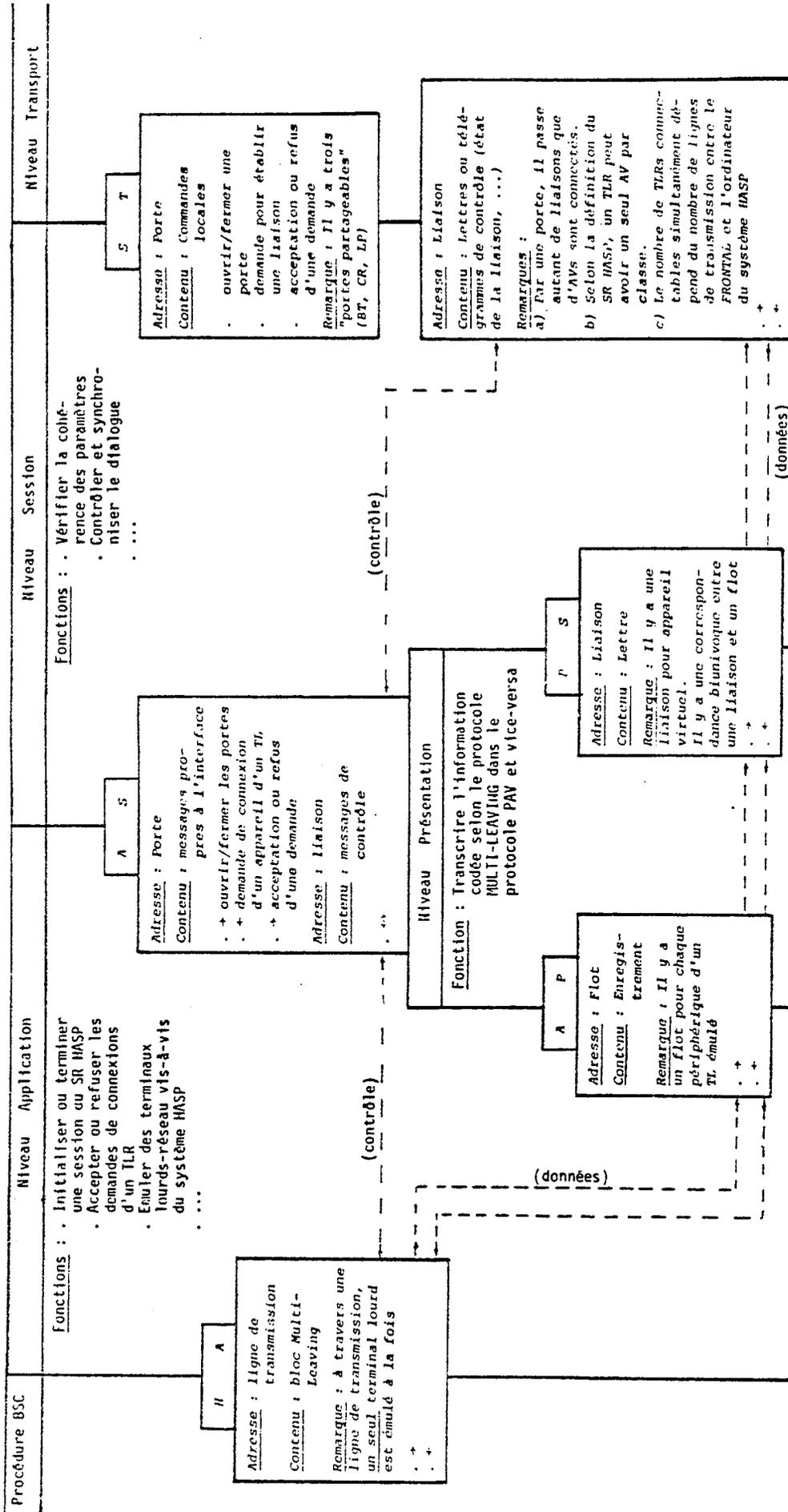
Les messages de contrôles reçus sont retransmis au Niveau Application (contrôle de flux, fermeture d'une liaison, ...), à l'exception de ceux concernant la gestion du dialogue qui sont traités uniquement au niveau local.

a.4 - Lettres

Parmi les lettres délivrées par la Station de Transport, seules celles qui contiennent des données sont soumises au Niveau Présentation pour leur traduction.

b - Messages reçus par l'interface Présentation (Interface P → T)

Les lettres délivrées par le Niveau Présentation sont émises sur le réseau à travers la Station de Transport.



AV - Appareil Virtuel
 SR - Serveur
 TLR - Terminal lourd-réseau
 TL - Terminal lourd

Tableau A.1 - Correspondance entre la couche "Application" (Réseau CYCLADES) et les Niveaux "Session", "Présentation" et "Application" (Modèle ISO)

ANNEXE B

LE POSTE D'ACCÈS ET LE MODÈLE DE RÉFÉRENCE ISO

Dans cette annexe, nous présentons un découpage fonctionnel de la couche Application du POSTE D'ACCES (cf. IV.5 - Partie I) dans les trois niveaux du Modèle de Référence ISO orientés vers le traitement de données : *Application, Session et Présentation*.

B.1 - Niveau Application

Fonctions : Le rôle du Niveau Application peut être assimilé à celui de "moniteur du POSTE D'ACCES", car il effectue le dialogue avec l'opérateur et gère les différents périphériques supportés (réels ou virtuels).

Les particularités de sa gestion sont fonction essentiellement de la définition adoptée pour le POSTE D'ACCES (périphériques réels supportés, appareils virtuels émulables simultanément, jeu de commandes-opérateur, ...) et aussi du protocole réseau défini pour le Service de Traitement par Lots (cf. II.4 - Partie I).

a - Dialogue avec l'opérateur

Le dialogue "opérateur-moniteur" s'effectue au travers de la console. Le rôle du moniteur est d'interpréter et d'exécuter les commandes-opérateur.

a.1 - Commandes opérateurs (Interface O → A)

L'opérateur possède la liste des appareils émulables par le POSTE D'ACCES. Il formule les commandes de connexion, déconnexion, interrogation en précisant le nom de l'appareil concerné (cf. IV.3 - Partie I).

Le moniteur vérifie la correction syntaxique de ces commandes et leur faisabilité, c'est-à-dire la disponibilité des appareils virtuels et réels pour satisfaire ces demandes.

Par exemple, quand une demande de connexion est acceptée, le moniteur réserve le périphérique réel et l'appareil virtuel nécessaires. Postérieurement, il ordonne au Niveau Session d'initialiser les démarches auprès de la Station de Transport pour l'établissement d'une liaison (entre la "porte" locale associée à l'appareil virtuel et celle du Serveur adressé dont le nom correspond à la classe d'appareil émulé localement) (Interface A → S).

Certaines commandes (attacher le clavier à une console virtuelle, ...) sont traitées seulement au niveau local, celles qui concernent le réseau sont systématiquement répercutées au Niveau Session.

a.2 - Messages à l'opérateur (Interface O ↔ A)

- * Toutes les commandes effectuées par l'opérateur sont acquittées. La réponse est immédiate seulement si la commande est traitée localement.
- * Un message prévient l'opérateur quand un appareil périphérique requiert son intervention.

b - Gestion des périphériques réels et virtuels

Le Niveau Application est chargé d'effectuer la gestion des périphériques réels et virtuels, c'est-à-dire d'enregistrer la correspondance entre les appareils réels actifs et les appareils virtuels émulés et de lancer les opérations d'entrée-sortie sur ces appareils.

b.1 - Opérations d'entrée-sortie (Interface P_e ↔ A)

Console : La console pouvant être partagée entre deux consoles virtuelles et servant aussi à rentrer les commandes-opérateur, une première analyse est faite pour déterminer si le texte frappé au clavier est une commande destinée au moniteur. Dans le cas où le message est adressé à un serveur, il est nécessaire de déterminer la console virtuelle à laquelle le clavier est rattaché.

Les messages rentrés par la console qui doivent être envoyés par le réseau ainsi que les cartes lues sont remis au Niveau Présentation pour leur traduction (Interface A → P).

Le Niveau Présentation délivre à son tour les données qui doivent être listées sur l'imprimante ou affichées sur l'écran (Interface A ↔ P).

B.2 - Niveau Présentation

Fonctions : Le Niveau Présentation fait la transcription des données codées dans le protocole de l'émetteur au protocole du récepteur.

Les cartes ou lignes lues, délivrées par le Niveau Application, sont traduites dans le protocole d'Appareil Virtuel et rangées dans des "lettres" remises au Niveau Session (Interface P → S).

Par ailleurs, les "lettres" provenant d'un serveur reçues à travers l'interface Session sont traduites dans le format adéquat pour leur présentation sur l'appareil réel concerné.

B.3 - Niveau Session

Fonctions : Le Niveau Session suit les normes établies dans le protocole CYCLADES du Service de Traitement par Lots (cf. II.4 - Partie I) pour les terminaux lourds-réseau.

a - Messages émis vers la Station de Transport (Interface S → T)

a.1 - Etablissement ou libération d'une liaison

Les commandes pour connecter ou déconnecter un appareil virtuel, enregistrées par le Niveau Application, sont traduites en une demande d'établissement ou de libération de liaison transmise à la Station de Transport locale.

a.2 - Emission de lettres

Les lettres délivrées par le Niveau Présentation sont remises à la Station de Transport pour leur transmission vers le réseau.

b - Messages reçus de la Station de Transport (Interface S ← T)

b.1 - Liaisons établies ou refusées

Le Niveau Session est prévenu par la Station de Transport quand une liaison est établie ou refusée et il en avertit à son tour le Niveau Application.

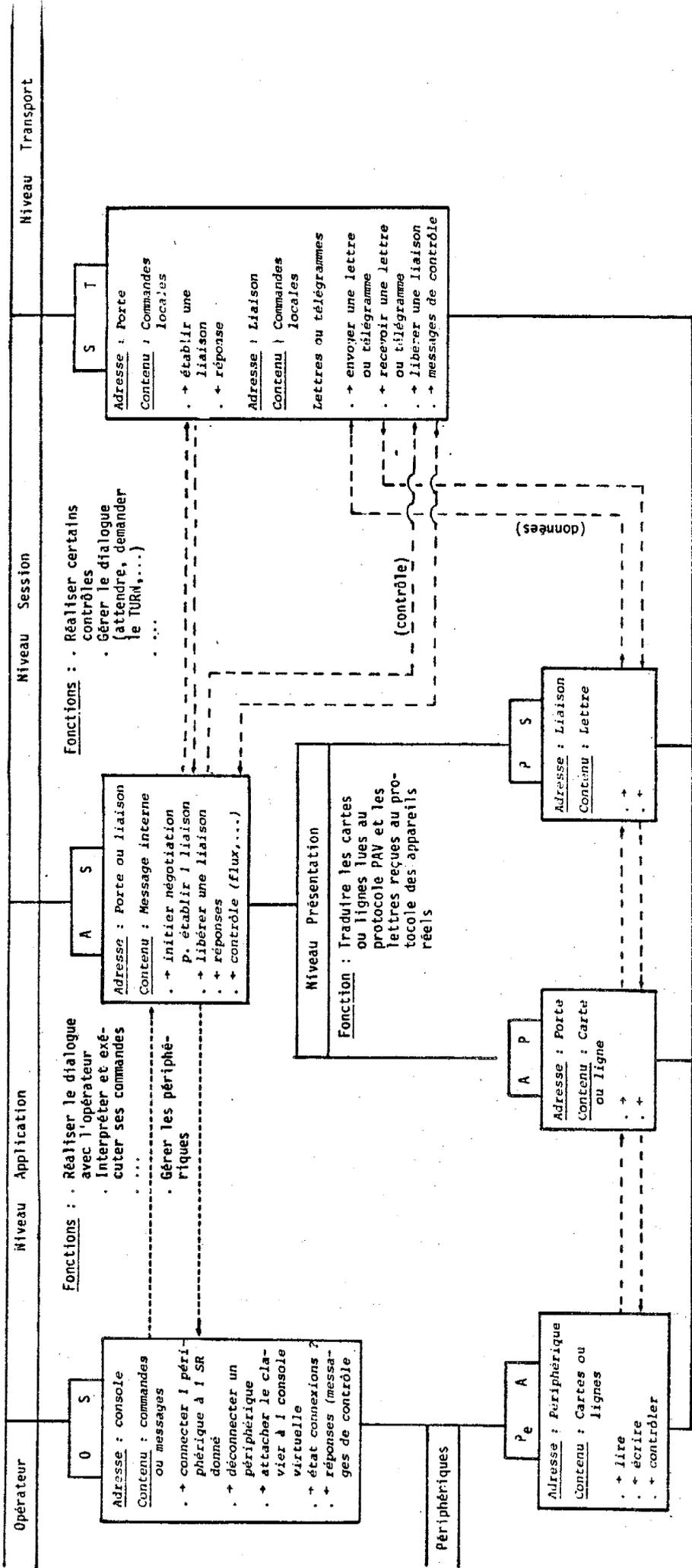


Tableau B.1 - Correspondance entre la couche "Application" (Réseau CYCLADES) et les niveaux "Session", "Présentation" et "Application" (Modèle ISO)

b.2 - Messages de contrôle

Les messages de contrôle reçus à travers cette interface concernent la *gestion du dialogue* (réception du droit d'émission pour la console ou le lecteur de cartes), la régulation du flux et la libération des liaisons (soit à l'initiative du serveur, soit en raison des anomalies détectées).

Dans tous les cas, le Niveau Application est informé de l'arrivée de ces messages.

b.3 - Données

Les lettres reçues du côté réseau contenant des données sont transférées au Niveau Présentation pour leur traduction.

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU le rapport de présentation de Monsieur :

- S. KRAKOWIAK, Professeur à l'Université Scientifique
et Médicale de GRENOBLE

Mademoiselle Silvia LUMELSKY

est autorisée à présenter une thèse en soutenance pour l'obtention du
titre de DOCTEUR de TROISIEME CYCLE, spécialité "Génie Informatique".

Grenoble, le 26 Novembre 1979

Le Président de l'I.N.P.G.

Ph. TRAYNARD
Président
de l'Institut National Polytechnique

