



HAL
open science

Architecture d'un frontal en environnement téléinformatique : application au réseau CYCLADES

Jean-François Chambon, Bernard Le Bihan

► **To cite this version:**

Jean-François Chambon, Bernard Le Bihan. Architecture d'un frontal en environnement téléinformatique : application au réseau CYCLADES. Réseaux et télécommunications [cs.NI]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1976. Français. NNT : 2000STET4023 . tel-00827637

HAL Id: tel-00827637

<https://theses.hal.science/tel-00827637>

Submitted on 29 May 2013

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.

N° d'ordre : 7 ii

THESE

présentée par

Jean François CHAMBON

Bernard LE BIHAN

pour obtenir

le grade de DOCTEUR – INGENIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

**ARCHITECTURE d'un FRONTAL
en ENVIRONNEMENT TELEINFORMATIQUE**

(APPLICATION AU RESEAU CYCLADES)

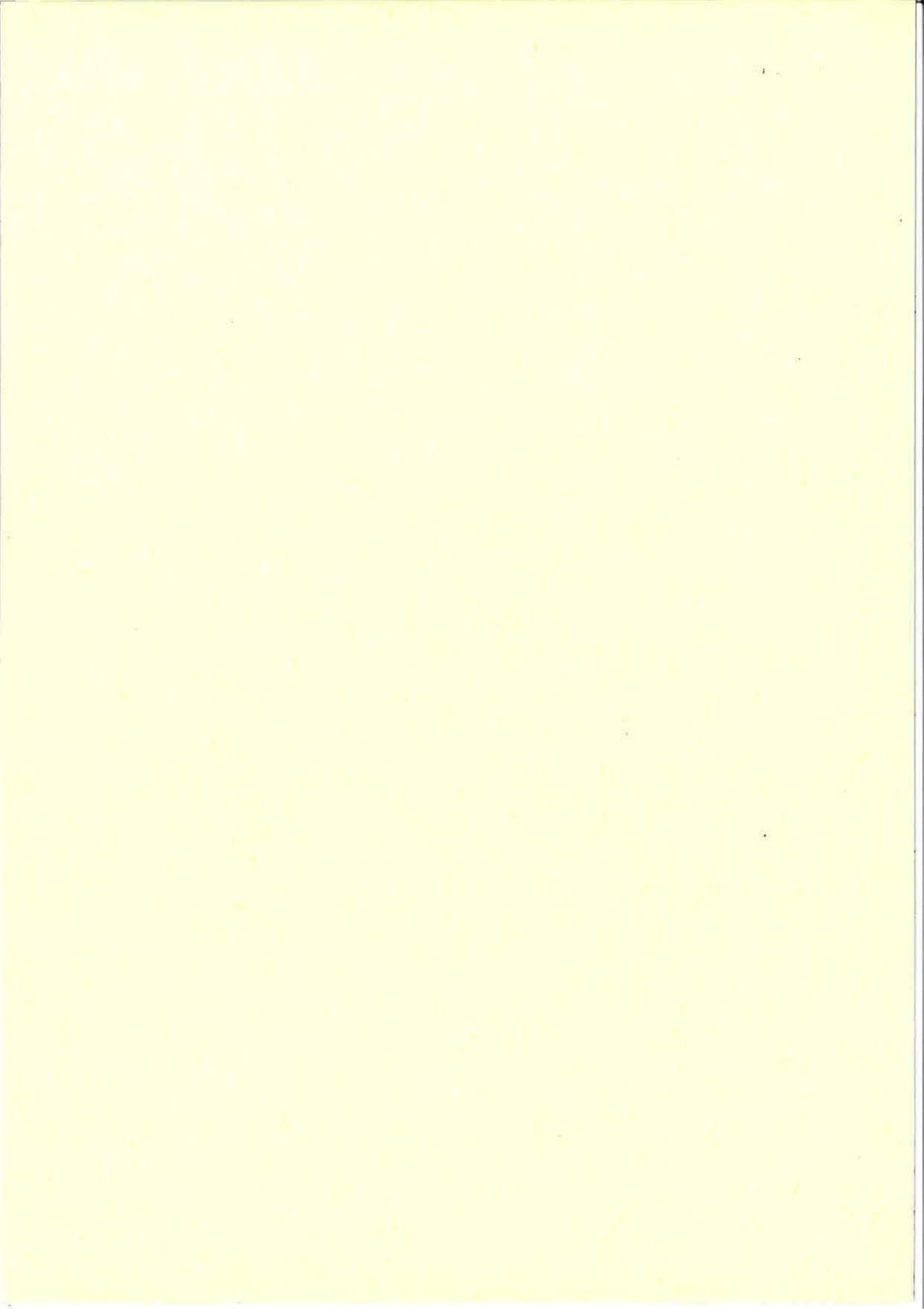
Thèse soutenue le 25 Octobre 1976 devant la commission d'examen :

MM. L. BOLLIET

Président

**G. BAZERQUE
S. GUIBOUD-RIBAUD
C. OTRAGE
R. MAHL**

Examineurs



N° d'ordre : 7 ii

THESE

présentée par

Jean François CHAMBON

Bernard LE BIHAN

pour obtenir

le grade de DOCTEUR – INGENIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

ARCHITECTURE d'un FRONTAL en ENVIRONNEMENT TELEINFORMATIQUE

(APPLICATION AU RESEAU CYCLADES)

Thèse soutenue le 25 Octobre 1976 devant la commission d'examen :

MM. L. BOLLJET

Président

G. BAZERQUE

S. GUIBOUD-RIBAUD

C. OTRAGE

R. MAHL

Examineurs

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT.ETIENNE

Directeur : M. Lucien VIELLEDENT

Directeur des Etudes : M. Jacques BOISSE

PROFESSEURS DE 1ère CATEGORIE

MM. COINDE Alexandre	Gestion
FORMERY Philippe	Mathématiques Appliquées
GOUX Claude	Métallurgie
LEVY Jacques	Métallurgie
RIEU Jean	Mécanique - Résistance des Matériaux
SOUSTELLE Michel	Chimie

PROFESSEURS DE 2ème CATEGORIE

MM. GUIBOUD-RIBAUD Serge	Informatique
LOWYS Jean-Pierre	Physique
TOUCHARD Bernard	Physique Industrielle

PROFESSEURS ASSOCIES DE 2ème CATEGORIE

MM. FONTEILLES Michel	Géologie
SEROR Denis	Informatique

DIRECTEUR DE RECHERCHE

M. LESBATS Pierre	Métallurgie
-------------------	-------------

MAITRES DE RECHERCHE

MM. BISCONDI Michel	Métallurgie
BOOS Jean-Yves	Métallurgie
Mlle. FOURDEUX Angéline	Métallurgie
MM. LALAUZE René	Chimie
LANCELOT Francis	Chimie
LE COZE Jean	Métallurgie
THEVENOT François	Chimie
TRAN MINH Canh	Chimie

DIRECTEURS DE RECHERCHE (au sens des arrêtés du 16 avril 1974)

MM. GUIRALDENQ Pierre	Métallurgie
KOBYLANSKI André	Métallurgie
GUILHOT Bernard	Chimie
THOMAS Gérard	Chimie
COUEIGNOUX Philippe	Informatique

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M. Philippe TRAYNARD
Vice-Président : M. Pierre-Jean LAURENT

PROFESSEURS TITULAIRES

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

PROFESSEUR ASSOCIE

M. ROUXEL Roland Automatique

PROFESSEURS SANS CHAIRE

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

MAITRES DE CONFERENCES

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

MAITRE DE CONFERENCES ASSOCIE

M. LANDAU Ioan Automatique

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

MM. FRUCHART Robert

Directeur de Recherche

ANSARA Ibrahim

Maître de Recherche

CARRE René

Maître de Recherche

DRIOLE Jean

Maître de Recherche

MATHIEU Jean-Claude

Maître de Recherche

MUNIER Jacques

Maître de Recherche

Nous tenons à remercier :

Monsieur Louis BOLLINET, Professeur à l'Université Scientifique et Médicale de Grenoble, qui nous a fait l'honneur de présider le jury de cette thèse.

Monsieur Georges BAZERQUE, Maître de Conférences à l'Université des Sciences Sociales de Toulouse, qui a toujours marqué son intérêt pour nos travaux et a accepté de participer au jury.

Monsieur Robert MAHL, Ingénieur des Mines, qui a été l'initiateur de cette étude alors qu'il était Directeur du département informatique de l'Ecole des Mines, et qui ne nous a jamais ménagé son aide ni ses conseils.

Monsieur Claude OTRAGE, Directeur de recherche à la TELEMECANIQUE, qui a bien voulu accepter de juger notre travail.

Monsieur Serge GUIBOUD-RIBAUD, Directeur du département informatique de l'Ecole des Mines, grâce auquel ce projet a pu aboutir.

Nous exprimons également notre reconnaissance à :

Monsieur Christian BERTIN, ingénieur INSA, qui a contribué pendant plus d'un an au développement de ce projet et permis une fructueuse collaboration dans le domaine des réseaux entre l'Ecole des Mines et l'INSA de RENNES.

Madame BONNEFOY , Messieurs LOUBET et DARLE qui ont assuré avec beaucoup de soins et de gentillesse la frappe et le tirage de cet ouvrage.



A V E R T I S S E M E N T

Cette thèse traduit un premier aboutissement de 3 années d'études et de réalisations dans le domaine de la télé-informatique en général et du réseau d'ordinateurs CYCLADES en particulier.

Nous avons abordé les différentes étapes de ce projet en complète et étroite collaboration. En particulier la conception général du frontal réalisé a été le fruit d'échanges constants d'idées et de critiques entre nous. Par suite il aurait été irréaliste de séparer artificiellement le travail effectué en deux parties distinctes, même si la réalisation, par souci d'efficacité, a été ponctuellement partagée, selon les goûts momentanés de chacun. C'est pourquoi, il en a résulté un document unique qui reflète mieux l'homogénéité de la recherche.

Ce document se veut non seulement une présentation du travail effectué, mais également une réflexion critique sur les domaines abordés. Sa rédaction a été partagée selon l'intérêt marqué par chacun pour présenter chacune des parties étudiées.

Ainsi, tandis que Jean-François CHAMBON rédigeait les chapitres IV, VI, VII et les annexes, Bernard LE BIHAN se consacrait aux chapitres I, II, III, V et VIII.

SOMMAIRE

	<u>Pages</u>
<u>CHAPITRE I : INTRODUCTION</u>	1
1.1. <u>Objectif général et intérêt de l'étude</u>	2
1.1.1. Le concept de frontal	3
1.1.2. Les solutions d'accès à un réseau.....	8
1.2. <u>Historique de l'étude</u>	15
1.3. <u>Plan de la thèse</u>	17
<u>CHAPITRE II : TERMINOLOGIE ET DEFINITION DES CONCEPTS</u>	19
2.1. <u>Les Télécommunications</u>	23
2.1.1. Support de transmission	23
2.1.2. Sens de transmission	24
2.1.3. Topologie d'une liaison	24
2.1.4. Format de transmission	25
2.1.5. Cadence de transmission.....	27
2.1.6. Procédures de transmission	28
2.1.7. Codes de transmission	29
2.2. <u>Classification des réseaux d'ordinateurs</u>	30
2.2.1. Réseaux centralisés	30
2.2.2. Réseaux distribués	32
2.3. <u>Caractéristiques d'un réseau général hétérogène</u>	35
2.3.1. Architecture de base	35
2.3.2. Architecture des protocoles-réseau.....	37
<u>CHAPITRE III : REALISATION ET OPTIMISATION DES AUTOMATES DE PROCEDURE</u>	45
3.1. <u>Principe des procédures de transmission</u>	49
3.1.1. Modèle de communication	49
3.1.2. Types de procédures de transmission	50
3.1.3. Langage de communication	54
3.1.4. Contrôle des erreurs de transmission	56
3.2. <u>Reconnaissance d'une procédure de transmission par automate</u>	57
3.2.1. Recherche d'un mode de représentation graphique	57
3.2.2. Graphes de la procédure multipoint ECMA-PHILIPS.....	58
3.2.3. Graphes de la procédure point à point TMM-UC.....	66
3.2.4. Avantages de la méthode	74

3.3.	<u>Langage de description d'un automate de procédure</u>	75
3.3.1.	Représentation de la syntaxe et de la sémantique.....	75
3.3.2.	Description de l'automate dans un langage.....	77
3.4.	<u>Optimisation du traitement</u>	87
3.4.1.	Traitement des erreurs	87
3.4.2.	Minimisation du temps de traitement d'un caractère...	88

<u>CHAPITRE IV :</u>	<u>EXPERIENCE SUR LES PROTOCOLES DE HAUT NIVEAU</u>	95
4.1.	<u>Problème du dialogue entre humains</u>	99
4.2.	<u>Protocoles</u>	100
4.2.1.	Protocole - Interface	100
4.2.2.	Négociation d'un échange d'information	102
4.2.3.	Séquencement des informations.....	102
4.2.4.	Contrôle d'erreur	102
4.2.5.	Contrôle de flux	103
4.2.6.	Evènements	104
4.3.	<u>Protocoles CYCLADES : choix et critiques</u>	105
4.3.1.	Cigale	107
4.3.2.	Stations de transport	107
4.3.3.	Abonnés.....	112
4.4.	<u>Comment modéliser le dialogue</u>	113
4.5.	<u>Protocole entre frontal et ordinateur de traitement</u>	118
4.5.1.	Hypothèses de base	118
4.5.2.	Construction du protocole	119

<u>CHAPITRE V :</u>	<u>CONCEPTION GLOBALE DU CALCULATEUR FRONTAL</u>	123
5.1.	<u>Rôle du frontal</u>	127
5.1.1.	Rappel des objectifs	127
5.1.2.	La fonction globale du frontal	128
5.1.3.	Les fonctions particulières d'accès aux ressources...	130
5.1.4.	Synoptique des fonctions nécessaires	135
5.2.	<u>Architecture du logiciel</u>	137
5.2.1.	La notion de "machine"	137
5.2.2.	Organisation générale de la machine "frontal".....	139
5.3.	<u>L'exemple de la machine de communication</u>	141
5.3.1.	La vision des serveurs	142
5.3.2.	L'architecture interne	145
5.3.3.	Flexibilité et extensibilité de la réalisation.....	152

<u>CHAPITRE VI :</u>	<u>INGENIERIE DU LOGICIEL.....</u>	157
	6.1. <u>Matériel.....</u>	161
	6.1.1. Généralités.....	161
	6.1.2. Tâches	161
	6.1.3. Interruptions.....	165
	6.1.4. Gestionnaire de ressources	166
	6.1.5. Entrées-sorties - coupleurs	169
	6.2. <u>Pourquoi partir d'un calculateur nu ?.....</u>	170
	6.3. <u>Contraintes et choix en résultant.....</u>	171
	6.3.1. Contraintes dues au matériel utilisé.....	171
	6.3.2. Réalisation des machines	177
	6.4. <u>Méthodologie</u>	178
	6.4.1. Programmation	178
	6.4.2. Outils de mise au point.....	181
<u>CHAPITRE VII :</u>	<u>RESULTATS ET PERFORMANCES</u>	183
	7.1. <u>Possibilités pratiques d'exploitation</u>	187
	7.1.1. Etat actuel de réalisation et utilisation.....	187
	7.1.2. Charge du frontal	189
	7.2. <u>Organisation et implantation du logiciel.....</u>	191
	7.2.1. Place occupée	191
	7.2.2. Transportabilité	192
<u>CHAPITRE VIII :</u>	<u>CONCLUSIONS.....</u>	195
	8.1. <u>Intérêt et limites du raccordement à un réseau à travers un frontal</u>	199
	8.1.1. Faisabilité	199
	8.1.2. Critiques	200
	8.2. <u>L'intégration au système d'un environnement téléinformatique</u>	201
	8.2.1. Nouveaux problèmes posés par la télé-informatique ...	201
	8.2.2. La solution apportée par le système.....	202
	8.3. <u>Les perspectives du système</u>	203
	8.3.1. Utilisation actuelle	203
	8.3.2. Améliorations à envisager.....	204
	8.3.3. Nouvelles perspectives d'utilisation.....	206

- ANNEXES
- Annexe I : Configuration du Centre de Calcul
 - Annexe II : Manuel Opérateur
 - Annexe III : Manuel utilisateur
 - Annexe IV : Exemples de sessions

BIBLIOGRAPHIE

CHAPITRE I

INTRODUCTION

1.1. - OBJECTIF GENERAL ET INTERET DE L'ETUDE

1.1.1 - Le concept de frontal

1.1.1.1. - Pour un seul ordinateur

1.1.1.2. - Pour un réseau d'ordinateurs

1.1.2 - Les solutions d'accès à un réseau

1.1.2.1. - Les solutions antérieures

1.1.2.2. - Les solutions étudiées en parallèle

1.1.2.3. - La solution retenue

1.2. - HISTORIQUE DE L'ETUDE

1.3. - PLAN DE LA THESE

1.1. - OBJECTIF GENERAL ET INTERET DE L'ETUDE

1.1.1. - LE CONCEPT DE FRONTAL

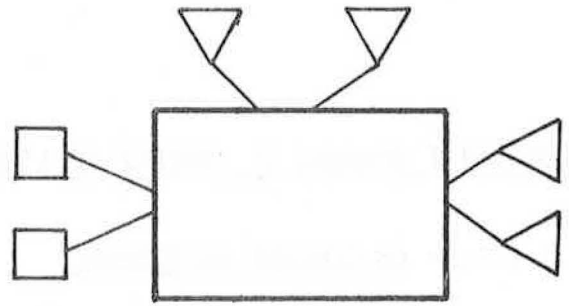
1.1.1.1. - Pour un seul ordinateur.

Ces dernières années, diverses techniques ont permis d'accroître considérablement le nombre des utilisateurs d'un ordinateur [GEN-9]. Le développement des systèmes de temps partagé a d'abord permis à plusieurs utilisateurs d'accéder simultanément aux ressources d'un même ordinateur, à partir de terminaux combinant un clavier et un organe d'affichage - du type imprimante ou écran de visualisation (fig. 1-b). L'interaction directe avec l'ordinateur permise à partir de ces terminaux a intéressé une nouvelle classe d'utilisateurs pour qui les systèmes traditionnels de traitement par lot à partir d'un lecteur de cartes et d'une imprimante s'avéraient inappropriés.

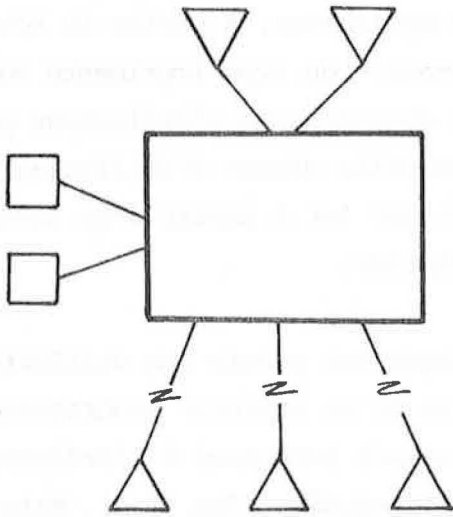
Puis les techniques de la télé-informatique ont permis aux utilisateurs d'un ordinateur de s'éloigner de cet ordinateur et de se répartir progressivement dans l'espace. On a commencé par relier plusieurs terminaux à l'ordinateur central par des lignes téléphoniques louées ou spécialisées (fig. 1-c). Mais à mesure que le nombre de ces terminaux éloignés augmentait, les lignes de communication prenaient une part de plus en plus importante dans le coût total de l'installation informatique. D'autre part, chaque terminal, du fait de sa fonction interactive, n'utilisait qu'une faible part de la capacité de transmission offerte par la ligne de communication le reliant à l'ordinateur. Des configurations multipoints ont alors contribué à réduire le coût de ces liaisons physiques en permettant le partage d'une même ligne de transmission par plusieurs terminaux - d'une manière similaire au partage d'un canal multiplexé entre plusieurs organes périphériques locaux. A cet effet un certain nombre de mécanismes de contrôle permettant la sélection des terminaux et la protection des données étaient inventés, augmentant par là-même le coût de gestion de ces lignes. Des unités câblées de contrôle des transmissions ont alors été introduites pour décharger l'ordinateur central des fonctions de communication répétitives et de bas niveau (fig. 1-d). Puis, pour prendre en compte des applications non interactives, de nouveaux terminaux, orientés vers des travaux de type traitement par lot, ont été attachés à ces lignes de transmission afin de dupliquer l'environnement local chez les utilisateurs éloignés.



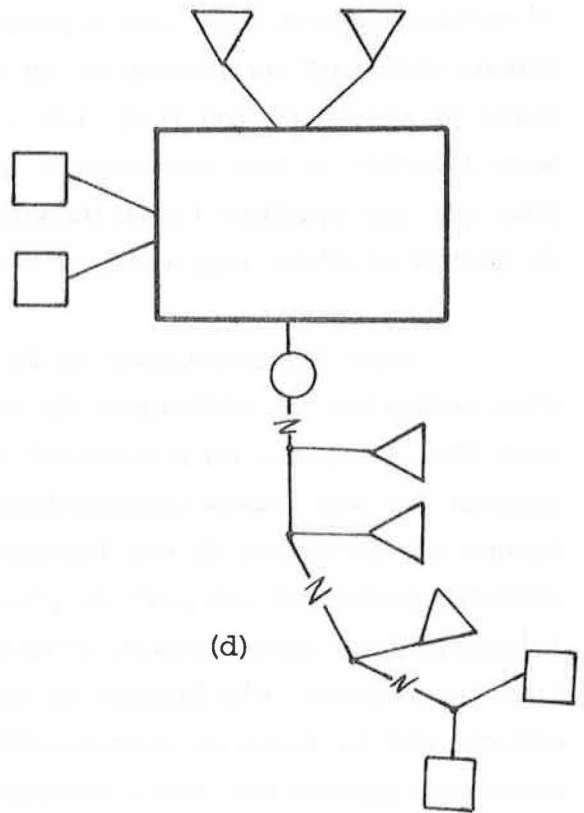
(a)



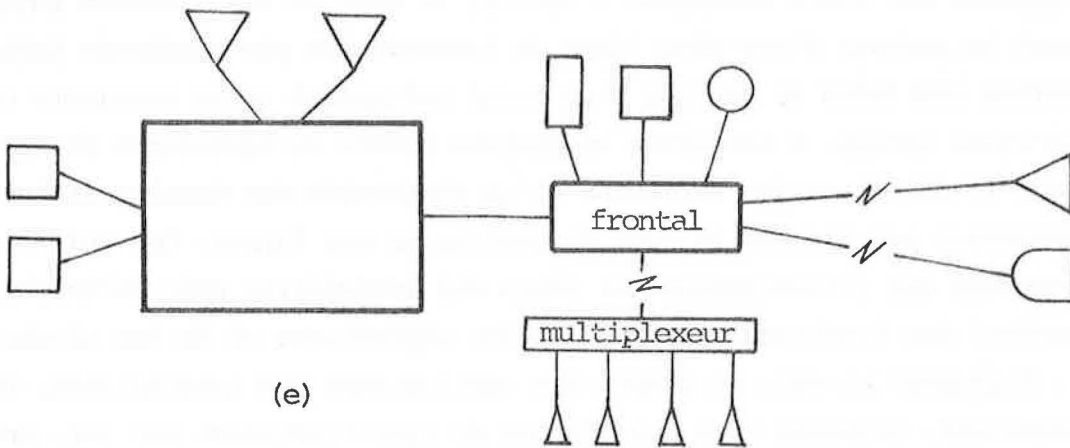
(b)



(c)



(d)



(e)

Figure 1 - Evolution de l'accès à un ordinateur

Cependant, des terminaux de plus en plus variés sont rapidement apparus sur le marché. Or le matériel et le logiciel de la plupart des gros ordinateurs limitent à une certaine classe les terminaux qu'il est possible de leur connecter : ces derniers sont soumis à des contraintes de vitesse, de jeu de caractères, de code, de mode de transmission, de contrôle d'erreurs etc... De nombreux terminaux, non fournis par un constructeur, sont difficilement utilisables sur les ordinateurs de ce constructeur. D'où la nécessité de transformer l'unité de contrôle évoquée précédemment en la rendant programmable. C'est ainsi qu'est apparue la notion de "frontal", généralement réalisé à partir d'un mini-ordinateur qui décharge l'ordinateur central de la gestion d'un certain nombre de terminaux locaux et distants (fig. 1-e). Cette évolution a été largement facilitée par la baisse de prix considérable des mini-ordinateurs, et par le fait qu'ils possèdent des interfaces d'accès matérielles et logicielles facilement adaptables à tout nouveau type de terminal. Le frontal est essentiellement utilisé dans un rôle de transcodage permettant de donner à l'ordinateur central une image "familiale" et uniforme des appareils plus ou moins hétéroclites qui lui sont connectés. On accroît ainsi la séparation entre les fonctions de communication et les fonctions de traitement au niveau de la répartition physique de ces fonctions. L'installation globale y gagne en flexibilité et en efficacité.

Vis à vis de l'ordinateur de traitement, le frontal se comporte donc comme un adaptateur. Remarquons que cette notion d'adaptateur est déjà commune en informatique et intervient à différents niveaux. Nous nous contenterons de citer deux exemples : le compilateur, qui adapte un programme écrit en un langage source externe au langage interne accepté par la machine ; la méthode d'accès, qui adapte un format général d'entrée/sortie à la spécificité d'un organe périphérique.

1.1.1.2. - Pour un réseau d'ordinateurs

Parallèlement au développement de ces systèmes orientés vers les terminaux, des liaisons étaient établies entre ordinateurs. Pour faciliter leur tâche de communication, un frontal pouvait à nouveau être interposé entre deux ordinateurs. Il permettait cette fois à chaque ordinateur de considérer son vis à vis comme un organe périphérique particulier ou une série d'organes périphériques (fig. 2).

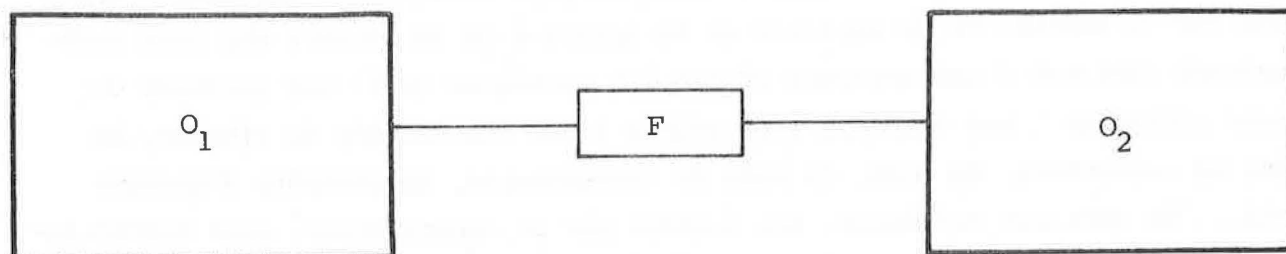


Figure 2

Plus récemment, les réseaux d'ordinateurs hétérogènes ont permis de franchir un nouveau pas dans la décentralisation des moyens informatiques. Non seulement ils autorisent une plus grande dispersion géographique des utilisateurs, mais ils permettent également l'accroissement et la dispersion des ressources auxquelles peuvent accéder ces utilisateurs. Nous faisons ici référence à des réseaux généraux du type ARPA aux Etats-Unis [ARP-1] ou CYCLADES en France [CYC-1]. Rappelons (*) brièvement que de tels réseaux possèdent généralement une architecture à deux niveaux :

- . Un premier niveau assure une connexion permanente entre tous les utilisateurs du réseau (ordinateurs ou terminaux). C'est le réseau de communication qui offre aux utilisateurs un service de transmission de messages par paquets (fig. 3).
- . Un second niveau utilise ce service de communication comme une "boîte noire" pour y déposer et en retirer des messages. C'est le réseau proprement dit, constitué de l'ensemble des ordinateurs et terminaux participants (fig. 4).

* Les concepts mis en jeu sont analysés avec plus de détail au chapitre II

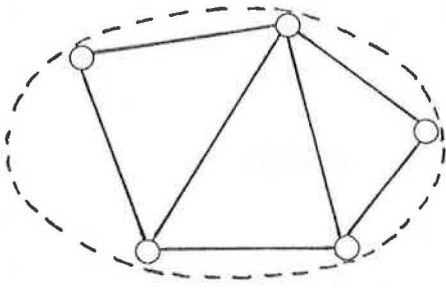


Figure 3

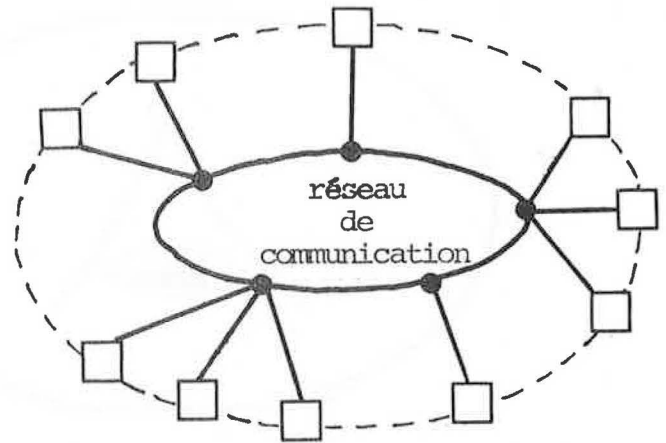
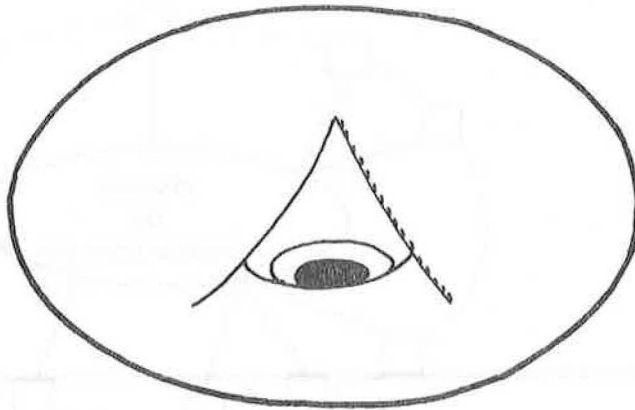


Figure 4

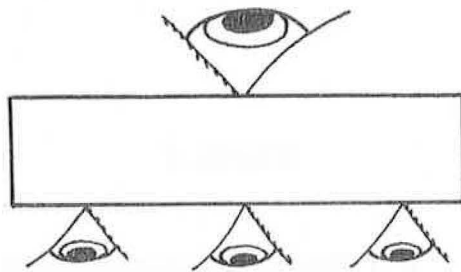
○ Noeud du réseau de communication □ Ordinateur participant

Cet avènement des réseaux d'ordinateurs hétérogènes a créé un nouveau besoin : celui de pouvoir raccorder rapidement un terminal ou un ordinateur (voire un réseau...) - nous dirons un appareil - à un réseau existant. Pour participer aux échanges du réseau l'appareil doit d'abord se connecter physiquement au réseau de communication utilisé, puis respecter un certain nombre de conventions (ou protocoles) en vigueur sur ce réseau. Or souvent, l'appareil à raccorder, quoique doté de vastes possibilités de traitement s'il s'agit d'un ordinateur, communique avec l'extérieur selon une procédure figée qui ne peut pas être modifiée à un coût raisonnable. Dès lors un adaptateur est ici encore nécessaire afin de faire concorder les visions que le réseau et l'appareil à y raccorder ont chacun de leur monde externe (fig. 5).

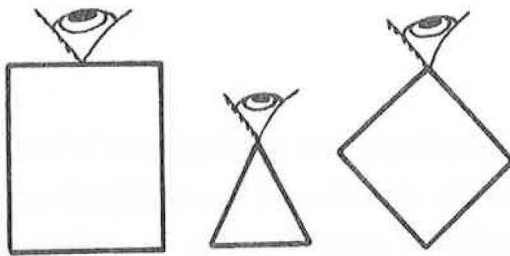
Cet adaptateur présente une analogie de fonction, au niveau réseau, avec l'adaptateur permettant l'accès à un ordinateur par divers terminaux. Pour cette raison, nous lui donnerons également le nom de frontal. La présente thèse fait état de l'expérience acquise par les auteurs en réalisant le logiciel d'un tel calculateur frontal.



RESEAU



ADAPTATEUR



APPAREILS

Figure 5

1.1.2. - LES SOLUTIONS D'ACCES A UN RESEAU :

Diverses méthodes permettant le raccordement d'un appareil à un réseau ont été mises en oeuvre, tant sur le réseau ARPA que sur le réseau CYCLADES. Certaines ont été élaborées avant que ne débute la présente étude, alors que d'autres ont été élaborées en parallèle (notamment sur CYCLADES). Enfin, parmi celles qui utilisent la technique de "frontal du réseau" il existe des variantes quant aux fonctions assumées par le frontal. Nous allons brièvement passer en revue ces diverses approches afin de mieux situer les nouvelles possibilités apportées par la solution que nous avons retenue.

1.1.2.1. - Les solutions antérieures :

- Connexion directe d'un ordinateur :

Cette première approche consiste à munir l'ordinateur participant d'un logiciel gérant les protocoles utilisés sur le réseau (fig. 6). Généralement on évite de modifier le système d'exploitation existant et on inclue ce logiciel en parallèle, en tant que "tâche-utilisateur" [CYC-11]. Cette méthode a été, pour des raisons d'opportunité, la première utilisée aussi bien sur ARPA que sur CYCLADES [CYC-10].

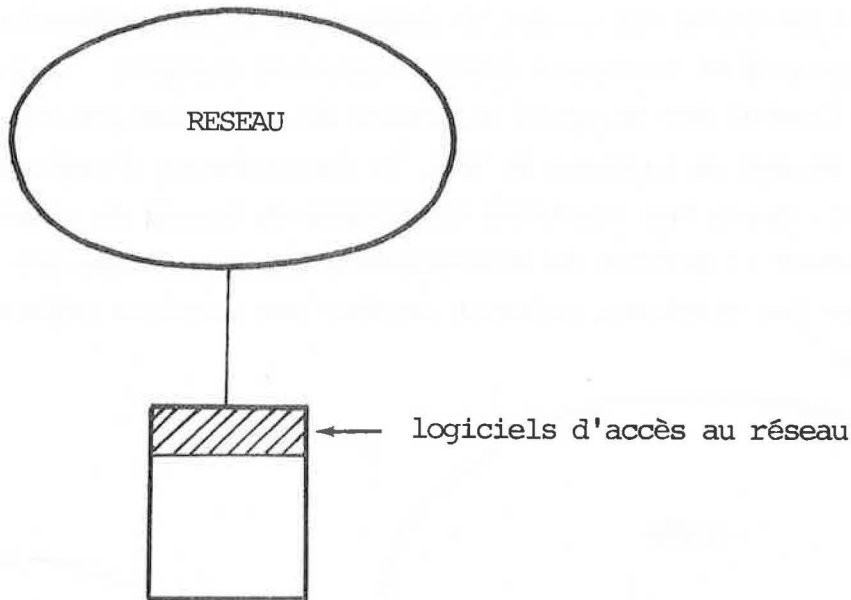


Figure 6

Elle a cependant mis en évidence un certain nombre de difficultés de réalisation, liées aux facilités plus ou moins grandes offertes par le système d'exploitation, et ne peut constituer qu'un palliatif en attendant l'arrivée de systèmes d'exploitation "orientés-réseau". De plus, son manque de souplesse est certain, car il faut reprendre l'ensemble des logiciels d'accès au réseau si l'on change d'ordinateur.

- Connexion de terminaux : la technique ARPA

L'accès d'un terminal aux services offerts sur le réseau peut s'opérer par son raccordement à un ordinateur participant si ce dernier possède un système convenable de temps partagé. Cette approche conduit cependant

rapidement à une charge excessive pour l'ordinateur à mesure que le nombre de terminaux augmente et fait reposer l'accès au réseau sur la disponibilité et la fiabilité de l'ordinateur participant. De plus, les terminaux que peut supporter l'ordinateur sont limités à une certaine classe (cf.1.1.1.) Enfin, certains sites peuvent même ne pas posséder de gros ordinateur et souhaiter accéder directement au réseau à partir de simples terminaux.

A cet effet, un premier modèle de calculateur frontal a été développé sur le réseau ARPA [ARP-3] : le TIP (ou "terminal IMP"). Un puissant contrôleur multi-lignes lui permet de supporter, selon différents modes de transmission, jusqu'à 64 terminaux divers, locaux ou distants. Il a été choisi d'implanter ce frontal sur un noeud du réseau de communication (ou IMP). Il admet par suite, de par sa fonction de IMP, le raccordement d'ordinateurs participants (fig. 7). Outre les fonctions classiques du réseau de communication, le logiciel assure la gestion du sous-ensemble des protocoles qui est nécessaire pour que les terminaux puissent accéder aux services offerts sur le réseau (fig. 8)

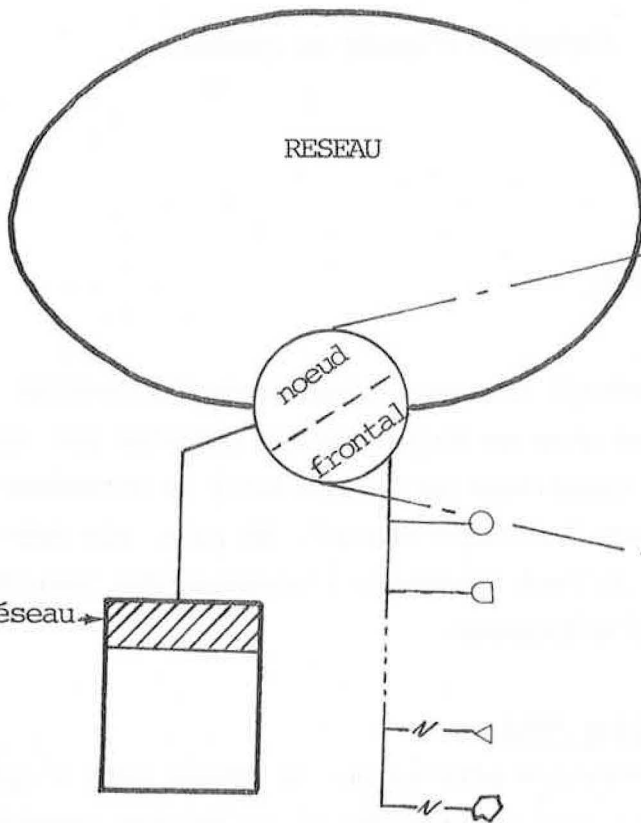
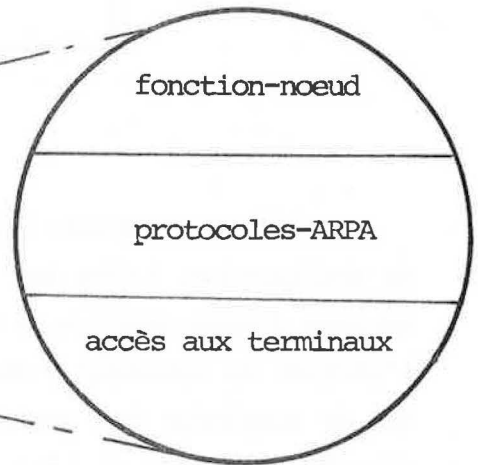


Figure 7



TIP

Figure 8

Il est à noter que cette solution ne dispense pas un ordinateur connecté au frontal d'avoir son propre logiciel d'accès au réseau. Par ailleurs, le choix - dicté par des raisons d'économie - de grouper les fonctions de noeud et de frontal sur un même calculateur entraîne plusieurs désavantages. D'une part, elle accroît les interférences - déjà sensibles sur ARPA - entre le sous-réseau de communication et le réseau utilisateur. D'autre part, en cas de surcharge du réseau de communication, la fonction-noeud du TIP prévaut et limite sérieusement le débit du frontal.

1.1.2.2. - Les solutions étudiées en parallèle

Parallèlement à notre étude, l'équipe CYCLADES de l'I.R.I.A. s'est attachée à développer les moyens d'accès au réseau. Cependant ils se sont presque exclusivement intéressés à l'accès des appareils du type terminal [CYC-8].

- Concentrateur de terminaux : (fig. 9 et 10)

Il permet à différents types de terminaux d'accéder aux différents services du réseau (services conversationnels ou services de traitement par lots). Un ordinateur peut être raccordé au réseau par son intermédiaire, mais il s'agit alors d'une simple "voie de passage" et la gestion de tous les logiciels d'accès au réseau reste à la charge de cet ordinateur.

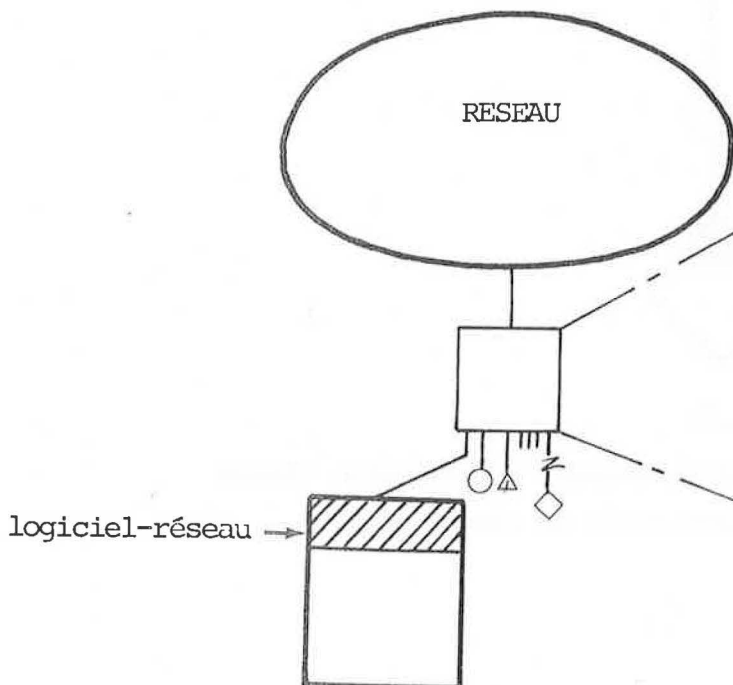
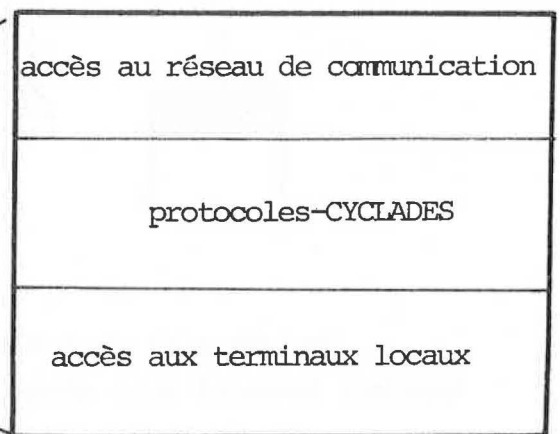


Figure 9



CONCENTRATEUR

Figure 10

Contrairement au TIP (1.1.2.1.), il est totalement situé hors du réseau de communication.

- Concentration décentralisée :

Pour mémoire, nous citons les études concernant le "Terminal paquet" [CYC-8] directement raccordable au réseau de communication. Ce terminal gère sa connexion au réseau de communication et le sous-ensemble des protocoles nécessaire pour ses communications avec les services du réseau. Ces fonctions sont réalisées par micro-programmes si le terminal est micro-programmable ou par un micro-processeur frontal s'il s'agit d'un terminal électro-mécanique.

Pour pouvoir partager une entrée du réseau de communication entre plusieurs utilisateurs un "concentrateur" de paquets a également été développé. Ce concentrateur est réalisé à partir d'un micro-processeur et multiplexe une "entrée-paquet" (fig. 11) vers 3 utilisateurs qui peuvent être soit un ordinateur, soit un terminal paquet, soit un nouveau concentrateur de paquets. Il est ainsi réalisé une grappe d'accès au réseau.

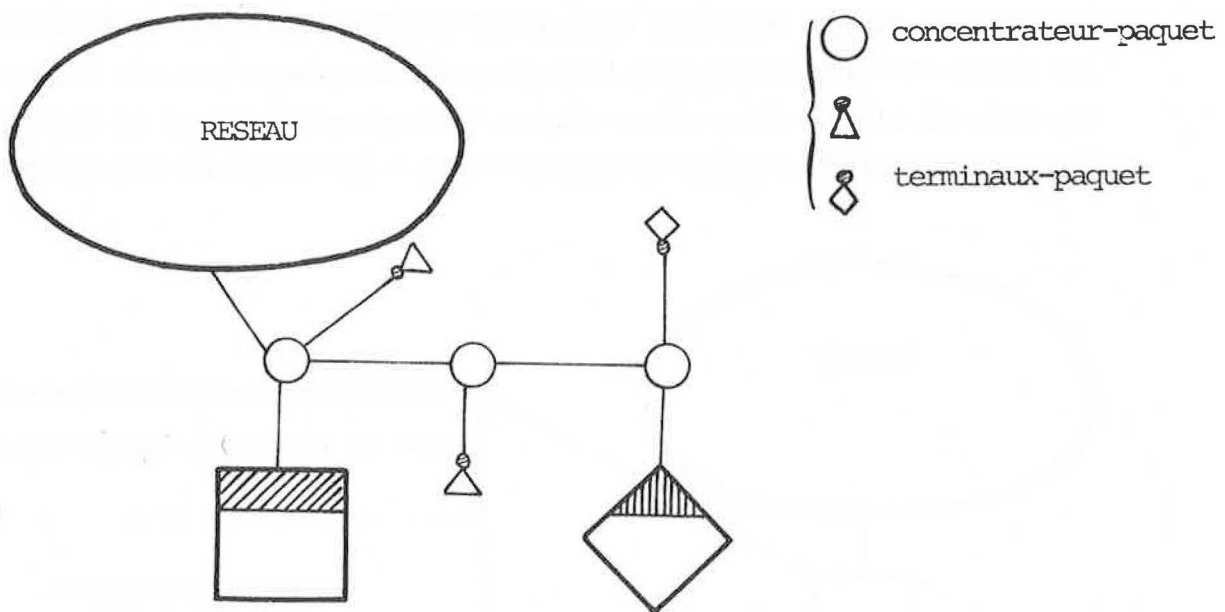


Figure 11

Toutefois ce concentrateur gère uniquement des paquets. Chaque appareil connecté doit gérer lui-même son accès au réseau général.

1.1.2.3. - La solution retenue :

L'examen de ces diverses réalisations nous conduit à formuler deux remarques :

- d'une part, aucune des approches précédentes n'envisage l'adaptation d'un appareil quelconque à un réseau. En particulier, pour un ordinateur, il n'a pas été étudié de solution autre que son raccordement direct au réseau, à la charge de cet ordinateur de gérer les protocoles d'accès au réseau. Les solutions évoquées se limitent toutes à résoudre le problème de l'accès des terminaux au réseau.
- d'autre part, toutes les réalisations précédentes sont exclusivement tournées vers l'accès aux ressources d'un réseau. Aucune d'entre elles ne prévoit la mise en commun des divers appareils d'un site pour une utilisation, locale et hors réseau, des ressources du site.

Partant de ces deux remarques, la solution que nous avons retenue tend à généraliser le concept de frontal en étendant ses fonctions selon deux grands axes (fig. 12 et 13).

a) Extension des accès aux ressources du réseau

Le frontal qui a été construit envisage le raccordement au réseau de tout appareil, qu'il soit du type terminal ou du type ordinateur. En particulier, un ordinateur connecté au frontal n'a plus à gérer son accès au réseau. Ce travail est entièrement réalisé par le frontal qui s'adapte à la vision du monde externe que possède l'ordinateur - si cette vision est trop restrictive, il convient, néanmoins, de la développer. Le réseau devient transparent pour l'ordinateur auquel les services offerts par le réseau apparaissent comme des services offerts par le frontal.

L'ordinateur connecté se contente de gérer ses rapports avec le frontal selon des règles locales. La connexion de tout nouvel ordinateur se limite à la re-définition de ces règles locales, l'ensemble des dialogues avec le réseau étant déjà pris en charge par le frontal.

b) Extension des accès aux ressources locales

Le frontal se propose d'offrir les ressources locales - par exemple, un service de traitement par lot sur l'un des ordinateurs directement connectés - aussi bien aux utilisateurs du réseau qu'aux utilisateurs locaux qui lui sont connectés. Pour cela, il se conforme aux conventions du serveur local. Un des aspects intéressants de cette approche réside, en dehors de toute "optique-réseau", dans son utilité locale pour la concentration et la banalisation des terminaux du site.

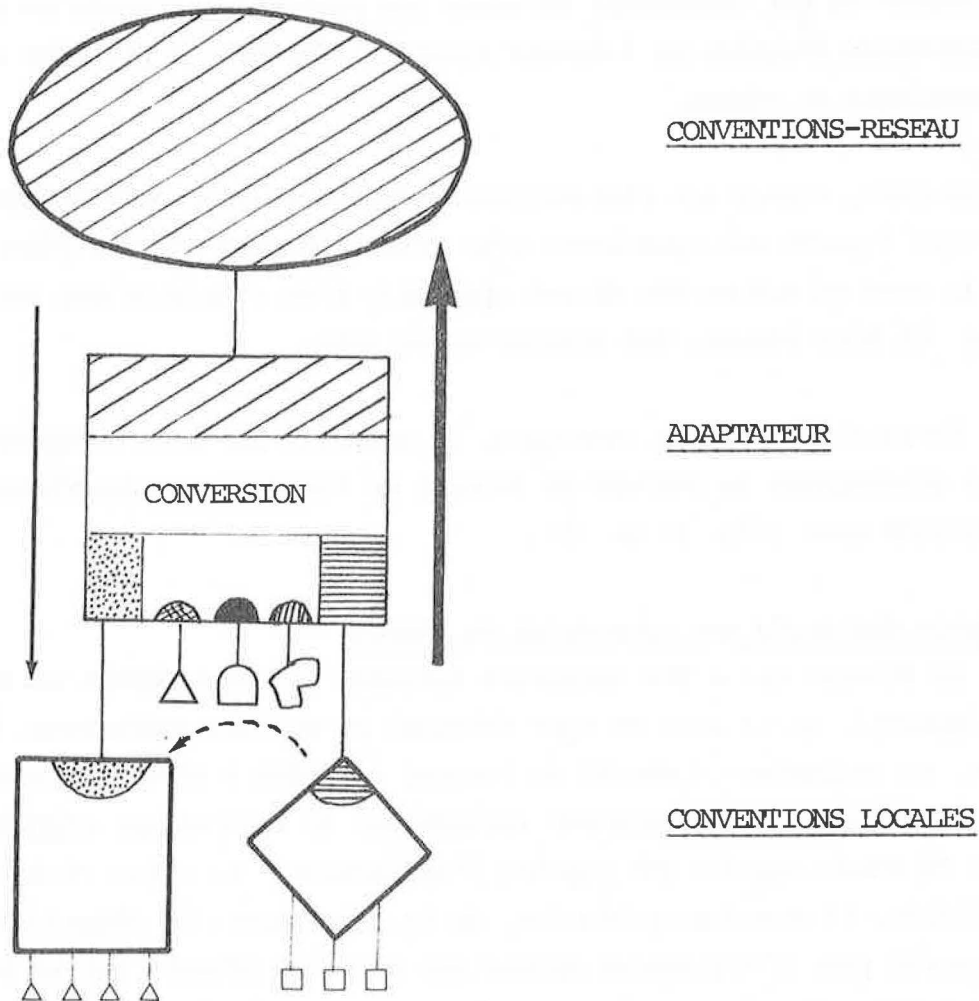
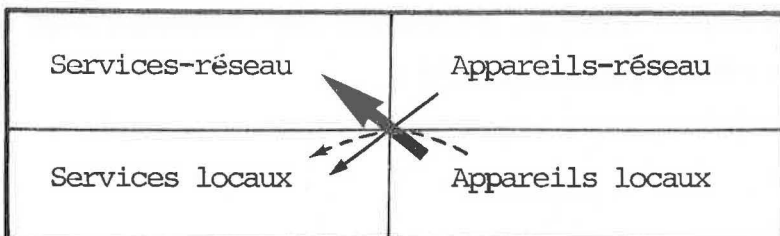


Figure 12



FONCTIONS DU FRONTAL

Figure 13

Le frontal ainsi défini est bien sûr non programmable par l'utilisateur et est utilisé comme une "boîte noire" par l'ensemble des appareils (terminaux, ordinateurs ou réseau) qui lui sont connectés. Il joue le rôle d'une "plaque tournante" avec la double fonction de :

- Commutateur permettant l'établissement simultané de plusieurs communications entre les divers appareils.
- Convertisseur d'interfaces et traducteur de protocoles permettant à deux appareils étrangers de dialoguer.

Bien entendu, nous ne prétendons pas avoir résolu tous les problèmes posés par l'accès d'un appareil à un réseau. La facilité d'accès reste fonction, principalement pour un ordinateur, du degré "d'ouverture vers l'extérieur" de cet appareil (cf. Chap. V). Nous apportons simplement, en attendant l'arrivée de systèmes d'exploitation orientés vers les réseaux, une méthode pratique de raccordement. Et nous pensons avoir ainsi contribué à mieux localiser les problèmes posés. [CYC-9]

1.2. - HISTORIQUE DE L'ETUDE :

Cette recherche s'inscrit dans le cadre du développement du réseau français d'ordinateurs hétérogènes CYCLADES, dont le Département d'Informatique de l'Ecole Nationale Supérieure des Mines de Saint-Etienne a été l'un des premiers centres de recherche participants. Dès l'été 1972, J.F. CHAMBON alors élève à l'Ecole des Mines, faisait un stage à l'I.R.I.A. où il participait à la définition de certains protocoles [CYC-4]. Les objectifs de la recherche étaient définis début 1973 en liaison avec l'équipe de M. POUZIN, directeur du projet CYCLADES, et avec MM. RABAULT et RAMSAY de la D.R.M.E. (Direction des Recherches et Moyen d'Essai). La D.R.M.E. a accepté de financer les travaux dans le cadre d'une convention qui s'étalait du 1er septembre 1973 au 1er Mars 1975. Dès le mois d'octobre 1973, la recherche débutait effectivement sous la responsabilité scientifique de M. R. MAHL.

L'étude s'est déroulée dans le cadre du Centre de Calcul de l'Ecole des Mines avec un PHILIPS P1175 comme ordinateur principal et un TELEMECANIQUE T1600 comme calculateur frontal (cf. ANNEXE 1)

Le premier travail a consisté à relier l'ordinateur principal au calculateur frontal. Pour cela, on a utilisé la faculté du P1175 de pouvoir gérer une ligne de transmission multipoint qui supportait initialement un certain nombre d'écrans alphanumériques. Il s'agissait alors de simuler sur le frontal le comportement d'une ou plusieurs adresses de cette ligne multipoint. Ce travail rendu fastidieux par la rigidité de la procédure de transmission utilisée - entièrement gérée par matériel sur les installations PHILIPS - fut encore compliqué par un manque évident de documentation. Parallèlement était mis au point sur le frontal un logiciel de gestion du coupleur synchrone utilisé, aucun produit n'étant alors fourni par la TELEMECANIQUE. En février 1974, la liaison T1600 - P1175 était opérationnelle. Elle fut immédiatement exploitée par les diverses équipes de recherche travaillant sur le T1600 pour le stockage et la maintenance de leurs programmes-source T1600 sur les bibliothèques du P1175. Par ailleurs, les terminaux du T1600 pouvaient se comporter comme les écrans alphanumériques de la ligne multipoint vis à vis du P1175.

Au cours de cette première réalisation il s'était avéré que le système standard développé par le constructeur sur le T1600 n'apportait pas suffisamment de possibilités, notamment du point de vue de la gestion des entrées-sorties. Il fut donc décidé de construire un système spécialement adapté à un environnement télé-informatique sur un T1600 nu. Parallèlement à sa conception globale (cf. chap V) un certain nombre de choix - dictée par les contraintes matérielles - étaient pris quant à sa réalisation (cf. Chap VI).

Bénéficiant de l'expérience acquise, la connexion au réseau de communication CIGALE fut rapidement établie. A cette occasion, des études plus poussées (cf. Chap. III) furent entreprises pour la production automatique des programmes de gestion des procédures de transmission. Les programmes étaient générés en langage source à l'aide du générateur de macro-instructions du P1175, puis télétransmis au T1600 où ils étaient compilés et testés. Malheureusement, il apparut rapidement que le gain de temps pour la mise au point des procédures était compensé par une perte d'efficacité des programmes non acceptable, ce qui obligea à abandonner la méthode.

Le nouveau système se mettant en place, il devenait possible, en Juillet 1974, de dialoguer, à partir de la console-opérateur, avec les consoles opérateurs des différents noeuds du réseau CIGALE. En octobre 1974, une première version de la station de transport CYCLADES, offrant un service de "lettres régulières" (cf. Chap IV) sur le réseau, était disponible. Un abonné de cette station de transport permettait un dialogue inter-opérateur avec les divers sites du réseau CYCLADES. En décembre 1974, des tests soutenus de transferts simultanés de plusieurs fichiers avaient lieu avec le Centre de Recherches ONERA-CERT de Toulouse, et permettaient de valider le logiciel.

A cette même époque, s'engageait une étroite collaboration avec l'INSA de RENNES qui, ne disposant que d'un ordinateur T1600, voulait profiter des services offerts sur le réseau. Nous leur livrions notre système et l'écriture des abonnés de haut niveau était répartie entre les deux centres. Les "abonnés-clients" du réseau étaient développés à l'INSA, cependant que nous mettions au point l'"abonné-serveur" capable d'offrir le service local de traitement par lot aux clients du réseau. De fréquents échanges eurent lieu sur le réseau pour les mises à jour des modules-source du système développé à distance. Parallèlement, nous définissions une interface standard entre le calculateur frontal et l'ordinateur principal (cf. Chap IV).

En Mai 1975, tout terminal du P1175 pouvait accéder aux services offerts sur le frontal de la même manière que les terminaux locaux au frontal. En septembre 1975, une nouvelle version du système d'exploitation du P1175 permettait d'accéder à son service de traitement par lot à partir du frontal. En décembre 1975, les services-réseau étaient accessibles en tout point du site. En juin 1976, un nouvel ordinateur (PDP 11/40) était connecté au frontal, la liaison étant essentiellement utilisée à des transferts de fichiers entre P1175 et PDP 11/40.

1.3. - PLAN DE LA THESE :

Plutôt que de présenter en détail tous les mécanismes mis en jeu par notre réalisation, nous avons préféré articuler cette thèse autour des quelques idées fondamentales que nous avons retirées de nos travaux.

- La présentation de ces idées requiert un vocabulaire propre que le chapitre II a pour but de préciser, afin d'éviter toute ambiguïté pour le non-spécialiste.

- La première partie de l'exposé se veut une réflexion critique sur les problèmes posés par la télé-informatique. Dans le chapitre III nous abordons en détail, et à partir d'exemples précis, le problème des procédures de transmission, à l'aide de la théorie élémentaire des automates d'état fini. Et nous en proposons le traitement dans un langage de haut niveau. Le chapitre IV expose les principes d'élaboration des protocoles utilisés sur un réseau, en s'appuyant sur les choix du réseau CYCLADES. Un exemple de protocole entre un calculateur frontal et un ordinateur principal y est proposé.

- La seconde partie de l'exposé concerne plus directement la réalisation du frontal. La conception globale du logiciel de ce frontal est présentée sous un angle fonctionnel dans le chapitre V. L'accent est mis sur la structure des composants plutôt que sur leur réalisation. Les principaux choix effectués au moment de la réalisation sont explicités dans le chapitre VI qui traite de l'ingénierie du logiciel. Enfin, le chapitre VII résume les possibilités pratiques d'utilisation du produit.

- En conclusion, nous nous interrogeons sur les perspectives d'avenir de ce frontal - notamment par rapport à CYCLADES - puis nous proposons quelques améliorations et des objectifs de recherche future.

CHAPITRE II

TERMINOLOGIE ET DEFINITION DES CONCEPTS

Depuis le début des années 60 la science informatique s'est attachée à lever l'obstacle que constitue l'éloignement géographique pour l'utilisation des ordinateurs. Et de nos jours, un nombre croissant d'ordinateurs opèrent en environnement télé-informatique, c'est-à-dire dans un environnement où certaines ressources matérielles et logicielles sont réparties géographiquement.

Dans ce chapitre, nous nous efforçons de préciser un certain nombre de concepts nouveaux introduits par la télé-informatique.

2.1. - LES TELECOMMUNICATIONS : [GEN-1,[GEN-5]

Lorsque le matériel est réparti il est nécessaire d'avoir un moyen rapide et sûr pour transmettre de l'information entre deux composants matériels distants - nous dirons deux stations -, qu'il s'agisse de deux ordinateurs interconnectés, ou d'un ou plusieurs terminaux connectés à un ordinateur.

2.1.1. - SUPPORT DE TRANSMISSION

Le support de transmission (fig.1) généralement utilisé est la ligne téléphonique.

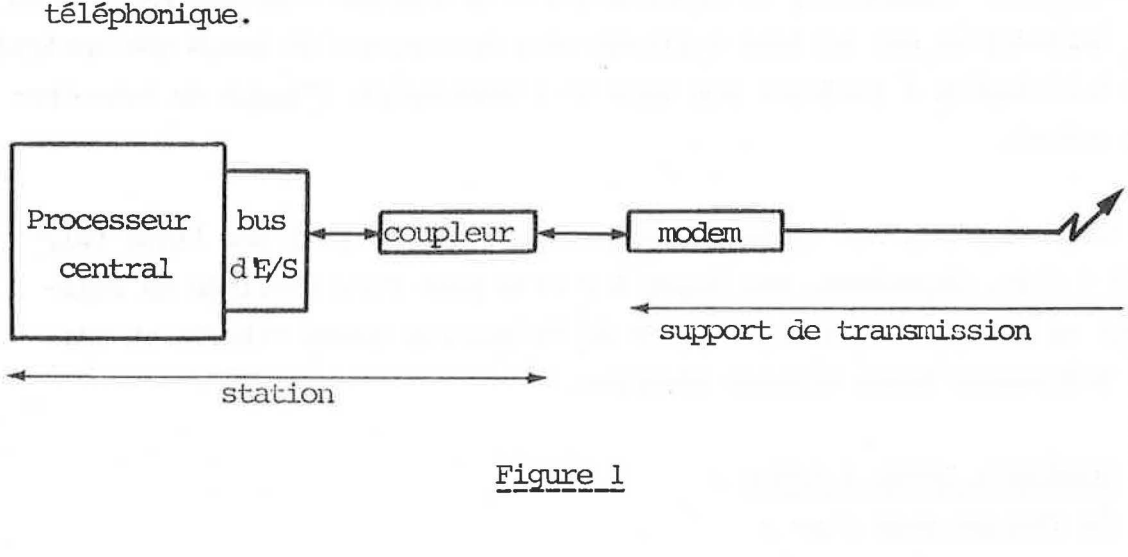


Figure 1

Cependant, le réseau téléphonique ayant été initialement conçu pour la transmission de la parole, il est nécessaire de convertir les données (digitales) à transmettre sous une forme (analogique) acceptable par la ligne. Cette adaptation est réalisée par des modems (de : "modulateur-démodulateur") situés aux deux extrémités de la ligne de transmission. Le modem de départ place les données binaires à émettre dans une plage de fréquences audibles qui peut être transmise sur la ligne téléphonique, par modulation en fréquence ou en phase (selon la vitesse de transmission) d'une onde porteuse. Le modem d'arrivée effectue l'opération inverse et restitue le signal binaire. Un coupleur de transmission, ou unité de contrôle, assure le couplage avec le bus d'Entrées/Sorties de la station. Il est chargé de sérialiser (mise de l'information sous forme d'une série de chiffres binaires) ou de désérialiser les bits composant les octets à émettre ou à recevoir.

2.1.2. - SENS DE TRANSMISSION

Un support de transmission est dit :

- . "simplex", s'il n'autorise la transmission que dans une seule direction.
- . "half-duplex", s'il autorise la transmission dans les deux directions, mais dans un seul sens à la fois.
- . "full-duplex" ou "duplex", s'il autorise la transmission dans les deux directions simultanément.

La transmission n'est full-duplex que si le coupleur permet ce mode de fonctionnement. Cependant, la combinaison d'un support full-duplex et d'un coupleur half-duplex est parfois utilisée pour économiser le temps que mettrait un modem half-duplex à inverser son sens de transmission ("temps de retournement" du modem).

Généralement, une ligne full-duplex est à 2 fils et une ligne full-duplex à 4 fils. Cependant, une ligne à 2 fils peut être utilisée en full-duplex par un modem utilisant une bande de fréquences haute vitesse et une bande de fréquences basse vitesse séparées.

2.1.3. - TOPOLOGIE D'UNE LIAISON :

La liaison peut être :

- point à point, si elle relie deux stations situées aux extrémités de la ligne (fig. 2).

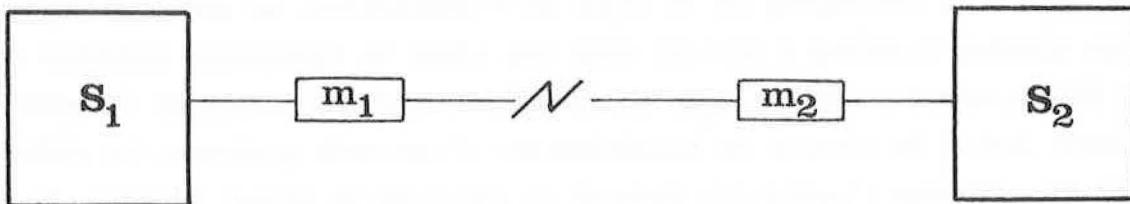
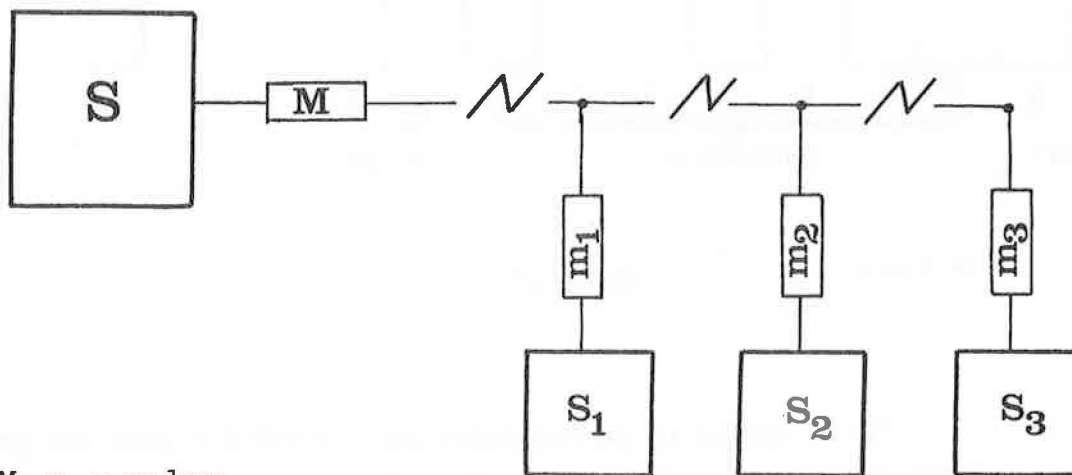


Figure 2

$\left\{ \begin{array}{l} S_i : \text{station} \\ m_i : \text{modem} \end{array} \right.$

- multipoint , si elle relie une station primaire à plusieurs stations secondaires (fig. 3). Les échanges d'information se produisent nécessairement entre une station secondaire et la station primaire.



$\left\{ \begin{array}{l} M, m_i : \text{modems} \\ S, S_i : \text{stations} \end{array} \right.$

Figure 3

2.1.4. - FORMAT DE TRANSMISSION :

La transmission peut s'effectuer en :

- format asynchrone, si chaque caractère est transmis en tant qu'entité indépendante. C'est-à-dire que l'intervalle de temps entre le dernier bit d'un caractère et le premier bit du caractère suivant peut être quelconque.

Pour être identifiée, la suite de bits définissant un caractère doit être transportée dans une enveloppe. La technique la plus couramment utilisée est celle du "Start-Stop" (fig. 4) qui consiste à faire précéder chaque caractère d'un signal de synchronisation (un bit de Start) et à le faire suivre d'un signal de fin (un ou deux bits de Stop). Le bit "Start" permet la synchronisation-caractère de l'équipement récepteur.

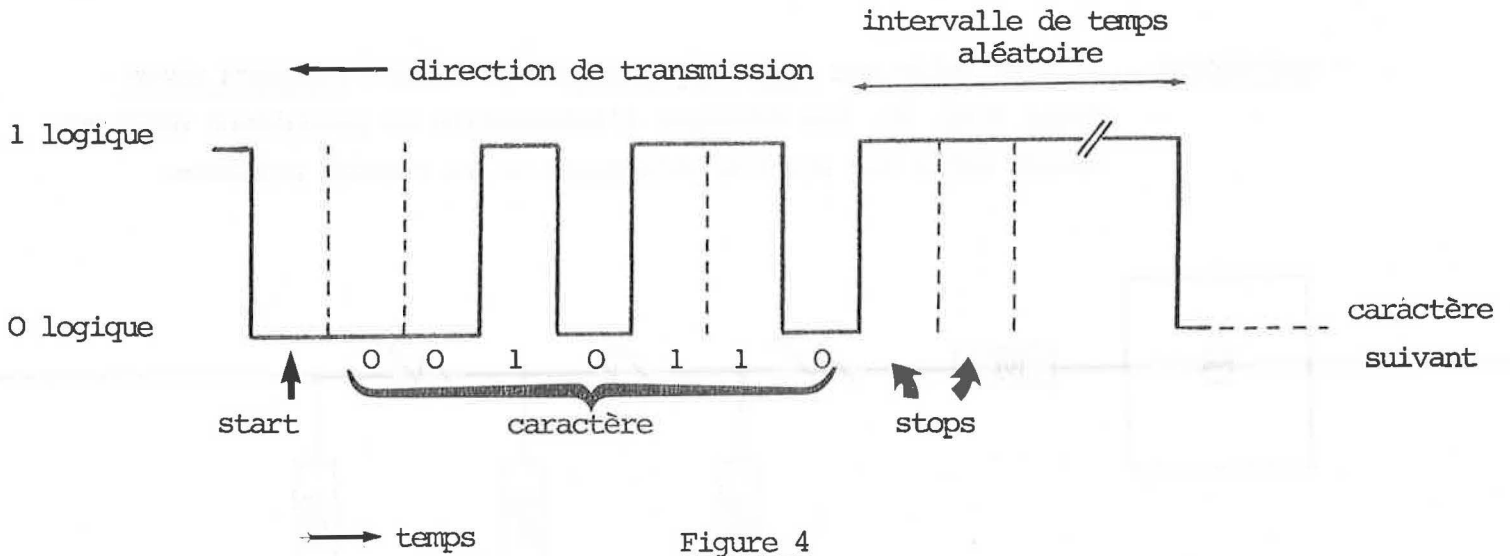


Figure 4

Cette technique de transmission, introduite pour les premiers télégraphes électromécaniques, n'utilise pas de façon optimale la capacité de la ligne de transmission puisque, sur 10 bits transmis pour un caractère en code ASCII, il n'y a que 7 bits d'information utile. Toutefois, plus simple à mettre en oeuvre au niveau d'un coupleur ou d'un terminal, elle reste en usage pour les terminaux lents opérant en mode caractère.

- Format synchrone, si les caractères sont transmis successivement et sans intervalle de temps entre eux à l'intérieur d'un bloc de caractères (fig. 5). L'intervalle de temps entre 2 blocs peut être quelconque.

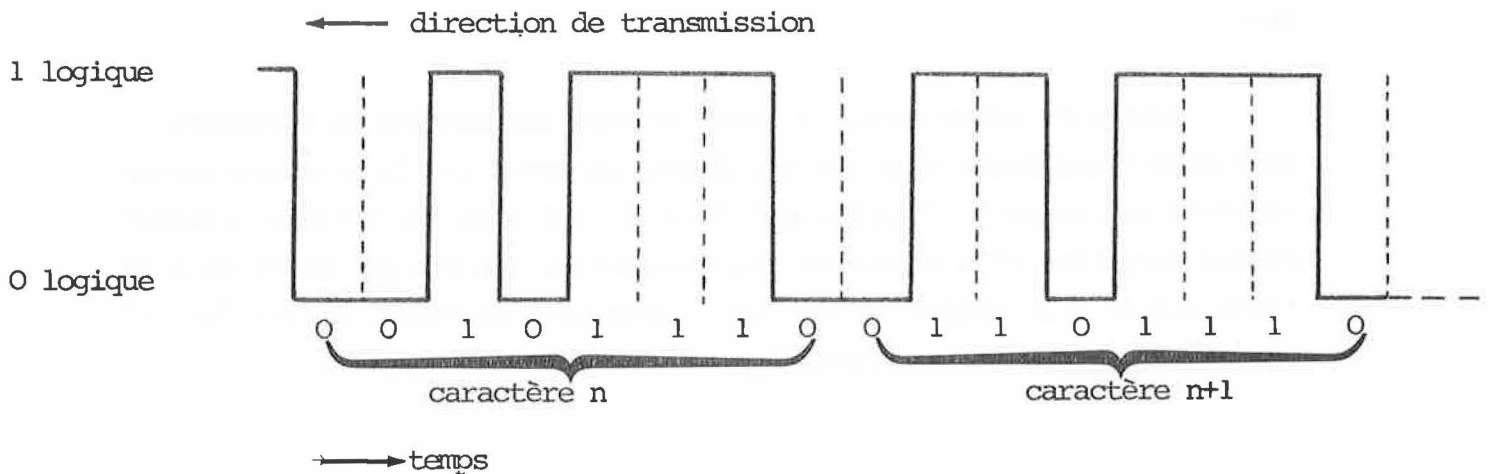


Figure 5

Chaque bloc doit être précédé d'une séquence de caractères prédéfinis de synchronisation (caractères SYN) permettant à l'équipement récepteur d'effectuer une synchronisation-caractère. Le coupleur récepteur filtre tous les caractères SYN consécutifs reçus jusqu'au premier caractère différent qui est considéré comme premier caractère significatif du bloc transmis. Le dernier caractère d'un bloc est également particularisé.

Ce mode de transmission nécessite l'accumulation et la mémorisation de tous les caractères reçus dans un bloc, soit au niveau du coupleur, soit au niveau de l'ordinateur ou du terminal en évitant toute erreur de cadence en réception. Il est plus efficace que le précédent mais requiert un équipement plus coûteux aux deux extrémités de la ligne.

La synchronisation-caractère des équipements émetteur et récepteur est donc liée au format de transmission utilisé.

2.1.5. - CADENCE DE TRANSMISSION :

La cadence de transmission (des bits à l'intérieur d'un bloc ou d'un caractère) est indépendante du format de transmission. La synchronisation-bit est liée au type des modems utilisés.

- Si les modems sont équipés d'une horloge, l'horloge du modem émetteur impose la cadence (synchrone) de transmission. Ceci est vrai quel que soit le format de transmission utilisé. Le signal modulé reçu permet à l'horloge du modem récepteur de se caler sur l'horloge du modem émetteur : c'est la synchronisation-bit. Cette horloge fournit à l'interface démodulateur la période d'échantillonnage du signal reçu. En format synchrone, on utilise presque toujours des modems équipés d'une horloge. En format asynchrone cela implique que les bits se succèdent de façon synchrone à l'intérieur d'un même caractère, et que les caractères se suivent de façon asynchrone tout en étant séparés par un multiple de la période du signal élémentaire.
- Pour les transmissions en format asynchrone avec des modems sans horloge, la synchronisation-bit est effectuée pour chaque caractère à la transition du bit Start. Pour la suite du caractère, la fréquence d'échantillonnage en réception est un multiple de la fréquence de transmission des données (fréquence qui a été approximée lors de la synchronisation-bit).

- Les vitesses de transmission se mesurent en nombre de bits par seconde (b.p.s) ou, par abus de langage, en bauds.

Les liaisons sont utilisées à des vitesses de 110 bps à 300 bps en mode asynchrone, et de 1,2 k bps à 48 k bps en mode synchrone. Cependant, certains terminaux de visualisation peuvent être reliés, sur courte distance, en mode asynchrone jusqu'à 9,6 k bps.

- Le taux d'erreur sur ces lignes varie de 10^{-5} à 10^{-6} . Le bruit provoquant les erreurs donne des rafales de 2 à 5 bits erronées en moyenne.

2.1.6. - PROCEDURES DE TRANSMISSION :

Sur une ligne de transmission reliant deux stations, on fait transiter non seulement des données mais également des informations de contrôle permettant aux deux stations de coordonner leurs activités et de s'entendre sur la bonne compréhension des messages échangés.

Ce partage d'une même voie pour véhiculer à la fois le flot des données et les informations de contrôle est un concept rarement utilisé en informatique. En effet, en architecture d'ordinateur, les lignes de données et les lignes de contrôle sont presque toujours distinctes, de même qu'en programmation (structurée !) on veille à séparer les données des instructions qui opèrent sur ces données.

Ainsi, on est amené à respecter des règles permettant de faire la séparation entre caractères de données et caractères de contrôle sur une base temporelle plutôt que spatiale. Une station ne peut émettre un message n'importe quand. Elle doit contrôler le trafic sur la ligne et émettre ses messages aux moments appropriés. Par ailleurs, ces messages doivent être structurés de telle sorte que l'on puisse distinguer les données des informations de contrôle. Ces "moments d'émission" et "structures de messages" sont définis par une procédure de transmission que chaque station doit respecter (cf. Chap. III).

Une telle procédure de transmission donne les moyens :

- d'identifier une station.
- de déterminer, pour un ou plusieurs échanges, la station émettrice et la station réceptrice.
- d'identifier un message.

- de fournir une information redondante de contrôle pour la détection des erreurs.
- de signaler, à la station émettrice, la bonne ou mauvaise réception d'un message.
- d'engager une procédure de récupération des erreurs.

La procédure de transmission peut être soit gérée totalement ou partiellement au niveau matériel par le coupleur, soit totalement ignorée par celui-ci, auquel cas sa gestion doit être assurée par programme sur la station.

2.1.7. - CODES DE TRANSMISSION :

- Le code de transmission est déterminé par le code de représentation binaire auquel appartiennent les caractères de contrôle de la procédure de transmission. Dans les deux principaux codes utilisés, le code ASCII et le code EBCDIC, un certain nombre de caractères sont réservés au contrôle des transmissions.

Le code ASCII utilise une représentation des caractères à 7 bits, ce qui permet 128 combinaisons.

Le code EBCDIC utilise une représentation des caractères à 8 bits, ce qui permet 256 combinaisons.

- Une procédure de transmission n'accepte normalement que les données appartenant au code de transmission qu'elle utilise. Cependant, certaines techniques permettent de transmettre des données indépendantes du code de transmission, et en particulier des données utilisant des configurations du code réservées pour le contrôle de la transmission. On dit alors que la procédure de transmission opère en mode transparent et permet de transmettre toutes les configurations binaires.
- La seule technique actuellement standardisée est le "doublement du DLE". Le caractère DLE est un caractère de contrôle particulier qui, suivi d'un autre caractère de contrôle, définit une fonction de contrôle pour le mode transparent. A l'intérieur d'un bloc transparent, aucun caractère de contrôle n'a d'effet s'il n'est pas précédé du caractère DLE. Pour éviter qu'une configuration occasionnelle "DLE - caractère de contrôle" dans les données ne soit interprétée comme fonction de contrôle, toutes les configurations DLE rencontrées dans les données sont doublées à l'émission et dédoublées à la réception.

2.2. - CLASSIFICATION DES RESEAUX D'ORDINATEURS

Le terme de réseau est généralement utilisé pour qualifier une installation informatique dont un certain nombre de composants matériels sont répartis géographiquement. Cette définition est suffisamment large pour recouvrir une grande variété d'installations, de propriétés assez distinctes [GEN-2, GEN-3, GEN-5]. Nous pouvons toutefois les regrouper en 2 types principaux.

2.2.1. - RESEAUX CENTRALISES :

Ce type de réseau est le plus ancien et encore le plus répandu. Il correspond simplement à une décentralisation des moyens d'accès.

2.2.1.1. - Réseaux étoilés :

Ces réseaux relient directement plusieurs terminaux distants à un ordinateur central (fig. 6). On peut y rencontrer une grande variété de terminaux depuis le simple "télétype" jusqu'au terminal lourd de "remote batch". Avec leur apparition se sont développées les techniques de temps partagé et de télétraitement par lot permettant le partage des ressources de l'ordinateur par plusieurs usagers indépendants. Le plus souvent, ces réseaux sont utilisés par des systèmes de temps partagé et dans des applications commerciales (interrogation de fichiers de commandes, transactions bancaires, réservation de places d'avions etc...).

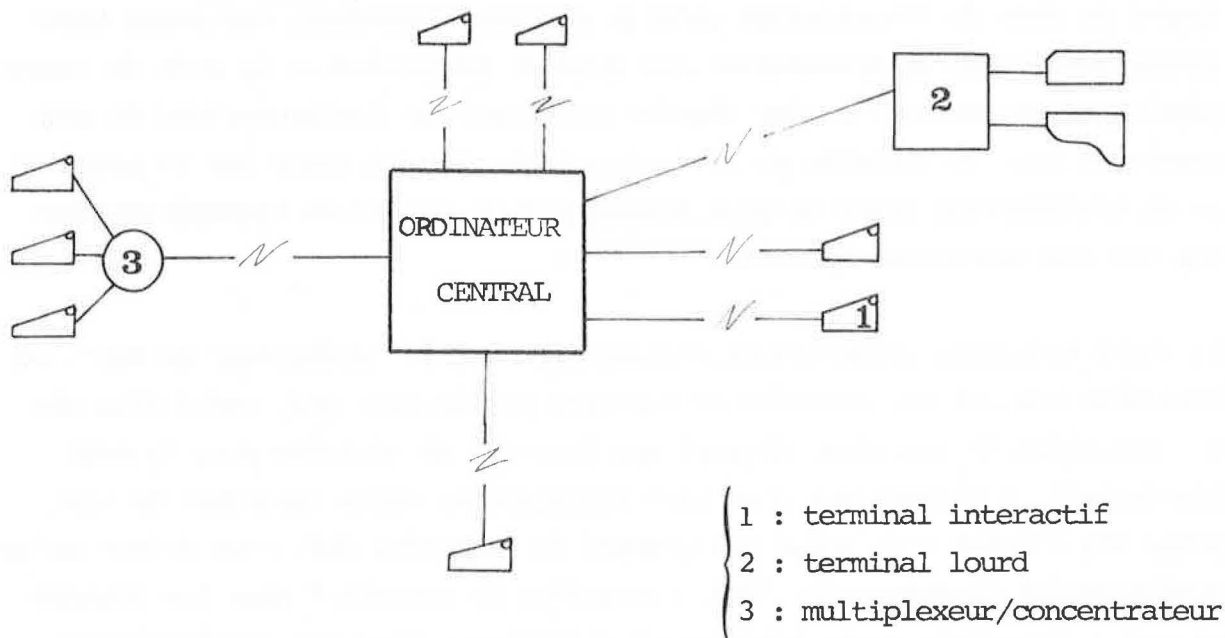


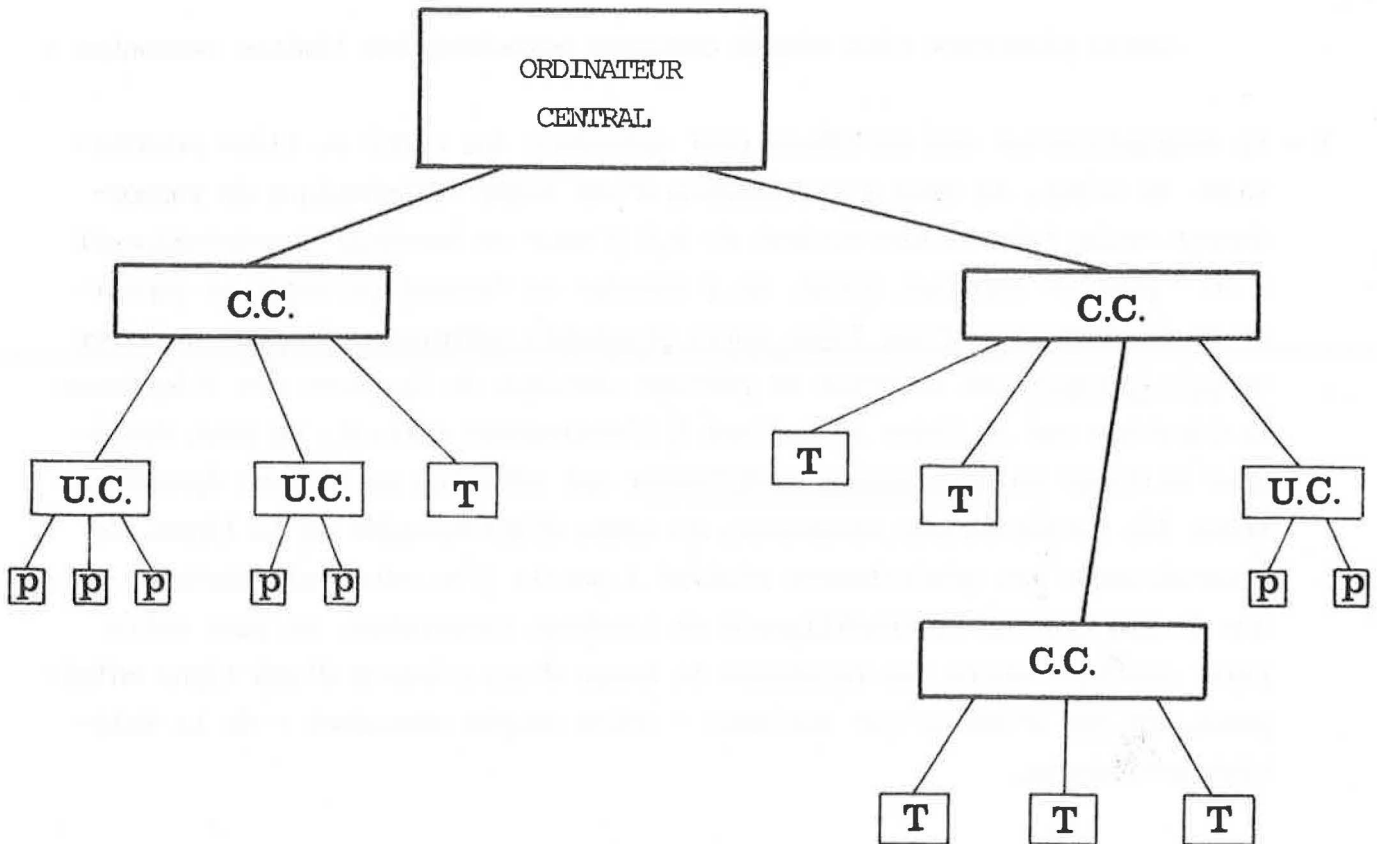
Figure 6

Cette structure très simple comporte cependant les limites suivantes :

- 1 - La multiplication des terminaux peut entraîner des coûts de ligne prohibitifs. En effet, le taux d'utilisation d'une ligne téléphonique de raccordement reste très faible variant de 0,5 % pour un terminal conversationnel à 50 % pour un terminal lourd. On y remédie en faisant partager la capacité de transmission d'une ligne entre plusieurs terminaux. On peut utiliser un multiplexeur qui effectue un partage statique de la plage des fréquences utilisables sur la ligne le reliant à l'ordinateur central. On peut également utiliser un concentrateur/diffuseur qui effectue un partage dynamique, selon les activités des terminaux, du temps d'utilisation de la ligne. Ce concentrateur est généralement réalisé à partir d'un mini-ordinateur en raison de ses besoins en intelligence et stockage temporaire. On peut enfin faire partager entre les terminaux le temps d'utilisation d'une ligne multipoint, ce qui n'est qu'une variante - moins souple cependant - de la solution précédente.
- 2 - Toute l'installation repose sur la fiabilité de l'équipement central.
- 3 - Il existe un chemin de communication unique entre un terminal et l'ordinateur central.
- 4 - L'accès au réseau est souvent limité à une certaine gamme de terminaux.

2.2.1.2. - Réseaux arborescents :

- Les premiers réseaux centralisés se sont souvent constitués au coup par coup, sans une stratégie-réseau bien définie. Actuellement apparaissent sur le marché des systèmes spécialement conçus dans une optique-réseau et pour des applications très diverses. Ainsi, le produit S.N.A. ("Systems Network Architecture") d'I.B.M. se veut un outil aussi bien matériel que logiciel pour la constitution de réseaux complets [IBM-1, IBM-2, IBM-3]. Ces réseaux restent cependant centralisés et n'éliminent pas les contraintes précédentes (fig. 7). Cependant l'architecture proposée facilite la distribution du contrôle des communications et même de certains traitements, en les partageant entre l'ordinateur central et des contrôleurs programmables situés aux noeuds du réseau.



- { C.C. Contrôleur de communication
- { U.C. Unité de contrôle
- { T Terminal
- { P Organe Périphérique

Figure 7

2.2.2. - RESEAUX DISTRIBUES

Il se différencie des précédents par une décentralisation des moyens de traitement et, généralement, leur grande dimension.

2.2.2.1. - Réseaux spécialisés

Plusieurs ordinateurs distants sont directement reliés par des lignes de transmission pour une application définie. Bien souvent c'est l'application qui définit la répartition. On peut ainsi citer le réseau de la SITA (Société Internationale de Télécommunications Aéronautiques) ou les réseaux "MARK III" (General Electric) et "TYMNET" (Tymshare).

La plupart de ces réseaux, plus ou moins maillés, utilisent le principe de la commutation de messages (fig. 8). Un ordinateur A qui désire émettre un message pour un ordinateur B le transmet à l'un de ses voisins et le processus est itéré jusqu'au destinataire. Cette structure procure une plus grande fiabilité de transmission et une meilleure utilisation des lignes que la commutation de lignes, utilisée sur le réseau téléphonique, qui mobilise simultanément toutes les lignes intermédiaires entre les deux correspondants. On peut ainsi répartir le trafic sur des itinéraires différents, selon la densité des messages et l'état des lignes.

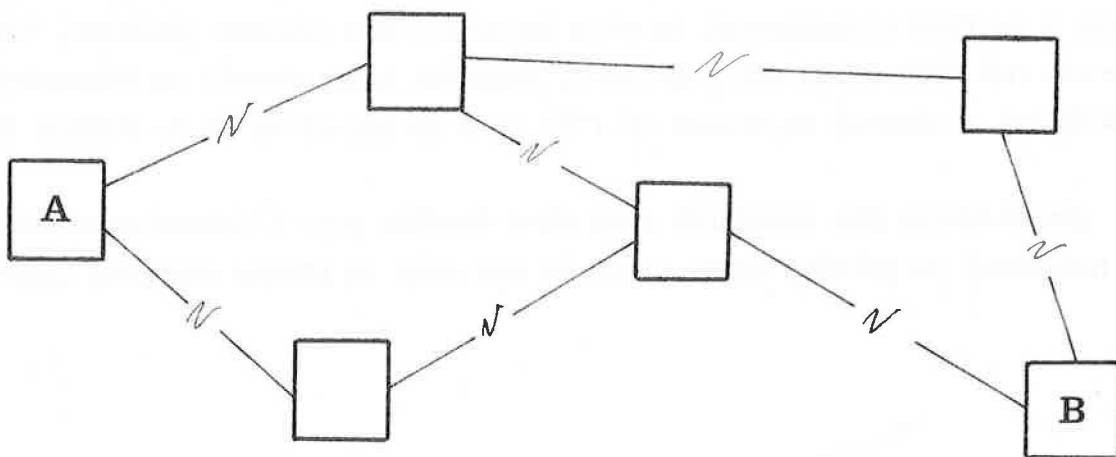


Figure 8

2.2.2.2. - Réseaux généraux

Ils ne sont plus conçus en fonction d'une application particulière. Il s'agit cette fois de construire un outil qui puisse servir de support à des applications diverses, en leur permettant éventuellement de dialoguer entre elles [GEN-4]. Ils en sont encore souvent au stade expérimental mais se sont fixés des objectifs très ambitieux. Les principaux buts sont :

- de permettre à un plus grand nombre d'utilisateurs d'accéder à un plus grand nombre de ressources, sans fixer à priori de restrictions sur la localisation et les caractéristiques des ressources et des utilisateurs.
- d'assurer une plus grande flexibilité, en rendant facile l'insertion sur le réseau ou le retrait du réseau d'un ordinateur quelconque.

- d'assurer une sécurité telle que le mauvais fonctionnement de l'un des sites ne perturbe pas le réseau tout entier.

Un certain nombre d'expériences ont été menées sur des réseaux homogènes, c'est-à-dire faisant coopérer des matériels et logiciels identiques aux différents sites. C'est le cas du réseau CYBERNET de la compagnie CDC [CDC-1], du réseau DECNET de D.E.C. [DEC-1] ou du réseau SOC en France. Mais les projets de grande taille, devant composer avec des installations existantes, ont conduit à des réseaux hétérogènes. C'est le cas du réseau ARPA ("Advanced Research Project Agency"), qui fut lancé aux Etats-Unis en 1968 sous la direction de M. L. ROBERTS et qui est à la fois l'ancêtre et le plus important des réseaux généraux. C'est aussi le cas des réseaux du NPL ("National Physical Laboratory") au Royaume-Uni et CYCLADES qui a démarré en France en 1972 sous la direction de M. POUZIN (fig.9).

Actuellement des standards sont même étudiés pour l'interconnexion de réseaux nationaux ou privés, comme c'est le cas pour le réseau européen COST-11.

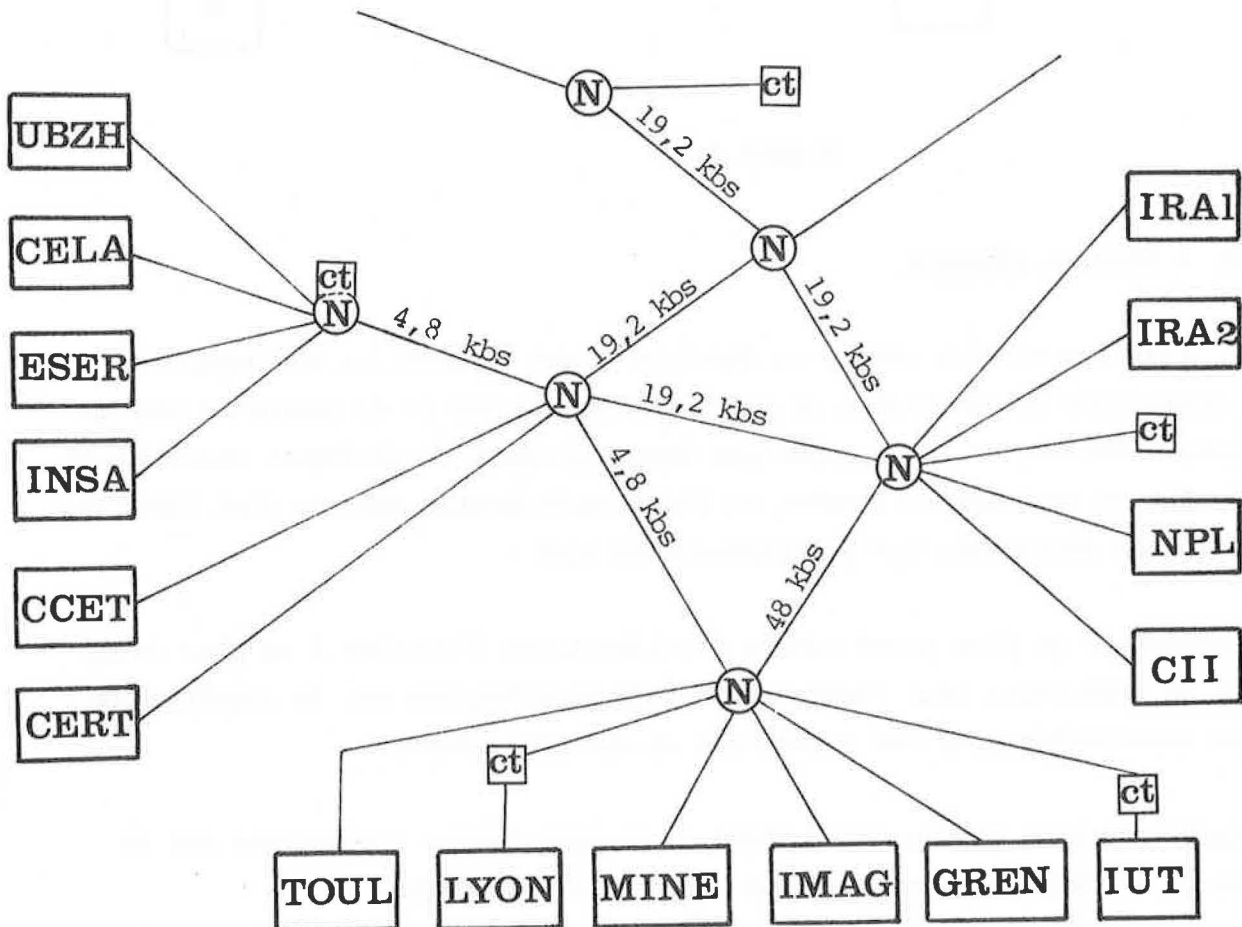


Figure 9

N : noeud du réseau CIGALE : MITRA-15

CT : concentrateur de terminaux/terminal de "remote-batch" : MITRA-15.

IRIA 1 : CII IRIS 80

IRIA 2 : CII IRIS 80

NPL : modular 1 (gateway)

CII : CII IRIS 80/CII 10070

IUT : CII IRIS 45

GREEN : CII IRIS 80

IMAG : IBM 360/67

MINE : TELEMECANIQUE T-1600 frontal de PHILIPS P1175 et PDP 11/40.

LYON : CII IRIS 80

TOUL : CII IRIS 80

CERT : CII 10070 - CII 10020 couplés

CCETT : CII IRIS 80 (centre de mesures)

INSA : TELEMECANIQUE T1600

ESER : HEWLETT-PACKARD 1200

CELAR : CII 10070

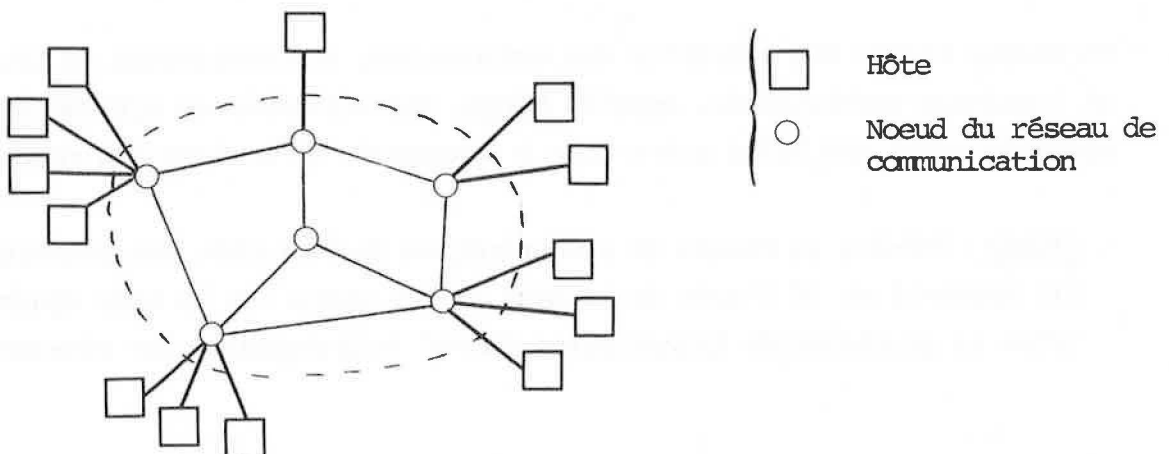
UBZH : CII 10070

2.3. - CARACTERISTIQUES D'UN RESEAU GENERAL HETEROGENE

Nous prenons comme références dans ce paragraphe le réseau CYCLADES [CYC-1] qui sert de support à notre étude et le réseau de l'ARPA qui a une structure et des objectifs comparables à CYCLADES.

2.3.1. - ARCHITECTURE DE BASE

La structure du réseau est à 2 niveaux (fig. 10)



- Un premier niveau assure une connexion permanente entre tous les utilisateurs du réseau (ordinateurs ou terminaux) et est constitué d'un réseau maillé dont les noeuds sont de petits ordinateurs et les arcs des lignes de transmission. C'est le réseau de communication qui offre aux utilisateurs un service de transmissions de blocs d'information. Ce réseau utilise la technique de commutation de paquets qui est dérivée de la commutation de messages cf. (2.2.2.1.). En effet, dans tout système de communication entre calculateurs, le choix d'une taille optimale pour l'unité de transfert tient compte de 2 objectifs apparemment contradictoires :

- . assurer un débit utile maximal, ce qui semble prêcher en faveur de grandes tailles.
- . minimiser les capacités de stockage nécessaires aux noeuds intermédiaires et la quantité d'informations à retransmettre en cas d'erreur, ce qui impose une limite sur la taille.

Il a été montré que la taille optimale, pour la technologie actuelle, était d'environ 2000 bits soit environ 250 octets. Pour cela, les messages à transmettre sont découpés en paquets à l'émission. Ils sont acheminés indépendamment puis réunis en ordre et réassemblés à la réception avant d'être délivrés au destinataire. Comme nous allons le voir dans ce qui suit, les lieux (hors du réseau de communication ou à l'entrée et à la sortie) de fragmentation/réassemblage et de remise en ordre de ces paquets, peuvent différer selon la politique choisie. Les paquets acheminés par le réseau de communication ont donc une taille limitée et comportent un certain nombre d'informations de contrôle dont l'adresse de l'utilisateur destinataire. Ils sont transmis de noeud en noeud suivant un itinéraire évolutif déterminé, à chaque noeud traversé, selon un algorithme de routage tenant compte de la densité du trafic et de l'occupation des diverses lignes.

- Un second niveau est constitué des ordinateurs, concentrateurs de terminaux et terminaux participants, appelés Hôtes, qui utilisent ce système de communication comme une boîte noire pour y déposer et en retirer des messages.
- . CIGALE [CYC-2], le réseau de communication de CYCLADES, est constitué de CII MITRA-15 de 16 K mots de 16 bits reliés entre eux en mode synchrone, selon la procédure de transmission TMM-UC full-duplex, à une vitesse variant

de 4,8 à 48 kbps. Les hôtes communiquent entre eux par paquets d'une taille maximale de 255 octets qu'ils délivrent au réseau. L'ordre des paquets n'est pas préservé par le réseau. Leur réordonnement éventuel est à la charge de l'hôte destinataire.

- . Le réseau de communication de l'ARPA [ARP-1] est constitué d'HONEYWELL DDP 516 et DDP 316 de 12 K mots de 16 bits appelés IMPs ("Interface Message Processor") généralement reliés entre eux à une vitesse de 50 kbps. Les hôtes communiquent entre eux par messages d'une taille maximale de 1012 octets qu'ils délivrent au réseau. Chacun de ces messages est découpé par l'IMP source en paquets qui sont transmis indépendamment jusqu'à l'IMP destinataire où ils sont réassemblés en un message délivré à l'hôte destinataire. L'ordre des messages est conservé par le réseau de communication mais l'IMP source doit veiller à ne pas accepter plus d'un message à la fois sur un circuit virtuel établi entre deux hôtes qui échangent des messages.

Il apparaît donc que le réseau de communication de l'ARPA est moins transparent que CIGALE au réseau des hôtes utilisateurs.

2.3.2. - ARCHITECTURE DES PROTOCOLES-RESEAU

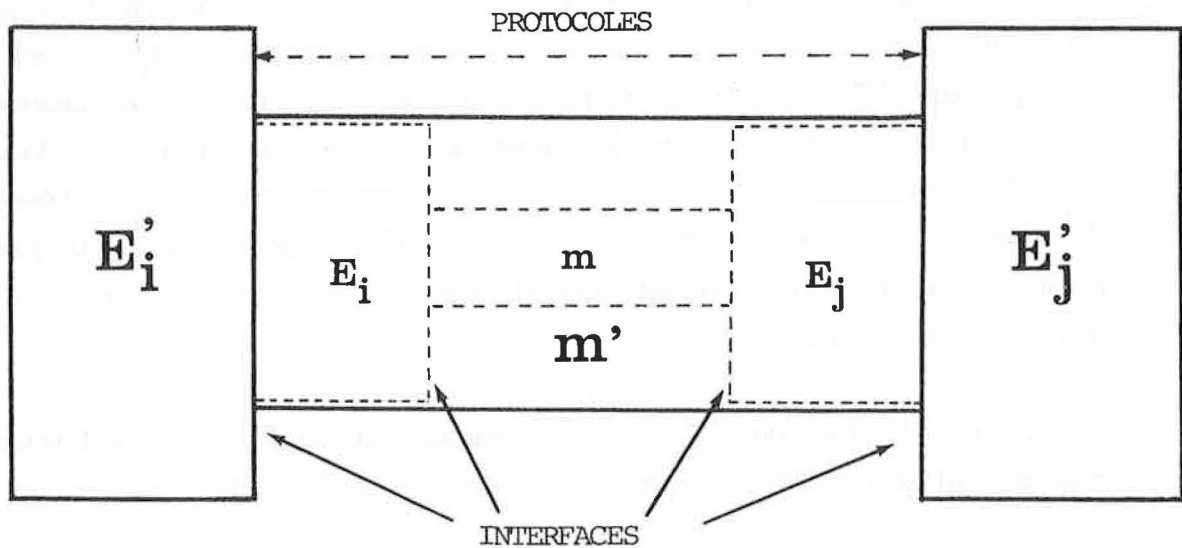
Vis à vis du réseau de communication un hôte a les fonctions d'une source susceptible d'émettre des messages et d'un puits susceptible de recevoir des messages. Cet hôte - qui peut être un ensemble d'ordinateurs, un ordinateur, une machine virtuelle sur un ordinateur ou un groupement de terminaux - est une installation virtuelle référencée par une adresse unique sur le réseau. Sur chacune de ces installations virtuelles se situent un certain nombre d'entités telles que programmes, processus, fichiers, organes périphériques..., qui doivent pouvoir communiquer entre elles, indépendamment de leur localisation, pour coopérer en une activité-réseau.

La coopération de deux entités correspondantes s'effectue par échange de messages :

- à travers une fonction de communication qu'elles considèrent comme une "boîte noire", ou une Machine, dont seules les caractéristiques externes leur importent (fig. 11).

- et en suivant un ensemble de règles de communication définies par un protocole (cf. Chap. IV).

Il y a bien lieu ici de distinguer entre un interface, qui définit une interprétation entre des entités situées à deux niveaux adjacents, et un protocole qui gouverne les communications entre des entités situées au même niveau [CYC-3].



$\{E, E'\}$ - entités réseau
 $\{m, m'\}$ - machines de communication

Figure 11

Situées dans des contextes hétérogènes souvent peu compatibles, ces entités ne sont pas explicitement spécifiées dans la définition des protocoles qui manipulent plutôt, comme nous allons le voir, des entités-réseau plus abstraites.

2.3.2.1. - Les protocoles-CYCLADES

- l'accès au réseau se fait par l'intermédiaire d'une Station de Transport (ST). Chaque hôte possède une station de transport dont l'accès est réservé à ses Abonnés (AB) (fig. 12 et 13). Les stations de transport gèrent les communications entre paires d'abonnés selon un protocole de transport qui permet à deux abonnés de s'échanger des informations :

- . soit par envoi de lettres régulières (RG). Les lettres sont acheminées indépendamment les unes des autres, avec ou sans accusé de réception.
- . soit par envoi de lettres sur voie virtuelle (VV). Si deux abonnés doivent échanger une grande masse d'information, ils établissent entre eux une ou plusieurs voies virtuelles. Chaque voie virtuelle permet d'acheminer un flux de données dans les deux sens de transmission en offrant les services additionnels suivants :
 - la fragmentation des lettres en éléments et leur réassemblage permettent d'assurer la conservation du séquençement de l'information entre l'émetteur et le récepteur.
 - un contrôle d'erreur permet de s'assurer que chaque élément d'une lettre a été reçu une et une seule fois.
 - un contrôle de flux permet d'asservir le rythme de l'émetteur aux disponibilités du récepteur.
 - un mécanisme d'adressage rapide permet d'accélérer l'identification des abonnés.
- . dans les deux cas, des informations prioritaires et inattendues, ou signaux d'interruption, peuvent être véhiculées sous forme d'événements.

La Station de Transport place les lettres, ou fragments de lettres, confiées par ses abonnés dans des plis pour les expédier vers les stations de transport des abonnés destinataires. Les plis sont eux-mêmes confiés, à l'intérieur de paquets, au réseau CIGALE qui se charge de leur acheminement.

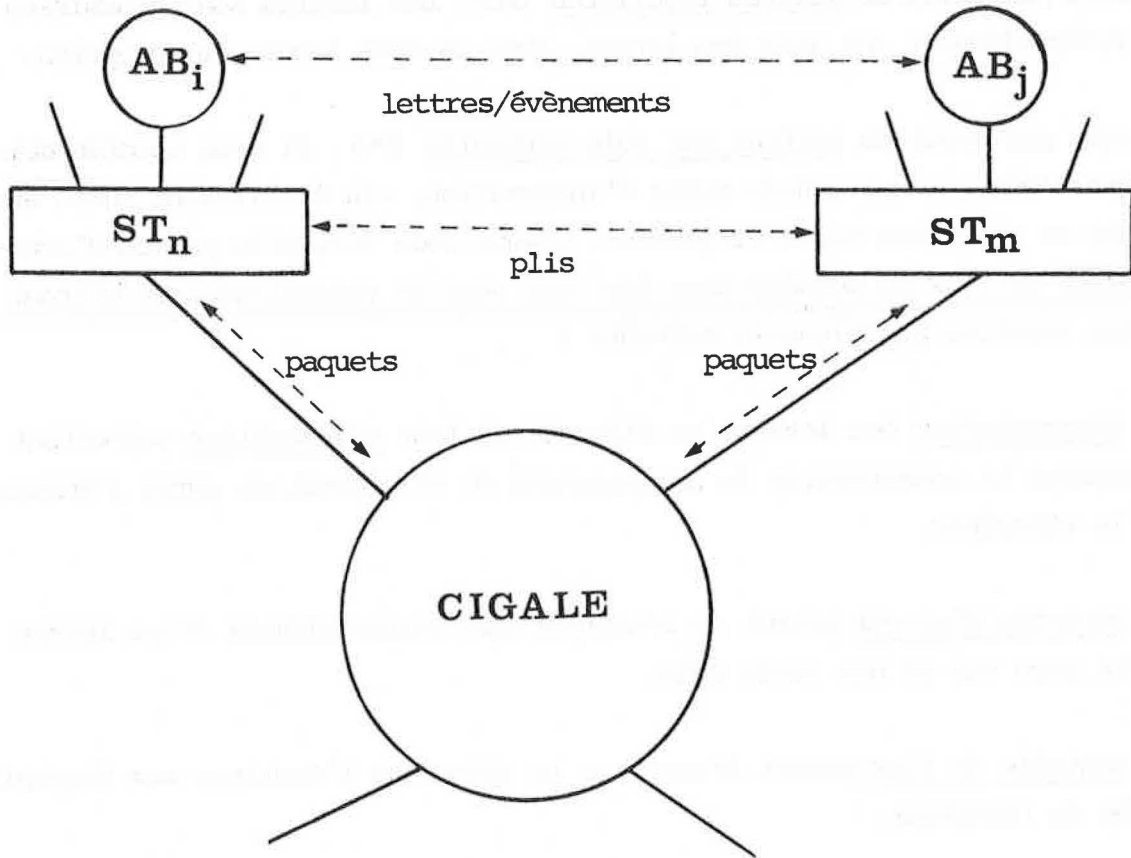


Figure 12

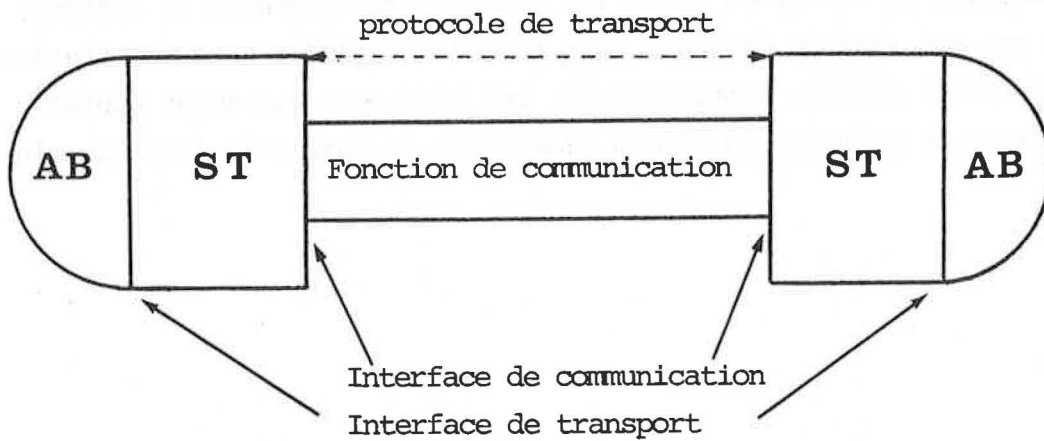


Figure 13

Dans le courant de l'année 1977, un nouveau protocole de transport [CYC-5], simplifié (et, pense-t-on, plus performant), doit être introduit sur le réseau. On en trouvera les principales caractéristiques au chap. IV (4.3.2.)

- Le concept d'abonné est suffisamment flexible pour recouvrir une large variété de ressources. Par exemple un abonné peut être un fichier, un utilisateur, un processus, un sous-système etc... Des protocoles de niveau supérieur, utilisant la fonction de transport, sont nécessaires pour que des activités-réseau, orientées vers des applications, puissent s'établir entre ces abonnés. Actuellement, 3 protocoles de haut niveau sont standardisés.

1°) Un protocole de dialogue inter-opérateurs permet l'échange de messages entre les opérateurs des différents hôtes.

2°) Un protocole client-serveur permet l'accès par des terminaux à des services distants de temps partagé ou de traitement par lot. On appelle serveurs les entités qui offrent de tels services et clients les entités qui les utilisent. Un client est un abonné du réseau et un serveur est un abonné ou un ensemble d'abonnés. Le déroulement d'une session de travail est le suivant [CYC-6] (figure 14) :

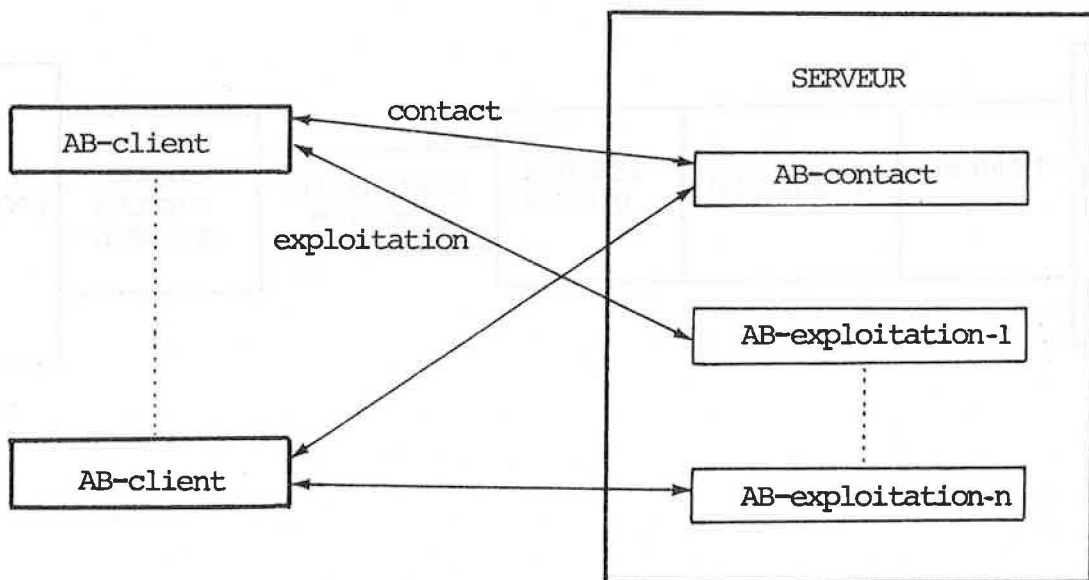


Figure 14

- prise de contact de l'abonné-client avec un abonné du serveur, connu de tous les clients potentiels et appelé abonné-contact. Les 2 parties s'accordent sur le protocole d'exploitation à utiliser.
- connexion de l'abonné-client à l'abonné-exploitation du serveur par une ou plusieurs voies virtuelles reliant les appareils du client aux entrées fonctionnelles du service. Par exemple un service de traitement par lot disposera de 3 entrées. : l'une pour la console-opérateur, l'autre pour un lecteur de cartes et la troisième pour une imprimante.
- exploitation du service selon le protocole convenu lors de la prise de contact.
- déconnexion à l'initiative de l'une des deux parties.

3°) Un protocole d'appareil virtuel, utilisé par le protocole client-serveur, définit un terminal virtuel unique et une méthode standard pour accéder à ce terminal [CYC-7]. Le terminal virtuel a des caractéristiques minimum susceptibles d'extension après négociation entre les deux parties.

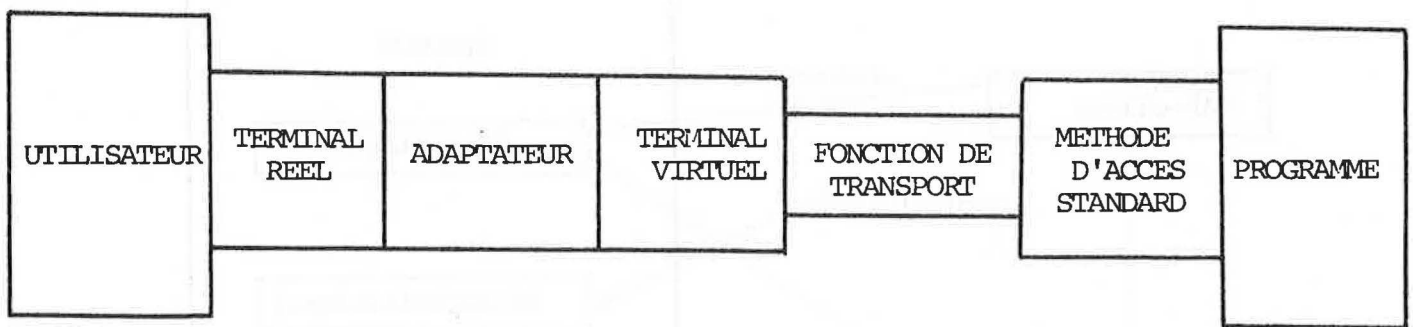


Figure 15

2.3.2.2. - Les protocoles ARPA

- L'accès au réseau se fait par l'intermédiaire d'un programme de contrôle-réseau ou NCP ("Network Control Program") situé sur chaque hôte (fig. 16). Pour communiquer entre eux les hôtes délivrent au réseau de communication des messages en respectant l'interface Hôte-IMP [ARP-1] (appelé aussi protocole par abus de langage) qui définit des circuits virtuels ou liens entre les NCP. Chaque lien joint une prise (ou "socket") émettrice d'un NCP à une prise réceptrice d'un autre NCP, de manière uni-directionnelle et en ne permettant pas le transfert de plus d'un message à la fois. Deux liens spéciaux ("liens zéro") permettent à un NCP donné d'échanger des messages de contrôle avec les autres NCP. La gestion de ces liens est partagée entre les NCP et les IMPs source et destinataire impliqués, ce qui constitue une différence essentielle avec CYCLADES qui utilise le réseau CIGALE de façon transparente.

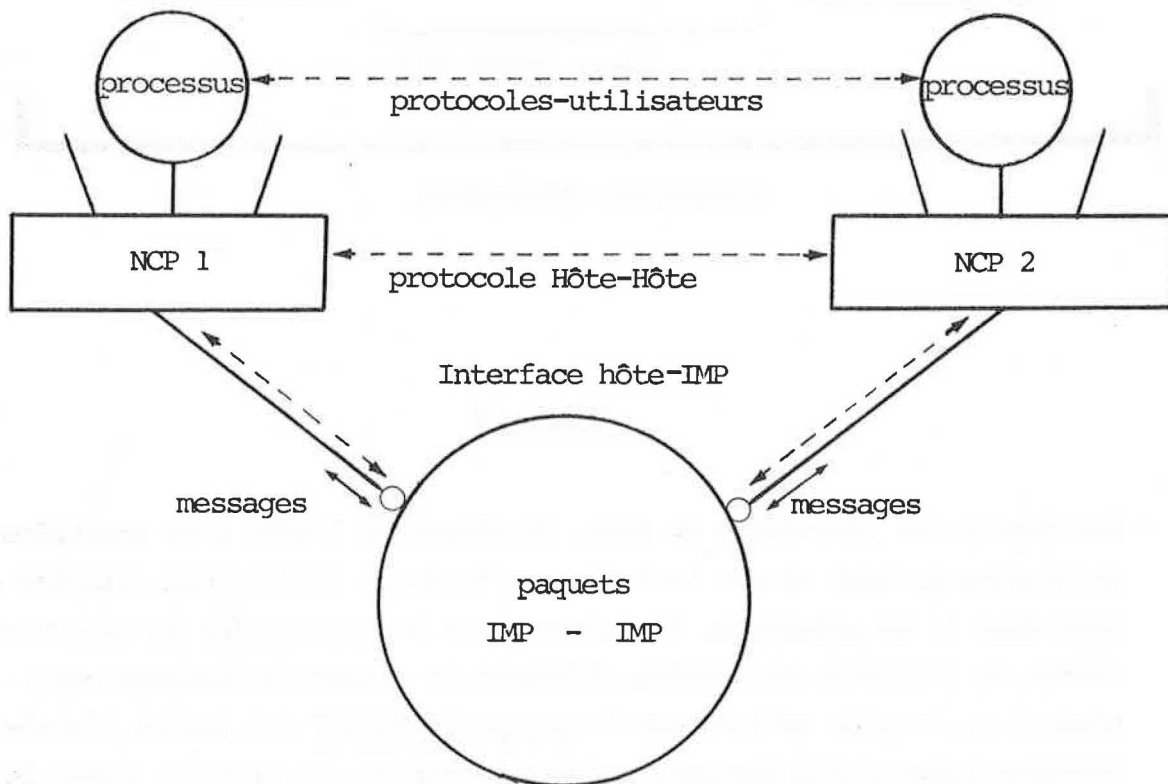


Figure 16

Les NCPs permettent à 2 processus distants de communiquer selon un protocole Hôte-Hôte [ARP-2]. Pour cela les NCPs associent chacun des ports de sortie ou d'entrée d'un processus à une prise émettrice ou réceptrice. Puis, ils établissent une connexion pour chaque paire de ports, à travers

le lien concerné. Le protocole Hôte-Hôte gère des échanges uni-directionnels sur chaque connexion reliant 2 processus. Chaque NCP est donc chargé d'effectuer la correspondance entre connexion-port et lien-prise (fig. 17)

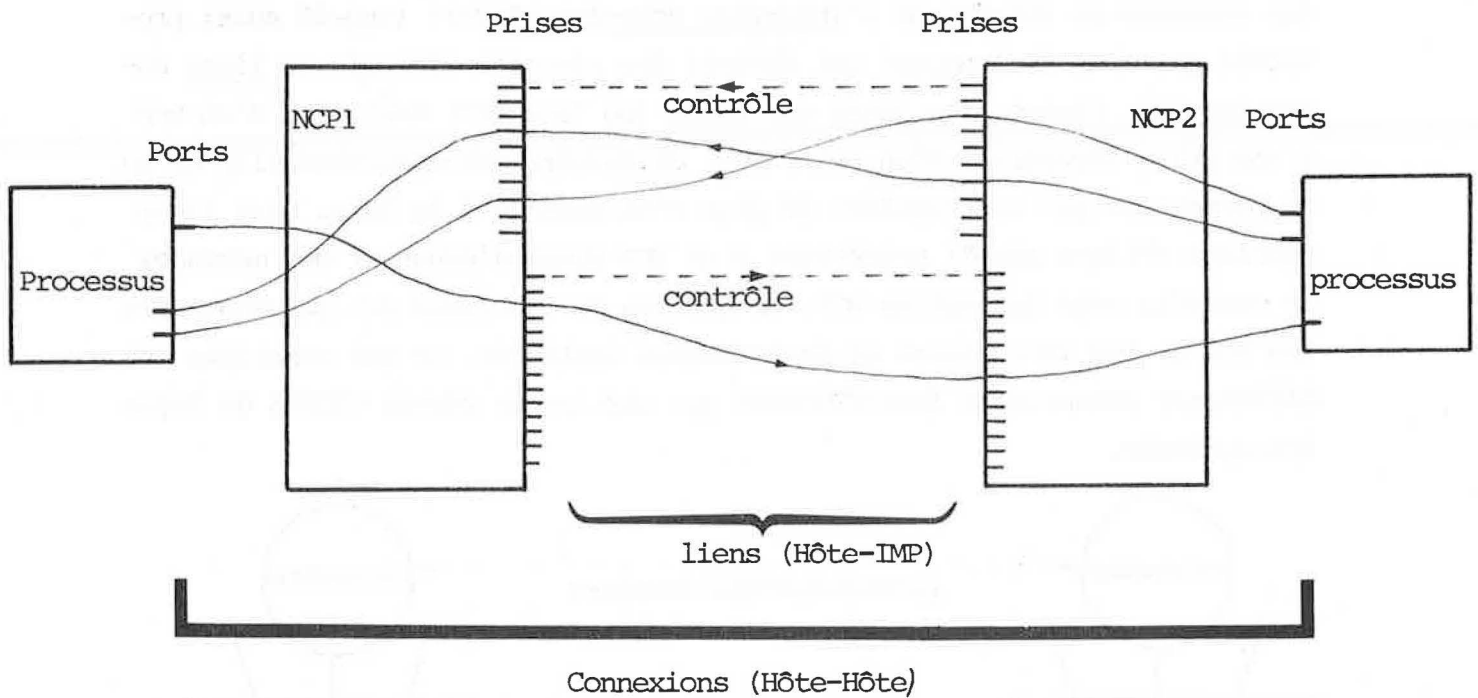


Figure 17

- Par dessus ces protocoles de base, le réseau de l'ARPA a vu proliférer les protocoles de haut niveau [ARP-4], pas toujours suffisamment simples et généraux dans leurs principes. Il existe ainsi des protocoles de transfert de fichiers, de transfert de données, d'entrée de travaux à distance etc... Le plus général et le plus utilisé est le protocole TELNET qui permet l'accès à des systèmes interactifs distants par des terminaux. Il utilise, comme le protocole Client-Serveur de CYCLADES, les notions de clients et serveurs et s'appuie également sur la définition d'un terminal virtuel.

CHAPITRE III

REALISATION ET OPTIMISATION DES AUTOMATES DE PROCEDURE

Dans ce chapitre, nous nous intéresserons aux problèmes posés par le contrôle des transmissions de données, selon une procédure de transmission. La longue histoire des transmissions de données a vu proliférer des techniques nombreuses et diverses conduisant à des conventions et des choix qui apparaissent complexes, et même souvent baroques, au non-spécialiste. Il en a résulté des implémentations de procédures de transmission par des programmes importants, confus, difficilement modifiables et d'une fiabilité douteuse.

Dans un souci de classification, nous rappelons d'abord les objectifs et les fonctions essentielles d'une procédure de transmission. Puis, à l'aide de quelques exemples, nous exposons une méthode simple de représentation graphique, sous forme d'automate d'état fini, conduisant à une définition compréhensible, complète, précise et sans ambiguïté de toute procédure. Ensuite, nous proposons un langage adapté à la nature d'une procédure de transmission et permettant de donner une description concise, lisible et opérationnelle des automates logiciels qui en assurent la gestion. Enfin, nous examinons plus en détail les problèmes posés par des cas pratiques d'implémentation, en nous appuyant sur les réalisations qui ont été menées sur le frontal étudié.

3.1. - PRINCIPES DES PROCEDURES DE TRANSMISSION

3.1.1. - Modèle de communication

3.1.2. - Types de procédures de transmission

3.1.2.1. - Procédures de transmission point à point

3.1.2.2. - Procédures de transmission multipoint

3.1.3. - Langage de communication

3.1.4. - Contrôle des erreurs de transmission

3.2. - RECONNAISSANCE D'UNE PROCEDURE DE TRANSMISSION PAR UN AUTOMATE

3.2.1. - Recherche d'un mode de représentation graphique

3.2.2. - Graphes de la procédure multipoint ECMA-PHILIPS

3.2.2.1. - niveau primaire-secondaire

3.2.2.2. - niveau grandes fonctions

3.2.2.3. - niveau bloc

3.2.2.4. - niveau caractère

3.2.3. - Graphes de la procédure point à point TMM-UC

3.2.3.1. - niveau primaire-primaire

3.2.3.2. - niveau maître-esclave

3.2.3.3. - niveau bloc

3.2.3.4. - niveau caractère

3.2.4. - Avantages de la méthode

3.3. - LANGAGE DE DESCRIPTION D'UN AUTOMATE DE PROCEDURE

3.3.1. - Représentation de la syntaxe et de la sémantique

3.3.2. - Description de l'automate dans un langage

3.3.2.1. - Sémantique du langage

3.3.2.2. - Syntaxe du langage

3.3.2.3. - Exemple de description

3.3.2.4. - Limites de la méthode

3.4. - OPTIMISATION DU TRAITEMENT

3.4.1. - Traitement des erreurs

3.4.2. - Minimisation du temps de traitement (maximum) d'un caractère

3.1. - PRINCIPES DES PROCEDURES DE TRANSMISSION

Nous ne considérons ici que les transmissions en mode synchrone sur des supports de type half-duplex ou full-duplex, le support simplex n'étant pas concerné par notre étude.

3.1.1. - MODELE DE COMMUNICATION

Deux stations qui échangent des messages à travers une ligne de transmission doivent chacune comporter un émetteur et un récepteur (fig. 1). Chaque émetteur attend un signal de sa station puis émet une chaîne de caractères qui constitue un message. Cette chaîne est transportée sur la ligne de transmission jusqu'au récepteur de la station vis-à-vis, qui, après un signal de sa station, essaie de reconnaître le message.

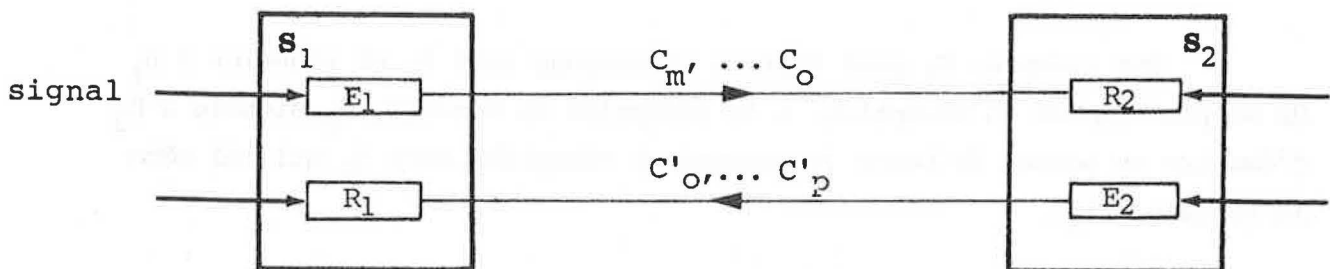


Figure 1

Ce modèle cependant ne suffit pas, car il ne tient pas compte :

- 1°) du fait que les messages de S₁ vers S₂ et les messages de S₂ vers S₁ doivent parfois se partager un même support physique de transmission (de type half-duplex) et donc transiter alternativement.
- 2°) du fait que les messages d'une station S₁ à une station S₂ appellent des confirmations de bonne réception de la station S₂ à la station S₁ et réciproquement.

Les deux stations doivent synchroniser leurs activités par des informations de contrôle. Pour cela, elles respectent une procédure de communication.

Et un modèle convenable (fig. 2) doit prendre en compte la synchronisation additionnelle qui en résulte entre le récepteur et l'émetteur d'une même station.

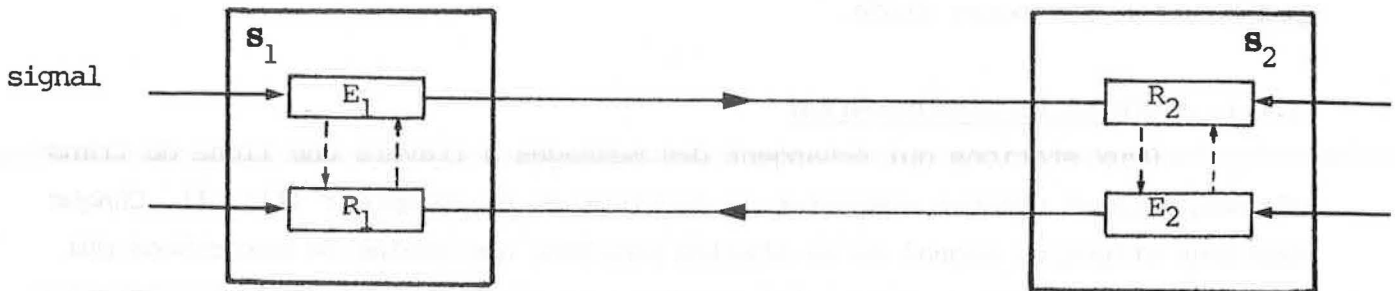


Figure 2 - Modèle de communication

Par exemple, E₁ peut émettre un message vers R₂ et signaler à R₁ de se positionner en réception. A la réception du message, R₂ signale à E₂ d'émettre un accusé de bonne (ou mauvaise) réception vers R₁ qui lui même le signale à E₁.

Remarquons bien qu'ici nous ne préjugeons absolument pas de la réalisation entièrement matérielle ou partiellement matérielle et partiellement logicielle des émetteur et récepteur de chaque station.

3.1.2. - TYPES DE PROCEDURES DE TRANSMISSION

Il existe de nombreuses procédures de transmission différentes, qui varient selon les caractéristiques du support de transmission, et surtout selon le constructeur du matériel employé. Toutefois, toutes les procédures utilisées sont des variantes de la procédure ISO ("International Standards Organization") publiée par l'ANSI ("American National Standards Institution") et reprise en Europe par le CCITT (Comité Consultatif International Télégraphique et Téléphonique).

On distingue généralement trois phases dans de telles procédures :

- l'initialisation des transferts qui détermine la station émettrice, ou station maîtresse, et la station réceptrice, ou station esclave.

- le transfert des messages qui englobe la transmission effective des messages de la station maîtresse à la station esclave, les accusés de bonne (ou mauvaise) réception renvoyés par la station esclave et les procédures de reprise sur erreur qui ne conduisent pas à un changement de statut (maître/esclave). La station maîtresse a l'initiative des reprises sur erreur.
- la fin des transferts qui remet chaque station au repos et survient :
 - . lorsque la station maîtresse, n'ayant plus rien à émettre, abandonne son statut maître et libère la station esclave.
 - . ou bien lorsque la station esclave demande l'interruption de la transmission.
 - . ou bien à la suite de mauvaises conditions de transmission.

Dans tous les cas, c'est la station maîtresse qui met fin aux transferts. Dans le détail l'organisation de chaque phase, et particulièrement de l'initialisation, est différente selon la topologie de la liaison (cf. 2.1.3.)

3.1.2.1. - Procédure de transmission point à point.

Dans ce cas, on ne préjuge pas, au départ, de la supériorité d'une station sur l'autre. La première ayant un message à émettre adresse une demande de statut maître (DSM) à sa vis à vis. Si elle est acceptée les statuts de maître et esclave sont respectivement donnés aux deux stations. Si le support de transmission est full-duplex, il n'y a pas de conflit possible puisque deux voies relient les stations qui, chacune, possèdent sur l'une des voies le statut maître et sur l'autre le statut esclave.

Par contre, si le support de transmission est half-duplex, il se peut que les deux stations tentent simultanément une demande de statut-maître. Alors, la première demande reconnue (au bout de plusieurs reprises éventuellement) prime l'autre. C'est le principe de la contention qui consiste à attribuer à chaque station des temps d'attente différents entre deux tentatives successives, par exemple dans les proportions 2/3 et 1/3.

La figure 3 donne un diagramme de temps simplifié de ce type de procédure. Les traits forts représentent le déroulement normal de la communication et les traits pointillés représentent les reprises en cas d'erreur ou de refus de l'une des stations. Des procédures d'erreurs spéciales peuvent être engagées en cas d'erreur persistante.

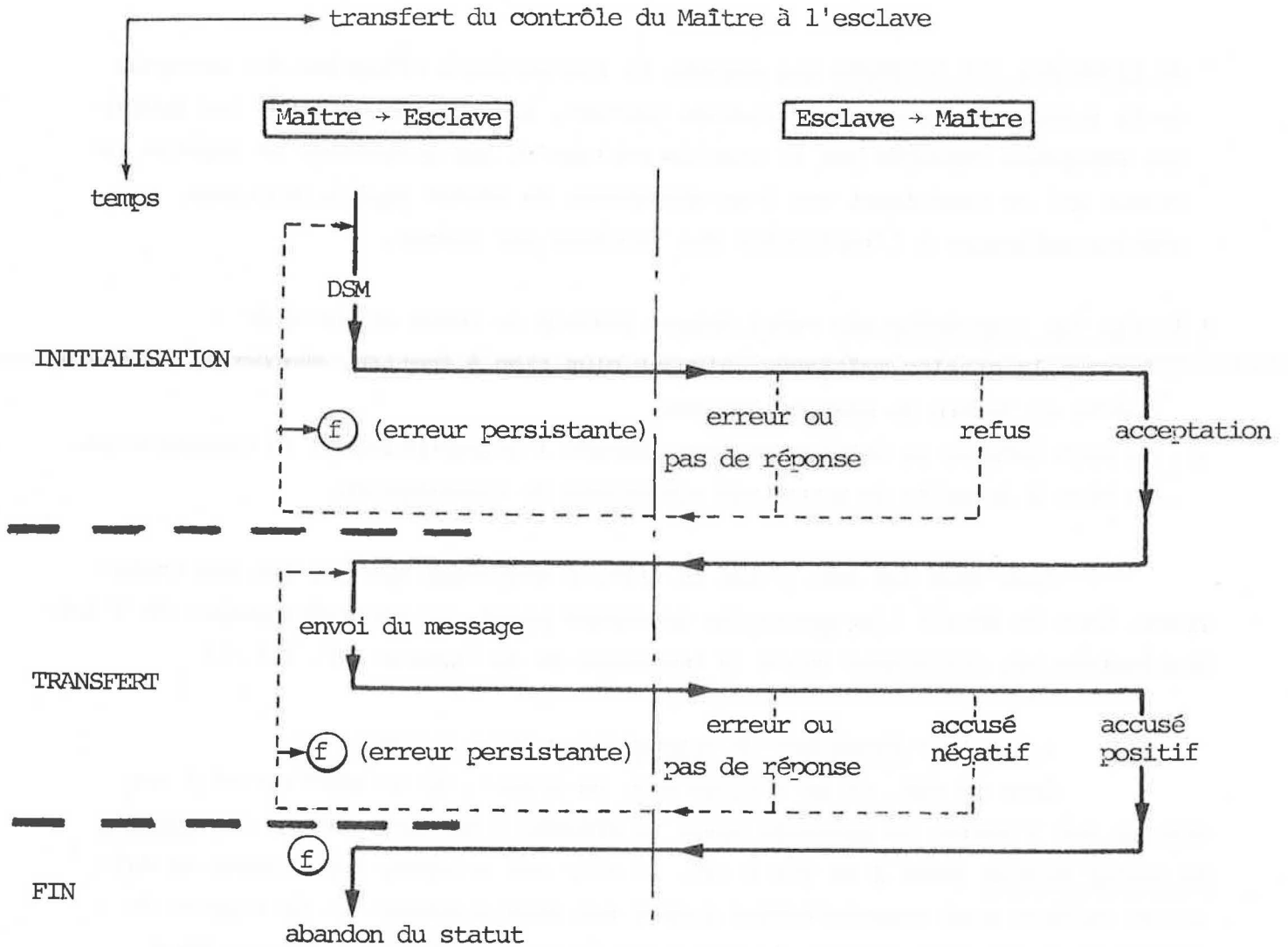


Figure 3 - Diagramme d'une procédure point à point

3.1.2.2 - Procédure de transmission multipoint

Dans ce cas, c'est la station primaire qui a l'initiative de toutes les communications. Toutes les stations secondaires reçoivent les messages émis par la station primaire, mais seule la station secondaire dont l'identification est contenue dans le message doit répondre. On distingue deux types d'initialisation de transfert.

- procédure de "polling" (invitation à émettre) : une station secondaire ne peut spontanément demander à émettre un message, mais doit attendre que la station primaire l'y invite. Pour cela, la station primaire émet une séquence de "polling" vers une station secondaire particulière. La station adressée répond en mettant fin à la communication si elle n'a rien à émettre ou répond par l'émission d'un message en prenant le statut maître (fig. 4).

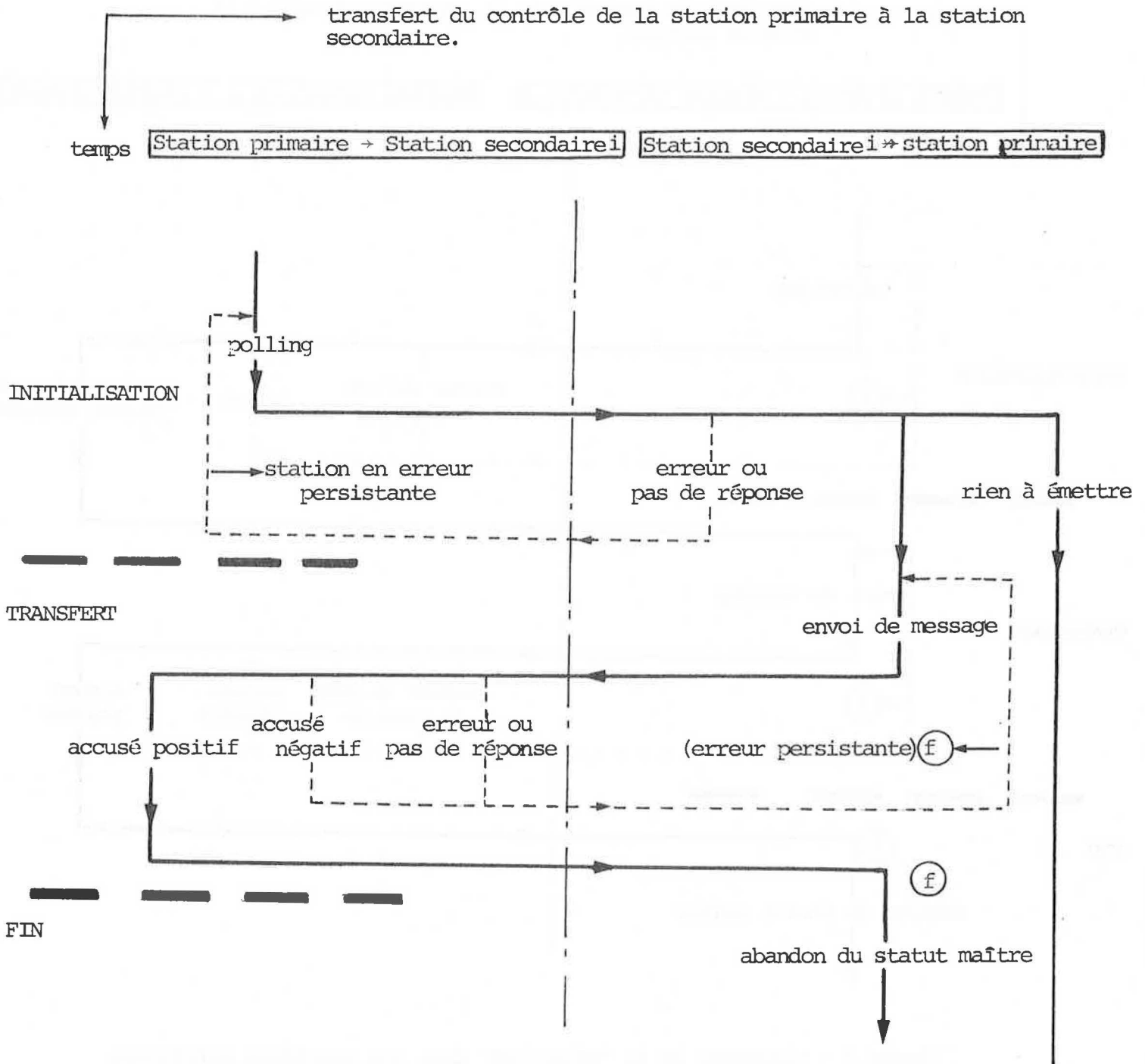


Figure 4 - diagramme du "polling" dans une procédure multipoint

- procédure de sélection (invitation à recevoir). La station primaire "sélectionne" la station secondaire à laquelle elle désire envoyer un message. Cette dernière peut ou non accepter le statut esclave. Dans l'affirmative, le transfert de message peut s'opérer.

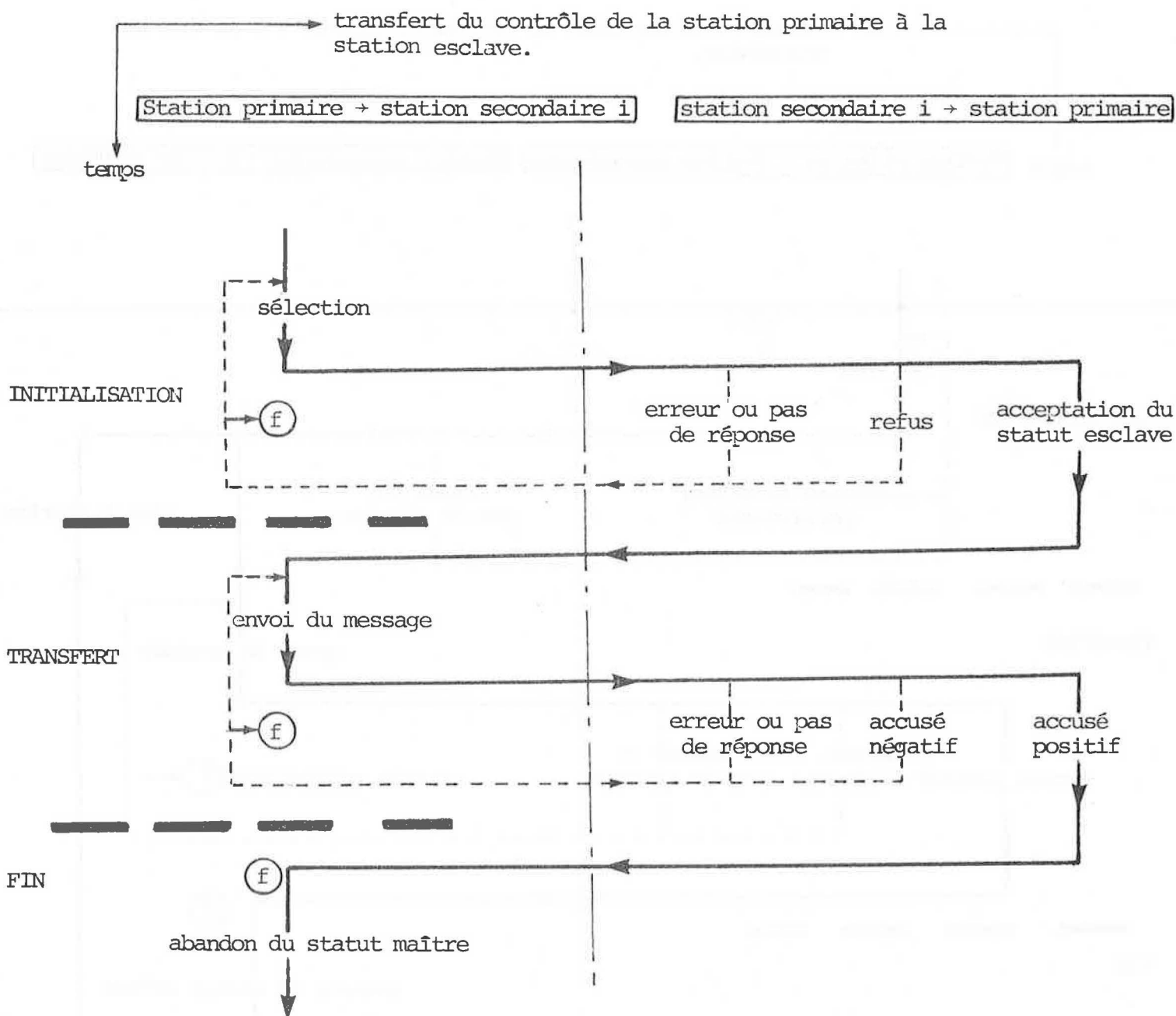


Figure 5 - Diagramme de la "sélection" dans une procédure multipoint

3.1.3. - LANGAGE DE COMMUNICATION

Comme nous l'avons vu deux stations distantes communiquent par échanges de blocs d'informations, constitués d'une suite de caractères. Ces caractères appartiennent à un "alphabet" et peuvent être regroupés en divers "mots" du "langage" utilisé par la procédure de transmission. Cet alphabet peut, par exemple, être constitué de l'ensemble des caractères représentables en code ASCII (128)

ou en code EBCDIC (256). Dans cet alphabet, une série de caractères est réservée au contrôle des échanges. Certaines associations de ces caractères de contrôle définissent des mots-clés du langage ("demande de statut maître", "séquence de polling", "séquence de sélection" etc...)

Les caractères de contrôle suivants ont été normalisés par l'ISO :

- SYN ("synchronous idle") : caractère de synchronisation pour les transmissions synchrones (cf. 2.1.4 et 2.1.5). Deux ou trois caractères SYN sont placés en tête de chaque bloc transmis.
- SOH ("start of heading") : indique le début d'un en-tête de message.
- STX ("start of text") : détermine la fin de l'en-tête du message et le début du texte du message.
- ETB ("end of transmission block") : marque la fin de l'un des blocs d'information du texte à transmettre.
- EXT ("end of text") : marque la fin de transmission d'un texte.
- EOI ("end of transmission") : émis par la station maîtresse pour mettre fin à la communication. Les statuts maître et esclave sont abandonnés. Egalement utilisé en tête de toute séquence de "polling" ou de "sélection" émise par la station primaire d'une ligne multipoint.
- ENQ ("Enquiry") : termine la séquence de supervision émise pour l'initialisation des transferts (demande de statut maître ou "polling"/sélection) et appelle une réponse de la station vis à vis.
- ACK ("Acknowledgement") : émis par une station esclave pour indiquer :
- l'acceptation du statut esclave, en phase d'initialisation
 - la bonne réception d'un texte, en phase de transfert.
- NAK ("negative acknowledgement") : émis par une station esclave pour indiquer :
- le refus du statut esclave, en phase d'initialisation.
 - la mauvaise réception d'un texte, en phase de transfert.
- DLE ("data link escape") : sert à modifier la signification du caractère qui le suit. Il est utilisé dans le mode de transmission transparent pour garantir sur le texte binaire les mêmes fonctions de procédure que sur le texte imprimable; (cf. 2.1.7).
Egalement utilisé pour les demandes de suspension de transmission émises par la station esclave.

Les blocs échangés entre deux stations peuvent, soit contenir uniquement des caractères de contrôle, soit envelopper un message utile (fig. 6).

S	S	S	S		S		E	B*
Y	Y	Y	O	en-tête	T	texte	T	C
N	N	N	H		X		X	C

* cf. 3.1.4.

Figure 6 - bloc enveloppant un message utile

3.1.4. - CONTROLE DES ERREURS DE TRANSMISSION :

Dans tout système de communication réel, les erreurs sont inévitables car la fiabilité du médium de transmission n'est pas absolue. Il est nécessaire d'utiliser un mécanisme permettant de détecter les erreurs éventuelles. En téléinformatique, on préfère en effet retransmettre sur détection d'une erreur plutôt que corriger à la réception les données erronées. Le taux d'erreur (10^{-5} à 10^{-6}) sur les lignes utilisées ne justifierait pas le coût des mécanismes complexes mis en jeu par la correction.

La méthode utilisée consiste à faire appliquer par l'émetteur une certaine fonction aux données à transmettre et à leur ajouter l'information redondante qui en résulte. De son côté, le récepteur applique la même fonction aux données reçues et compare le résultat obtenu avec l'information redondante reçue. S'il n'y a pas concordance, le récepteur acquitte négativement (ou n'acquiesce pas) le message reçu, ce qui provoquera sa re-transmission.

Ces contrôles de redondance sont différents selon le code de transmission utilisé. On distingue 2 types de contrôle :

- un contrôle local au niveau de chaque caractère : le VRC ("vertical Redundancy Check"). Cette redondance verticale, ou parité, consiste à ajouter un bit à chaque caractère, tel que le nombre de bits égaux à 1 dans le caractère soit impair (ou pair). Toutefois, ce type de contrôle s'avère insuffisant car les erreurs ont tendance à se produire par rafales de bits successifs. Il est alors inutile de chercher un contrôle élaboré au niveau du caractère.
- un contrôle global au niveau du bloc : le BCC ("Block Check Character"). Le BCC peut être un contrôle par redondance longitudinale, ou LRC ("Longitudinal Redundancy Check"), qui consiste à évaluer la fonction "ou exclusif" des caractères composant le message. Le caractère résultant est ajouté en fin de bloc.

Ce BCC peut être également un contrôle par redondance cyclique, ou CRC ("Cyclic Redundancy Check"). Chaque bloc à émettre constitué de n bits est associé à un polynôme $B(x)$ de degré $n-1$ dans le corps des entiers modulo 2. Le bloc émis $B'(x)$ doit être divisible par un polynôme générateur donné $G(x)$ de degré k . Pour constituer $B'(x)$ on ajoute au bloc initial $B(x)$ le reste de sa division par $G(x)$. En effet si $B(x) = Q(x) \cdot G(x) + R(x)$ alors $B'(x) = x^k B(x) + R(x)$ est divisible par $G(x)$. Le polynôme générateur étant généralement de degré 16, le reste est représenté sur 2 octets : les caractères CRC. La procédure de transmission détermine les caractères du bloc qui entrent dans l'élaboration du BCC.

On utilise généralement en code ASCII, un contrôle par VRC associé à un contrôle par LRC, et en code EBCDIC, un contrôle par CRC.

Remarquons que ces procédés ne font que diminuer la probabilité d'erreur. En effet, les informations redondantes sont transportées par le même médium de transmission que les données qu'elles protègent.

3.2. - RECONNAISSANCE D'UNE PROCEDURE DE TRANSMISSION PAR UN AUTOMATE :

3.2.1. - RECHERCHE D'UN MODE DE REPRESENTATION GRAPHIQUE :

Une procédure de transmission est généralement trop complexe pour pouvoir être décrite de façon satisfaisante et sans ambiguïté par la simple narration des divers cas possibles et des actions à prendre pour chacun de ces cas. (Toutefois, la plupart des constructeurs s'en tiennent encore à ce type de présentation). Un algorithme est encore moins adapté à une telle description. Par contre, une représentation graphique est plus précise et bien plus compréhensible. Ainsi les diagrammes de temps utilisés en 3.1.2 constituent une méthode intuitive qui a l'avantage de fournir un schéma d'ensemble des relations entre stations, et de montrer la direction de chaque flot d'informations. Cependant, ce type de graphe perd rapidement de sa clarté et reste imprécis lorsque la procédure se complique.

D. BJORNER [PRO-1] a le premier suggéré une méthode plus adaptée qui s'impose naturellement si nous nous interrogeons sur le comportement des appareils qui régissent la procédure de transmission. En effet, le récepteur et

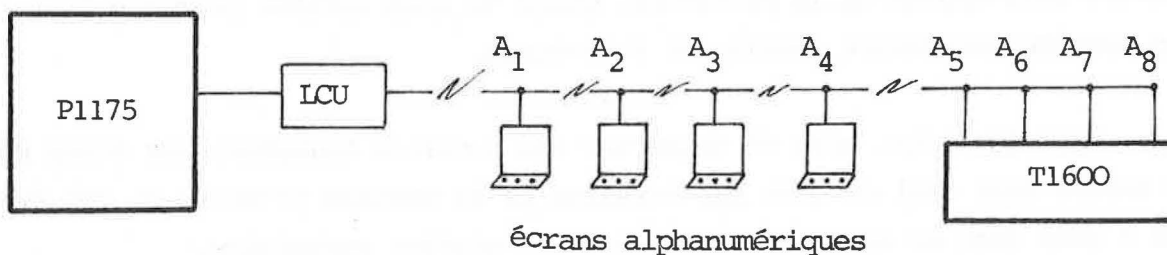
l'émetteur d'une station sont essentiellement des machines déterministes, c'est-à-dire qu'à tout instant elles sont, du point de vue de la procédure dans un état donné et, dans cet état, elles réagissent toujours de la même manière à un évènement donné. Ces machines se comportent donc comme des automates d'état fini. La méthode la plus naturelle, et sans aucun doute la plus précise, consiste à décrire le comportement de ces automates par un graphe de transition entre un nombre fini d'états. Le graphe représente ainsi la syntaxe ou la grammaire du langage de communication (cf.3.1.3.) et définit l'ensemble des séquences de caractères valides, ou l'ensemble des "productions" que reconnaît l'automate. L'aspect sémantique peut également figurer sur ce graphe en y ajoutant les actions à prendre pour chaque séquence de caractères valide ou invalide reçue.

Enfin, un tel graphe est utilisable pour toute implémentation de la procédure, que ce soit au niveau matériel, au niveau de micro-programmes ou au niveau logiciel. D'autre part, comme nous allons le montrer sur des exemples que nous avons réalisés sur le T1600, il est possible d'établir ce graphe à plusieurs niveaux d'abstraction successifs, et de s'en inspirer pour une implémentation "structurée" [DIJKSTRA] respectant cette hiérarchie (états puis sous-états de ces états).

3.2.2. - GRAPHE DE LA PROCEDURE MULTIPPOINT ECMA-PHILIPS.

Cette procédure est utilisée par les ordinateurs PHILIPS de la série P1000 pour gérer, par l'intermédiaire d'une unité de contrôle câblée (station primaire) un certain nombre d'adresses (station secondaires) sur une ligne multipoint. Ces adresses peuvent être attachées à divers organes physiques : télétype, écran alphanumérique, terminal lourd etc... ; mais ces organes physiques proviennent nécessairement du même constructeur car la procédure possède la particularité d'être entièrement gérée au niveau matériel, tant du côté de la station primaire que du côté des stations secondaires.

Désirant relier le frontal T1600 à l'ordinateur P1175 par l'intermédiaire de cette ligne multipoint (fig. 7), nous avons été amenés à lui faire simuler le comportement de plusieurs stations secondaires. Pour cela, nous avons dû reconstituer le graphe de la procédure afin d'en déduire le logiciel de transmission à implémenter sur le frontal. Cette reconstitution n'a pas été facilitée par la description, touffue et imprécise, donnée par le constructeur et a nécessité des recours fréquents à un programme d'espionnage de l'activité sur la ligne de transmission.



LCU ("Line Control Unit")

Figure 7

Tirant profit de notre expérience, nous allons montrer sur cet exemple une façon efficace de décrire une procédure selon une technique descendante en construisant les graphes correspondants à chaque niveau d'abstraction rencontré.

3.2.2.1. - Niveau primaire-secondaire.

Soit \textcircled{P} l'état du système de communication lorsque la station primaire émet une séquence de caractères \underline{m} vers une ou plusieurs stations secondaires. La fin du transfert de \underline{m} fait passer le système à un état \textcircled{S} dans lequel l'une des stations secondaires répond par une séquence de caractères \underline{r} à la station primaire. La réception de \underline{r} par la station primaire provoque le retour à l'état initial \textcircled{P} . Certaines séquences \underline{m}' émises par la station primaire peuvent ne pas appeler de réponse des stations secondaires : le système reste alors dans l'état \textcircled{P} .

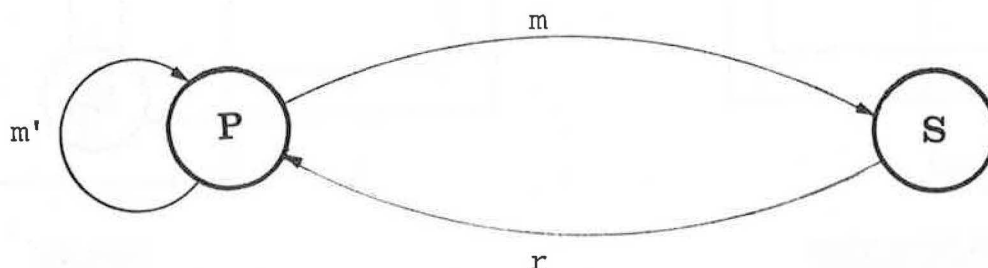


Figure 8

Le graphe (fig. 8) illustre l'exclusion mutuelle de ces 2 états due à la nature half-duplex de la procédure. Quand le sous système primaire émet le sous système secondaire reçoit, et vice-versa.

Dans la suite, afin de conserver une notation homogène, les états de type P seront ceux dont on sort par émission de la station primaire et les états de type S ceux dont on sort par émission d'une station secondaire.

3.2.2.2. - Niveau grandes fonctions

Les états (P) et (S) peuvent être éclatés en sous-états (fig. 9). En effet, les transactions entre la station primaire et les stations secondaires peuvent être groupées en 3 classes : "polling", interrogation/sélection d'une station secondaire et "flash write" ou émission inconditionnelle vers une ou plusieurs stations secondaires.

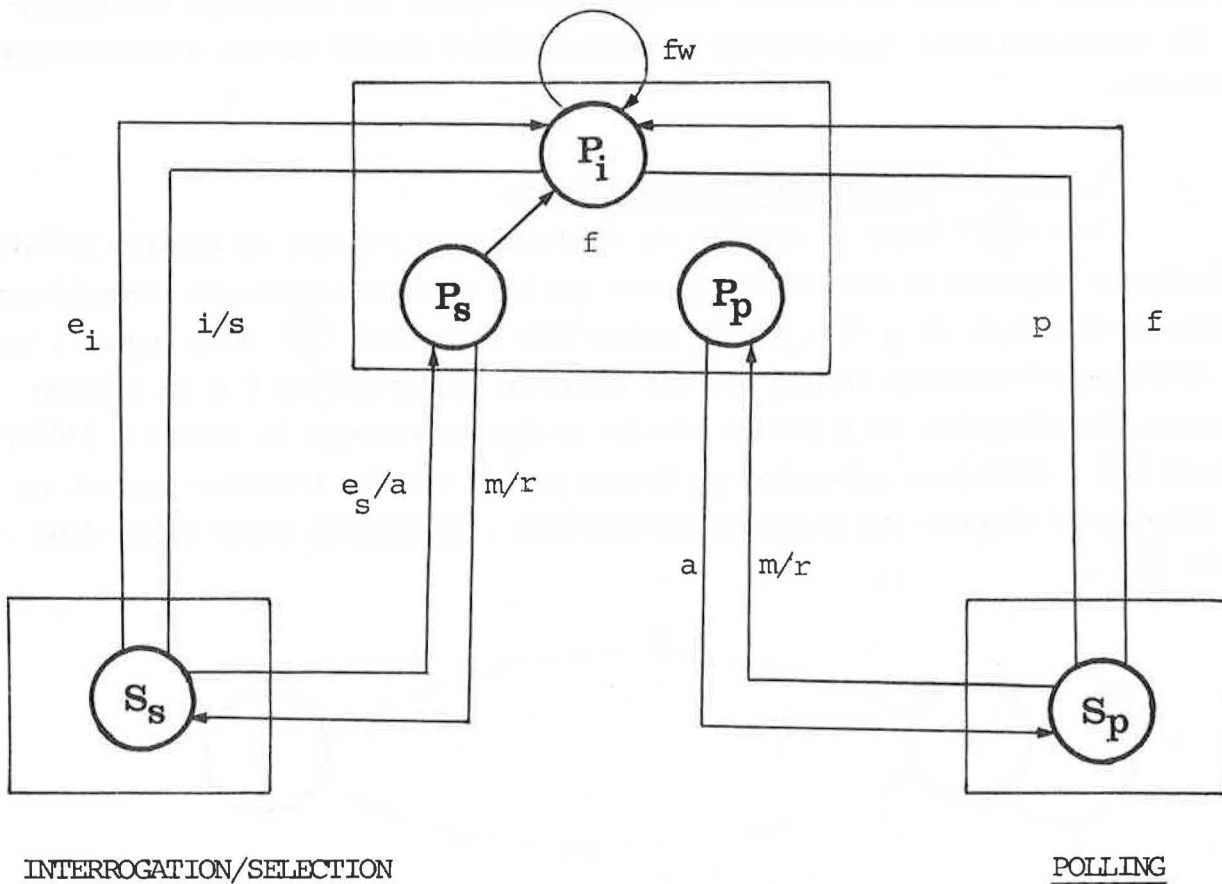


Figure 9

NB. Les séquences m, r n'ont pas la même signification sur les figures 8 et 9.

- 1 - La station primaire peut interroger une station secondaire par l'envoi d'une séquence \underline{i} qui fait passer le système d'un état (P_j) à l'état (S_s) . La station secondaire adressée donne son état ("occupée", "prête à recevoir", "prête à émettre" etc...) en renvoyant une séquence $\underline{e_i}$ qui fait retourner le système à l'état initial (P_j) .

La station primaire peut également inviter à recevoir une station secondaire par l'envoi d'une séquence de sélection \underline{s} , identique à la séquence \underline{i} . De la même manière, la station secondaire adressée renvoie son état dans une séquence $\underline{e_s}$, identique à la séquence $\underline{e_i}$ mais qui conduit cette fois à l'état (P_s) . Si l'état de la station secondaire le permet la station primaire lui adresse un message \underline{m} qui fait passer le système à l'état (S_s) , sinon elle émet une fin de communication \underline{f} qui reconduit le système à l'état initial (P_j) . L'accusé de réception \underline{a} du message, renvoyé par la station secondaire, fait repasser le système de l'état (S_s) à l'état (P_s) d'où il retourne à l'état initial (P_j) par une fin de communication \underline{f} émise par la station primaire. En cas de mauvaise réception de l'accusé-réception, la station primaire peut émettre une demande de répétition \underline{r} .

- 2 - Lorsque la station primaire veut inviter une station secondaire à émettre, elle transmet une séquence de "polling" \underline{p} qui fait passer le système de l'état (P_j) à l'état (S_p) . La station secondaire adressée peut alors émettre une fin de communication \underline{f} , provoquant le retour à l'état (P_j) , ou émettre un message \underline{m} qui conduit le système à l'état (P_p) . L'acquiescement \underline{a} du message par la station primaire fait repasser le système à l'état (S_p) d'où il retourne à l'état (P_j) par une fin de communication \underline{f} émise par la station secondaire. En cas de mauvaise réception de l'acquiescement, la station secondaire peut émettre une demande de répétition \underline{r} .

- 3 - Le "flash write" est l'émission inconditionnelle d'un message \underline{fw} de la station primaire vers une ou plusieurs secondaires. Il n'appelle pas de réponse et ne conduit pas à un changement d'état.

Pour la clarté de l'exposé, nous ne considérerons dans la suite ni le "flash-write", ni l'émission d'un message vers plusieurs stations secondaires sélectionnées simultanément.

3.2.2.3. - Niveau bloc

En descendant à un niveau plus fin, les sous-états précédents peuvent eux-mêmes être éclatés en nouveaux sous-états plus élémentaires, comme le montre la figure 10.

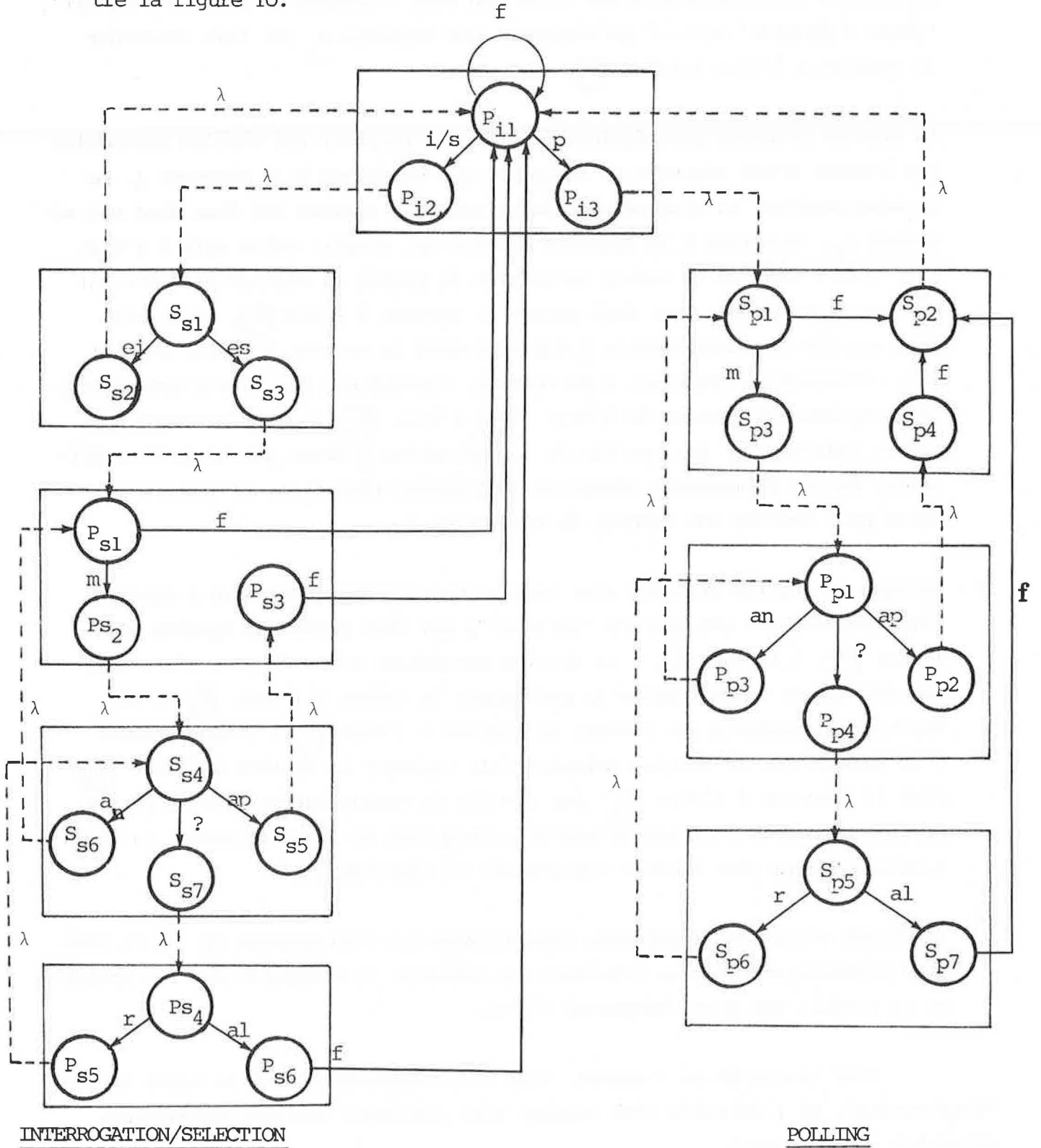


Figure 10

A ce stade, nous introduisons un nouveau type de transition : la transition λ qui symbolise un retournement du sens de transmission entre la station primaire et la station secondaire adressée. Nous passons ainsi d'un état global du système (fig. 9) aux états des stations (fig. 10).

Les différentes transitions ont les significations et structures suivantes :

- i/s : séquences d'interrogation/sélection

A	A	A	A	A	S	S	E	A	(A)*	E	S	S
Z	O	O	O	O	Y	Y	O	D	(D)	N	Y	Y.....
C	C	C	C	C	N	N	T	R _{S1}	(R _{S2})	Q	N	N



synchronisation initiale des modems au démarrage d'une session.

ADR_S est l'adresse de la station secondaire en format interrogation/sélection.

ADRS₂ n'est présent que si l'espace des adresses est à 2 niveaux hiérarchiques.

- p : séquence de "polling"

A	A	A	A	A	S	S	E	A	(A)*	E	S	S
Z	O	O	O	O	Y	Y	O	D	(D)	N	Y	Y.....
C	C	C	C	C	N	N	T	R _{P1}	(R _{P2})	Q	N	N



synchronisation initiale

ADR_P est l'adresse de la station secondaire en format "polling".

- ei/es : réponses à une interrogation/sélection

S	S	I	(I)	S	A	N	S	S
Y	Y	D	(D)	I	C	A	Y	Y.....
N	N	T ₁	(T ₂)	C	K	K	N	N

IDT identifie la station secondaire qui répond.

SIC donne l'état de la station secondaire.

ACK (NAK) indique la disponibilité (l'indisponibilité) à recevoir.

* optionnel

- m : envoi d'un message

.....	S	S	S	E	B	S	S
	Y	Y	TT	C	Y	Y
	N	N	X	X	C	N	N

Le texte ne peut pas contenir de caractères de procédure (non transparent).

- f : fin de communication

.....	S	S	E	S	S
	Y	Y	O	Y	Y
	N	N	T	N	N

- ap/an : acquittements positif/négatif d'un message

.....	S	S	A	N	S	S
	Y	Y	C	A	Y	Y
	N	N	K	K	N	N

- ? : réponse invalide ou non comprise/temps d'attente dépassé

- r : demande de répétition de l'acquiescement

La procédure correspondante qui comprend une resynchronisation est complexe et apparaîtra au 4ème niveau.

- al : procédure d'alarme :

Elle est engagée après un certain nombre de séquences invalides et/ou d'essais de reprises sans succès. Elle se termine toujours par une fin de communication et un retour à l'état initial. Nous la faisons figurer seulement en deux points du graphe afin de ne pas le surcharger.

Remarque :

1 - Le support de transmission est full-duplex mais la procédure half-duplex.

La station primaire émet continuellement des caractères SYN lorsqu'elle n'a rien d'autre à émettre. Par contre, au repos, une station secondaire n'émet

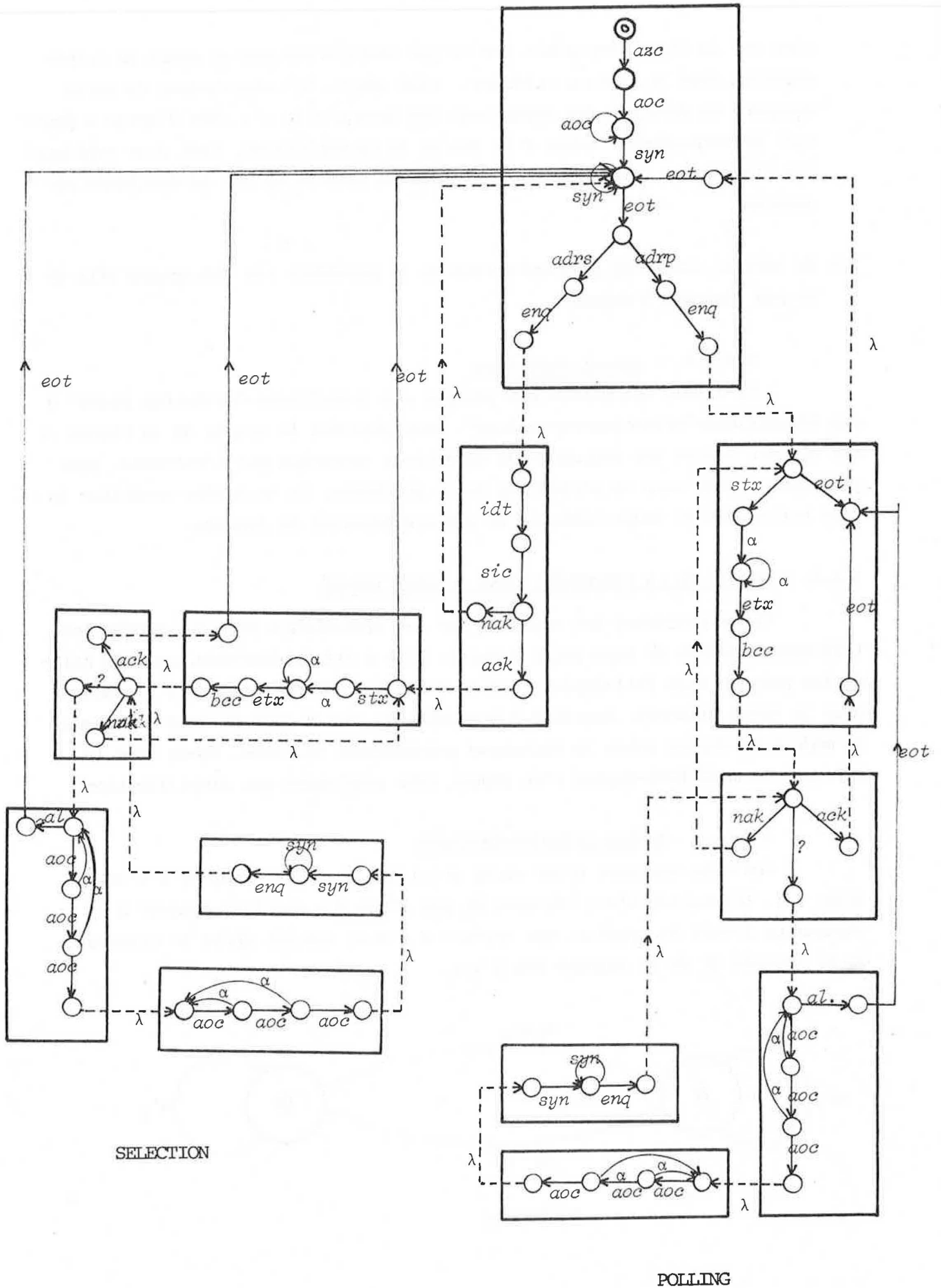


Figure 11

rien sur la ligne. Cependant dès qu'une station secondaire entre en communication avec la station primaire, elle adopte le comportement de cette dernière en émettant des caractères SYN lorsqu'elle n'a rien d'autre à émettre. Ce comportement cesse à la fin de la communication. Ceci pour expliquer les tiretés que nous avons fait figurer en tête et en fin de certaines séquences.

2 - Un certain nombre de particularités de la procédure ont été omises afin de ne pas alourdir l'exposé.

3.2.2.4. - niveau caractère

Ce niveau est obtenu par passage des transitions-"orientées blocs" à des transitions-"orientées-caractères". Nous obtenons le graphe de la figure 11 qui définit toutes les séquences de caractères reconnues par l'automate. Dans ce graphe, comme dans la définition de la procédure, le caractère constitue la plus petite entité indivisible ou le symbole terminal du langage.

3.2.3. - GRAPHE DE LA PROCEDURE POINT A POINT TMM-UC

Cette procédure est utilisée par les ordinateurs CII pour toutes les télé-transmissions de type point à point. Elle a été implémentée, en mode half-duplex puis en mode full-duplex, sur le frontal T1600 pour gérer la liaison avec le noeud MITRA-15, localisé à Grenoble, du réseau CIGALE. Nous décrivons le mode half-duplex selon la technique précédemment utilisée. Comme nous le verrons, le mode full-duplex s'en déduit très simplement par simplification.

3.2.3.1. - niveau primaire-primaire

Les deux stations n'ont aucun privilège l'une par rapport à l'autre (fig. 12). L'émission d'une séquence m_i par l'une des stations, conduit à un changement d'état du système, qui revient à l'état initial après le transfert de la réponse r_i de la station vis à vis.

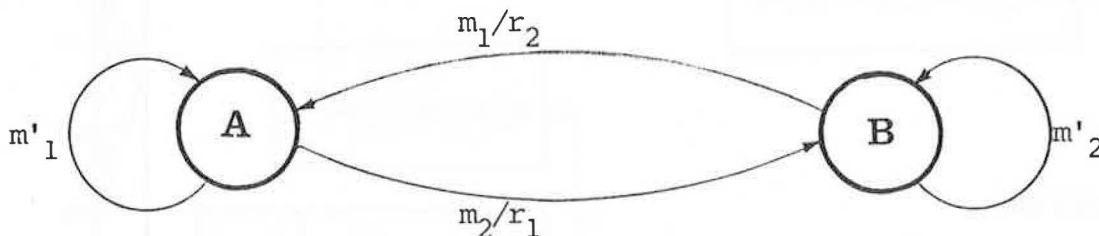


Figure 12

Certaines séquences m'i peuvent ne pas appeler de réponse : le système ne change pas d'état.

3.2.3.2. - niveau maître/esclave

Lorsque la station A(B) est au repos et désire émettre, elle envoie une "demande de statut maître" DSM à sa vis à vis. La station B(A) renvoie une réponse REP par laquelle :

- soit elle refuse le statut esclave, auquel cas la station A(B) met fin à la communication par l'envoi d'une séquence FIN;
- soit elle accepte le statut esclave, auquel cas la station maîtresse A(B) envoie des messages MES à la station esclave B(A) dont elle reçoit des accusés-réception ACC, jusqu'à une fin de communication émise par la station maîtresse.

La figure 13 illustre la symétrie des relations entre les 2 stations. Le conflit résultant de 2 "demandes de statut maître" opposées et simultanées est résolu selon le principe de la contention (cf. 3.1.2.1).

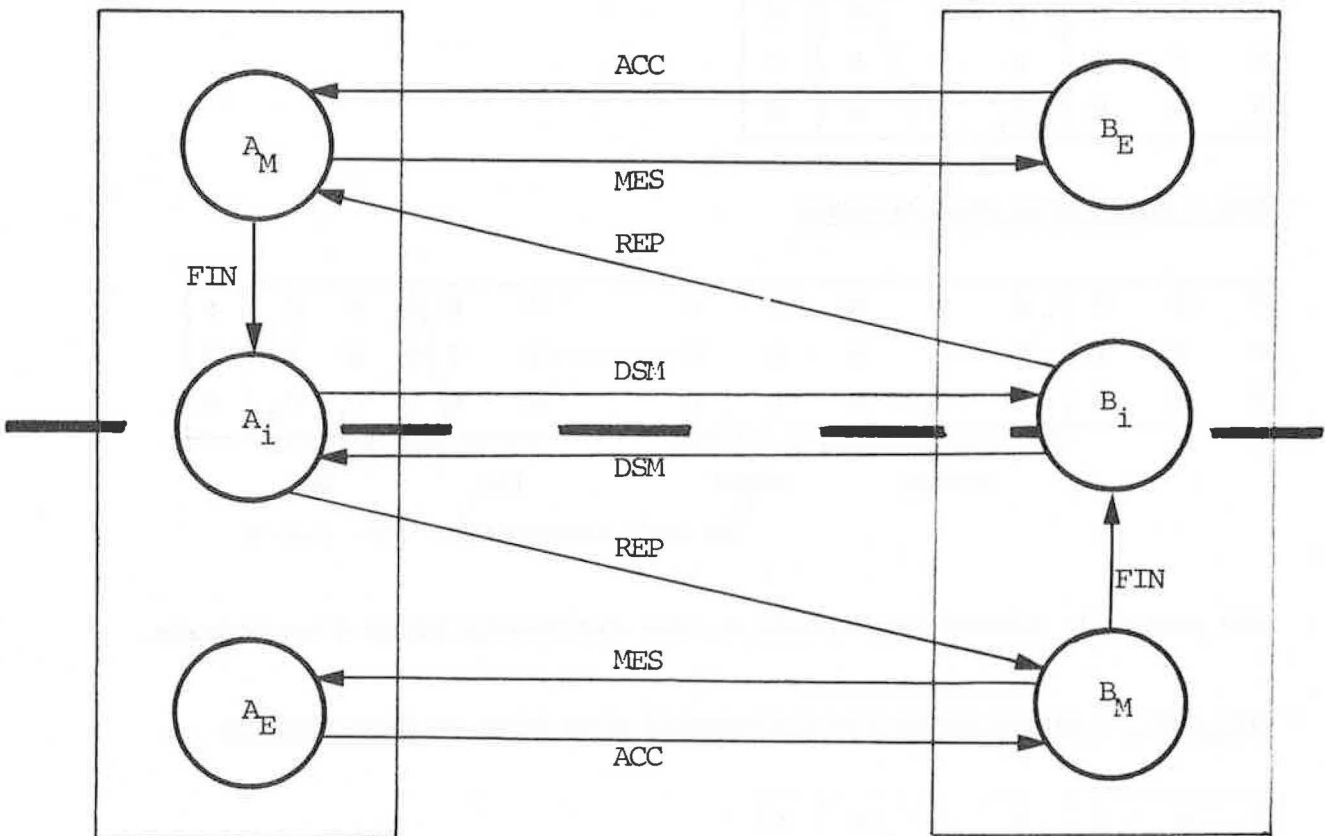


Figure 13

3.2.3.3. - niveau bloc

Comme précédemment, nous introduisons à ce niveau (fig. 14) la transition λ et nous considérons l'ensemble des différentes "transitions-blocs" possibles.

- DSM : demande de statut-maître

S	S	S	S	S	E	S
Y	Y	Y	E	E	N	Y
N	N	N	L ₁	L ₂	Q	N

SEL₁ et SEL₂ sont 2 caractères d'adresse utilisés pour rester compatible avec la procédure TMM multipoint. Dans notre cas SEL₁ et SEL₂ ont une configuration identique choisie à l'avance.

- REP_p/REP_n : acceptation/refus du statut esclave

S	S	S	S	A	N	S
Y	Y	Y	E	C	A	Y
N	N	N	L ₁	K	K	N

- MES : envoi d'un bloc/message

S	S	S	S	S	N	D	S	D	E	E	C	C	S	
Y	Y	Y	O	E	O	L	T	---	L	T	T	R	R	Y
N	N	N	H	L ₁	B	E	X	E	B	X	C ₁	C ₂	N	

← entête
← début
← fin
← BCC

↓ du mode transparent (cf. 2.1.7)

NOB permet le numérotage, modulo 8, des différents blocs d'un message.

- ACC_p/ACC_n : acquittement positif/négatif d'un bloc ou d'un message

S	S	S	N	A	N	S
Y	Y	Y	O	C	A	Y
N	N	N	B	K	K	N

NOB donne le numéro du dernier bloc/message correctement reçu.

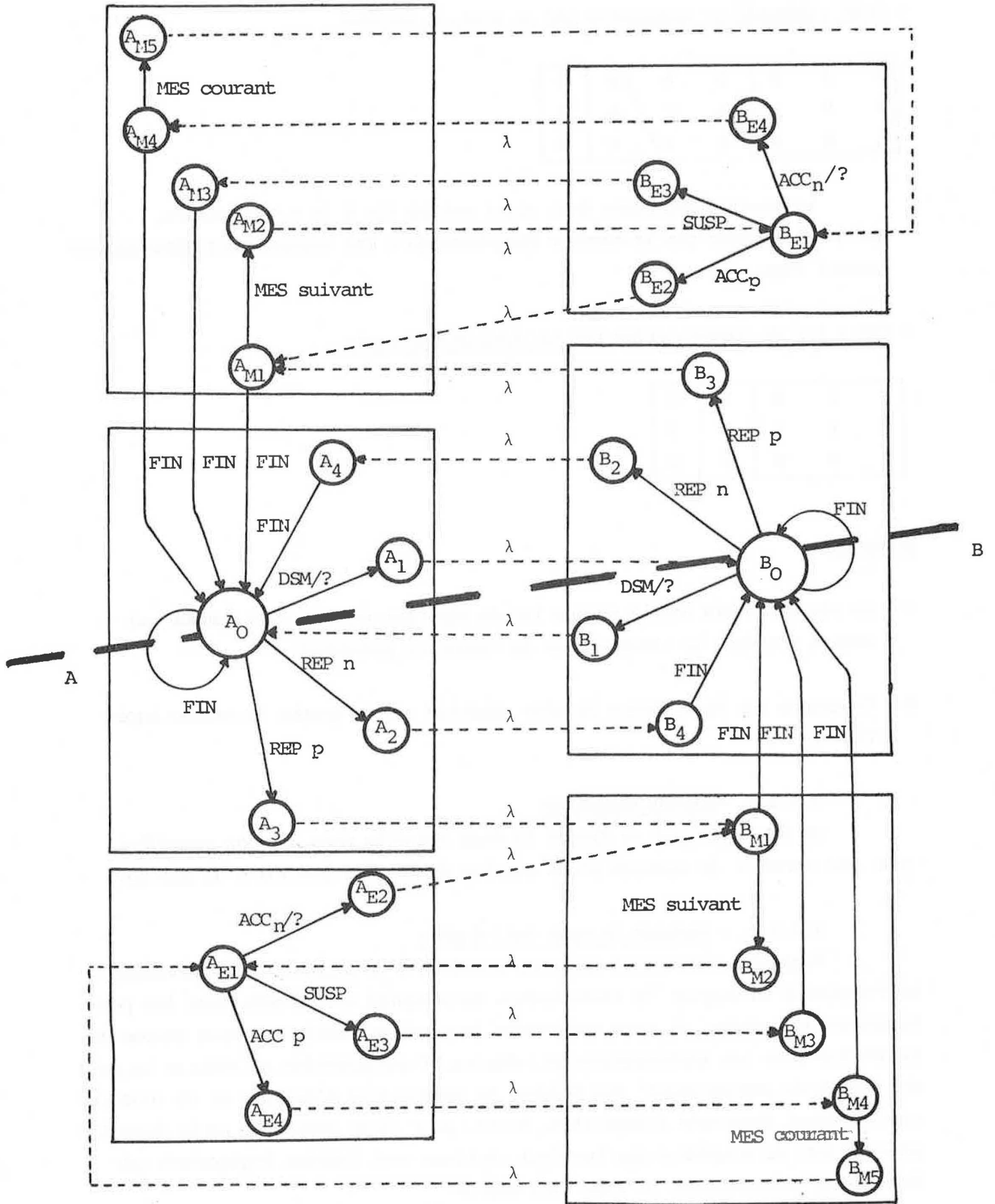


Figure 14

- SUSP : demande de suspension par la station esclave

S	S	S	D	A	/	N	S
Y	Y	Y	L	C	/	A	Y
N	N	N	E	K	/	K	N

La station maîtresse doit alors mettre fin à la communication. ACK (NAK) indique que le dernier bloc/message a été correctement (non correctement) reçu.

- FIN : fin de communication par la station maîtresse

S	S	S	E	S
Y	Y	Y	O	Y
N	N	N	T	N

Remarque :

- 1 - Le caractère SYN émis à chaque fin de bloc est un caractère tampon qui sert à protéger la transmission du caractère précédent.
- 2 - Ce graphe met en évidence la même symétrie que le graphe de niveau supérieur.

3.2.3.4. - niveau caractère

Il est donné par la figure 15 dans laquelle nous n'avons considéré que les envois de message d'une station A vers une station B (A maître).

3.2.3.5. - Passage au mode full-duplex

Supposons maintenant que nous passions d'un support de transmission half-duplex à un support de transmission full-duplex afin d'améliorer les performances du système. Si nous conservons la procédure telle que nous venons de la décrire pour une exploitation half-duplex, l'amélioration se réduira au gain des "temps de retournement" des modems, la possibilité d'émettre et de recevoir simultanément demeurant inexploitée. Aussi, pour tirer partie de cette dernière possibilité, on considère que les deux stations sont reliées logiquement par deux voies indépendantes de type half duplex.

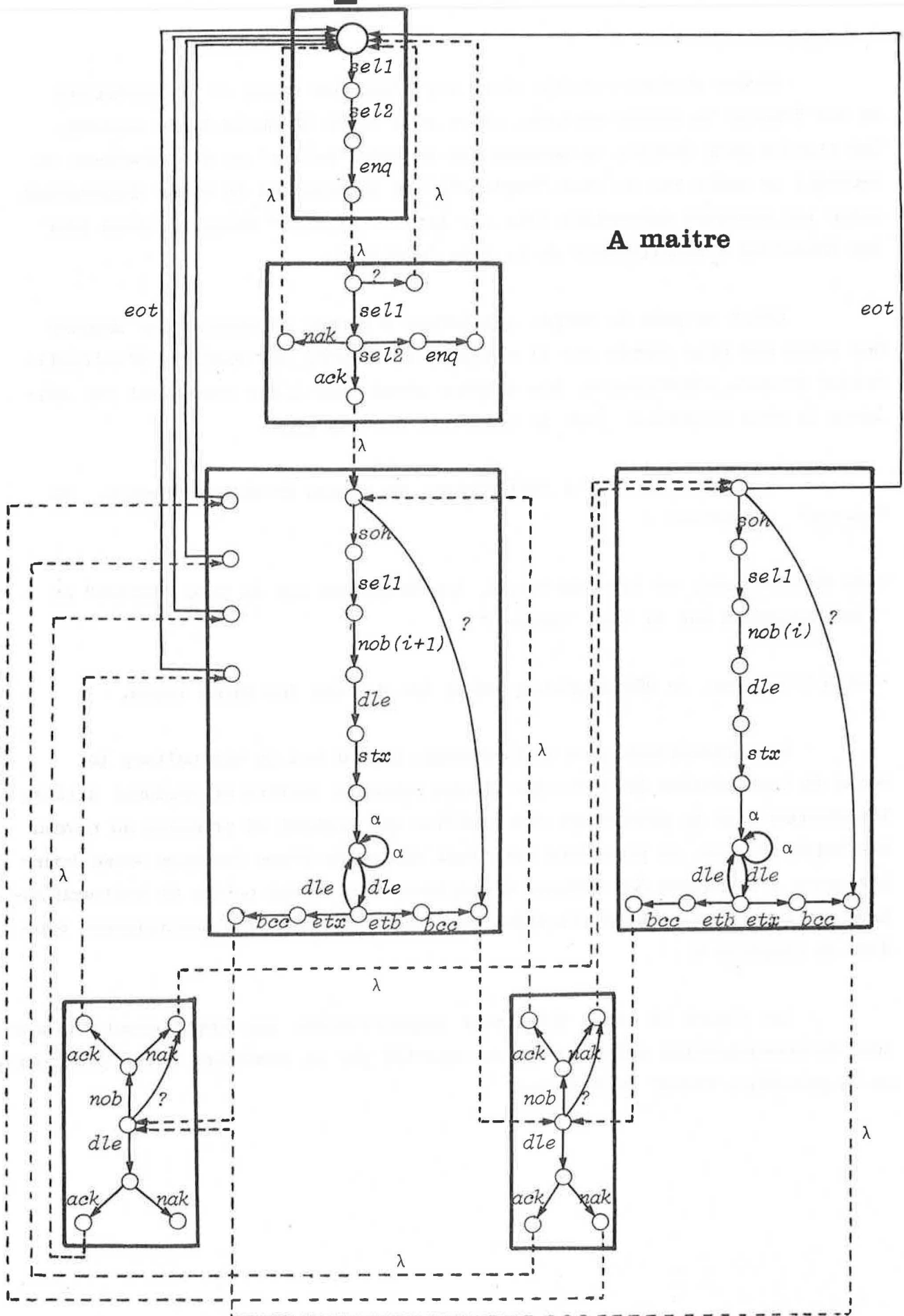


Figure 15

Chaque station possède alors sur l'une des voies le statut-maître et sur l'autre le statut esclave, et ce pour toute la durée d'une session. Une station peut émettre un message sur sa voie "maître" et simultanément en recevoir un autre sur sa voie "esclave", les intervalles de temps disponibles entre les messages successifs émis sur la voie "maître" étant utilisés pour les émissions d'acquiescement de la voie "esclave".

Outre ce gain de temps, l'automate à mettre en oeuvre pour chacune des voies est plus simple car il n'a plus à résoudre les conflits d'initialisation évoqués précédemment. Les graphes ainsi simplifiés conservent par ailleurs la même structure pour le transfert des messages.

Il reste cependant à implémenter, au niveau de chaque station, un "verrou" permettant :

- de synchroniser, en les imbriquant, les émissions sur la voie "maître" et les émissions sur la voie "esclave".
- et à l'inverse, de démultiplexer entre les 2 voies les blocs reçus.

Ce verrouillage , ou multiplexage, a pour but de virtualiser les voies de transmission de sorte que chaque automate (maître et esclave) utilise les siennes sans se préoccuper des conflits qui peuvent se produire au niveau des voies réelles. La procédure est ainsi déchargée d'une fonction somme toute inhérente à la nature des organes d'entrée-sortie (cette notion de hiérarchisation des fonctions et simplification des interfaces est plus spécialement étudiée au Chapitre V).

Les figure 16 et 17 illustrent respectivement les comportements du système de communication selon qu'il est exploité par la procédure TMM-UC half-duplex ou la procédure TMM-UC full-duplex.

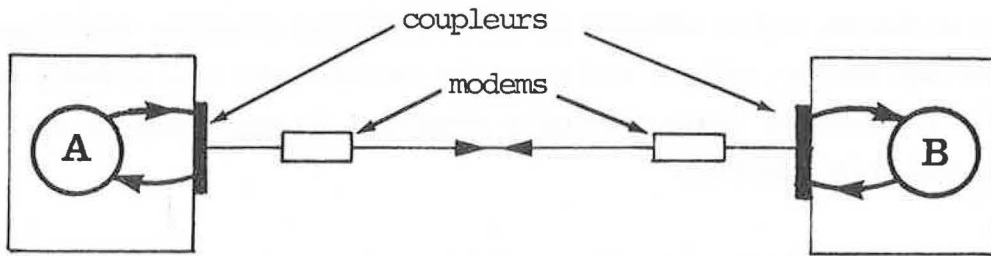


Figure 16 - Système half-duplex

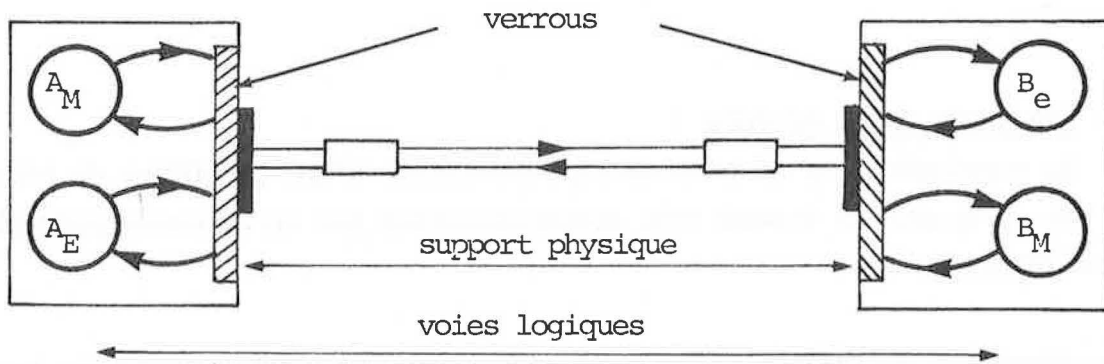


Figure 17 - Système full-duplex

Une seconde solution plus simple à réaliser, mais peu usitée parce que plus coûteuse, consisterait à relier physiquement les deux stations par deux supports de transmission half-duplex entièrement indépendants (fig. 18). Le problème du verrou ne se pose plus mais, en contrepartie, cette solution nécessite deux fois plus d'équipements (avec toutefois cette nuance qu'un matériel half-duplex - ligne, modem ou coupleur - est généralement moins onéreux que le matériel full-duplex correspondant).

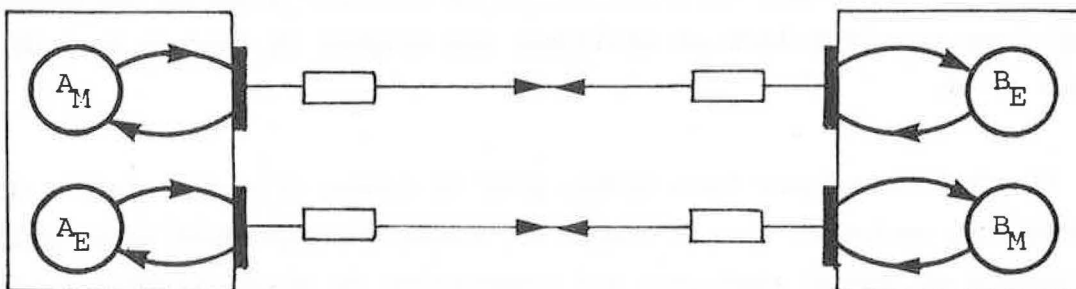


Figure 18 - Système full-duplex

Par ailleurs, cette seconde solution présente d'autres avantages :

- la rapidité est accrue, malgré les temps de retournement des modems, car en émission il n'y a plus de temps d'attente résultant du multiplexage de 2 voies logiques sur une voie physique.
- la fiabilité est améliorée. En effet, si l'une des voies physiques est défaillante, il est toujours possible, après un changement d'automate au niveau de chaque station, de se replacer dans le contexte de la figure 16 sur la voie restante.

3.2.4. - AVANTAGES DE LA METHODE :

En conclusion, la méthode de représentation d'une procédure de transmission par un graphe de transitions reconnaissables par un automate présente plusieurs avantages :

- 1 - Elle donne une définition précise et sans ambiguïté de la procédure. Le graphe permet d'envisager tous les cas de figures possibles parce qu'il comporte un nombre fini de noeuds (ou états de l'automate) et que chaque noeud autorise un nombre fini de transitions (ou évènements).
- 2 - Le graphe donne le comportement du système de communication tout entier. Si nous partons d'un même graphe pour implémenter l'émetteur et le récepteur de chaque station, nous sommes assurés de la concordance des comportements des différentes stations.
- 3 - Il est possible de choisir le niveau de décomposition du graphe en fonction du niveau auquel se situe l'implémentation. Par exemple, si l'on dispose d'un coupleur fonctionnant en mode bloqué et capable de reconnaître les caractères de contrôle de la procédure, on implémentera un automate logiciel reconnaissant des transitions au niveau bloc en utilisant des graphes du type de ceux des figures 10 et 14.

Il est d'autre part très utile, pour un niveau d'implémentation donné de s'appuyer non seulement sur le graphe de niveau correspondant mais également sur les graphes de niveau supérieur qui permettront de mieux structurer l'implémentation. Ainsi sur le frontal T1600, qui est équipé de coupleurs fonctionnant

en mode caractère, nous avons implémenté des automates logiciels au niveau caractère. Mais la structure du logiciel est telle que les états de chacun de ces automates apparaissent comme des sous états d'un automate plus global opérant au niveau du bloc. De cette manière, le remplacement des coupleurs orientés-caractères par des coupleurs orientés-blocs devient très facile puisque le nouveau logiciel se déduit par simplification naturelle de l'ancien. Il est ainsi possible de suivre aisément l'évolution d'un matériel prenant en charge des fonctions jusqu'alors assurées par le logiciel.

4 - Enfin tout changement apporté à la procédure peut être facilement pris en compte. Il suffit en effet de le refléter sur tous les noeuds qui en sont affectés. On sait qu'il n'en est généralement pas de même pour un certain nombre de logiciels de gestion de ligne bâtis au coup par coup...

3.3. - LANGAGE DE DESCRIPTION D'UN AUTOMATE DE PROCEDURE.

Dans toute la suite de ce chapitre, nous nous cantonnons au domaine sur lequel a porté notre étude, c'est-à-dire à l'implémentation logicielle d'automates reconnaissant le caractère comme information élémentaire ou symbole terminal.

3.3.1. - REPRESENTATION DE LA SYNTAXE ET DE LA SEMANTIQUE.

Le graphe nous fournit une méthode claire de représentation de la syntaxe, ou de la grammaire du langage de procédure. Il définit en particulier toutes les séquences de caractères valides. Mais l'implémentation requiert que soient également précisées les actions sémantiques à prendre sur chaque transition, valide ou invalide.

Ces actions peuvent être indiquées :

- sur le graphe : pour toute transition t qui fait passer le système d'un état I à un état J , on note l'action $A(t)$ à prendre par le récepteur à la réception de cette transition (fig. 19).

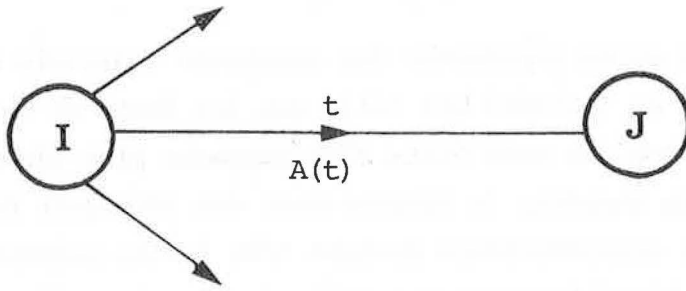


Figure 19

Cette méthode conduit rapidement à un graphe trop surchargé pour être lisible.

- dans une matrice de transitions : chaque ligne de la matrice est associée à un état de l'automate et donne, pour chacune des diverses transitions possibles, l'action à entreprendre par le récepteur et le nouvel état de l'automate (fig. 20)

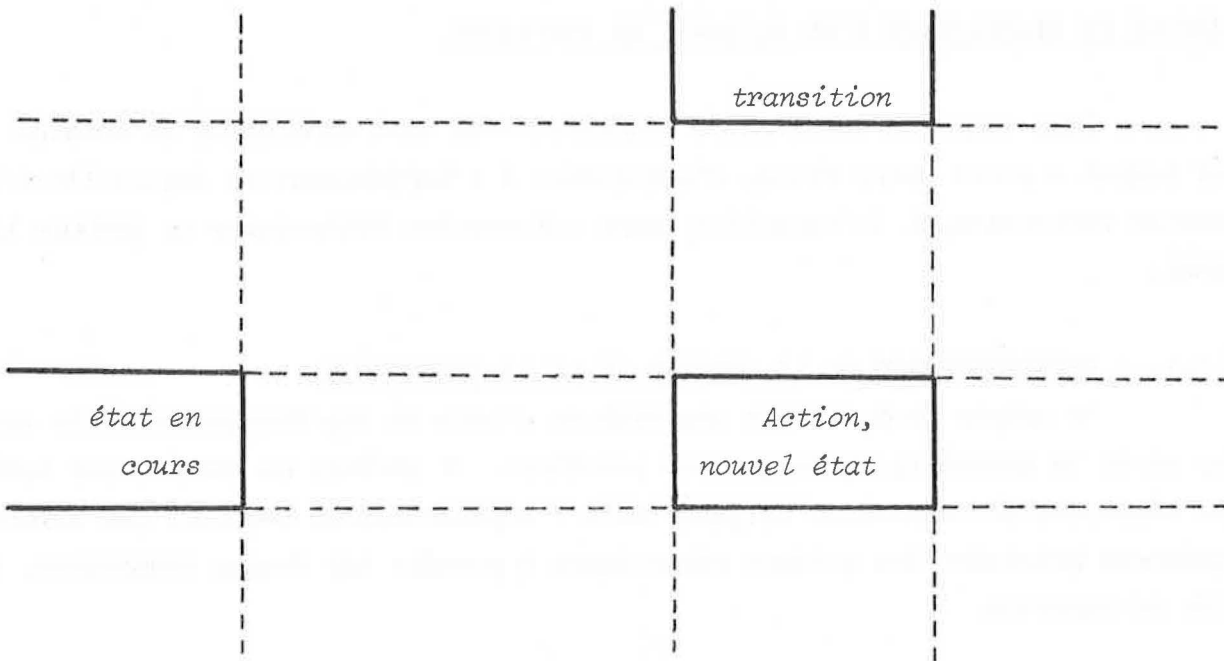


Figure 20

Les transitions non attendues (invalides) conduisent toutes à un état d'erreur qui varie avec l'état en cours.

Il est alors possible de choisir un langage existant (par exemple PL1600 sur le frontal T1600) pour l'implémentation et d'y décrire la matrice des transitions sous forme de tableaux successifs. Dans ce cas, le logiciel

comporte un moniteur qui identifie chaque transition et en déduit, à partir de l'état courant de l'automate et par consultation de ces tableaux, le nouvel état de l'automate et le module à appeler pour réaliser l'action appropriée.

Mais nous allons montrer qu'il est également possible de parvenir à une description opérationnelle de tout automate de transmission dans un langage défini à cet effet et mieux adapté à la nature d'une procédure de transmission. C'est l'objet du paragraphe suivant.

3.3.2. - DESCRIPTION DE L'AUTOMATE DANS UN LANGAGE

Nous utilisons un langage algolique permettant une description concise, lisible, cohérente et opérationnelle de toute procédure de transmission. Après avoir présenté les principaux aspects sémantiques et syntaxiques de ce langage, nous l'illustrons par la description de la procédure TMM-UC "half-duplex" (cf 3.2.3.4). Nous décrivons l'automate de l'une des stations, qui fonctionne tantôt en générateur, tantôt en analyseur avec inclusion des actions sémantiques en cours de production ou en cours d'analyse.

3.3.2.1. - Sémantique du langage

Le langage se caractérise par sa concision et certaines propriétés de base adaptées à la nature des procédures de transmission qu'il doit décrire. Nous nous proposons d'examiner ces propriétés :

- 1 - Outre des objets de type "entier" ou "booléen", le programme de description de l'automate doit pouvoir manipuler des objets de type "caractère", que nous pouvons décrire syntaxiquement de la façon suivante:

```
<caractère> ::= <caractère de contrôle> | <autre caractère> | <événement> | bcc
```

NB : dans la définition des règles de grammaire, on utilisera les règles suivantes :

- | sépare différentes possibilités.
- [] dénote une occurrence optionnelle.
- { } signifie que l'unité syntaxique renfermée peut apparaître un nombre quelconque de fois (y compris 0).

Les objets de type caractère constituent en effet l'alphabet terminal de la procédure. Cet alphabet comprend l'ensemble des symboles représentables dans le code de transmission utilisé (soit 2^8 symboles si ces symboles utilisent un code de représentation à 8 bits). Ces symboles peuvent être divisés en 2 classes :

<caractères de contrôle> et <autres caractères>. L'alphabet comprend en outre les <événements> suivants :

- . "time" : temps d'attente fini
- . "timeout" : écoulement d'un temps d'attente.
- . "défaut" : erreur matérielle détectée par le coupleur
- . "nilb" : fin d'un bloc de texte à émettre
- . "nilm" : fin d'un message (suite de blocs) à émettre

On étend le type "caractère" à ces <événements>.

Enfin bcc représente le (ou les) caractère(s) résultant du calcul de redondance effectué sur les caractères transmis et ajouté en fin de bloc.

- 2 - Une part importante des opérations effectuées par l'automate concerne ses relations avec l'extérieur (fig. 21) : tampons de l'utilisateur (contenant des messages à émettre ou destinés à recevoir des messages) et ligne de transmission.

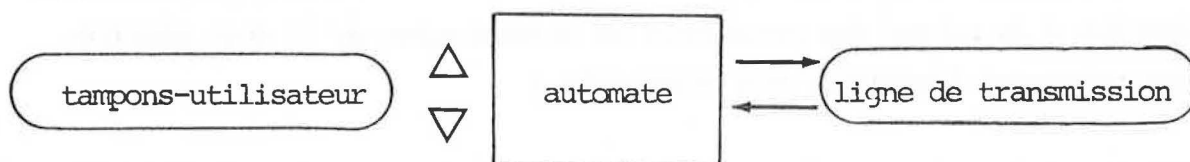


Figure 21

Pour cela le langage possède les instructions d'entrée/sortie généralisées suivantes :

- . instruction d'émission :

`<émission>::=><caractère>[↑] {,<caractère>[↑]}`

qui permet d'émettre sur la ligne les caractères cités en séquence, avec accumulation ou non, dans le bcc, de chaque caractère émis selon que le symbole ↑ est présent ou non.

→ bcc a pour résultat l'émission du bcc

→ time a pour résultat l'armement d'un réveil à la valeur donnée par "time"

. instruction de rangement :

```
<rangement> ::= ∇ <caractère>
```

qui permet de ranger, dans le tampon de l'utilisateur, un caractère courant du message utile reçu.

. instruction de réception avec test et action :

```
<action sur reception> ::= + either <caractère>[+] { , <caractère>[+] } ; <bloc>  
                       or <caractère>[+] { , <caractère>[+] } ; <bloc>  
                       { or <caractère>[+] { , <caractère>[+] } ; <bloc> }  
                       end
```

qui permet de recevoir des caractères de la ligne, de les tester et, suivant le résultat du test, d'exécuter l'un des <blocs> d'instructions. Pour un caractère donné, il peut y avoir ou non l'indication qu'il faut l'accumuler dans le bcc si on le reçoit, selon que le symbole + est présent ou non. La structure du test est équivalente à celle du case <expression> of rencontrée dans plusieurs langages. Nous pouvons l'illustrer par l'exemple suivant :

```
+ either car1+ ; bloc 1  
    or   car2, car3 ; bloc 2  
    or   x, x ; bloc 3  
    end
```

qui peut se traduire par : recevoir 1 caractère

```
si   caractère = car 1  
alors l'accumuler dans le bcc ; exécuter bloc 1  
sinon si   caractère = car 2  
      alors recevoir 1 caractère si   caractère = car 3  
          alors exécuter bloc 2  
          sinon exécuter bloc 3 fin  
      sinon recevoir 1 caractère ; exécuter bloc 3  
fin   fin
```

. instruction d'alimentation avec test et action :

<pre><action sur alimentation> ::= Δ <u>either</u> <caractère>; <bloc> <u>or</u> <caractère>; <bloc> { <u>or</u> <caractère>; <bloc> } <u>end</u></pre>

qui permet d'obtenir, du tampon de l'utilisateur, le caractère suivant du message utile à transmettre, de le tester, et d'exécuter le <bloc> associé au caractère reçu. Sa structure est équivalente à celle de l'instruction précédente.

Le problème des relations entre les opérations de base \rightarrow , \leftarrow , \uparrow , \downarrow , et les commandes à adresser au coupleur est du domaine de la réalisation interne de ces opérations. Il en est de même pour les relations entre les opérations Δ , ∇ et les tampons de l'utilisateur. On suppose en particulier qu'à leur première occurrence dans le programme, Δ et ∇ possèdent un mécanisme qui les positionne en tête du tampon de l'utilisateur (qui peut être un nouveau tampon ou le même tampon s'il y a retransmission, le changement de tampon pouvant par exemple être provoqué à la réception du compte rendu du transfert par l'utilisateur).

3 - Par ailleurs, le langage possède des instructions classiques d'affectation, de contrôle (IF - THEN - ELSE, GOTO...) et d'appel de procédure.

3.3.2.2. - Syntaxe du langage :

Nous décrivons la syntaxe du langage utilisé dans un métalangage de type ENF ("Backus - Naur - Form")

```
<programme> ::= [ <déclaration> { <déclaration> } ]  
                <bloc> { <bloc> }  
                [ <définition de procédure> { <définition de procédure> } ]  
<définition de procédure> ::= procédure <identificateur> [ ( <paramètre> ) ]  
                                <bloc> { <bloc> } end  
<déclaration> ::= <type> <liste d'identificateurs>  
<type> ::= integer | boolean | character | procédure  
<liste d'identificateur> ::= <identificateur> | <liste d'identificateur>, <identificateur>  
<bloc> ::= [ <identificateur> : ] <instruction> { ; <instruction> }
```

```
<instruction> ::= <variable> := <expression> | <émission> | <rangement>
                | if <expression booléenne> then <bloc> [else <bloc>] end
                | goto <identificateur> | cycle <bloc> end
                | <action sur réception> | <action sur alimentation>
                | <appel procédure> | continue
<émission> ::= → <caractère> [↑] { , <caractère> [↑] }
<rangement> ::= ∇ <caractère>
<appel procédure> ::= <identificateur> [ ( <paramètre> ) ]
<action sur réception> ::= ← either <caractère> [↑] { , <caractère> [↑] }; <bloc>
                        or <caractère> [↑] { , <caractère> [↑] }; <bloc>
                        { or <caractère> [↑] { , <caractère> [↑] }; <bloc> }
                        end
<action sur alimentation> ::= Δ either <caractère> ; <bloc>
                        or <caractère> ; <bloc>
                        { or <caractère> ; <bloc> }
                        end
<expression> ::= <expression booléenne> | <expression arithmétique>
<expression booléenne> ::= <constante booléenne> * <variable> | <expression de relation>
<expression de relation> ::= <primaire> <opérateur de relation> <primaire>
<expression arithmétique> ::= <primaire> | <expression arithmétique> + <primaire>
                        | <expression arithmétique> - <primaire>
<primaire> ::= <variable> | <entier sans signe>
<variable> ::= <identificateur>
<paramètre> ::= <liste d'identificateurs>
<entier sans signe> ::= <chiffre> | <entier sans signe> <chiffre>
<identificateur> ::= <lettre> | <identificateur> <lettre>
<caractère> ::= <caractère de contrôle> | <autre caractère> | <événement> | bcc
<caractère de contrôle> ::= sel1 | sel2 | eng | soh | nob | dle | stx | etb | etx | ack | nak | eot |
<autre caractère> ::= x
<événement> ::= time | timeout | défaut | nilb | nilm
<opérateur de relation> ::= > | ≥ | = | ≠ | ≤ | <
<constante booléenne> ::= true | false
<lettre> ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z
<chiffre> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

3.3.2.3. - Exemple de description :

Nous prenons comme exemple la description dans ce langage de la procédure TMM-UC "half-duplex", dont nous reprenons le graphe (fig. 22) déjà donné en 3.2.3.4.

Les trois phases de l'automate sont décrites : initialisation, émission en statut maître, réception en statut esclave. Le programme fait appel à un certain nombre de procédures dont le détail des actions n'est pas essentiel à la compréhension et qui ne sont pas explicitées afin de ne pas alourdir l'exposé. Par exemple la procédure "décision" teste les demandes de l'utilisateur et positionne en conséquence les booléens "émission" et "réception".

3.3.2.4. - Limite de la méthode :

Le langage présenté nous semble être un bon outil de description des procédures de transmission. Une phase ultérieure consiste à le rendre opérationnel, ce qui nécessite :

- soit l'écriture d'un compilateur spécialisé.
- soit l'utilisation d'un macro-langage puissant permettant de traduire le langage de description en un langage existant. Cette méthode nécessite d'écrire dans le macro-langage choisi, les macro-définitions des primitives du langage de base. Outre qu'elle est plus rapide à implémenter, elle permet de plus l'extension du langage (pour la description d'une procédure de transmission plus fantaisiste !) par l'écriture de nouvelles macro-définitions.

On peut ainsi obtenir un programme capable de gérer une procédure de transmission telle qu'elle a été décrite formellement. Toutefois, cette description formelle ne prend pas en compte les contraintes de temps auxquelles est soumis l'automate et que nous évoquons au paragraphe suivant. Il en résulte un produit aux performances limitées, c'est-à-dire un produit non utilisable au delà d'une certaine vitesse de transmission (cf. 3.4.2.).

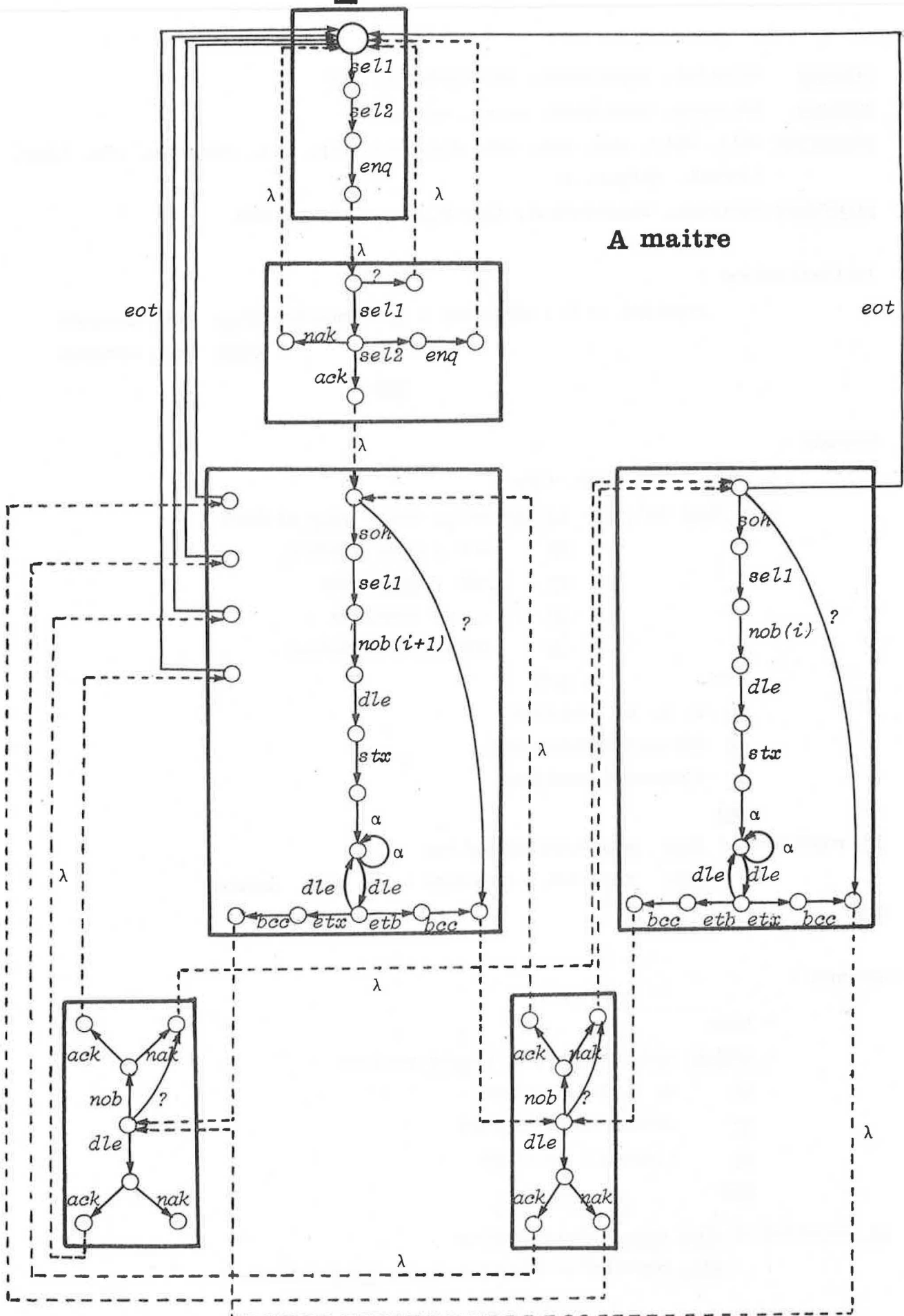


Figure 22

integer repetini, repetenvoi, repetrec, OK, NOK

booleen émission, réception, suite, valeur

character sel1, sel2, enq, soh, nob, dle, stx, etb, etx, ack, nak, eot, time,
timeout, défaut, x

procédure décision, resetdefaut, ininob, avancenob, crendu

initialisation :

```
repetini := 0 ; décision ; if émission then goto demande  
                                else goto attente  
                                end
```

demande :

```
→ sel1, sel2, enq, time ;  
← either sel1 ; ← either sel2, enq ; goto esclave  
    or    ack ; goto maître  
    or    nak ; goto fin  
    or    x, x ; continue  
    or    défaut ; resetdefaut  
    end  
or x, x, x ; continue  
or défaut ; resetdefaut  
or timeout ; continue  
end
```

```
if repetini ≥ 3 then goto initialisation  
    else repetini := repetini + 1 ; goto demande  
end
```

attente :

```
→ time  
← either sel1, sel2, enq ; goto esclave  
    or    x, x, x ; continue  
    or    défaut ; resetdefaut  
    or    timeout ; continue  
    end
```

```
if repetini ≥ 3 then goto initialisation  
    else repetini := repetini + 1 ; goto attente
```

maître :

ininob

suiivante : repetenvoi := 0 ; avancenob

émission : → soh, sell ↑ , nob ↑ , dle ↑ , stx ↑ ;

cycle

Δ either dle ; → dle ↑ , dle

or x ; → x ↑

or nilb; → dle, etb ↑ , bcc, time ; suite:=true ; goto réponse

or nilm; → dle, etx ↑ , bcc, time ; suite:=false; goto réponse

end

end

réponse :

← either dle ; ← either ack ; crendu (OK) ; goto fin

or nak ; crendu (NOK); goto fin

or x ; continue

or défaut ; resetdefault

end

or nob ; ← either ack ; crendu (OK); if suite then goto suivante
else goto fin

or x ; continue end

or défaut ; resetdefault

end

or x, x ; continue

or timeout ; continue

end

if repetenvoi > 3 then crendu (NOK) ; goto fin

else repetenvoi := repetenvoi + 1 ; goto émission

end

fin :

→ eot ; goto initialisation

esclave :

if réception then → sel₁, ack ; ininob ; goto suivants
else → sel₁, nak ; goto attentefin

end

suivants : repetrec: = 0 ; avancenob

réception : → time

← either soh, sel₁↓ , nob ↓, dle ↓, stx ↓;

cycle

← either dle ; ← either dle↓ ; ∇ dle

or etb↓ ; ← either bcc; valeur:=true

goto accuse

or x; valeur:=false

goto accuse

end

or etx↓ ; ← either bcc; valeur:=true

goto accuse

or x; valeur:=false

goto accuse

end

or x ; valeur:=false ; goto accuse

or défaut ; resetdefault; valeur:=false; goto accuse

end goto accuse

or x↓ ; ∇ x

or défaut ; resetdefault ; valeur:=false ; goto accuse

end

end

or sel₁, sel₂, enq ; goto esclave

or eot ; goto initialisation

or x ; valeur:=false ; goto accuse

or défaut ; resetdefault ; valeur:=false ; goto accuse

or timeout ; valeur:=false ; goto accuse

end

accuse :

if valeur then crendu (OK) ; if réception then → nob, ack ; goto suivants

else → dle, ack ; goto attentefin

end

else if repetrec >3 then crendu (NOK) ; → dle, nak ; goto attentefin

else repetrec = repetrec + 1 ; → nob, nak ; goto réception

end

end

```
attente fin : → time
              → either eot ; goto initialisation
                 or      x   ; goto attente fin
                 end
```

3.4. - OPTIMISATION DU TRAITEMENT :

Des cas pratiques d'implémentation que nous avons rencontrés, nous avons déduit que le logiciel de traitement d'une procédure de transmission pouvait être optimisé à 2 niveaux.

3.4.1. - TRAITEMENT DES ERREURS

- Pour une meilleure efficacité de l'automate il est souhaitable qu'une station ne s'engage pas dans un rattrapage d'erreur tant que l'erreur n'est pas établie du point de vue de la procédure de transmission. Or, concernant un transfert d'informations, une erreur ne peut être établie qu'à la fin du-dit transfert. En effet, de même qu'il n'y a que la station réceptrice qui puisse déclarer valide le bloc qu'elle a reçu, de même il n'y a qu'elle qui puisse le déclarer invalide et prendre les mesures appropriées. En particulier, un bloc peut, du point de vue de l'émetteur, avoir été émis incorrectement mais être reçu correctement et finalement déclaré valide par le récepteur.

Nous avons donc pris l'option de ne pas prendre en compte les erreurs détectées à l'émission et de laisser l'initiative de toutes les reprises sur erreur au récepteur de la séquence erronée. Dans le pire des cas, la station qui détecte à l'émission une erreur également détectée à la réception, doit attendre, pour réagir, que la station réceptrice lui confirme cette erreur (en renvoyant un acquittement négatif si le bloc erroné était un message, ou en répétant son message si ce bloc erroné concernait l'acquiescement de ce message). Dans tous les cas, on évite, particulièrement pour une procédure de type half-duplex, les conflits pouvant résulter d'un rattrapage d'erreur engagé simultanément par les deux stations (ces conflits sont inévitables pour une procédure basée sur l'alternance "message-acquittement").

- Outre une amélioration de l'efficacité des reprises sur erreurs, un tel choix simplifie avantageusement la conception de l'automate. Bien des programmes de gestion de lignes de transmission sont confus et peu fiables pour n'avoir pas su faire ce choix, qui découle de la nature d'une procédure de transmission bien comprise.

3.4.2. - MINIMISATION DU TEMPS DE TRAITEMENT MAXIMUM D'UN CARACTERE.

- Un automate logiciel reconnaissant le caractère comme information élémentaire est soumis à une contrainte de temps qui n'apparaît pas pour un automate logiciel dont l'information élémentaire est le bloc (la contrainte est alors masquée par le niveau matériel). En effet, un tel automate doit, en réception, absorber les caractères successifs d'un même bloc à la cadence imposée par la vitesse de transfert des informations sur la ligne de transmission. Pour prendre en compte tout caractère α_i , reçu par le coupleur à la date t_i , l'automate dispose, à partir de cette date t_i , d'un intervalle de temps $\Delta t = \frac{1}{v}$ seconde si v est la vitesse de transfert mesurée en caractères/seconde (fig. 23). L'intervalle de temps Δt correspond à la durée d'assemblage d'un caractère par le coupleur, à partir des 8 bits qui le composent et qui sont transmis en série. Si le caractère α_i n'a pas été pris en compte à la date $t_{i+1} = t_i + \Delta t$, il est perdu et détruit (dans la mémoire interne du coupleur) par le caractère suivant α_{i+1} du bloc. Il en résulte alors une erreur de cadence en réception, qui est signalée par le coupleur et qui invalide entièrement le bloc transmis.

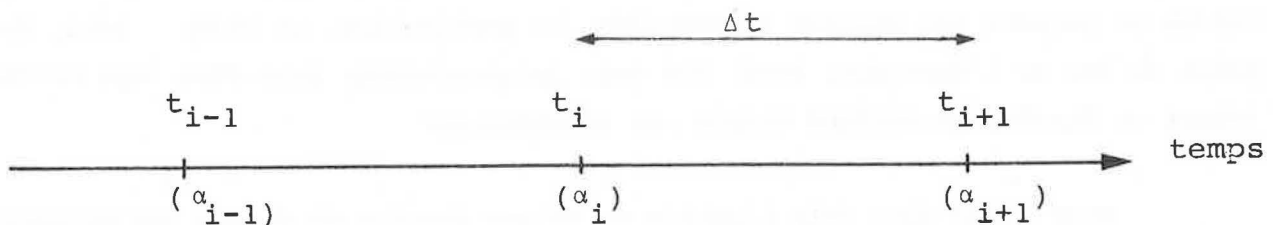


Figure 23

*
Soit θ_i la durée du traitement logiciel associé à la réception d'un caractère de transition α_i . Les figures 24-a et 24-b correspondent à une situation acceptable, ce qui n'est pas le cas de la figure 24-c. Si θ_m désigne le maximum des θ_i , la contrainte apparaît donc comme une borne sur θ_m . Une bonne mesure de sécurité consiste à rendre θ_m inférieur à Δt .

* y compris le changement de contexte ou temps d'activation de l'automate.

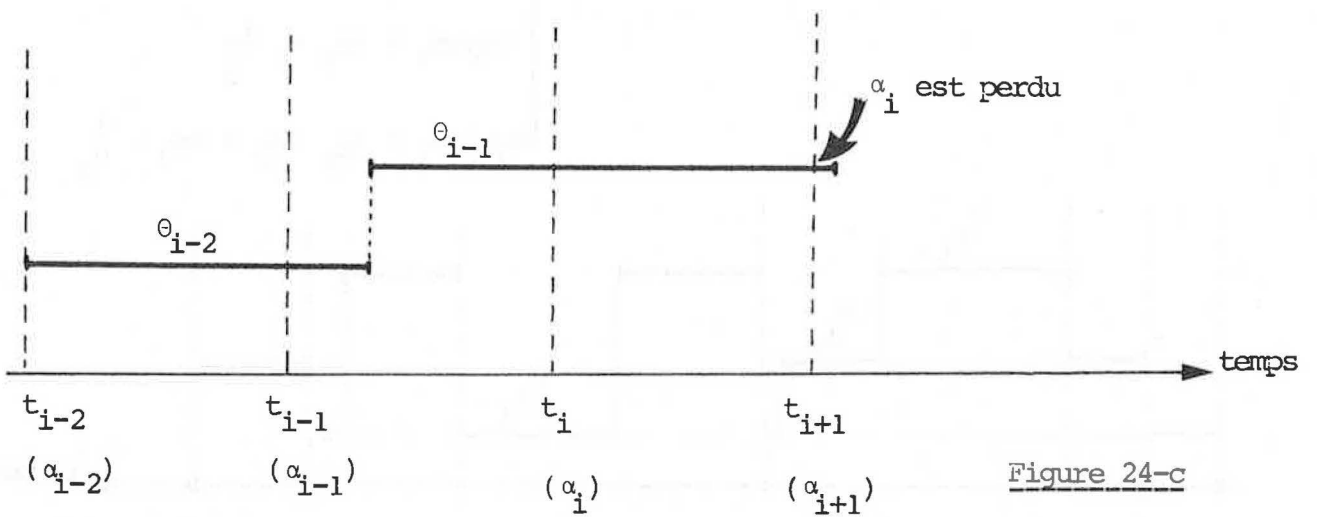
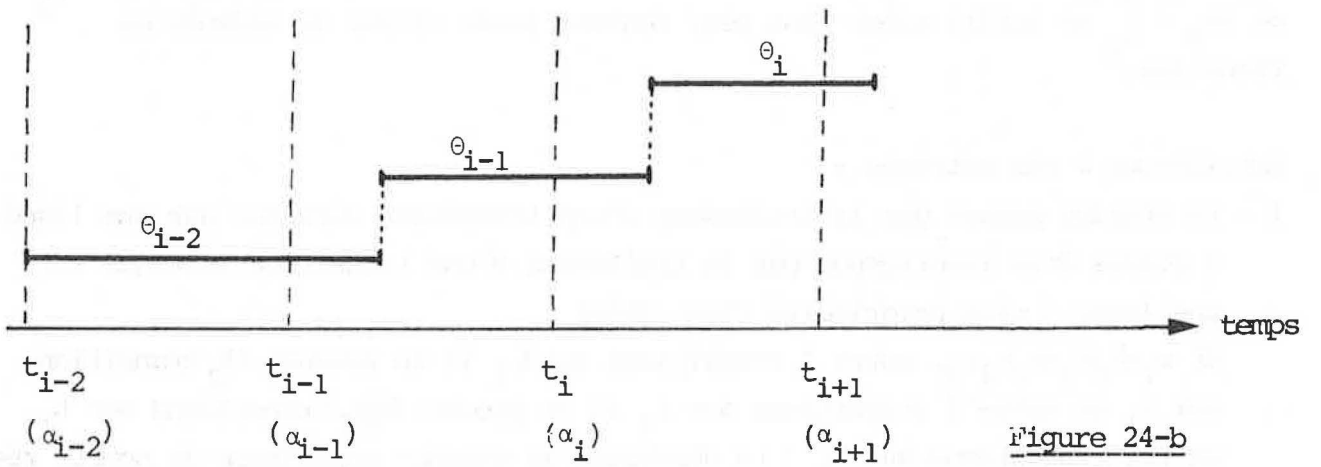
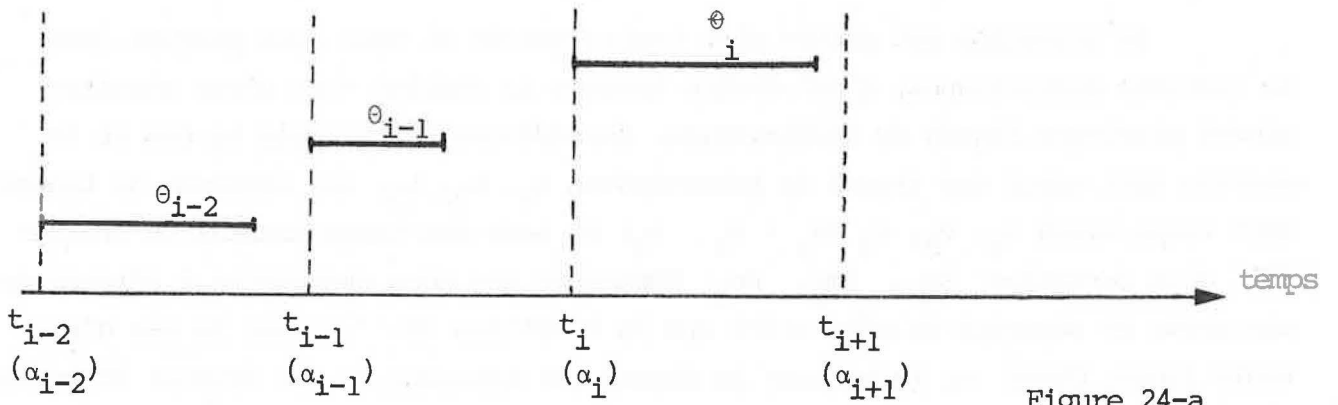


Figure 24

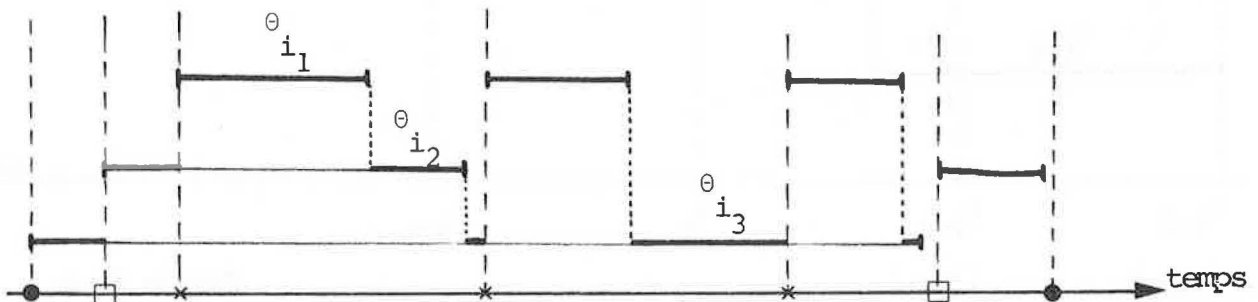
La situation est encore plus contraignante si nous nous plaçons dans un contexte multi-lignes, c'est-à-dire lorsque la station doit gérer simultanément plusieurs lignes de transmission. Considérons par exemple le cas où la station doit gérer des lignes de transmission L_1, L_2, L_3 , aux vitesses de transfert respectives v_1, v_2, v_3 ($v_1 > v_2 > v_3$) et avec des temps maximum de traitement d'un caractère $\theta_{m_1}, \theta_{m_2}, \theta_{m_3}$. Supposons que nous cherchions à déterminer une norme de sécurité du même ordre que la condition $\theta_m \leq \Delta t$ pour le cas d'une seule ligne. Comme va le montrer la discussion suivante, le cas le plus défavorable correspond à la réception quasi-simultanée des 3 transitions dans l'ordre $\alpha_{i3}, \alpha_{i2}, \alpha_{i1}$. A l'évidence, la simultanéité des conditions $\theta_{m_1} \leq \frac{1}{v_1}, \theta_{m_2} \leq \frac{1}{v_2}$ et $\theta_{m_3} \leq \frac{1}{v_3}$ ne suffit alors plus pour écarter toute erreur de cadence en réception.

Considérons 2 cas extrêmes :

1 - La station permet que le traitement d'une transition survenue sur une ligne i puisse être interrompue par le traitement d'une transition survenue sur une ligne j plus prioritaire (fig. 25-a)

Si $v_1 = k_2 v_2 = k_2 k_3 v_3$, entre 2 transitions sur L_2 il se produit $\lceil k_2 \rceil$ transitions sur L_1 et entre 2 transitions sur L_3 il se produit $\lceil k_2 k_3 \rceil$ transitions sur L_1 et $\lceil k_3 \rceil$ transitions sur L_2 ($\lceil a \rceil$ désignant la puissance supérieure du nombre réel a , c'est-à-dire le plus petit entier supérieur au nombre a).

La norme de sécurité* désirée impose alors

$$\left\{ \begin{array}{l} \theta_{m_1} \leq \frac{1}{1} \\ \lceil k_2 \rceil \theta_{m_1} + \theta_{m_2} \leq \frac{1}{2} \\ \lceil k_2 k_3 \rceil \theta_{m_1} + \lceil k_3 \rceil \theta_{m_2} + \theta_{m_3} \leq \frac{1}{v_3} \end{array} \right.$$


- transition de type α_{i3}
- transition de type α_{i2}
- * transition de type α_{i1}

Figure 25-a

* condition suffisante mais non nécessaire.

2 - La technologie de la station ne permet pas l'interruption d'un traitement par un autre traitement (c'est-à-dire qu'il n'y a pas de priorité entre les lignes de transmission) (fig. 25-b).

La norme de sécurité est alors respectée si $\theta m_1 + \theta m_2 + \theta m_3 \leq \frac{1}{v_1}$, ce qui est évidemment beaucoup plus restrictif.

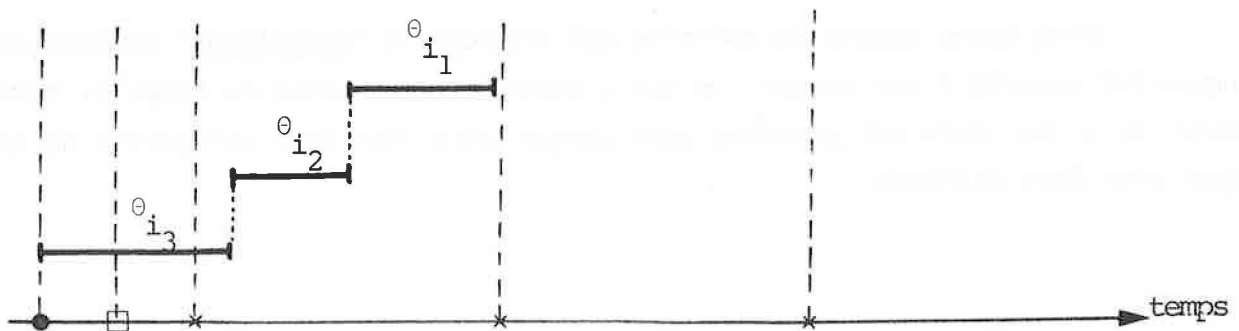


Figure 25-b

Dans un tel cas, la station qui veut gérer 3 lignes de transmission L_1, L_2, L_3 doit être capable de gérer simultanément 3 lignes de transmission à une vitesse de transfert qui est la plus grande des 3 vitesses de transfert. Ainsi une station qui doit gérer une ligne à 2400 bps, une ligne à 4800 bps et une ligne à 9600 bps doit, en fait, être capable de gérer 3 lignes à 9600 bps.

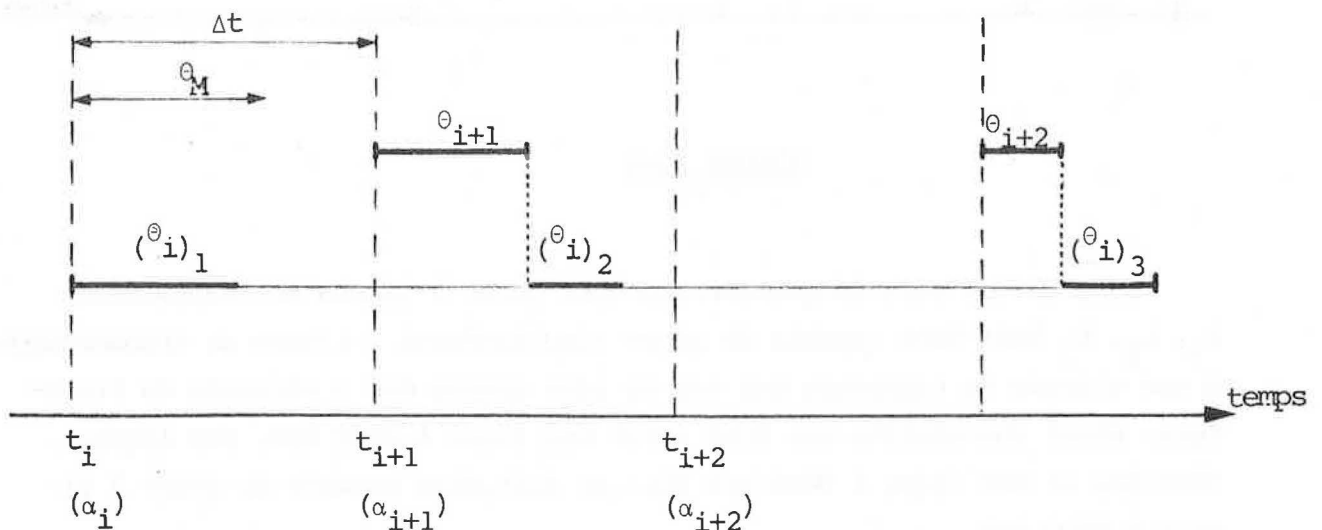
Le système d'interruption du frontal T1600 se situe en fait entre ces deux cas, car il n'existe qu'une seule interruption pour l'ensemble des entrées-sorties (donc le traitement d'une ligne donnée ne peut être interrompue pour traiter une autre ligne) mais au lancement d'un traitement, il est possible de choisir prioritairement parmi plusieurs sous niveaux d'interruption déjà déclenchés.

Si la nécessité de minimiser le temps maximum de traitement d'un caractère est ainsi mise en évidence, il importe également de minimiser le temps de traitement de tout caractère afin que la station conserve un temps disponible suffisant pour effectuer d'autres travaux (traitement propre aux messages par opposition à leur transmission).

Comment minimiser le temps de traitement d'un caractère ?

Soit θ_M la durée maximale permise pour le traitement d'un caractère. Comme on vient de le voir, θ_M dépend du nombre de lignes de transmission gérées simultanément et de leurs vitesses - Soit Δt la durée d'une "transition-caractère", c'est-à-dire l'intervalle de temps séparant les arrivées de 2 caractères successifs.

Nous avons retenu la solution qui consiste à "distribuer" un traitement important associé à une transition sur plusieurs intervalles de temps Δt successifs. (fig 26). Cela est possible pour chaque caractère dont une partie du traitement peut être différée.



$$\theta_i = (\theta_i)_1 + (\theta_i)_2 + (\theta_i)_3$$

Figure 26

Or, dans une procédure de transmission, les traitements qui consomment du temps d'Unité Centrale sont ceux qui sont associés à la réception du dernier caractère d'un bloc. En effet, ces traitements englobent :

- la décision de l'action à mener et de la réponse à émettre pour la station vis à vis.
- le compte rendu éventuel à l'utilisateur.
- la préparation du bloc-réponse et l'initialisation de son émission.

Et précisément aucun de ces traitements partiels n'est urgent. En particulier, l'émission de la réponse peut sans danger être retardée de plusieurs intervalles de temps Δt , car le temps d'attente de la station vis à vis (qui est la durée maximale de la transition λ ou "retournement de ligne" cf. 3.2.2 et 3.2.3) est de l'ordre de 10^3 fois plus grand que Δt . La technique consiste donc, à la réception du dernier caractère d'un bloc, à n'effectuer qu'une partie du traitement correspondant, à condition de recevoir des "caractères-tampons" au rythme des interruptions et à effectuer le reste du traitement à la réception de ces caractères tampons (généralement 1 à 2). Ces caractères tampons sont fournis régulièrement par le coupleur qui continue à échantillonner le signal reçu tant que l'on ne lui a pas adressé une commande de fin de réception.

On a ainsi pu améliorer les performances des logiciels de transmission dans un rapport de 1 à 2.

CHAPITRE IV

EXPERIENCES SUR LES PROTOCOLES DE HAUT NIVEAU

Partant de la communication entre humains, nous isolons les mécanismes fondamentaux d'un dialogue.

Nous détaillons ensuite les différents concepts entrant dans la définition des protocoles en donnant leurs aspects fonctionnels. Puis, à partir d'un exemple concret : le réseau CYCLADES, nous émettons quelques critiques sur les choix pris lors des spécifications.

Enfin, après avoir donné un certain nombre d'outils permettant la modélisation d'un dialogue, nous proposons un exemple de protocole entre un calculateur frontal et un calculateur principal.

4.1. - PROBLEME DU DIALOGUE ENTRE HUMAINS

4.2. - PROTOCOLE

4.2.1. - Protocole-Interface

4.2.2. - Négociation d'un échange d'information

4.2.3. - Séquencement des informations

4.2.4. - Contrôle d'erreur

4.2.5. - Contrôle de flux

4.2.6. - Evènements

4.3. - PROTOCOLES CYCLADES : CHOIX ET CRITIQUES

4.3.1. - Cigale

4.3.2. - Stations de transport

4.3.3. - Abonnés

4.4. - COMMENT MODELISER LE DIALOGUE

4.5. - PROTOCOLE FRONTAL-ORDINATEUR DE TRAITEMENT

4.5.1. - Hypothèses de base

4.5.2. - Construction du protocole

4.1. - PROBLEME DU DIALOGUE ENTRE HUMAINS.

Un individu au sein d'une société, amené à entrer en relation avec ses pairs, doit se plier à certaines règles (sans que ce soit obligatoirement conscient pour lui).

- S'il veut s'adresser à un individu faisant partie d'un groupe, il est amené à attirer son attention en l'appelant par son nom . S'il y a ambiguïté (deux noms identiques), par son prénom.
- Le contact n'est possible que si le partenaire écoute et s'il comprend le même langage (deux sourds pourront se "parler" par gestes).
- On ne peut être certain qu'il a compris (sinon entendu) que s'il répond.
- Les outils ou les moyens utilisés pour communiquer peuvent être très divers et n'interviennent pas dans la compréhension. (parole ↔ audition - geste ↔ vue, ...)
- Le partenaire ayant mal compris demandera de répéter.
- Si celui qui parle a un débit trop rapide, on lui demandera de ralentir (voir même de répéter plus lentement).
- Le contact étant pris, on peut demander quelque-chose (un service, un objet, ...) et attendre la réponse. Si elle ne vient pas, on peut répéter la question.
- Le partenaire est libre d'accepter ou de refuser ou encore de discuter.

Ce sont des concepts de base comme ceux-ci, qui régissent la vie sociale des ordinateurs dans un réseau. Les entités réparties dans un réseau (travaux, processus, tâches, programmes, terminaux...) ont besoin d'échanger des informations. Pour ce faire, elles utilisent un langage ayant sa syntaxe et sa sémantique : un Protocole.

4.2. - PROTOCOLE

4.2.1. - PROTOCOLE-INTERFACE

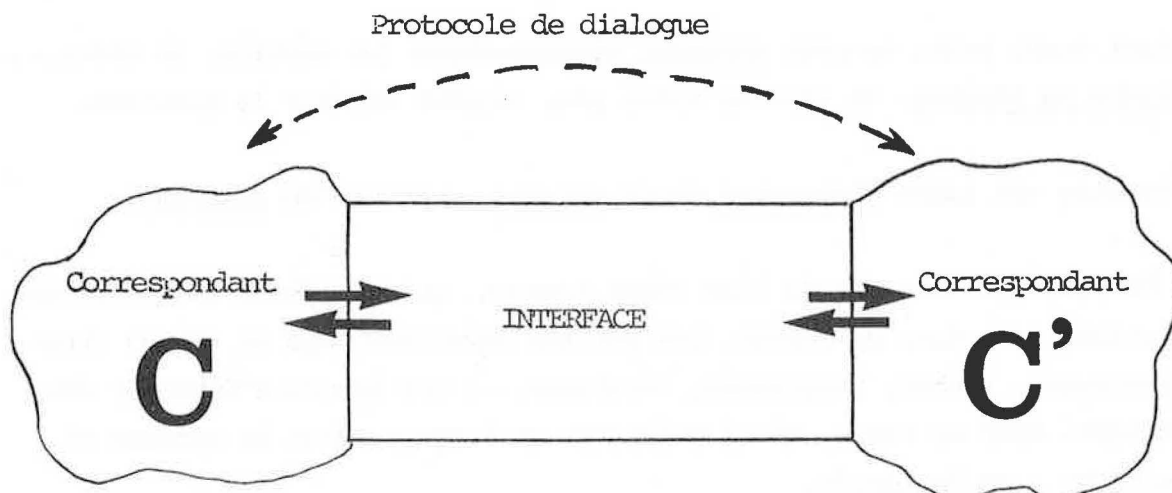
Un protocole est l'ensemble des conventions qui règlent les échanges entre plusieurs objets de même niveau.

Ces objets pouvant éventuellement s'exécuter en parallèle sur des ordinateurs différents.

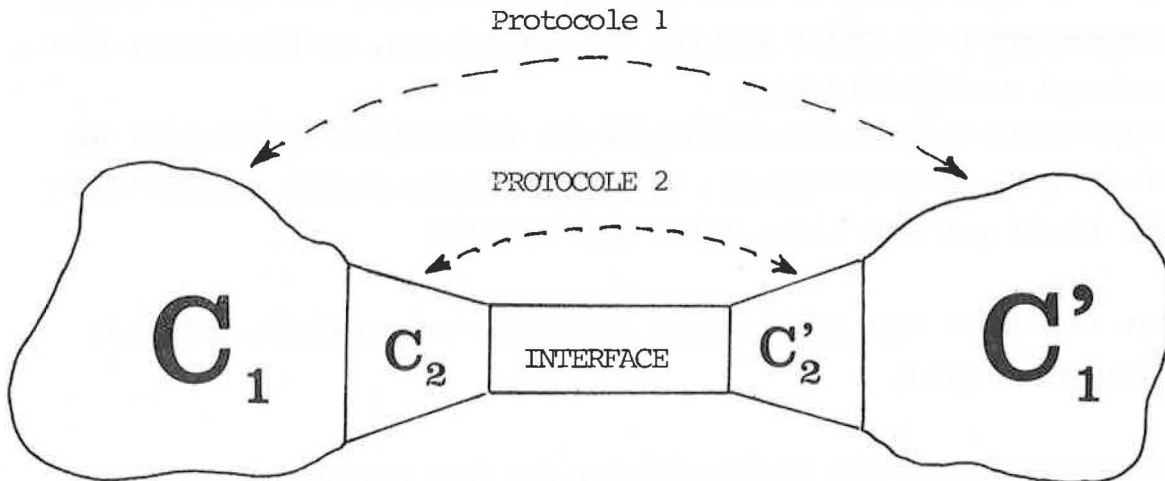
Nous utiliserons le terme "correspondants" pour signifier deux objets de même niveau engagés dans un dialogue.

Les correspondants échangent des informations par l'intermédiaire d'une interface virtuelle, mais n'ont aucun lien direct si ce n'est à travers les éléments intermédiaires régissant les communications.

Nous distinguons bien la notion de protocole de la notion d'interface. Cette dernière constituant tous les rapports directs d'un correspondant avec le milieu qui le sépare de son homologue.



Le rôle de deux correspondants peut être d'assurer une interface entre deux correspondants de niveaux supérieurs. En effet, pour des soucis de modularité, il est souvent préférable de répartir les services en plusieurs couches (ou niveau).



Différents niveaux de protocole peuvent ainsi se superposer, les niveaux inférieurs déchargeant les niveaux supérieurs d'un certain nombre de problèmes et assurant pour eux des services. Voici un exemple de hiérarchie de protocoles.

Niveaux de protocole	Système de banque de données		Langage de commande réseau	
	Soumission de travaux par lots		Accès à des services de temps partagé	Protocole d'appareil virtuel
	Protocole Boîte aux lettres et communication inter-opérateurs		Lancement de travaux	Transfert de fichiers
	Station de transport : permet l'échange de messages (avec contrôles divers).			
	Commutation de paquets : permet l'échange de paquets.			
	Transmission de données : permet l'échange de suites de bits (suivant une procédure de transmission : BSC, TMM,...)			
	Réseau téléphonique : permet la transmission de signaux - Protocole imposé par la réglementation des PTT.			

Nous allons passer en revue les principaux services standards que peut réaliser un protocole de dialogue.

4.2.2. - NEGOCIATION D'UN ECHANGE D'INFORMATION

- Elle assure un contact entre deux correspondants après accord de chacun d'eux. C'est-à-dire qu'elle permet de créer un lien en associant les noms de chacun des correspondants à un numéro logique. Suivant les cas, ce lien pourra être unidirectionnel ou bidirectionnel.

Les correspondants étant amenés à échanger des informations de contrôle (ne serait-ce que pour se synchroniser), il est préférable d'avoir un lien bidirectionnel plutôt que deux liens (comme dans ARPANET)

- Elle rompt le contact soit après accord (l'échange est terminé), soit à la suite d'une erreur grave.

Cette négociation permet la synchronisation des deux correspondants et est nécessaire si l'on veut effectuer les contrôles d'erreur ou de flux sur les informations échangées.

4.2.3. - SEQUENCEMENT DES INFORMATIONS

Les différentes interfaces utilisées dans les communications requièrent souvent une limite à la longueur des messages. Par exemple, un commutateur de paquets (petit ordinateur) véhicule des paquets de petite taille (de l'ordre de 1000 bits). Si on veut transférer un fichier disque, il faudra fragmenter l'information initiale.

Les fragments n'étant pas sensés parvenir dans le bon ordre à destination (erreur, duplication, chemins différents, perte...) Il faudra un mécanisme pour réassembler les messages à l'arrivée et restituer l'information telle qu'elle a été émise à l'origine.

4.2.4. - CONTROLE D'ERREUR

Le but est de détecter les pertes, les altérations, les répétitions de messages. Mais il s'agit également de remédier à l'erreur : c'est-à-dire qu'il faut un mécanisme pour redémarrer la transmission en un point sûr. Cela implique que l'émetteur garde une copie d'un certain nombre de messages parmi les derniers envoyés.

Pour éviter une conservation indéfinie des messages, il faut également que le récepteur avertisse l'émetteur qu'il a bien reçu les messages et que celui-ci peut libérer la place occupée par les copies.

Tout cela constitue en fait un mécanisme de producteur-consommateur : l'émetteur produit des copies de messages qui sont peu à peu consommées par le récepteur. A ceci près que l'émetteur a besoin d'une réponse de contrôle (feed back) de la part du récepteur, pour lui indiquer les messages effectivement consommés.

On pourrait concevoir un mécanisme de contrôle d'erreur au plus haut niveau. Mais les délais de transit de l'information et l'importance de l'entête (overhead) sont prohibitifs. Par contre, au bas niveau, le contrôle et le recouvrement d'erreur ne sont pas trop coûteux puisqu'on opère sur des fragments de petite taille.

D'autre part, le contrôle à un niveau intermédiaire ne garantit pas l'absence d'erreur au niveau supérieur. Tout au plus, il diminue le taux d'erreur.

Il n'y a pas de critère absolu pour déterminer à quel niveau le contrôle d'erreur est le plus efficace. On sait qu'il faut de toute façon en faire un au niveau le plus haut. Pour les niveaux intermédiaires de protocole, le plus rationnel est d'en faire une option que l'on peut mettre en service ou non.

4.2.5. - CONTROLE DE FLUX

C'est l'asservissement de l'émetteur par la capacité du récepteur. C'est un mécanisme qui régit la cadence et le volume des messages transférés à un instant donné. Si ceux-ci ne varient pas, ce contrôle n'est pas nécessaire. Le récepteur alloue de la place à l'émetteur (peu importe la façon dont cette place est allouée). Ce dernier l'interprète comme une autorisation d'envoi.

Le mécanisme peut se décrire d'une manière similaire au contrôle d'erreur, par un modèle de type : producteur-consommateur

Le récepteur produit des "vides" que l'émetteur peut consommer, lorsque la transmission s'est opérée correctement. On peut dire que la sémantique du contrôle de flux est différente de celle du contrôle d'erreur, mais que la syntaxe est la même. (Très souvent, on utilisera le même mécanisme dans les réalisations).

De même que le contrôle d'erreur, on ne sait à quel niveau de protocole, il est préférable de placer un contrôle de flux. On choisira donc d'en faire un mécanisme que l'on pourra à volonté mettre ou non en service à chaque niveau.

4.2.6. - EVENEMENTS

Comme dans les mécanismes d'entrée/sortie, on a besoin de liens dédiés à des signaux autres que des données (genre "interruption"). Ces signaux devant être immédiatement transférés à l'autre extrémité. On appelle ces signaux des évènements.

Cette fonction s'obtient soit en privilégiant un lien uniquement destiné aux évènements soit en instituant des priorités dans les messages et en donnant la plus forte priorité aux évènements.

Un exemple d'évènement est la répercussion de l'appel ("attention") d'un terminal à travers le réseau.

* Tout ou partie de ces fonctions est réalisé à chacun des niveaux de protocole. Les protocoles devant cependant tous satisfaire deux critères fondamentaux :

- La transparence par rapport aux autres niveaux : il ne doit y avoir aucun "effet de bord" entre niveaux. De plus, on doit pouvoir changer la définition d'un protocole (en conservant les services rendus et les interfaces) sans perturber les autres niveaux.

- l'indépendance vis à vis des autres niveaux :

Un protocole doit être autolimitatif, c'est-à-dire qu'une fonction doit nécessiter une quantité finie de ressources (pas de "boucle" infinie ou d'attente perpétuelle), même en cas de mauvais fonctionnement des niveaux inférieurs.

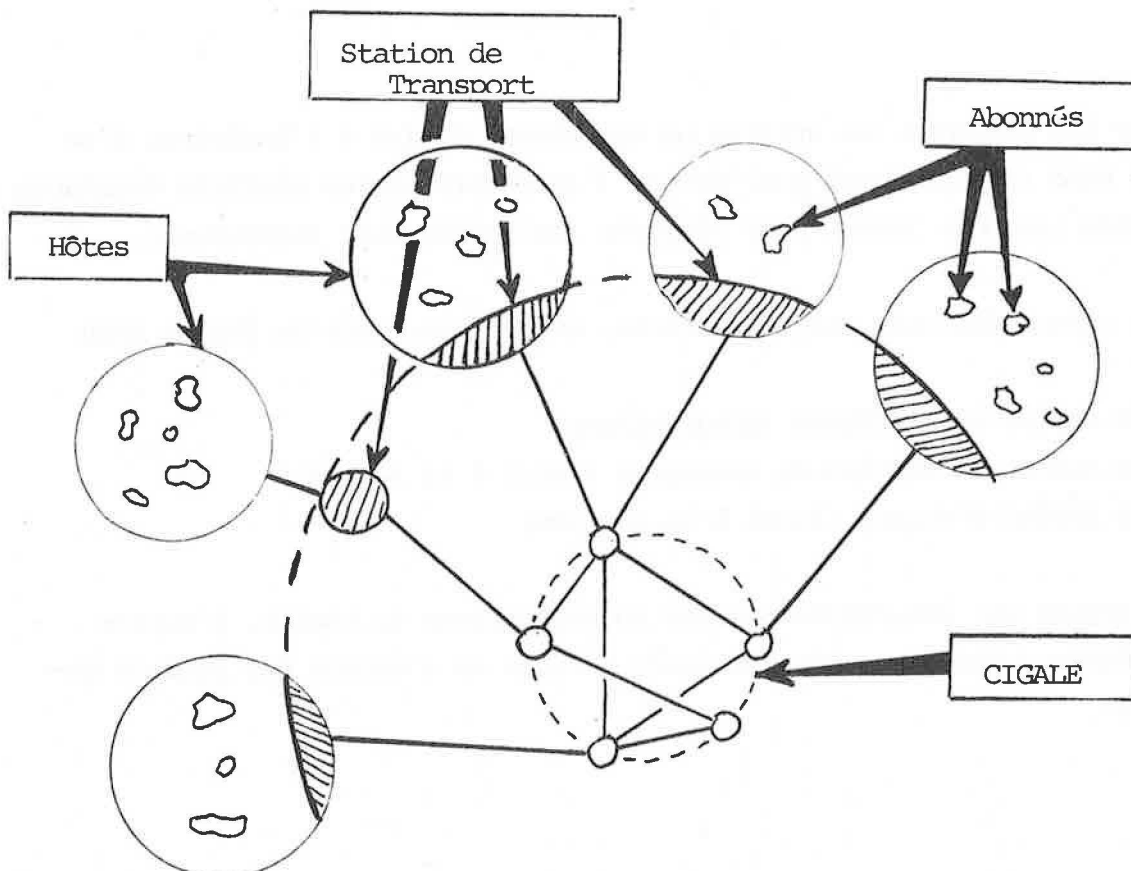
Nous allons reprendre ces différents concepts sur un exemple concret : les protocoles de dialogue dans le réseau CYCLADES.

4.3. - PROTOCOLES CYCLADES : CHOIX ET CRITIQUES

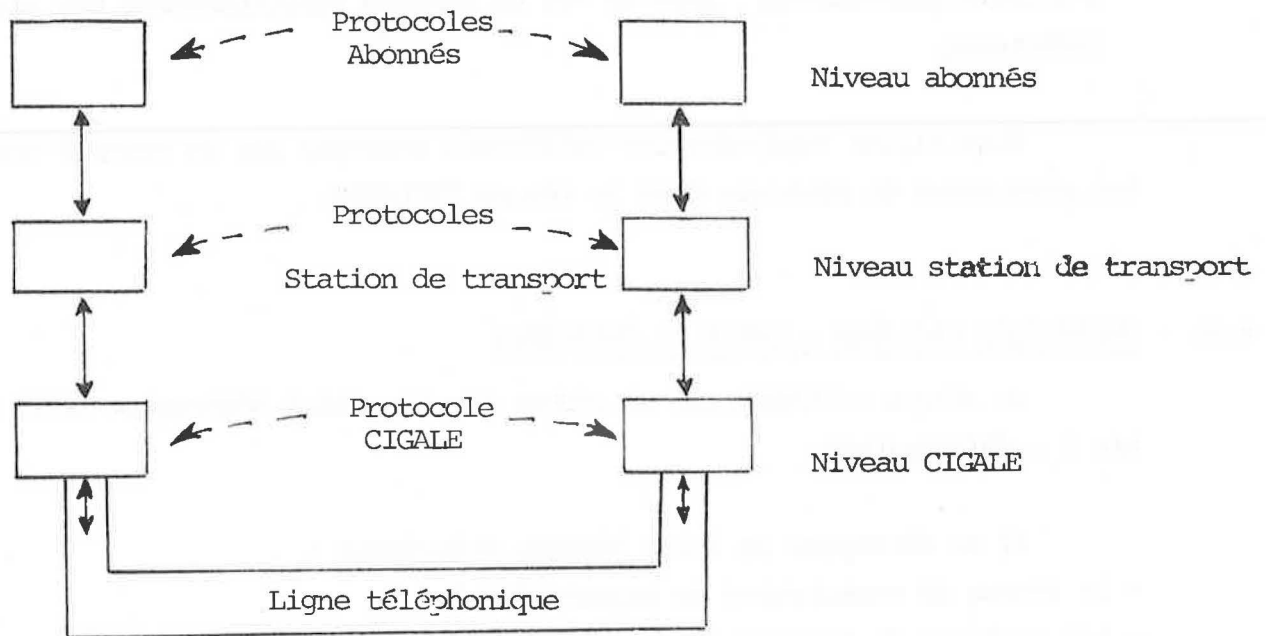
Le réseau CYCLADES est un réseau d'ordinateurs hétérogène (voir chapitre I : définitions).

Il se décompose en trois niveaux principaux :

- le réseau de commutation de paquets, CIGALE
- les stations de transport
- les abonnés dans les sites (hôtes).



Ces différents niveaux sont régis par des protocoles que nous allons détailler dans la suite.



Pour les échanges, on utilise un nom réseau défini à l'intérieur d'un espace des noms réseaux (ceci pour éviter l'ambiguïté et les conflits éventuels si on utilise les noms locaux pour désigner les différentes ressources).

Cet espace des noms est hiérarchisé, chaque nom ayant un format bien défini :

- le numéro de la région (géographique)
- le numéro de station de transport (local à la région)
- le numéro d'abonné (local à la station)

La région est déterminée par des considérations de trafic. L'espace des noms réseau a une structure similaire à celle de l'espace des numéros téléphoniques.

4.3.1. - CIGALE

Le réseau CIGALE est un sous réseau spécialisé qui assure la commutation de paquets avec un routage plus ou moins sophistiqué (à l'heure actuelle, c'est un routage adaptatif c'est-à-dire que pour un paquet, le chemin à parcourir entre le noeud origine et le noeud extrémité est redéfini dynamiquement soit sur apparition d'un évènement modifiant les conditions de transmissions, soit au bout d'un certain laps de temps).

Dans l'élaboration des protocoles CIGALE, on a bénéficié de l'expérience du réseau ARPA (voir ch. I : définitions). Beaucoup de fonctions assumées par le commutateur de paquets dans ARPA sont rejetées aux niveaux supérieurs dans CYCLADES :

- séquençement des paquets et recombinaison à l'arrivée.
- contrôle du flux
- contrôle d'erreur et demande de renvoi de paquets erronés.

Plusieurs avantages à ce rejet :

- Les fonctions dont il est question occupent beaucoup de place mémoire (en particulier la recombinaison des paquets). Or les noeuds de CIGALE sont de petits ordinateurs. Si on augmente leur capacité de mémoire, on augmente leur coût et on risque de réduire leur efficacité.
- CIGALE allégé de ce fait, peut être remplacé par tout autre réseau de commutation de paquets, sans qu'il soit nécessaire de modifier CYCLADES. En particulier, on peut imaginer une connexion inter-réseaux par l'intermédiaire de commutateurs tels que CIGALE, dont les standards sont simples (Mécanismes d'adressage, taille des paquets et procédures de transmission).
- On peut ajouter à cela, que l'utilisateur n'est pas tenté (comme c'est le cas dans ARPANET) d'inclure ses propres logiciels dans les noeuds de commutation.

4.3.2. - STATIONS DE TRANSPORT

Les protocoles entre stations de transport (ST) permettent d'acheminer des messages (lettres) d'une ST à une autre en passant par le réseau CIGALE.

La ST assure un premier niveau de service supplémentaire aux usagers.

Deux types d'échanges sont prévus :

Echanges sur lettres régulières (RG)

- Les lettres de taille limitée (250 caractères) sont envoyées par la station émettrice indépendamment les unes des autres.
- Il n'y a pas de contrôle de flux.
- On peut demander (ou non) un "accusé réception" de la lettre émise.
- Une commande spéciale permet l'envoi d'un évènement.

* Ce mode d'échange ne voit son utilité que dans sa simplicité de mise en oeuvre.

Echanges sur voies virtuelles (W)

Ils comprennent :

- a) Un mécanisme d'établissement et de fermeture sous forme d'automate, avec l'attribution d'un diminutif. Par couple d'abonné, on assure la correspondance des noms réseaux avec des noms locaux à chacune des stations, et un numéro de lien.

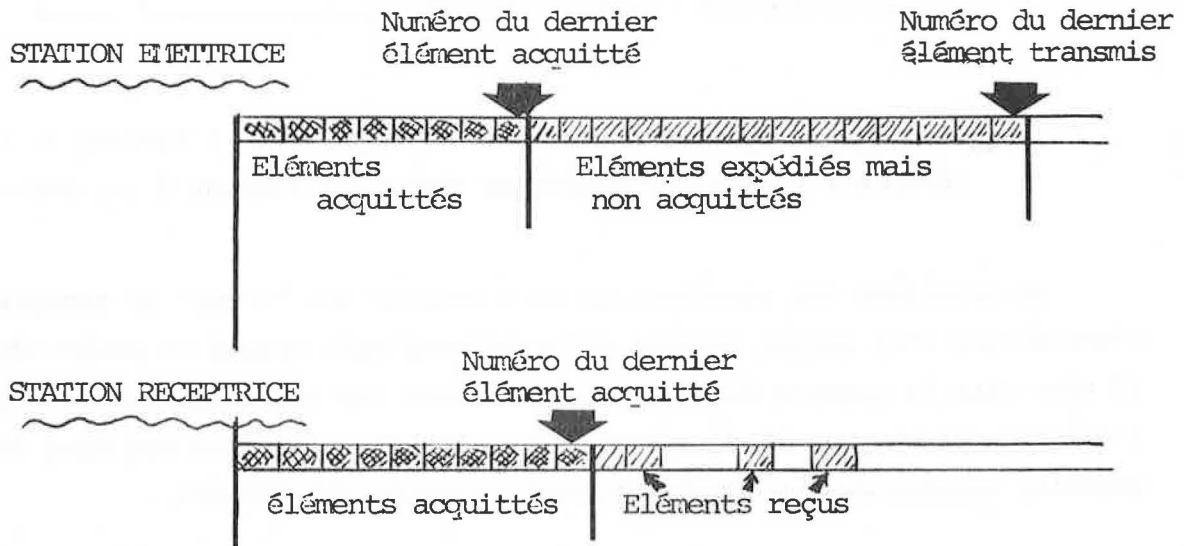
* Ceci permet de diminuer l'entête de chaque envoi. D'autre part, les échanges peuvent se faire simultanément dans les deux sens (symétrie)

- b) Des mécanismes d'exploitation assurant

- b.1) Une restauration du séquençement et un contrôle d'erreur.

- les lettres d'une taille quelconque, sont découpées en éléments de taille fixe numérotés d'une façon contigüe. (le cycle de numérotation étant de 256). La station émettrice envoie un certain nombre d'éléments contigus sous forme de fragments.

- La gestion est fondée sur la notion de pointeurs mémorisant un numéro d'élément.

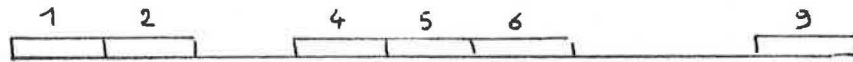


- La station réceptrice acquitte un élément lorsque tous les éléments dont les numéros précédent (modulo le cycle) ont été reçus.
- Si l'élément dont elle reçoit l'acquittement ne correspond pas au dernier élément qu'elle a transmis, la station émettrice renvoie un fragment commençant par le premier élément non acquitté.

* On est certain ainsi qu'à ce niveau, la lettre reçue est correcte (sans erreur, ni doublement, ni "trou") jusqu'à un endroit connu de l'émetteur. En cas de panne, on pourra toujours reprendre l'envoi à partir de cet endroit.

* D'autre part, on minimise le trafic en ne renvoyant que les "trous".

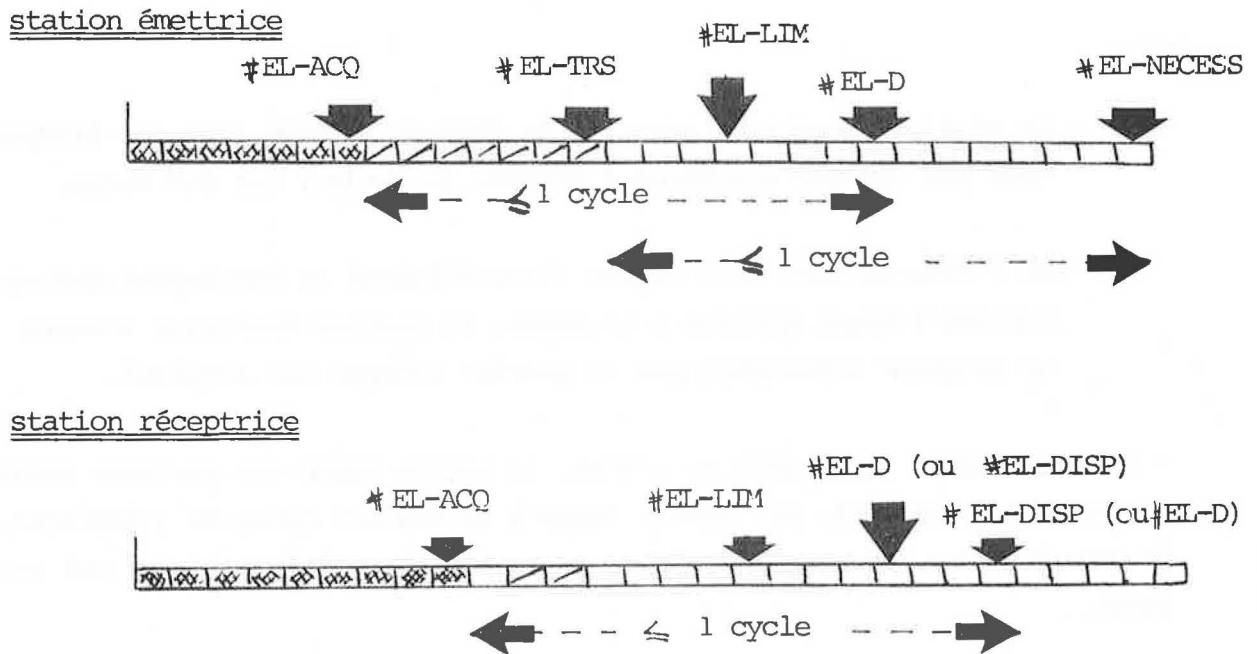
Exemple : Supposons que le récepteur ait reçu correctement les éléments 1, 2, 4, 5, 6 et 9 d'une lettre. Il renvoie l'acquiescement de l'élément 2.



L'émetteur renvoie un fragment commençant par l'élément 3. Le récepteur peut alors acquiescer jusqu'à l'élément 6 ... etc...

Ce mécanisme est excellent si on a souvent des "trous" en réception, c'est-à-dire s'il arrive souvent qu'un élément soit erroné ou perdu. Mais il nécessite la gestion de deux pointeurs pour les acquiescements : un pour l'émission et un pour la réception. (on verra dans la suite que dans une nouvelle version des protocoles ST, ces pointeurs ont disparu).

b.2) Un contrôle de flux réalisé par l'adjonction de six pointeurs supplémentaires. A un instant donné, on a la situation suivante :



#EL-ACQ : Numéro du dernier élément acquitté.

#EL-TRS : Numéro du dernier élément transmis.

#EL-LIM : Numéro du dernier élément qu'on est autorisé à émettre
#EL-NECESS : Numéro du dernier élément que l'émetteur désire émettre
#EL-D : Numéro du dernier élément dont l'émetteur demande l'autorisation d'émission.
#EL-DISP : Numéro du dernier élément disponible dans le récepteur.

} gestion des ressources en mémoire

- A chaque envoi de fragment, l'émetteur demande des ressources au récepteur (#EL-D) en nombre d'éléments.

- Compte tenu de la place disponible chez lui (#EL-DISP), le récepteur octroie des ressources à l'émetteur (#EL-LIM) à chaque envoi d'acquiescement.

* La gestion des ressources est assez lourde surtout du fait qu'elle se fait au niveau des éléments (très petits). Elle requiert en outre deux processus supplémentaires : un pour l'émetteur (calcul du nécessaire) et un pour le récepteur (calcul du disponible).

c) Une commande spéciale permet l'envoi d'un évènement.

* Récemment (avril 1974), l'équipe de coordination de CYCLADES a proposé de nouveaux protocoles. Ces protocoles font l'objet de discussions internationales, en vue de l'adoption de standards. La gestion des protocoles est beaucoup allégée par rapport à la version précédente et s'appuie sur trois constatations :

- les délais de transfert d'un paquet de 1000 bits dans le réseau CIGALE sont inférieurs à la demi-seconde.

- la probabilité pour qu'un bit soit erroné est de 10^{-10}

- la probabilité pour qu'un paquet soit erroné est de 10^{-4}

Cette Station de Transport ainsi définie est appelée Station type 2 (par opposition à la précédente : Station type 1).

En premier lieu, la notion d'abonné n'intervient plus au niveau du service de transport.

La Station de Transport ne connaît plus que des portes. (Les abonnés s'attribuant ces portes ne sont pas connus de la Station).

On peut associer un lien à deux portes : un flot dont l'identification est la concaténation des identifications des deux portes.

L'existence d'un flot peut être comparée à l'existence d'une voie virtuelle dans la station de transport type 1.

Là encore, deux types d'envois :

- les lettres de longueur variable mais limitée sont échangées sur flots
- les télégrammes courts et de longueur fixe ont remplacé la notion d'évènement, et sont envoyés sur portes

Les lettres sont découpées en fragments de longueur fixe et recomposées à l'arrivée. L'envoi sur lettres bénéficie en outre de trois services additionnels qui sont l'objet d'options prises lors de l'initialisation de la session (négociation). Ce sont :

- contrôle d'erreur - opérant sur la lettre entière (et non sur les fragments)
- contrôle de flux - par allocation de "crédits" entre stations au cours des échanges. Un crédit étant un nombre de lettres que l'émetteur peut envoyer et le récepteur recevoir.
- adressage réduit - l'attribution d'un diminutif.

La symétrie entre stations est conservée (Lien bidirectionnel)

- * Le contrôle d'erreur opérant sur la lettre, il n'est plus besoin de mémoriser les séquences de fragments chez l'émetteur : en cas d'erreur, il renvoie toute la lettre. De même, chez le récepteur, l'acquiescement se fait sur la lettre, et non sur une séquence de fragments.
- * Le contrôle de flux revient pour le récepteur, à répercuter les "demandes de lecture" de l'abonné.

4.3.3. - ABONNES

Les protocoles abonnés, ou protocoles de niveaux supérieurs ne seront pas détaillés dans ce chapitre.

Parmi eux, seul le protocole d'appareil virtuel a fait l'objet de spécifications précises (dans CYCLADES), ainsi que le protocole Client-Serveur.

4.4. - COMMENT MODÉLISER LE DIALOGUE

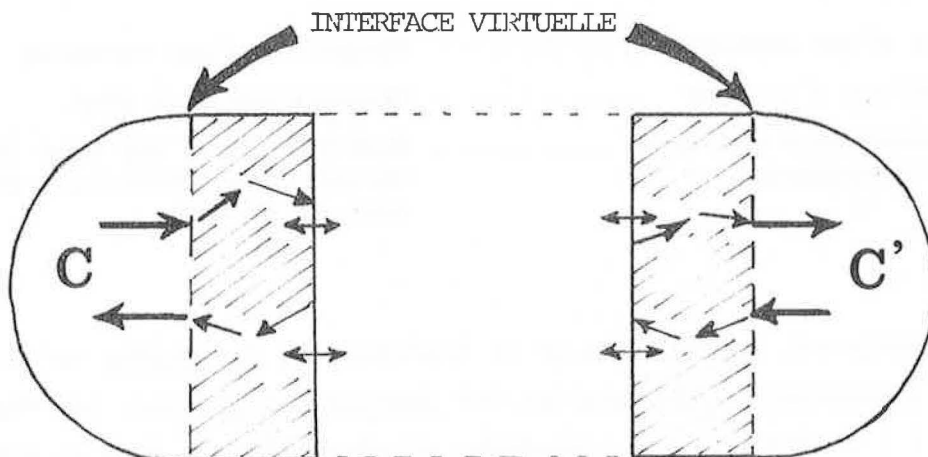
Deux correspondants communiquent en agissant sur leur interface commune. Chacun d'eux a sa propre vision de l'autre, et cette vision est l'ensemble des moyens d'interactions qu'il possède sur son homologue.

Le protocole de dialogue définit le comportement de chacun des correspondants, c'est-à-dire l'ensemble des actions produites par l'un et des effets induits dans l'autre [CYC-3]. (Lesquels effets n'entraînent pas obligatoirement de réactions).

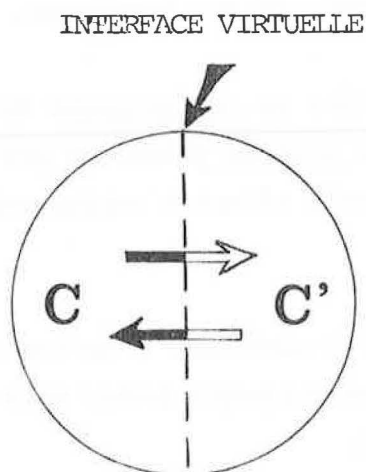
Les moyens mis en oeuvre pour transformer l'action en effet ne concernent pas les protocoles entre les deux correspondants, mais les rapports qu'ont chacun d'eux avec sa propre interface.

- * Un exemple de protocole est la coopération de deux processus par échange de messages.
 - Le protocole ne concerne que la structure des messages échangés
 - Le moyen d'échange ou interface peut être réalisé à l'aide d'un sémaphore avec lequel chaque processus a des rapports de type P et V.(voir DJIKSTRA [SYS-2])

Pour spécifier un protocole de dialogue en faisant abstraction des contraintes relatives aux interfaces, il faut définir une interface virtuelle de niveau supérieur. Cette interface est la vision réelle qu'a un correspondant C de son homologue C' .



Ce qui, du point de vue fonctionnel, revient à :

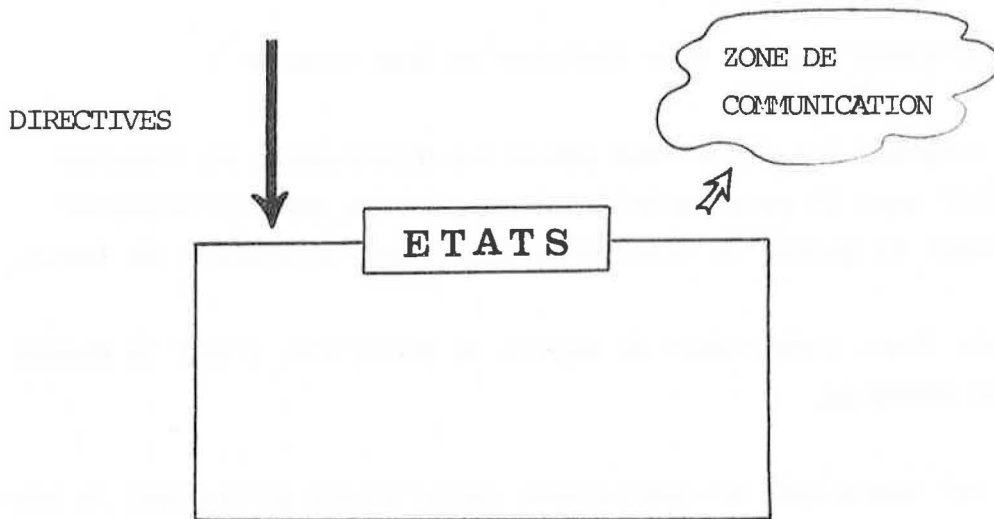


Ce modèle d'interface doit être suffisamment général pour s'appliquer à chaque cas particulier et trouver sa traduction naturelle dans chaque environnement.

Le modèle utilisé dans CYCLADES [CYC-4] fait intervenir trois types d'actions élémentaires auxquelles correspondent des effets :

Envoi d'une commande	Réception d'une commande
Affichage d'un état	Observation d'un état
Ecriture dans une zone.....	Modification d'une zone de communication et accessibilité de cette zone en lecture.

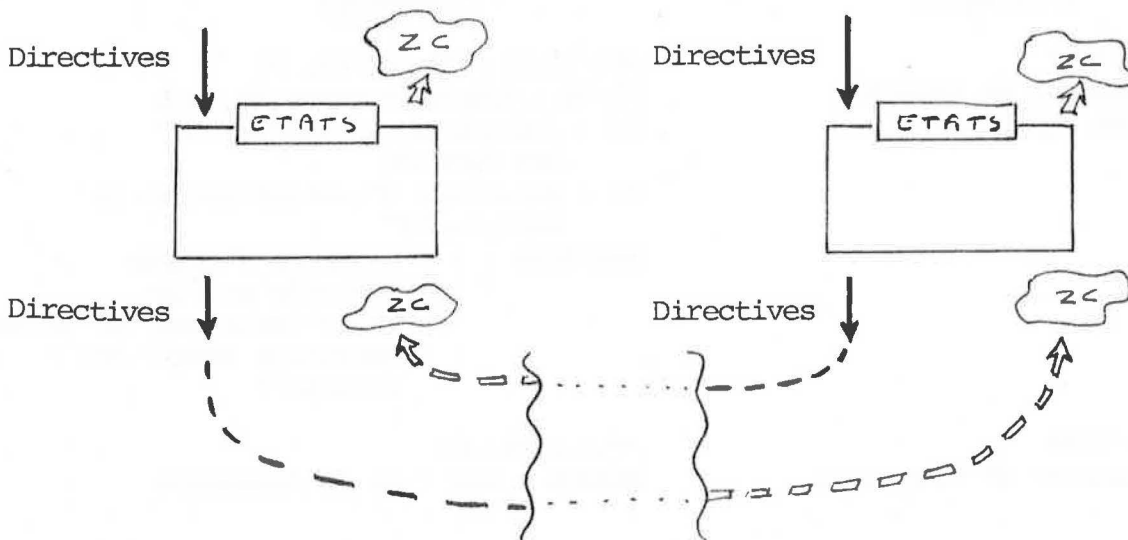
Ce modèle est capable d'agir en fonction des directives qu'il reçoit. Ces actions provoquent en particulier des changements d'états. Les états sont observables à l'extérieur. Les transferts d'information se font au moyen d'un espace mémoire partagé appelé zone de communication. L'ensemble fonctionne selon un automate d'états finis.



Le modèle dispose en outre d'un mécanisme interne de réveil. Il pourra enclencher le réveil qui, au bout du temps indiqué adressera une directive et fera ainsi sortir d'un état d'attente.

La façon dont les directives sont adressées, les moyens d'observation des états, la structure du réveil, de même que les modes de gestion et d'accès des zones de communication peuvent être adaptés à chaque réalisation et ne sont pas précisés.

Les rapports entre deux correspondants pourront se schématiser ainsi : Le correspondant A émet des directives (ou des informations) que le correspondant B peut récupérer dans sa zone de communication avec le niveau inférieur, et réciproquement.



Les directives peuvent être divisées en deux classes :

- La première comprend les directives associées directement au transfert de texte. Elles sont là pour contrôler l'exactitude, et pour replacer (éventuellement) la partie de texte à sa place dans l'ensemble du texte.

Le texte étant indépendant du niveau de protocole, c'est la classe des variables globales.

- La deuxième est constituée des mécanismes particuliers permettant la synchronisation entre les deux correspondants. C'est une partie du contexte d'exécution du protocole : elle constitue les variables locales. Ces dernières n'apparaissent pas aux autres niveaux, sauf peut être les variables utilisées pour les réveils, lorsque le contrôle passe à un autre niveau (au dessus ou en-dessous).

On pourra donc exprimer les traitements relatifs à ces directives dans un langage à structure de bloc type ALGOL. Les directives étant synthétisées par un mnémonique suivi d'un certain nombre de paramètres.

Les changements d'états et les événements associés aux transitions étant représentés sous forme de graphe.

Pour illustrer ce qui précède, nous allons prendre un exemple tiré des spécifications de la station de transport CYCLADES type II [CYC-5] : ouverture et fermeture d'un flot.

Directives

FL-INIT
(demande de création
d'un flot)

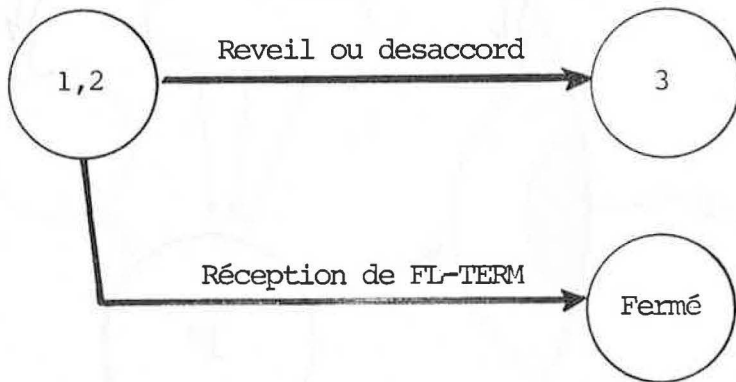
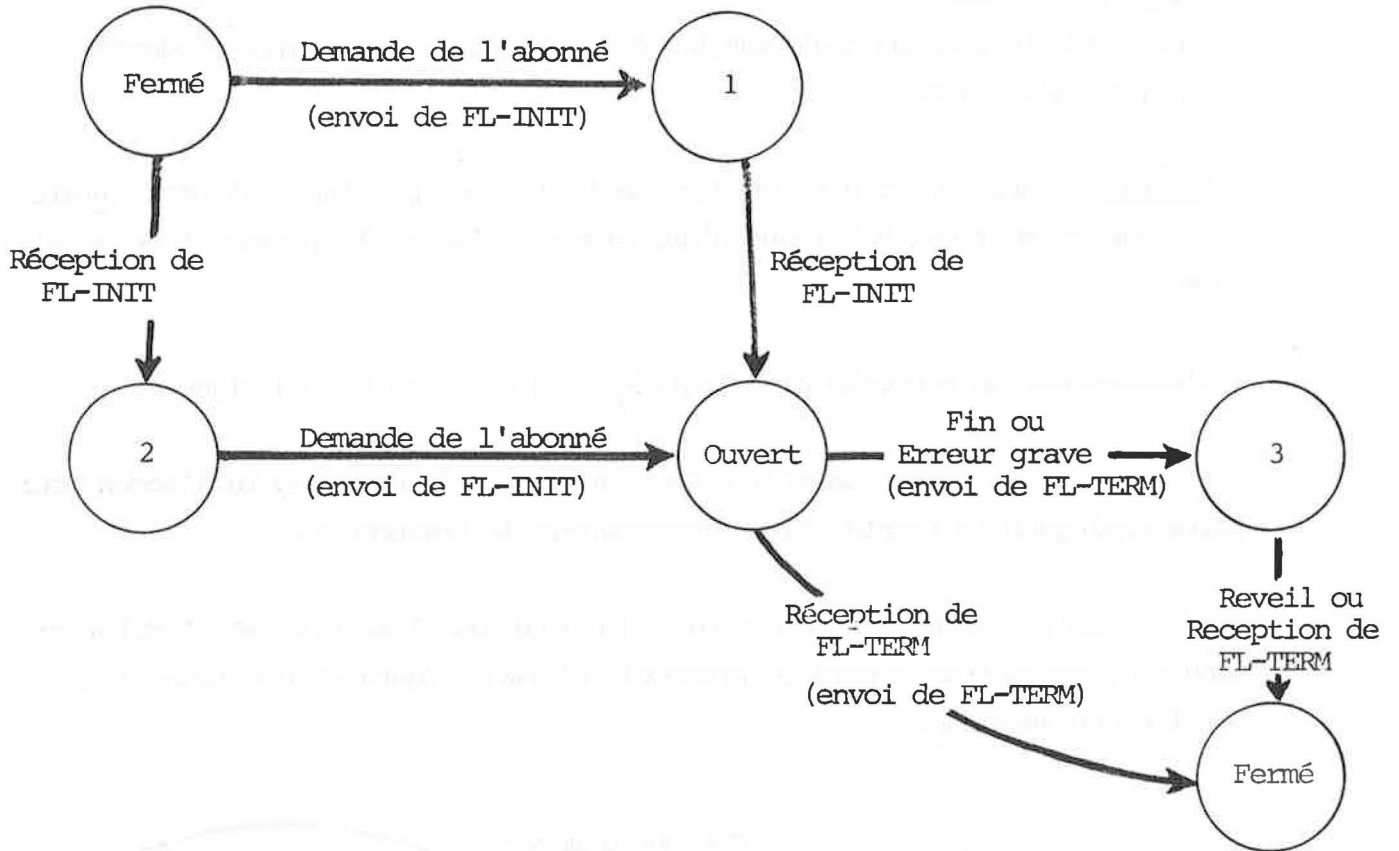
FL-TERM
(demande de fermeture)

Paramètres

/FL-ID/S1, S2, S3/P2, P3
FL-ID : identificateur du flot
SJ : indique si le service "J"
est demandé
PJ : paramètre d'initialisation du
service "J"
Services : 1 : contrôle d'erreur
2 : contrôle de flot
(P2 = table maxi de lettre)
3 : adressage réduit (P3 :
diminutif)

/FL-ID/RAISON
RAISON : motif de la fermeture

AUTOMATE
D'OUVERTURE ET DE FERMETURE D'UN FLOT



4.5. - PROTOCOLE FRONTAL - ORDINATEUR DE TRAITEMENT

4.5.1. - HYPOTHESES DE BASE

Pour utiliser, à partir d'un frontal, les services d'un ordinateur de traitement connecté à lui, deux cas sont à considérer :

- l'ordinateur supporte, sur la ligne de transmission, les entrées aux services qu'il offre.

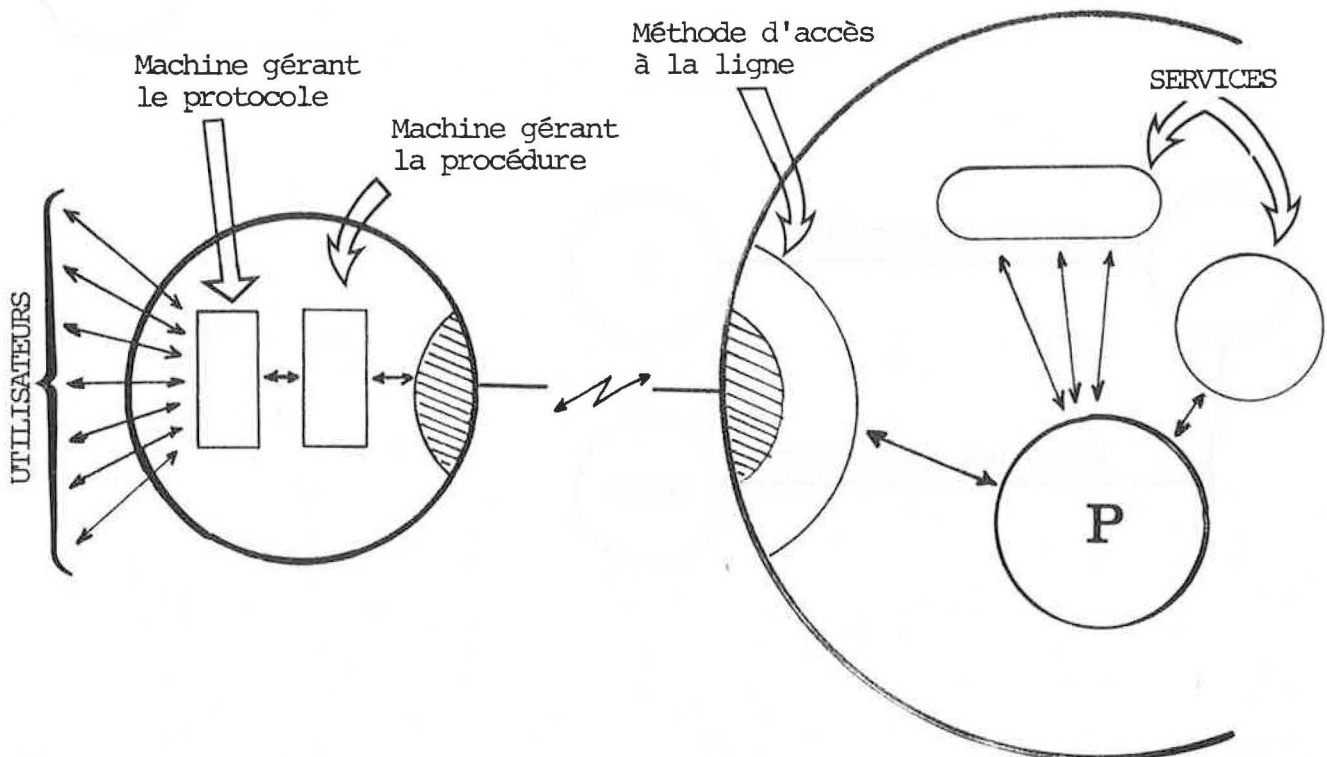
En ce cas, le frontal doit simuler ces entrées, c'est-à-dire s'adapter au protocole existant.

Exemple : Dans le cas d'un service de traitement par lots, il peut simuler le comportement des terminaux d'une station d'accès à distance à ce service.

- l'ordinateur ne supporte pas d'entrées aux services sur la ligne.

On doit alors construire de toutes pièces un protocole d'accès pour faire dialoguer le frontal avec l'ordinateur de traitement.

Un programme (P) se déroulant parallèlement au système d'exploitation de l'ordinateur gèrera ce protocole et se chargera d'interfacer les services en question.



On suppose qu'on se trouve dans ce dernier cas. C'est-à-dire qu'il faut construire un protocole pour accéder les services de l'ordinateur de traitement.

On suppose également qu'il existe, en même temps que la ligne de transmission reliant les deux ordinateurs, une méthode d'accès gérant cette ligne et permettant d'envoyer des informations, via cette ligne, à un programme se déroulant dans l'ordinateur de traitement.

4.5.2. - CONSTRUCTION DU PROTOCOLE

Les échanges entre le frontal et l'ordinateur de traitement s'effectuent suivant trois niveaux hiérarchiques :

Niveau 1 : Communications de "trains d'information" sur la ligne de transmission avec pour support une procédure de transmission :
C'est la méthode d'accès du système d'exploitation qui les gère.

Niveau 2 : Premier niveau de protocole qu'on appellera "Service de communication par paquets".
Il s'agit des services généraux au niveau des échanges entre le programme (P) et son homologue dans le frontal.

Niveau 3 : Echanges proprement dit entre un service particulier (ou une entrée à ce service) parmi les services offerts de l'ordinateur de traitement et un utilisateur du frontal.
Il font l'objet d'un deuxième niveau de protocole : "le protocole d'accès au frontal" (PAF)

PREMIER NIVEAU

Ce protocole très simple peut se schématiser comme suit :

SRV	LG	Texte du paquet
-----	----	-----------------

SRV définit ce que le programme (P) doit faire du texte qui suit. C'est l'en-tête de contrôle.

Exemple : - Trace du paquet demandée
- Acquittement du paquet demandé
- Echo demandé
- Commandes sans texte propre au programme (P)

LG est la longueur du texte

DEUXIEME NIVEAU

De nombreuses possibilités s'offrent à nous pour ce protocole d'échange entre "interlocuteurs", c'est-à-dire entre utilisateurs et services.

Pour faciliter le choix, on établit les conventions suivantes. De ces conventions découleront la structure des éléments échangés.

1°) Dans un même paquet, on peut mettre plusieurs éléments de protocole.

Il est alors nécessaire d'indiquer la longueur de chaque élément dans l'en-tête de celui-ci.

2°) Pas de trafic parallèle pour les acquittements.

On les inclue dans les envois courants.

3°) Pas d'acquittement systématique au coup à coup.

On admet une anticipation dans les envois et on doit ordonner les messages.

4°) Pas d'acquittement négatif.

On est amené à acquitter le dernier élément reçu dans l'ordre.

5°) Contrôle d'erreur et de flux optionnels.

6°) Les messages ont une longueur maximum fixée.

Les paramètres nécessaires sont alors :

- IDENT Identification de l'interlocuteur, c'est-à-dire :
- l'utilisateur dans le cas des échanges Ordinateur → Frontal
 - le service (ou l'entrée dans le service) dans le cas des échanges Frontal → Ordinateur.
- LG Longueur de l'élément de protocole.
- CTRL On regroupe dans ce paramètre les informations de contrôle :
- ouverture, fermeture, contrôle d'erreur, contrôle de flux demandés.
 - "ATTENTION" de la part d'un terminal.
 - "A vous" (ou déblocage du clavier) de la part du service.
 - Crédits accordés, c'est-à-dire place disponible chez le récepteur en nombre de messages.
- REF-ACK Référence du dernier message reçu correctement.
- REF-MSG Référence du message transporté dans l'élément de protocole

On peut schématiser un élément de protocole comme suit :

IDENT	LG	CTRL	REF-ACK	REF-MSG	Texte du message
-------	----	------	---------	---------	------------------

CHAPITRE V

CONCEPTION GLOBALE DU CALCULATEUR FRONTAL

Ce chapitre met en relief les idées fondamentales qui ont présidé à la conception du frontal réalisé.

Après avoir défini le rôle que nous assignons au frontal dans un environnement télé-informatique, nous en déduisons les fonctions hiérarchiques nécessaires qu'il doit assurer. Partant de cette analyse, nous indiquons la méthodologie que nous avons suivie pour construire le logiciel et qui nous a conduit à une architecture "structurée" et flexible du système.

5.1. - RÔLE DU FRONTAL

5.1.1. - Rappel des objectifs

5.1.2. - La fonction globale du frontal

5.1.3. - Les fonctions particulières d'accès aux ressources

5.1.3.1. - L'accès aux ressources du réseau

5.1.3.2. - L'accès aux ressources du site

5.1.4. - Synoptique des fonctions nécessaires

5.2. - ARCHITECTURE DU LOGICIEL

5.2.1. - La notion de "machine"

5.2.2. - Organisation générale de la machine "frontal"

5.3. - L'EXEMPLE DE LA MACHINE DE COMMUNICATION

5.3.1. - La vision des "serveurs"

5.3.2. - L'architecture interne

5.3.2.1. - Le niveau technologique

5.3.2.2. - La machine - interruption

5.3.2.3. - Les machines de transmission

5.3.2.4. - Les machines de gestion des transmissions

5.3.2.5. - Les machines de gestion des protocoles

5.3.2.6. - La boîte aux lettres

5.3.3. - Flexibilité et extensibilité de la réalisation

5.3.3.1. - Flexibilité

5.3.3.2. - Extensibilité

5.1. - ROLE DU FRONTAL

5.1.1. - RAPPEL DES OBJECTIFS

Considérons la situation d'un site (type centre de calcul), disposant d'un ou de plusieurs ordinateurs locaux, vis à vis d'un réseau d'ordinateurs externe (type CYCLADES). Sur le réseau de même que sur les ordinateurs locaux pré-existent un certain nombre de services - nous entendons ici des services macroscopiques (services "visibles" au niveau des utilisateurs) tels que services de temps partagé, services de traitement par lots etc... Par ailleurs, un certain nombre de terminaux sont raccordés soit à ces ordinateurs locaux (terminaux locaux), soit au réseau d'ordinateurs (terminaux distants).

L'objectif est alors de permettre (cf. 1.1.2.3) :

- 1 - Aux terminaux locaux d'accéder aux services distants disponibles sur le réseau.
- 2 - Aux terminaux distants connectés au réseau d'accéder aux services locaux disponibles sur les différents ordinateurs du site.
- 3 - Au terminaux locaux d'accéder indifféremment aux divers services locaux offerts par les ordinateurs du site.

La première idée qui vient à l'esprit consiste à employer une méthode brutale. Les objectifs 1 et 2 peuvent en effet être réalisés par connexion directe de chaque ordinateur local au réseau externe, ce qui nécessite l'introduction de tous les logiciels d'accès au réseau sur chacun de ces ordinateurs locaux [CYC-10]. Puis l'objectif 3 peut être atteint soit avec une connexion par couple des ordinateurs locaux, ce qui nécessite autant de logiciels de communication particuliers qu'il existe de couples, soit par la définition d'un réseau local complet. Ces solutions ne peuvent être satisfaisantes dans la mesure où elles engendrent rapidement une multiplication des logiciels avec une grande dispersion des efforts et de nombreuses duplications.

Notre concept de frontal simplifie considérablement les problèmes. Il consiste en effet, à interposer un mini-calculateur d'une part entre chaque ordinateur local et le réseau, d'autre part, entre chaque couple d'ordinateurs

locaux. Bien entendu, ce mini-calculateur est unique. Il concentre les logiciels de communication avec le réseau et les logiciels de communication avec chaque ordinateur local, et est utilisé :

- par chaque ordinateur local en frontal du réseau,
- par le réseau en frontal de chaque ordinateur local,
- par chaque ordinateur local en frontal des autres ordinateurs locaux.

La superposition de ces fonctions frontales au niveau du mini-calculateur définit un service frontal "orientable", qui généralise la notion classique de frontal "orienté" depuis une série de terminaux vers l'accès à un ordinateur [GEN-6]. Cette méthode présente l'avantage de centraliser les efforts de programmation dans le frontal plutôt que dans les ordinateurs à connecter, où sont minimisés les investissements logiciels.

5.1.2. - LA FONCTION GLOBALE DU FRONTAL

Le frontal doit offrir les moyens de mettre en relation les terminaux qu'il connaît avec les services qu'il connaît (fig. 1). Tous les services - ou programmes d'application - considérés sont réalisés hors du frontal alors que les terminaux peuvent ou non lui être directement connectés.

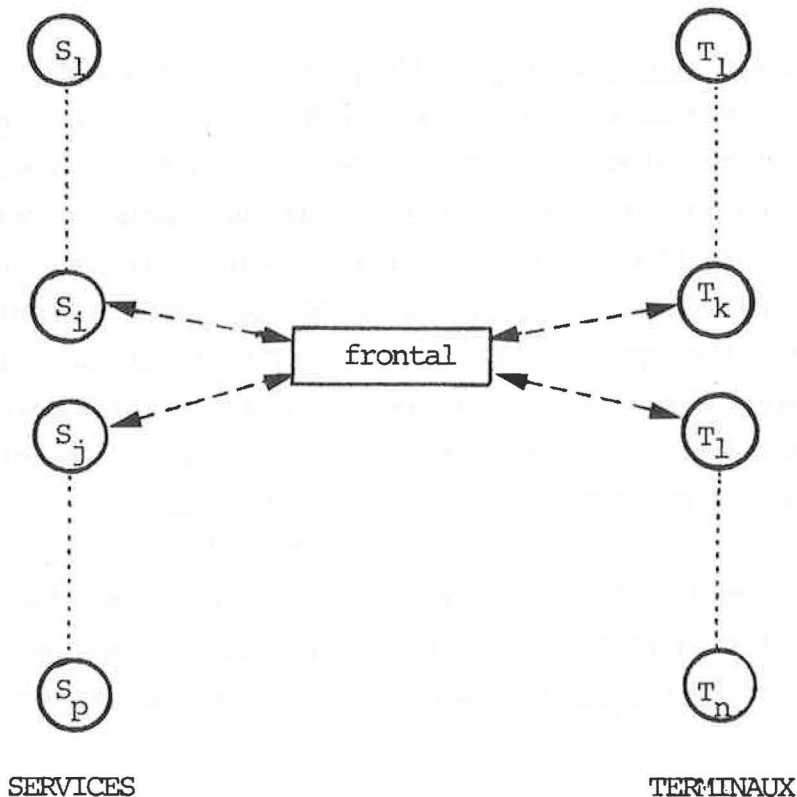


Figure 1

Un service peut mettre en jeu différents types de terminaux ou d'organes périphériques (type conversationnel, type lecteur de cartes, type imprimante etc...). A chacun de ces types de terminaux correspond une entrée dans le service. Par exemple, un service de traitement par lots disposera généralement de 3 entrées, l'une pour la console-opérateur, l'autre pour un lecteur de cartes, et la troisième pour une imprimante. Par suite, le frontal doit, plus précisément, permettre de "connecter" un terminal à une entrée fonctionnelle d'un service en leur offrant une "voie de communication" (fig. 2).

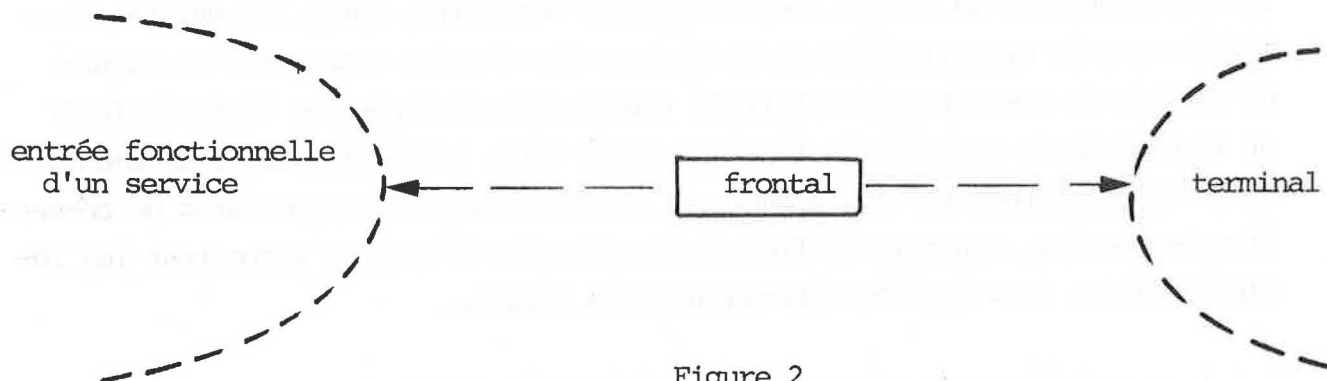


Figure 2

Vis à vis de l'ensemble des terminaux, le frontal joue un rôle de serveur virtuel qu'il remplit en gérant les accès aux terminaux et aux services et en dirigeant les opérations de contact entre eux (fig. 3)

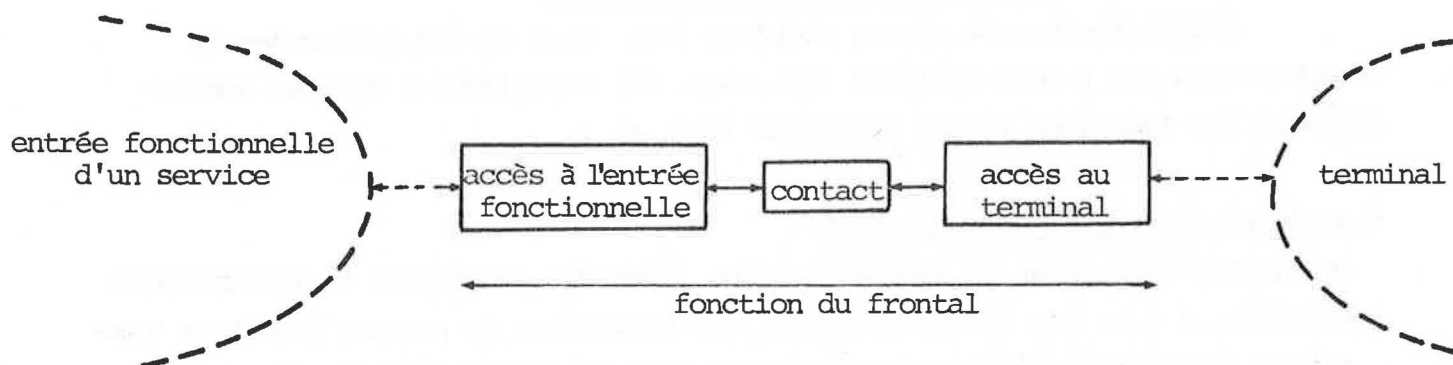


Figure 3

Remarquons que fonctionnellement, pour le frontal, un "terminal" et une "entrée fonctionnelle d'un service" constituent des entités logiques de même nature. Il leur offre des voies de communication non pas permanentes mais établies sur la base de "sessions". L'établissement et l'exploitation de ces sessions varient selon les entités terminales impliquées et leur localisation physique (cf. 5.1.2) mais le scénario de base demeure invariant. L'établissement d'une session consiste à réserver, pour la durée de la session, un certain nombre de ressources appropriées telles que tampon-mémoire, voies logiques, capacité de traitement etc... L'établissement d'une session doit également conduire à un accord, éventuellement implicite, entre les entités terminales sur la signification et la syntaxe des données échangées, la nature de la voie de communication utilisée (simplex, "half-duplex" ou "full-duplex") et la méthode de contrôle du flot sur cette voie. L'exploitation proprement dite est constituée par les transferts effectifs avec des procédures de conversion de données, contrôle de flot et récupération d'erreurs respectant les règles définies lors de l'établissement de la session.

5.1.3. - LES FONCTIONS PARTICULIERES D'ACCES AUX RESSOURCES

Les ressources à atteindre sont du type "terminal" ou "service". Les fonctions que doit assurer le frontal se déduisent de l'analyse des moyens d'accès à ces différentes ressources. Ces dernières sont de deux classes différentes selon qu'elles sont situées sur le site même ou en un autre point du réseau.

5.1.3.1. - L'accès aux ressources du réseau.

L'architecture du réseau CYCLADES (cf. 2.3) et des protocoles de communication qui y sont utilisés (cf. chap. IV) conduisent à assurer successivement les fonctions réseau suivantes (fig. 4) :

- accès au service de communication :

Le frontal doit gérer sa connexion, selon l'une des procédures de transmission offertes, à l'un des noeuds du réseau de commutation de paquets CIGALE et respecter l'interface d'échange par paquets avec ce réseau [CYC-2]. Ainsi le frontal T1600a successivement été raccordé au MITRA 15 noeud de GRENOBLE selon une procédure de transmission TMM-UC "half-duplex" puis "full-duplex" (cf. 3.2.3)

- accès au réseau CYCLADES

Le frontal doit pouvoir échanger des informations avec les entités adressables du réseau possesseurs des ressources qu'il veut atteindre : les abonnés distants. Pour ce faire, il doit posséder une Station de Transport (ST) lui assurant un service de transport d'informations entre Abonnés du réseau (AB). [CYC-4, CYC-5].

- accès aux ressources terminales :

L'accès à des services-réseau par les appareils du site nécessite la présence sur le frontal d'au moins un Abonné-client capable de contacter les abonnés-serveurs distants pour connecter les terminaux aux entrées-fonctionnelles des services désirés. Il peut y avoir un abonné-client par terminal local - solution retenue pour le concentrateur CYCLADES MITRA 15 (cf.1.1.2.2)-, ou un abonné-client unique concentrant tous les terminaux usagers locaux - solution retenue pour le frontal T1600 - Parallèlement, l'offre au réseau de services locaux nécessite la présence d'autant d'Abonnés-serveurs qu'il est offert de services. Ces abonnés serveurs jouent le rôle de serveurs virtuels vis à vis du réseau alors que les services réels sont localisés sur les ordinateurs du site connectés au frontal. Les prises de contact entre ces deux types d'abonnés doivent être conformes au protocole "Client-Serveur" [CYC-6]. Par ailleurs, les échanges de données doivent respecter le Protocole d'Appareil Virtuel (PAV) en vigueur qui standardise le format des échanges entre un service et les terminaux qui s'y connectent [CYC-7].

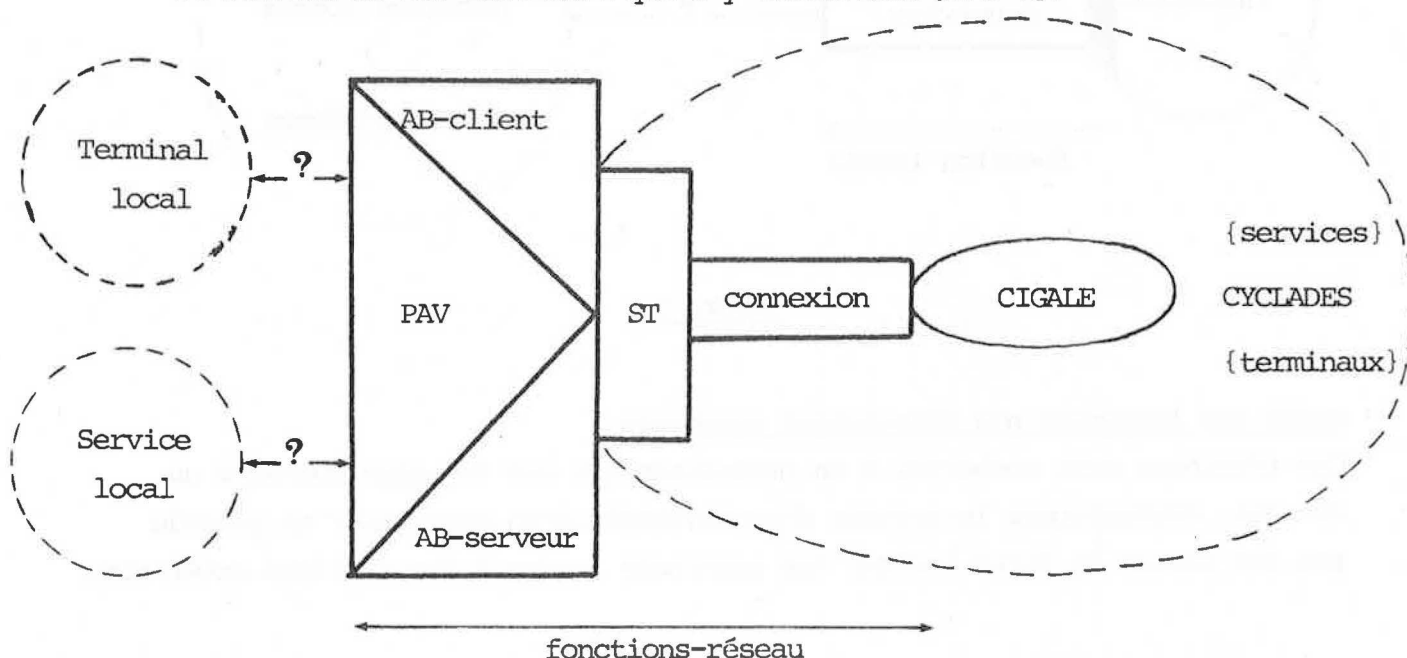


Figure 4

La localisation de ces fonctions-réseau sur le frontal évite leur duplication sur chaque ordinateur local, rendant plus rapide et moins coûteuse la connexion d'un nouvel ordinateur au réseau, ou la modification d'un niveau de protocoles sur le réseau (cf. passage d'une station de transport type 1 à une station type 2 sur CYCLADES). L'optimisation est ici réalisée au niveau de la répartition physique des fonctions logiques, qui restent identiques quel que soit le mode choisi pour le raccordement d'un ordinateur au réseau.

5.1.3.2. - L'accès aux ressources du site

Ces fonctions dépendent largement de l'"emplacement" des ressources par rapport au frontal. Nous pouvons toutefois distinguer 3 types de fonctions locales à assurer :

- accès aux appareils directement connectés :

Le frontal doit gérer la connexion de l'appareil, qui peut être un terminal ou un ordinateur (fig. 5). Cette fonction est plus ou moins complexe selon la nature de la liaison (avec ou sans modem, synchrone ou asynchrone, avec ou sans procédure de transmission etc...) et le type de l'appareil (terminal standard ou non, terminal plus ou moins intelligent etc...).

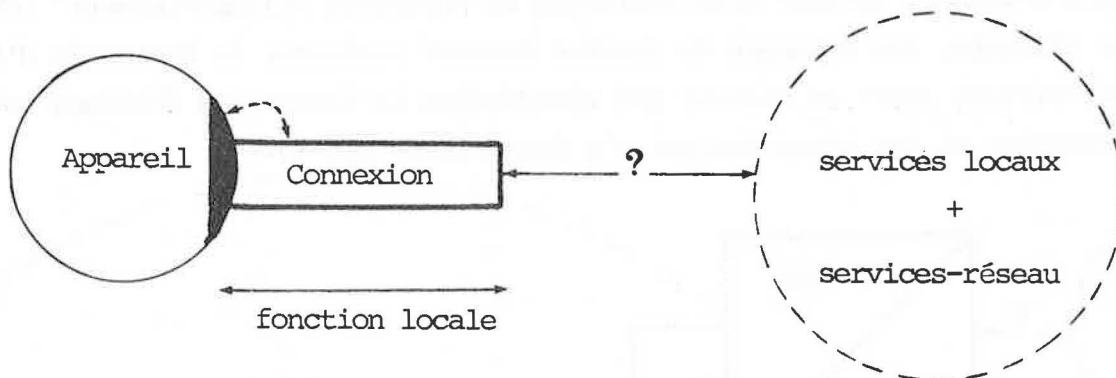


Figure 5

- accès aux terminaux non directement connectés :

Ces terminaux sont connectés à un ordinateur qui est lui-même connecté au frontal. Généralement le système d'exploitation d'un ordinateur ne possède pas les moyens de faire accéder les terminaux locaux à des services extérieurs.

Pour pallier cet inconvénient, nous avons développé un Protocole d'Accès au Frontal (PAF) permettant aux terminaux de l'ordinateur connecté d'accéder au frontal comme tout terminal directement connecté (cf. 4.5., TECH-1). Un programme (de niveau utilisateur et écrit en FORTRAN par exemple) de l'ordinateur connecté communique avec le frontal selon ce protocole et est chargé d'interfacier les terminaux réels (fig. 6). Ce protocole, inspiré mais sous une forme très simplifiée du PAV(cf. 5.1.3.1), est identique pour tous les ordinateurs locaux connectés au frontal. Les services-réseau et les services locaux situés sur d'autres ordinateurs du site apparaissent aux terminaux comme offerts par le frontal.

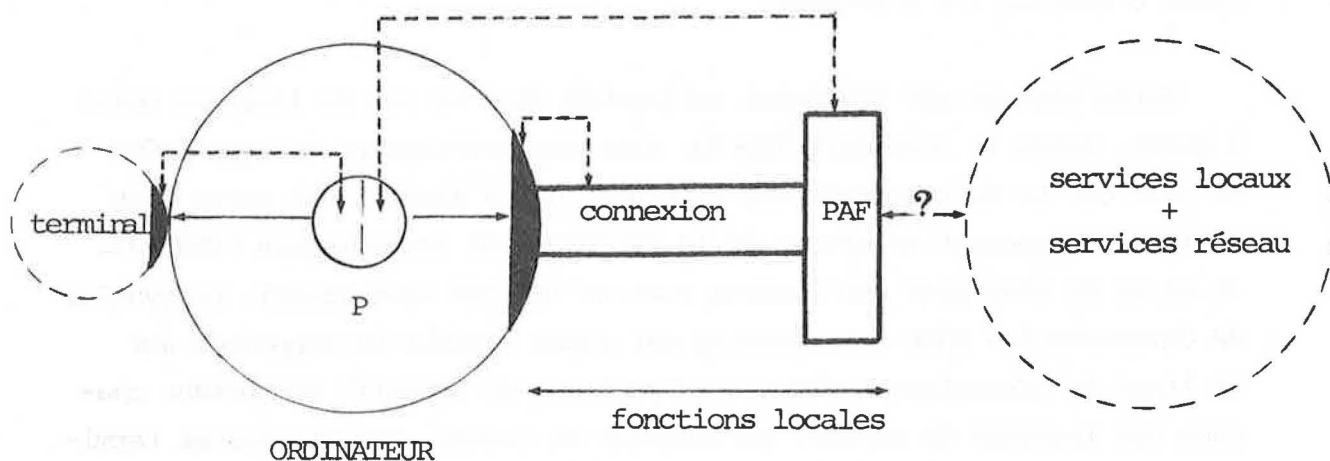


Figure 6

- accès aux services locaux :

Ces services locaux sont situés sur les ordinateurs du site directement connectés au frontal. Deux cas sont alors à considérer :

- 1°) ou bien, l'ordinateur local supporte, sur la ligne de transmission le reliant au frontal, les entrées au service qu'il offre. En ce cas, le frontal doit simuler le comportement des entrées voulues, c'est-à-dire s'adapter à un Protocole d'Accès à l'Ordinateur (PAO) pré-existant (fig. 7). Il peut simuler, par exemple, le comportement des terminaux (console-opérateur, lecteur de cartes et imprimante) d'une station éloignée d'accès à un service de traitement par lot.

Ainsi, cette simulation a été facilement réalisée pour l'accès au service de traitement par lots du P1175. En effet, son système d'exploitation supporte ce type de stations éloignées sur la ligne multipoint reliant le P1175 au frontal (cf. 3.2.2.), chaque entrée fonctionnelle du service étant représentée par une adresse sur cette ligne. Le seul inconvénient est de devoir accepter une taille des messages transférés, imposée par le système hôte, ceci pouvant représenter une forte charge pour le frontal (particulièrement pour la réception des listes d'imprimante par blocs d'environ 1/2 K octets).

Cette méthode est également utilisable dans le cas de liaisons point à point, selon la procédure TMM-RB, avec des ordinateurs du type 10070 ou IRIS 80. Le multiplexage des terminaux de la station est cette fois réalisé logiquement au niveau de la procédure de transmission [TECH-3]. Il en va de même pour une liaison avec un IBM 360 sous OS.MVT. Le système de communication HASP multileaving est alors capable de supporter sur la ligne de transmission plusieurs "stations de travail" comportant chacune des lecteurs de cartes, perforateur de cartes, imprimantes et terminaux interactifs [ARP-7].

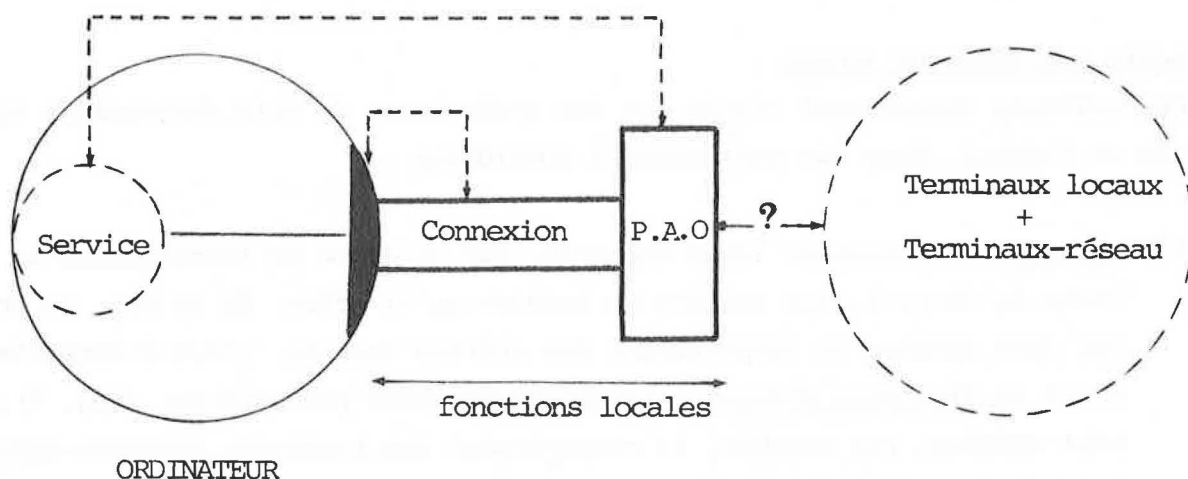


Figure 7

2°) ou bien, l'ordinateur local ne supporte pas d'entrées au service sur cette ligne de transmission. Il s'agit en ce cas de construire de toutes pièces ce protocole d'accès à l'ordinateur. Un programme, parallèle au système d'exploitation, gère ce protocole sur l'ordinateur hôte et se charge d'interfacer le service désiré. Pour le frontal, le problème reste inchangé par rapport au cas précédent. La mise en oeuvre de cette méthode sur l'ordinateur hôte, plus coûteuse que la précédente, est largement fonction des facilités offertes par son système d'exploitation. Elle présente cependant le grand intérêt de permettre au frontal de maîtriser complètement les transferts d'information (taille des messages, choix de la méthode de contrôle de flot etc...)

Nous avons primitivement employé cette méthode pour accéder au service de traitement par lot du P1175 sous une version antérieure du système d'exploitation qui ne permettait pas l'accès à distance. Le programme gérant le protocole P.A.O. sur le P1175 soumettait au service les fichiers constitués à partir des entrées fournies par le frontal. Ce procédé est facilement réalisable sur tout système où il est possible de lancer un "job" à partir d'un autre programme (généralement par l'intermédiaire d'un fichier). Par contre la récupération des listes d'imprimante correspondantes avait nécessité une modification du système d'exploitation. Enfin, il n'était même pas envisageable de "détourner" une entrée console-opérateur pour le contrôle des travaux lancés depuis le frontal.

La méthode continue à être employée pour des services "édition de textes" et "transferts de fichiers" créés de toutes pièces sur le P1175. Ces services sont réalisés au niveau de programmes utilisateurs P1175, dialoguant avec le frontal selon un protocole P.A.O., et interfaçant le système de gestion de fichiers du P1175. Ceci permet d'avoir sur le site un système unique d'édition de textes et de gestion de bibliothèques, accessible depuis tout point du site (cf. Chap. VII).

5.1.4. - SYNOPTIQUE DES FONCTIONS NECESSAIRES.

Un exemple de réalisation peut être schématisé par la figure 8. Il correspond au prototype - récemment passé en exploitation - réalisé dans le cadre de la présente étude au Centre de Calcul de l'Ecole des Mines de Saint-Etienne (cf. Annexe 1). Ce modèle est facilement extensible et les chaînes d'accès aux entités terminales (terminaux et services) peuvent être encore diversifiées.

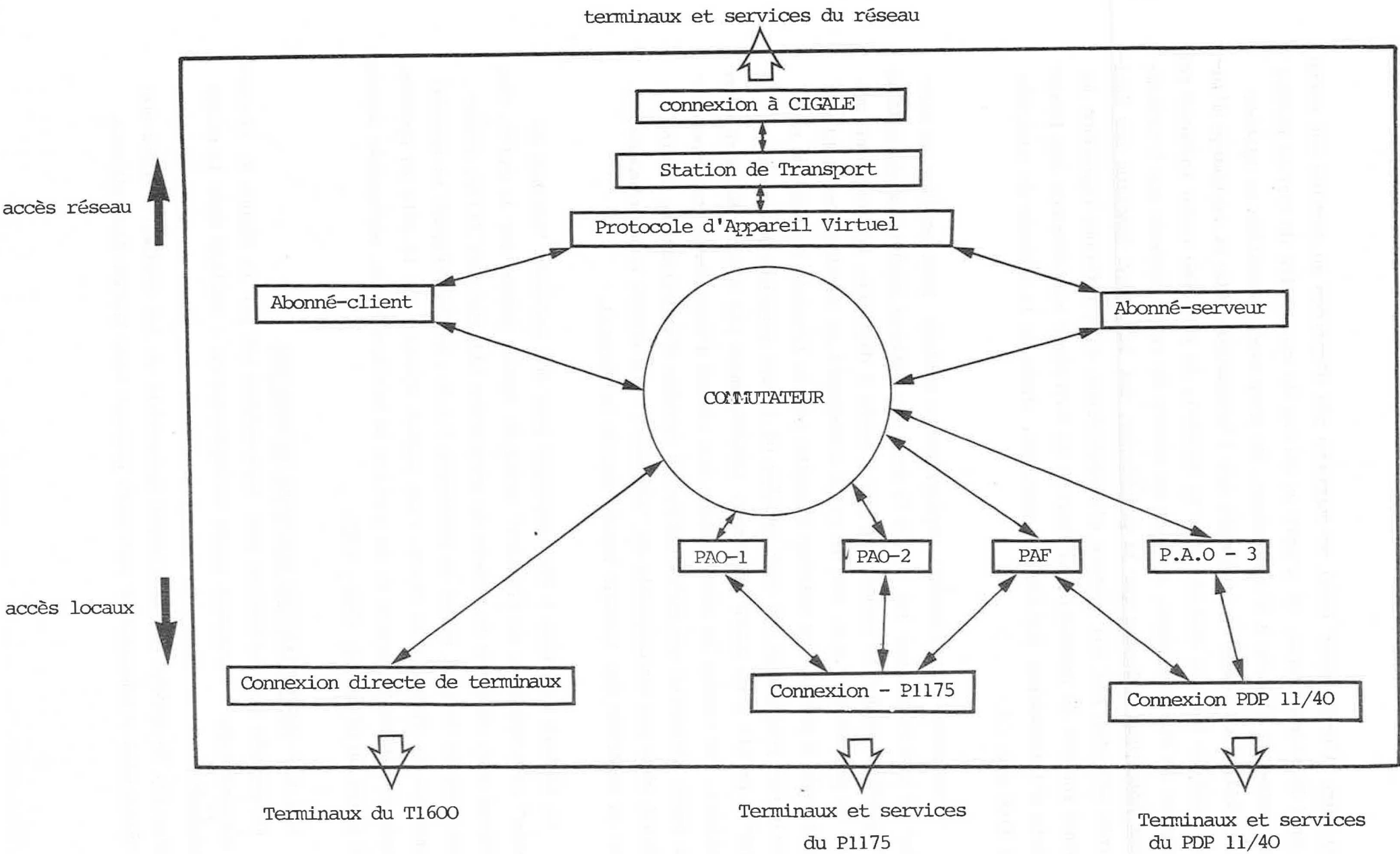


Figure 8

5.2. - ARCHITECTURE DU LOGICIEL

Nous nous intéressons dans ce paragraphe aux concepts de base retenus pour la conception du logiciel du frontal et qui nous ont conduit à le compartimenter en niveaux successifs bien distincts [GEN-1, SYS-4]. Les détails d'implémentation pratique, fonctions du matériel - et d'autres choix éventuellement discutables - sont envisagés au chapitre suivant.

5.2.1. - LA NOTION DE MACHINE

Le frontal est essentiellement non-programmable par l'utilisateur qui s'en sert comme d'une "boîte noire" dans laquelle il n'a jamais à intervenir. A ce niveau, il est suffisant d'indiquer que le terme "utilisateur" recouvre toute entité "connectée" au frontal et qui en utilise les services : le réseau, un ordinateur local, un usager à son terminal etc... Nous dirons que ce frontal est une machine aux fonctions strictement définies et limitées.

La conception de tout le logiciel repose sur cette notion de machine que nous pouvons définir de la manière suivante :

Une machine est un ensemble fonctionnel spécialisé dans l'exécution d'une fonction (plus ou moins complexe). Toute machine se situe à un niveau hiérarchique donné. Elle reçoit des commandes des machines de niveau supérieur et leur signale des évènements. Pour exécuter ces commandes, elle peut elle-même adresser des commandes aux machines de niveau inférieur dont elle est également susceptible de recevoir des évènements. Enfin, elle est de nature séquentielle et possède un état courant qui évolue depuis un état initial. (fig. 9)

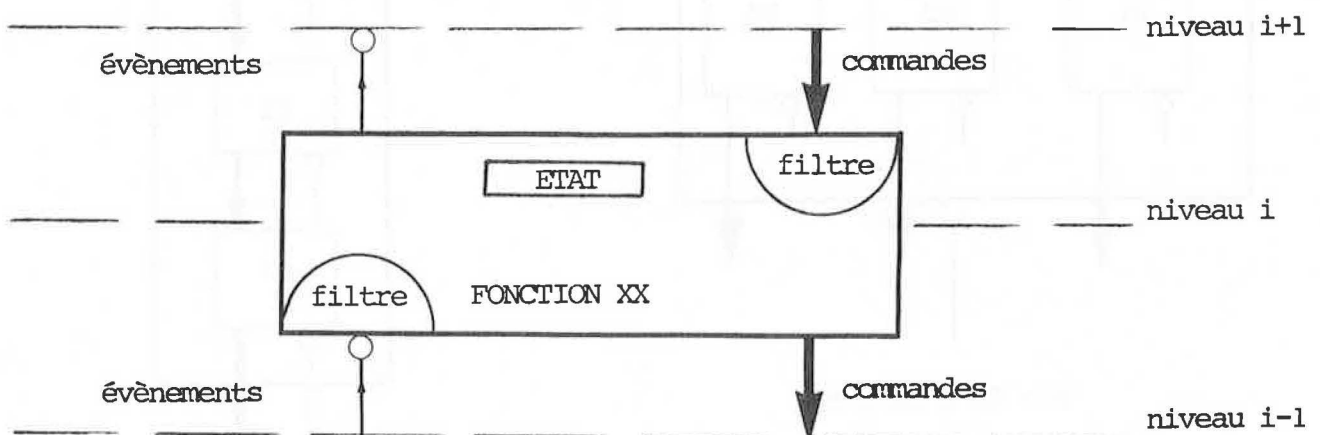
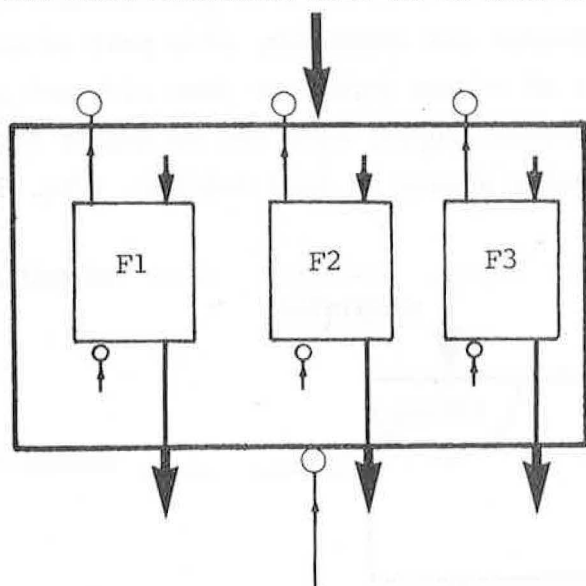


Figure 9 : machine notée M:XX

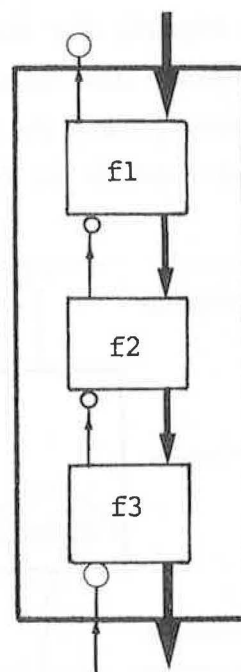
Chaque machine possède un filtre en entrée pour contrôler la validité des commandes reçues et se protéger d'un mauvais fonctionnement éventuel des machines environnantes. Toute commande invalide est rejetée. Il en va de même pour les événements non reconnus qui sont détruits. Une commande déclenche à l'intérieur de la machine commandée un mécanisme d'exécution dont le résultat est attendu par la machine origine de la commande. Ce résultat peut lui être fourni de manière synchrone par un compte rendu immédiat - commande exécutable ou non, commande exécutée, valeur de l'état observé, etc... - ou lui être transmis de manière asynchrone par un canal événement. Par contre un événement peut entraîner ou non une réaction de la part de la machine réceptrice : il peut être ignoré, être mémorisé, être traduit puis répercuté au niveau supérieur ou déclencher en retour une nouvelle commande. La machine qui signale un événement n'en attend aucun résultat.

Lorsqu'une machine réalise une fonction complexe, elle peut elle-même renfermer des machines plus élémentaires réalisant chacune un sous-ensemble de la fonction. Ces machines élémentaires s'ignorent et évoluent parallèlement au même niveau, sous le contrôle d'un aiguilleur interne à la machine englobante, lorsque la fonction globale est l'union de fonctions indépendantes de même niveau (fig. 10-a). Elles coopèrent, en propageant une action à des niveaux hiérarchiques internes successifs, lorsque la fonction globale se décompose en une série de fonctions élémentaires successives (fig. 10-b). Ces machines élémentaires sont inconnues hors de la machine englobante.



$$F = F1 \cup F2 \cup F3$$

10-a



$$F = f1 . f2 . f3$$

10-b

Figure 10 - Décomposition d'une machine

Une fonction élémentaire peut être commune à plusieurs fonctions indépendantes. Dans ce cas, elle est réalisée par une seule machine réelle - c'est-à-dire physique - sur laquelle sont multiplexées autant de machines virtuelles - c'est-à-dire logiques - que nécessaire.

L'ensemble des commandes offertes par les machines d'un niveau donné constitue un langage de programmation de ces machines qui est offert au niveau supérieur. Chaque niveau interprète le langage qu'il a défini pour le niveau supérieur. A mesure que l'on remonte vers les niveaux hiérarchiques les plus élevés, ces langages offrent des commandes - ou instructions - de plus en plus puissantes (cf. 5.3.). Corollairement, cette remontée dans les niveaux s'accompagne de ce que D.L. PARNAS [SYS-4] nomme une "perte de transparence" par rapport aux niveaux inférieurs. Ce qui veut dire que parmi les actions qu'il est possible de déclencher par programmation d'un niveau donné, certaines de ces actions ne peuvent plus être déclenchées par programmation du niveau supérieur.

5.2.2. - ORGANISATION GENERALE DE LA MACHINE "FRONTAL"

Pour jouer son rôle de serveur virtuel (cf. 5.1.2) le frontal dispose d'une machine "serveur" par service accessible. Ces machines "serveurs" sont logiquement indépendantes et évoluent en parallèle, chacune gérant ses propres usagers. Les usagers du type terminaux - et en particulier l'opérateur - peuvent passer d'un serveur à l'autre par l'intermédiaire d'un serveur "commutateur" capable de communiquer avec l'ensemble des autres serveurs. Les services réels n'étant généralement pas directement réalisés sur le frontal, chaque serveur est chargé d'établir, de contrôler puis de fermer les voies de communication entre les terminaux et les entrées fonctionnelles du service qu'il représente. Ces terminaux et entrées fonctionnelles sont uniformément désignés sous le terme d'unités symboliques. Les accès à ces unités symboliques étant généralement complexes, les serveurs sont totalement déchargés de leur gestion. Ils utilisent pour cela une machine "calculateur étendu" qui offre un sur-ensemble des possibilités de base du T1600 (fig. 11).

SERVEURS

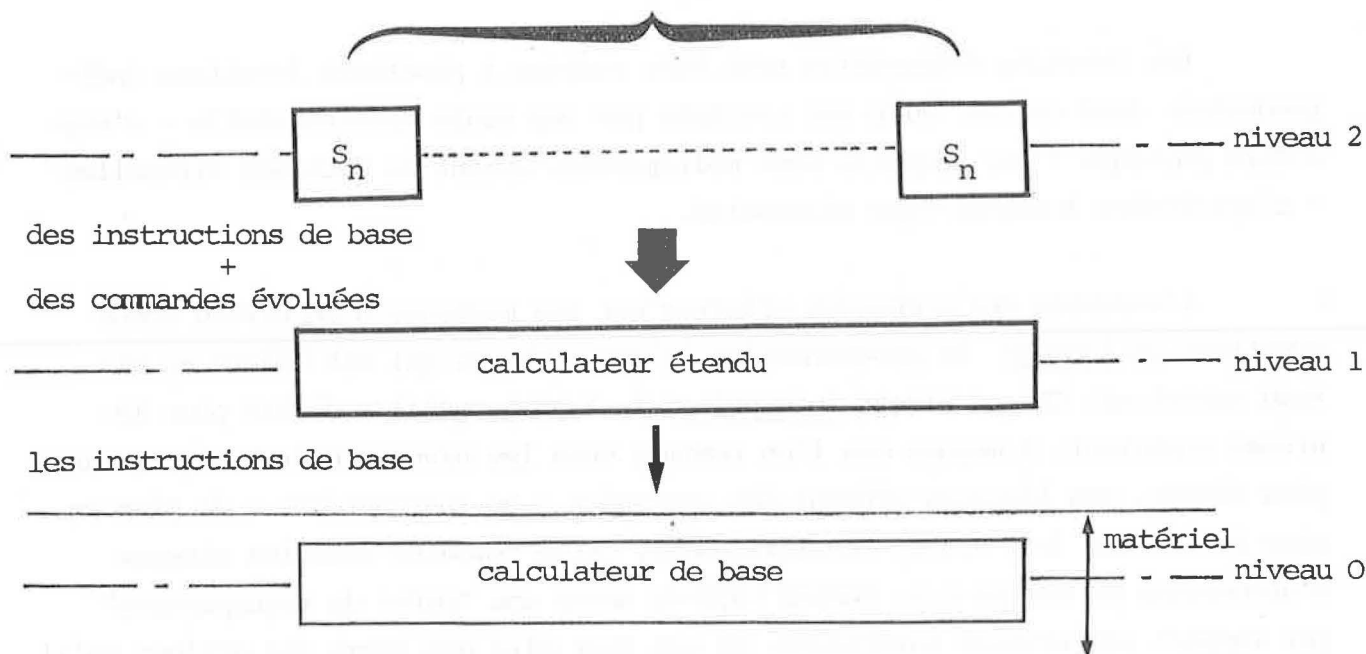


Figure 11

Le calculateur de base est commandé par l'intermédiaire des instructions appartenant au code d'ordre du T1600 et exécute des fonctions de complexité variable (transferts entre registres, entre registres et mémoire, opérations de tests, d'entrées/sorties sur le bus, allocation de l'unité centrale entre les tâches etc...). Les fonctions du type instructions de base sont elles-mêmes décomposées en fonctions plus élémentaires exécutées par des machines internes spécialisées (processeur mémoire vive, esclave d'entrée/sortie etc...) commandées par des micro-instructions. Ce calculateur de base signale des événements du type : interruption matérielle, progression du compteur ordinal d'instructions, positionnement d'indicateurs (par exemple, pour indiquer l'occupation d'un organe périphérique à la suite d'une instruction SIO) etc...

L'objectif du calculateur étendu est d'offrir, par logiciel, un certain nombre de fonctions supplémentaires complexes et d'utilisation fréquente. Ainsi, pour un calculateur T1600 ne disposant pas de l'option "scheduler micro-programmé", il aurait été intéressant de construire une machine logicielle "gestionnaire de tâches" offrant des commandes de synchronisation entre tâches. Dans notre cas, ce calculateur étendu offre aux serveurs :

- l'ensemble des fonctions du calculateur de base dont certaines sont exclues (perte de transparence) : commandes directes d'entrée/sortie sur le bus par SIO, commandes d'accès au bus général par BIO etc...
- les fonctions évoluées suivantes :
 - . émission/réception d'un message vers/à partir d'une unité symbolique (cf. 5.3).
 - . gestion dynamique de la mémoire principale (avec obtention de tampon-mémoire à la demande)
 - . gestion d'une mémoire auxiliaire sur disque (avec diverses méthodes de stockage temporaire)
 - . gestion de files d'attente
 - . gestion de réveils
 - . fonctions élémentaires diverses (décalage, transfert, transcodage de zones d'octets, positionnement de bits dans une file de bits etc...)

Ces possibilités du calculateur étendu facilitent grandement l'introduction de tout nouveau serveur.

Les serveurs sont divers et de complexité variable :

- service de commutation et d'entrée dans le système ("LOGGER")
- concentrateur de terminaux client de CYCLADES
- serveur sur le site du traitement par lot du P1175
- serveur "éditeur de textes du site"
- serveur "transfert de fichiers" sur le site
- serveur de l'APL local sur CYCLADES (en préparation)
- serveurs spécifiques d'accès à des programmes utilisateurs particuliers sur le P 1175 :
 - . acquisition de données sur ruban papier
 - . visualisation de courbes sur écran graphique

5.3. - L'EXEMPLE DE LA MACHINE DE COMMUNICATION

Nous nous intéressons dans ce paragraphe à la machine la plus complexe du calculateur étendu (cf. 5.2.2.) : la machine de communication, composant fondamental du logiciel du frontal. Nous montrons sur cet exemple comment ont été mis en pratique les principes de construction du logiciel exposés au paragraphe 5.2. Dans la suite cette machine sera notée M:DM (pour machine "Data Management").

5.3.1. - LA VISION DES "SERVEURS"

Les serveurs virtuels du frontal échangent des données avec les usagers et les services réels extérieurs qu'ils mettent en relation. La machine M:DM combine les fonctions de gestion et de contrôle des données et les fonctions de transmission de ces données. Son but est d'isoler les serveurs du frontal de tous les problèmes de communication liés à la localisation des usagers et des services réels à atteindre. Ces derniers sont représentés par des unités symboliques auxquelles chaque serveur a des accès standards. Les particularités d'accès aux entités finales et les détails de transmission en résultant sont totalement à la charge de la machine M:DM (fig. 12).

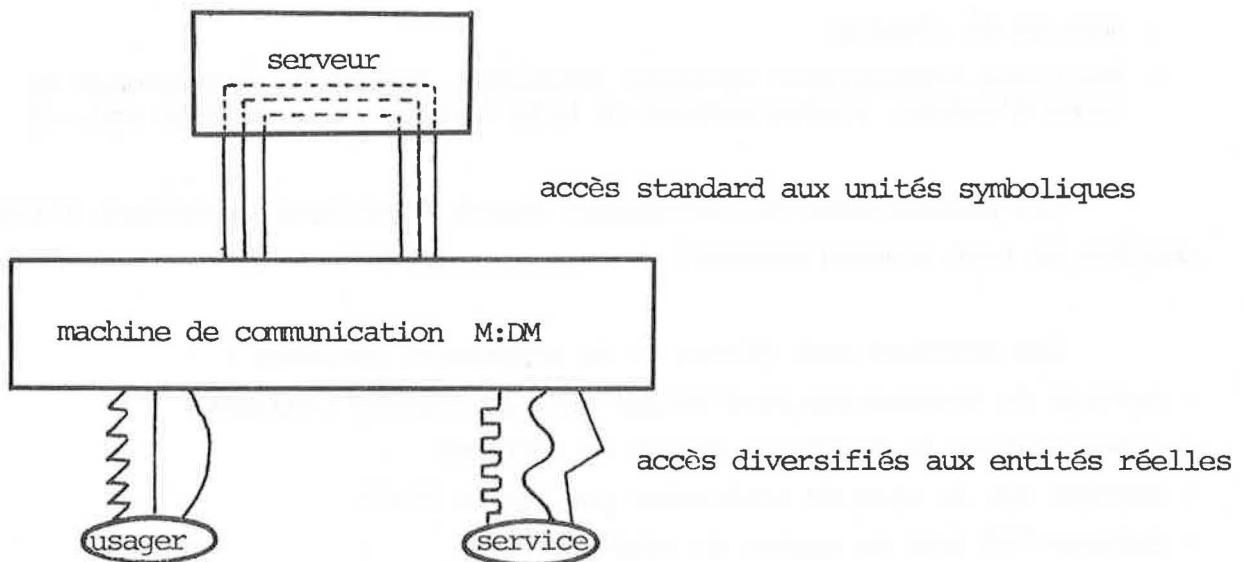


Figure 12

- les unités symboliques (notées SU) peuvent être classées en :

SU Locales : Chaque SU correspond à un terminal local ou à une entrée fonctionnelle d'un service local. Une SU locale est identifiée par un numéro unique choisi arbitrairement à la génération du système .

SU réseau : Rappelons que sur le réseau, les échanges d'information se font entre une "porte" d'une Station de Transport et une porte d'une autre station de transport, l'association de ces 2 portes constituant un "flot" (cf. ST type 2 chap. IV). Chaque porte est elle

même associée localement à un terminal ou à une entrée fonctionnelle de service. Par suite, il existe des SU de type porte et des SU de type flot.

Les SU de type porte correspondent aux portes-réseau du frontal. Elles servent à recevoir du réseau (la porte émettrice pouvant être quelconque) et à initialiser un flot avec une porte de station de transport distante. Une SU-porte est identifiée par un préfixe Cyclades (CYC) concaténé à un numéro local de porte (PT) : CYC/ID-PT.

Les SU de type flot correspondent aux flots existants, établis au préalable par l'intermédiaire d'une SU-porte. Une SU flot est identifiée par un préfixe CYCLADES (CYC) concaténé à un numéro local de flot (FL), attribué dynamiquement par M:DM à l'ouverture du flot : CYC/ID-FL.

- La figure 13 schématise l'environnement de la machine M:DM

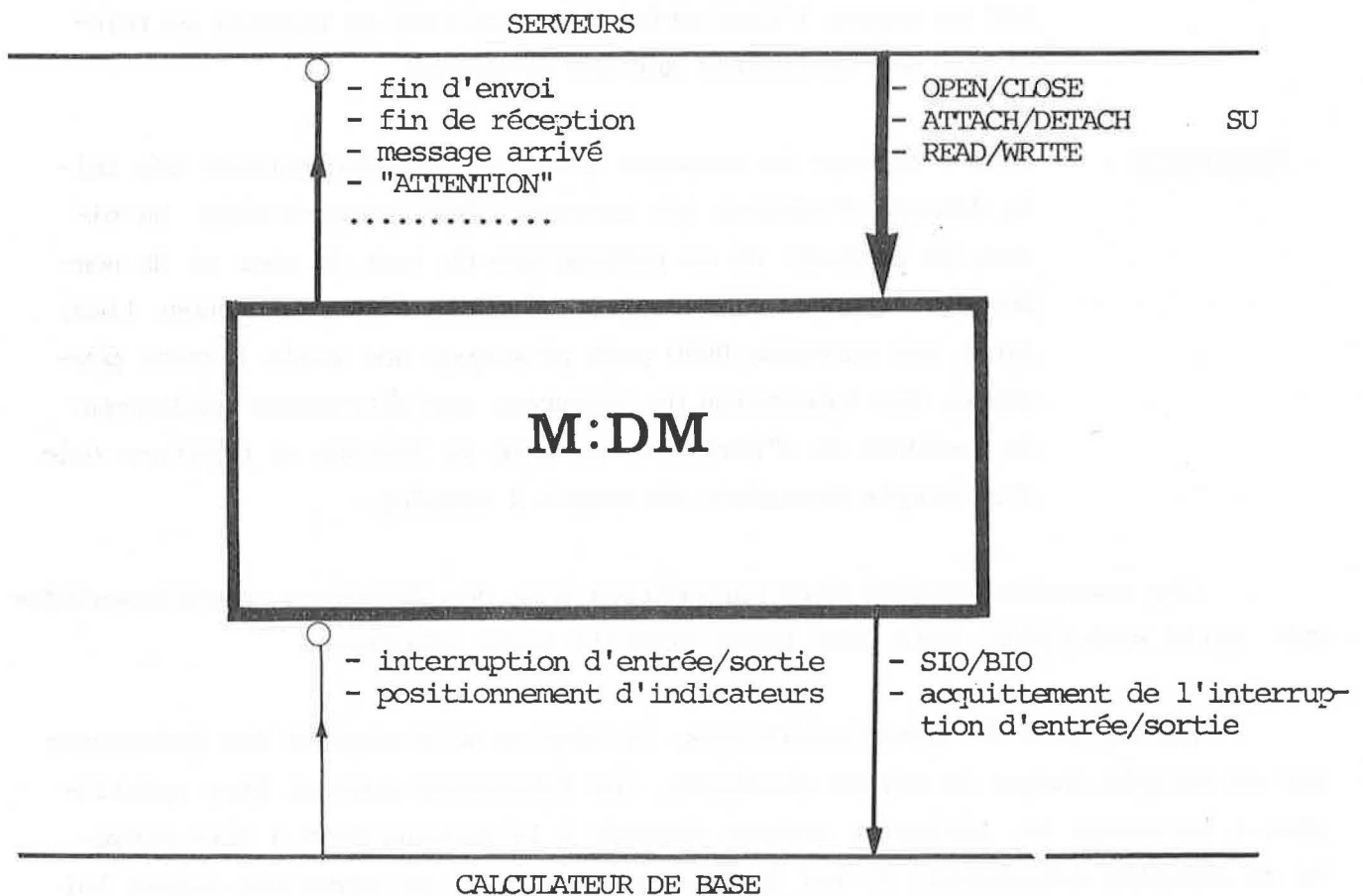


Figure 13

Les commandes suivantes sont offertes aux serveurs :

- . OPEN/CLOSE : Pour l'ouverture/la fermeture d'une SU. Ainsi l'ouverture d'une SU provoque l'initialisation d'une voie logique à travers la machine M:DM, l'activation des machines élémentaires associées à cette voie logique et la réservation des ressources correspondantes notamment en tampon-mémoire. Les mécanismes d'ouverture et de fermeture sont de complexité variable selon les SU concernées. Dans les cas les plus complexes (SU-réseau) ils peuvent même conduire à des échanges d'information avec l'extérieur.

- . ATTACH/DETACH : Pour l'attachement/détachement d'une SU à un serveur. L'attachement permet en particulier de répercuter jusqu'au serveur attaché un certain nombre d'évènements, par exemple une "ATTENTION" provoquée par un usager sur un terminal interactif ou encore l'indication d'une arrivée de message ou télégramme non sollicités sur une SU-réseau.

- . READ/WRITE : Pour l'échange de messages avec une SU. Bien entendu une telle demande d'échange est macroscopique (c'est-à-dire du niveau du serveur) et ne préjuge pas du tout du sens ni du nombre des échanges élémentaires auxquels elle peut donner lieu. Ainsi une commande READ peut provoquer une seule lecture physique, une succession de lectures, une alternance quelconque de lectures et d'écritures ou même ni lecture ni écriture (cas d'un simple transfert de tampon à tampon).

Ces commandes peuvent être paramétrées pour des fonctions spécifiques à chaque unité symbolique, mais leur forme générale reste identique.

Enfin, sous certaines conditions, la machine M:DM signale des évènements par un certain nombre de moyens standards, ces évènements pouvant être spécifiques à certaines SU. Lorsqu'un serveur demande à la machine M:DM à être prévenu de certains évènements, il lui fournit en même temps un moyen par lequel lui passer ces évènements. Ainsi une commande READ ou WRITE peut ou non indiquer

un moyen d'être prévenu de la fin de réception ou de la fin d'envoi. A la réception de cet évènement le serveur peut observer de manière standard le compte rendu d'exécution de sa commande (état de sa machine M:DM pour la SU concernée).

5.3.1. - L'ARCHITECTURE INTERNE :

Comme nous l'avons mis en évidence au chapitre 5.1.3, l'accès aux unités symboliques est hiérarchisé. Cette hiérarchie est globalement constituée des 3 niveaux fonctionnels suivants :

- un premier niveau permet d'accéder aux coupleurs utilisés par le frontal,
- un second niveau permet d'accéder aux appareils raccordés à ces coupleurs par une ligne de transmission,
- un troisième niveau permet d'accéder aux unités symboliques situées sur ces appareils.

Cette hiérarchie est reflétée par les niveaux d'interprétation internes successifs de la machine M:DM (fig. 14).

5.3.2.1. - Le niveau technologique (niveau 0)

C'est le niveau primitif de notre construction, fourni par le matériel du constructeur.

- évènements signalés : - interruption d'entrée/sortie, positionnement d'indicateurs (pour coupleur libre, occupé, etc...).
- commandes offertes : - instructions SIO (accès à l'esclave d'E/S). BIO (accès au bus général).
 - reconnaissance du niveau de l'interruption produite, acquittement de cette interruption.

5.3.2.2. - La machine-interruption (niveau 1.0)

La machine-interruption (M:IT) est chargée des relations de M:DM avec le système matériel d'entrée/sortie. En particulier, elle gère complètement l'interruption d'E/S qu'elle démultiplexe en identifiant le coupleur originai-
re de l'appel. Elle détermine par ailleurs (cf. 6.3.1) si le niveau supérieur

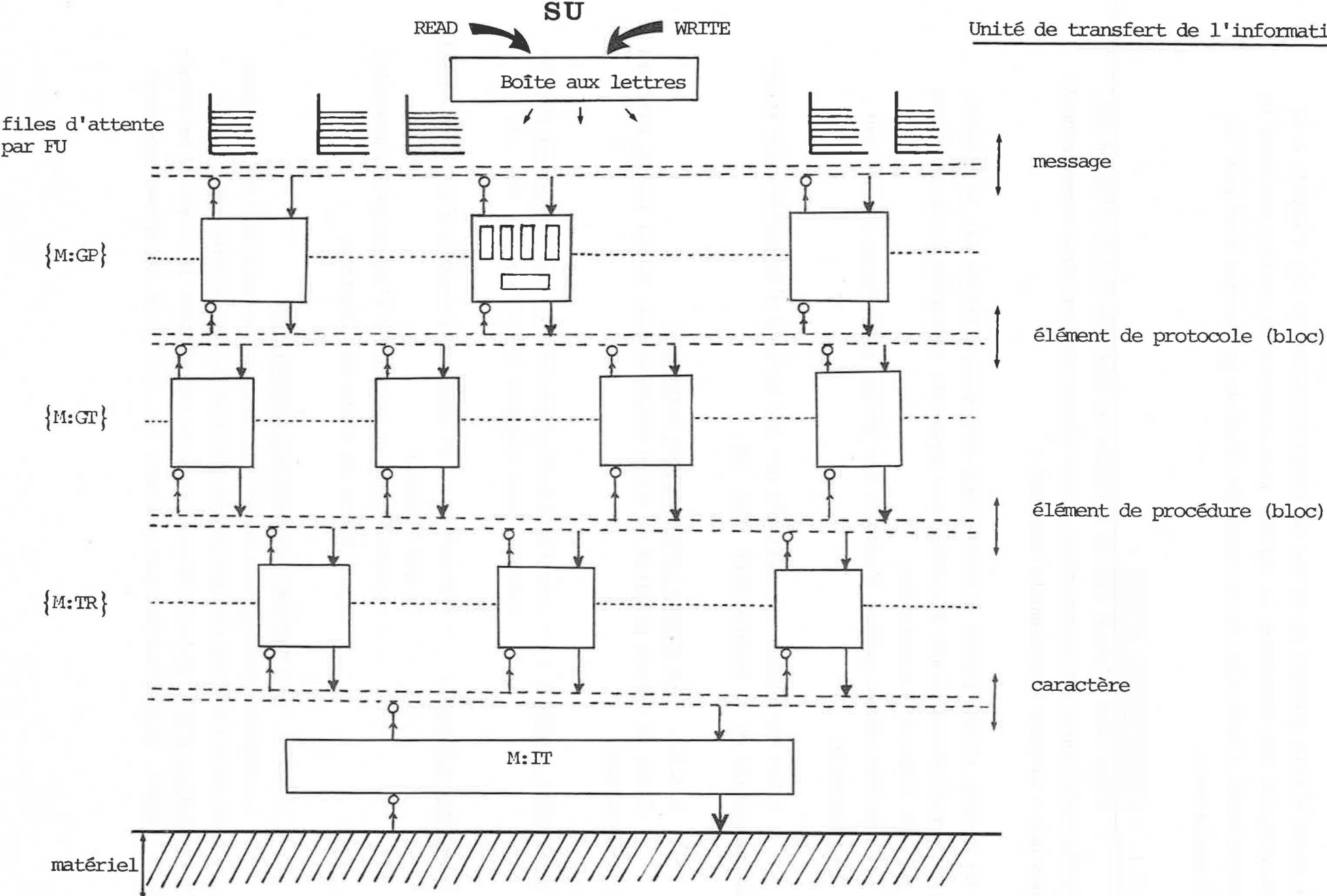


Figure 14

doit être activé immédiatement (coupleur de ligne synchrone rapide) ou si cette activation peut être différée dans le temps (coupleur de ligne asynchrone lente).

- Evènements signalés : interruption de tel coupleur d'E/S, positionnement d'indicateurs
- Commandes offertes : SIO et BIO du niveau précédent

Ce niveau introduit une première perte de transparence pour le niveau supérieur (cf. 5.2.1) en ce qui concerne la prise en compte et le traitement des interruptions.

5.3.2.3. - Les machines de transmission (niveau 1.1)

Les machines de transmission (M:TR) gèrent chacune un type de coupleur d'entrée/sortie, plusieurs machines virtuelles pouvant être multiplexées sur une seule machine réelle dans le cas de coupleurs identiques. Une machine M:TR contrôle l'ensemble du support physique de transmission constitué du coupleur, du modem et de la ligne elle-même. Pour une machine M:TR donnée, une certaine paramétrisation du support est permise : transmission avec ou sans modem, type de modem, nature de la ligne (half duplex ou full duplex). Seul le coupleur a des caractéristiques fixées. Ces machines M:TR sont chargées d'émettre et de recevoir des blocs de caractère - généralement en mode caractère par caractère - sans intervenir au niveau de la syntaxe ni de la sémantique de ces blocs. Elles assurent en outre des fonctions spéciales diverses (connexion, déconnexion des ensembles coupleurs/modems, armement de chiens de garde, trace de l'activité sur la ligne etc...) et détectent les erreurs dues au médium de transmission (défauts du modem, perte de synchronisme, écoulement de chiens de garde, altération des données avec code de redondance reçu non valide etc...)

- évènements signalés :
 - connexion/déconnexion effective de la ligne
 - anomalies matérielles
 - réception d'un bloc de caractères
 - erreurs diverses en réception ("time-out" etc...)
 -

- commandes offertes : - connexion/déconnexion physique de la ligne
- émission/réception de blocs
- armement de chiens de garde
-

Ce niveau introduit une transparence complète vis à vis du système matériel d'entrée/sortie du calculateur, des coupleurs utilisés et de leur mode de fonctionnement (mode bloqué ou mode-caractère).

Exemples de machines M:TR réalisées :

- gestion d'un coupleur asynchrone half-duplex, mode caractère
- gestion d'un coupleur asynchrone full-duplex, mode caractère
- gestion d'un coupleur asynchrone multiplexé (8 voies) full-duplex, mode caractère
- gestion d'un coupleur synchrone full-duplex, mode caractère
- gestion d'un coupleur accès direct mémoire, mode micro-canal.

5.3.2.4. - Les machines de gestion des transmissions (niveau 1.2.)

Les machines de gestion des transmissions (M:GT) gèrent chacune l'accès à un appareil directement connecté (terminal, ordinateur ou noeud de raccordement à un réseau). Chaque machine M:GT est chargée des échanges de données avec un appareil connecté en respectant une procédure de transmission qui fixe les conventions d'échange à travers le médium de transmission. Cette procédure de transmission est simple si l'appareil est un organe périphérique local (télétype, lecteur de cartes, imprimante, etc...) Elle se résume alors à la prise en compte de ses particularités physiques. Elle est plus complexe lorsque l'appareil est un ordinateur connecté par liaison téléphonique point-à point ou multipoint. Dans ce dernier cas, les données doivent être insérées dans une enveloppe (en-tête et queue) définie par la procédure, et les échanges doivent être insérés dans le dialogue de procédure. La machine M:GT se charge en particulier de l'initialisation des statuts (maître et esclave) sur la ligne, de l'acquiescement des transferts et des répétitions en cas d'erreur. Un échange de données commandé à une machine M:GT se traduit alors par plusieurs échanges de blocs de caractères sur la ligne mais ces échanges sont transparents au niveau supérieur.

- évènements signalés : - appareil connecté/déconnecté
 - fin d'émission de données/réception de données
 - erreurs de transfert (après un certain nombre de répétitions)

- commandes offertes : - connexion/déconnexion de l'appareil
 - émission/réception de données

Ce niveau introduit une transparence complète vis à vis du mode de raccordement aux appareils (liaison directe ou téléphonique, point à point ou multipoint, procédures différentes etc...).

Remarque :

Dans le cas d'une liaison multipoint avec un ordinateur, le frontal peut reconnaître un certain nombre d'adresses sur cette ligne (cf. chap. III). La machine M:GT qui gère la procédure de ligne est alors multiplexée en autant de machines virtuelles que d'adresses reconnues, chaque adresse se présentant au niveau supérieur comme un appareil.

Exemple de machines M:GT réalisées :

- . gestion de la procédure point à point TMM-UC half-duplex
- . gestion de la procédure point à point TMM-UC full-duplex
- . gestion de la procédure multipoint ECMA-PHILIPS half-duplex
- . gestion des accès aux organes périphériques directement connectés.

5.3.2.5. - Les machines de gestion des protocoles (niveau 1.3)

Les machines de gestion des protocoles (M:GP) gèrent chacune l'accès à une unité symbolique située sur un appareil connecté. Chaque machine M:GP est chargée des échanges de messages avec une unité symbolique en respectant un protocole de dialogue. Il s'agit ici des messages directement émis ou reçus par un serveur virtuel du frontal. Le protocole contrôle l'exploitation de la voie logique établie entre l'unité symbolique et le serveur.

Ce protocole est très simple s'il s'agit d'un organe périphérique local. Il se borne alors à écouter les demandes d'émission/réception du serveur selon un séquençement strict (pas de recouvrement) avec une possibilité

d'interruption venant d'une unité symbolique de type terminal interactif ("break" ou "attention" de l'utilisateur).

Ce protocole est un peu plus complexe dans le cas d'unités symboliques situées sur un ordinateur directement connecté. Ainsi, il peut consister à simuler (cf. 5.1.3.2) une console interactive, un lecteur de cartes ou une imprimante sur une adresse de la ligne multipoint reliant le frontal au P 1175. Il s'agit alors de présenter (ou de recevoir) les messages sous une forme spécifique à l'appareil simulé.

Dans le cas extrême de la gestion du protocole de transport CYCLADES, qui gère l'accès aux SU-réseau, la machine M:GP (Station de Transport cf. Chapitre IV) est très complexe. Elle fait à son tour l'objet d'une décomposition en machines plus élémentaire. Une machine (M:IR) interface le réseau de commutation de paquets CIGALE, au niveau des paquets échangés. Elle est utilisée par une série de machines de gestion de flots (M:FL) et de gestion de portes (M:PT), à raison d'une machine par flot et d'une machine par porte, pour l'échange de lettres et de télégrammes avec les SU-réseau (portes et flots du réseau). Les échanges de messages commandés à la machine M:GP se traduisent alors par plusieurs échanges de données sur le réseau, avec éventuellement fragmentation/réassemblage de ces messages sur une mémoire auxiliaire (disque). Ces opérations intermédiaires sont transparentes au niveau supérieur.

- évènements signalés :
 - unité symbolique ouverte/fermée
 - fin d'émission de message
 - réception d'un message
 - erreurs de transfert (après un certain nombre de répétitions)
 - "ATTENTION" sur un terminal/télégramme du réseau
 -

- commandes offertes :
 - ouverture/fermeture d'une unité symbolique
 - émission/réception de messages (lettres et télégrammes sur le réseau)

Ce niveau introduit une grande transparence vis à vis de la localisation (et donc de leur mode d'accès) des unités symboliques connues des serveurs. Toutefois, cette transparence n'est pas complète puisque nous n'avons pas intégré le Protocole d'Appareil Virtuel (PAV) CYCLADES dans la machine de communication. Cette intégration nous a semblé difficile à réaliser et aurait nécessité une extension générale assez lourde d'un format "Appareil Virtuel" pour l'ensemble des unités symboliques (et non plus simplement pour les unités symboliques de type réseau). Le PAV est donc géré au niveau des serveurs du frontal lorsqu'il y a lieu.

Exemples de machines M:GP réalisées

- . gestion du protocole de transport CYCLADES
- . gestion du protocole d'accès au frontal (PAF) pour les terminaux d'un ordinateur connecté au frontal (cf. 5.1.3.2).
- . gestion des protocoles d'accès aux ordinateurs connectés (PAO) pour des entrées du type lecteur de cartes, imprimante, console-opérateur, terminaux spécifiques à une application (cf. 5.1.3.2).
- . gestion des terminaux locaux au frontal

5.3.2.6. - La boîte aux lettres (niveau 1.4)

La machine de communication possède au niveau supérieur une boîte aux lettres qui reçoit directement - après une première validation - les commandes des serveurs. Cette boîte aux lettres est constituée en files d'attente où viennent s'approvisionner, en parallèle, les machines M:GP, l'approvisionnement étant séquentielle pour une machine M:GP donnée. Les files d'attente - de type LIFO - sont constituées par unité fonctionnelle (ou FU). A chaque unité symbolique peuvent correspondre :

- 1 FU fonctionnant aussi bien en émission qu'en réception si la SU est du type half-duplex.
- 1 FU fonctionnant seulement en émission (réception) si la SU est du type simplex réceptrice (émettrice).
- 1 FU émission et 1 FU réception si la SU est du type full-duplex.

Un mécanisme unique de tables permet de connaître la série des machines (M:GP, M:GT, M:TR) associées à une voie logique donnée. La figure 15 donne un aperçu de l'effet multiplicateur introduit par ces niveaux d'interprétation successifs quant au nombre des échanges induits par une seule commande de transfert d'un serveur.

5.3.3. - FLEXIBILITE ET EXTENSIBILITE DE LA REALISATION

5.3.3.1. - Flexibilité

Nous entendons par flexibilité la possibilité de modifier facilement la réalisation d'une fonction existante (pour des raisons de coût économique par exemple) sans perturber le service offert. Cette flexibilité peut ici être mise en évidence à 2 niveaux :

- Introduction de processeurs d'entrée/sortie spécialisés : L'architecture de la machine de communication facilite les compromis entre logiciel et matériel pour la réalisation d'une fonction donnée. Or ces compromis, qui dépendent de l'état de la technologie à un instant donné, ne peuvent qu'évoluer dans le temps et particulièrement en faveur du matériel car la chute des prix dans le domaine des composants n'a pas d'équivalent dans le domaine du logiciel. En particulier pour le frontal, un gain considérable serait obtenu par la décentralisation, au niveau de processeurs périphériques spécialisés, de la gestion des transmissions de bas niveau. Par exemple l'introduction sur une ligne donnée d'un coupleur capable de fonctionner en mode bloqué (ou mode canal) permettrait de faire l'économie des traitements, par le processeur central, de chaque caractère échangé sur la ligne. Cela nécessiterait simplement la modification de la machine M:TR correspondante dont l'essentiel de fonctions serait reporté au niveau matériel. L'introduction d'unités micro-programmées capables de gérer une procédure de transmission pour une ligne synchrone présente un intérêt encore plus évident. Ces unités seront même indispensables pour les nouvelles procédures de transmission, type HDLC [PRO-2], normalisées par l'I.S.O., pour lesquelles la gestion s'effectue au niveau du bit et n'est donc pas envisageable au niveau logiciel. Le choix de la répartition physique du contrôle d'une ligne - entre processeur central et processeur périphérique - est possible par simple modification des machines M:TR et M:GT correspondantes. Nous pouvons même dire que ces machines restent identiques sous un angle purement fonctionnel quelle que soit cette répartition. Seules sont modifiées les parts respectives du matériel et du logiciel dans la réalisation de chacune d'entre elles.

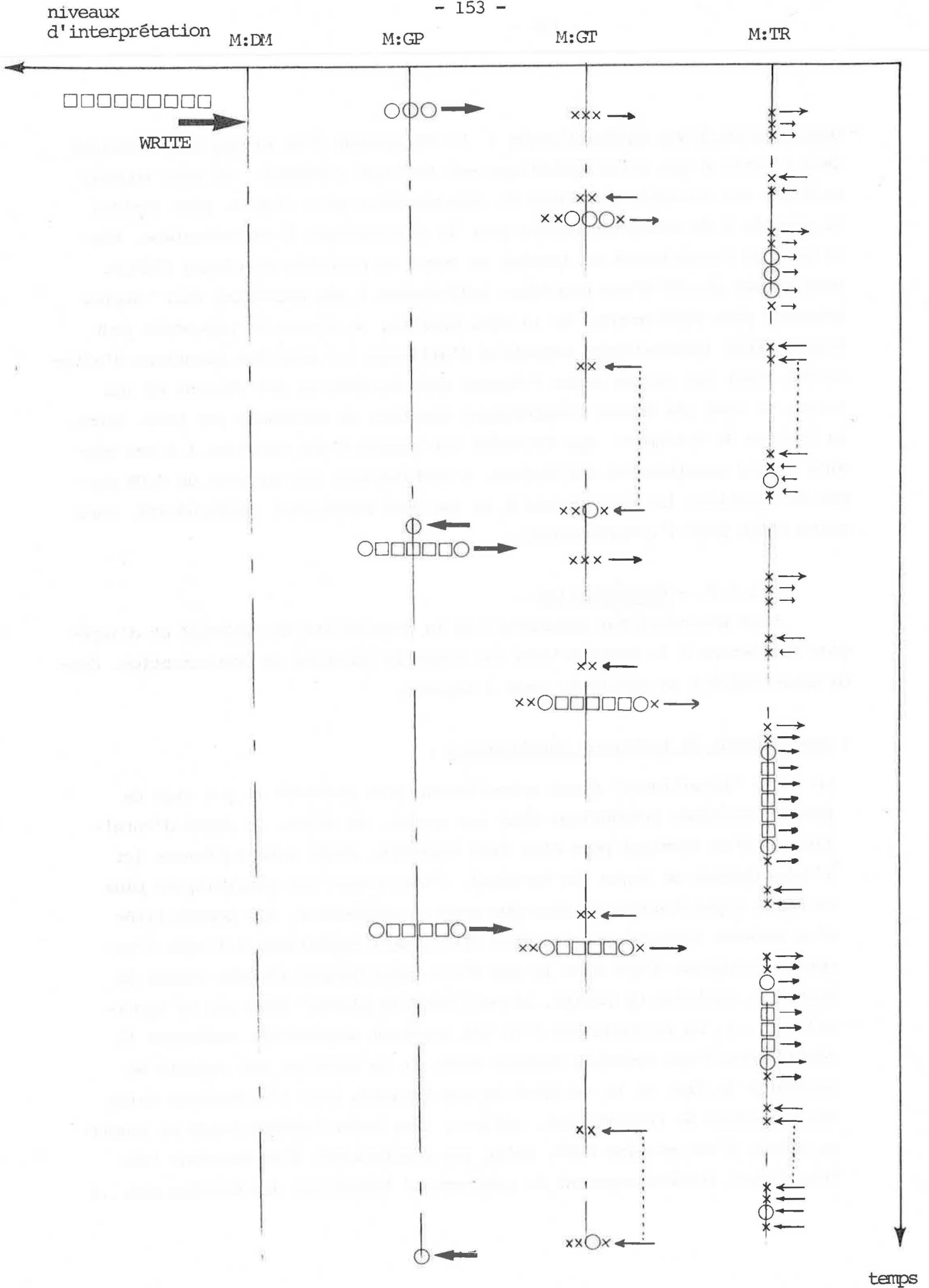


Figure 15': Effet multiplicateur des échanges

- Modification d'une chaîne d'accès : Le changement d'un niveau intermédiaire dans l'accès à une unité symbolique est toujours possible. On peut vouloir utiliser une nouvelle procédure de communication plus fiable, plus rapide, ou adaptée à de nouvelles normes pour le raccordement à un ordinateur. Ainsi pour le raccordement du frontal au noeud de Grenoble du réseau CIGALE, nous sommes passés d'une procédure half-duplex à une procédure full -duplex beaucoup plus performante. De la même manière, un niveau de protocole peut être modifié (généralement simplifié d'ailleurs...) pour des questions d'efficacité. Ceci est encore assez fréquent dans le domaine des réseaux où les normes ne sont pas encore complètement établies et reconnues par tous. Ainsi, la Station de Transport sur CYCLADES est passée d'une version 1 à une version 2 très sensiblement différente. L'architecture par niveaux de M:DM permet de localiser les changements à la (ou aux) machine(s) concernée(s), sans autre effet pour l'environnement.

5.3.3.2. - Extensibilité.

Nous entendons par extensibilité la possibilité de réaliser et d'intégrer facilement à la machine M:DM une nouvelle fonction de communication. Cette extensibilité se manifeste sous 3 aspects.

- raccordement de terminaux intelligents :

Le terme "intelligent" étant actuellement très galvaudé il y a lieu de prendre quelques précautions dans son emploi. En effet, le degré d'intelligence d'un terminal peut être très variable. Seule nous intéresse ici "l'intelligence en ligne" du terminal, c'est-à-dire ses possibilités plus ou moins sophistiquées de dialogue avec un ordinateur. Les possibilités d'un nouveau terminal peuvent être directement exploitées à l'aide d'un nouveau coupleur d'E/S dans le cas d'une intelligence limitée (codes de contrôle, contrôle de parité, signal "mémoire pleine" émis par le terminal etc...). Le raccordement d'un tel terminal nécessitera seulement la réalisation d'une nouvelle machine M:TR. Si le terminal est capable de contrôler le flot et la validité de ses échanges avec l'ordinateur selon une procédure de transmission, celle-ci sera naturellement prise en compte au niveau d'une machine M:GT. Enfin les possibilités d'un terminal très intelligent (télé-chargement de programmes, formatage des données etc...)

pourront être pris en compte au niveau d'une machine M:GP. Cette extensibilité de la machine de communication est essentielle si l'on veut profiter de la prolifération actuelle de ce type de terminaux.

- raccordement de nouveaux ordinateurs-hôtes :

De tels raccordements impliquent l'accès à de nouvelles unités symboliques (cf. 5.1.3.2). Or à chaque niveau d'interprétation (TR, GT, GP) de la machine de communication, il existe une bibliothèque de machines disponibles. L'addition d'une nouvelle unité symbolique consiste à définir sa chaîne d'accès et à lui associer les machines correspondantes à partir des bibliothèques existantes. Si les fonctions nécessaires à un certain niveau ne sont assurées par aucune des machines de ce niveau, il suffit (ce qui ne veut pas dire que ce soit simple...) de construire une nouvelle machine et de l'inclure à la bibliothèque. Les problèmes de mise au point sont ainsi considérablement localisés.

Ainsi, si l'on connecte un nouvel ordinateur au frontal :

- . on utilisera vraisemblablement la machine M:TR existante capable de gérer un coupleur synchrone.
- . si la procédure de transmission utilisée par cet ordinateur est déjà gérée par le frontal, on utilisera la machine M:GT correspondante ; sinon on en développera une nouvelle.
- . le protocole d'accès au frontal (PAF) restera identique. Il faudra le gérer par un programme-utilisateur sur l'ordinateur-hôte.
- . par contre, les protocoles d'accès aux entrées fonctionnelles des services offerts sur ce nouvel ordinateur, seront vraisemblablement nouveaux et il faudra développer de nouvelles machines M:GP.

- raccordement de nouveaux réseaux :

Il s'agit là d'une éventualité à beaucoup plus long terme mais qui n'est plus utopique avec la probable prolifération des réseaux. En ce cas, le frontal peut servir de "gateway" entre 2 réseaux ayant adopté des techniques similaires de transport, en se chargeant de la conversion d'un protocole de transport à l'autre. La machine de communication fournirait alors une machine M:GP de type Station de Transport par réseau connecté.

Pour conclure, nous rapprocherons l'architecture de la machine de communication de l'architecture-réseau S.N.A. ("Systems Network Architecture") d'IBM [IBM-1], qui est toutefois de plus grande envergure. Dans les 2 cas, le choix d'un découpage en niveaux fonctionnels bien distincts vise à obtenir une structure flexible et extensible. Le sous-système de communication de S.N.A. [IBM-2] possède une structure logique comparable. La différence tient à la décentralisation des divers composants de ce sous-système qui peuvent être répartis physiquement sur l'ordinateur central et sur des contrôleurs de communication intermédiaires répartis de façon arborescente à partir de l'ordinateur central (cf. 2.2.3). Nous remarquerons toutefois que le découpage fonctionnel de la machine M:DM, schématisé par la figure 14, ne préjuge absolument pas de la réalisation logicielle ou matérielle de chaque machine élémentaire.

CHAPITRE VI

INGENIERIE DU LOGICIEL

Le chapitre précédent décrivait le Frontal sous son aspect fonctionnel. Ici, nous nous intéressons plus particulièrement aux aspects techniques.

Compte tenu des facilités offertes par le matériel et des contraintes inhérentes à son utilisation, nous avons été conduits à certains choix dans la réalisation du logiciel. Nous explicitons ces choix, puis nous esquissons quelques éléments des méthodes de travail que nous avons utilisées.

6.1. - MATERIEL

6.1.1. - Généralités

6.1.2. - Tâches

6.1.3. - Interruptions

6.1.4. - Gestionnaire de ressources

6.1.5. - Entrées-Sorties - Coupleurs

6.2. - POURQUOI PARTIR D'UN CALCULATEUR NU ?

6.3. - CONTRAINTES ET CHOIX EN RESULTANT

6.3.1. - Contraintes dues au matériel utilisé.

6.3.2. - Réalisation des machines

6.4. - METHODOLOGIE

6.4.1. - Programmation

6.4.2. - Outils de mise au point

6.1. - MATERIEL

6.1.1. - GENERALITES

L'ordinateur utilisé est un petit calculateur Télémécanique T1600. Il est organisé autour de deux bus :

- un bus mémoire vive
 - un bus général
- (voir figure)

Il comprend principalement :

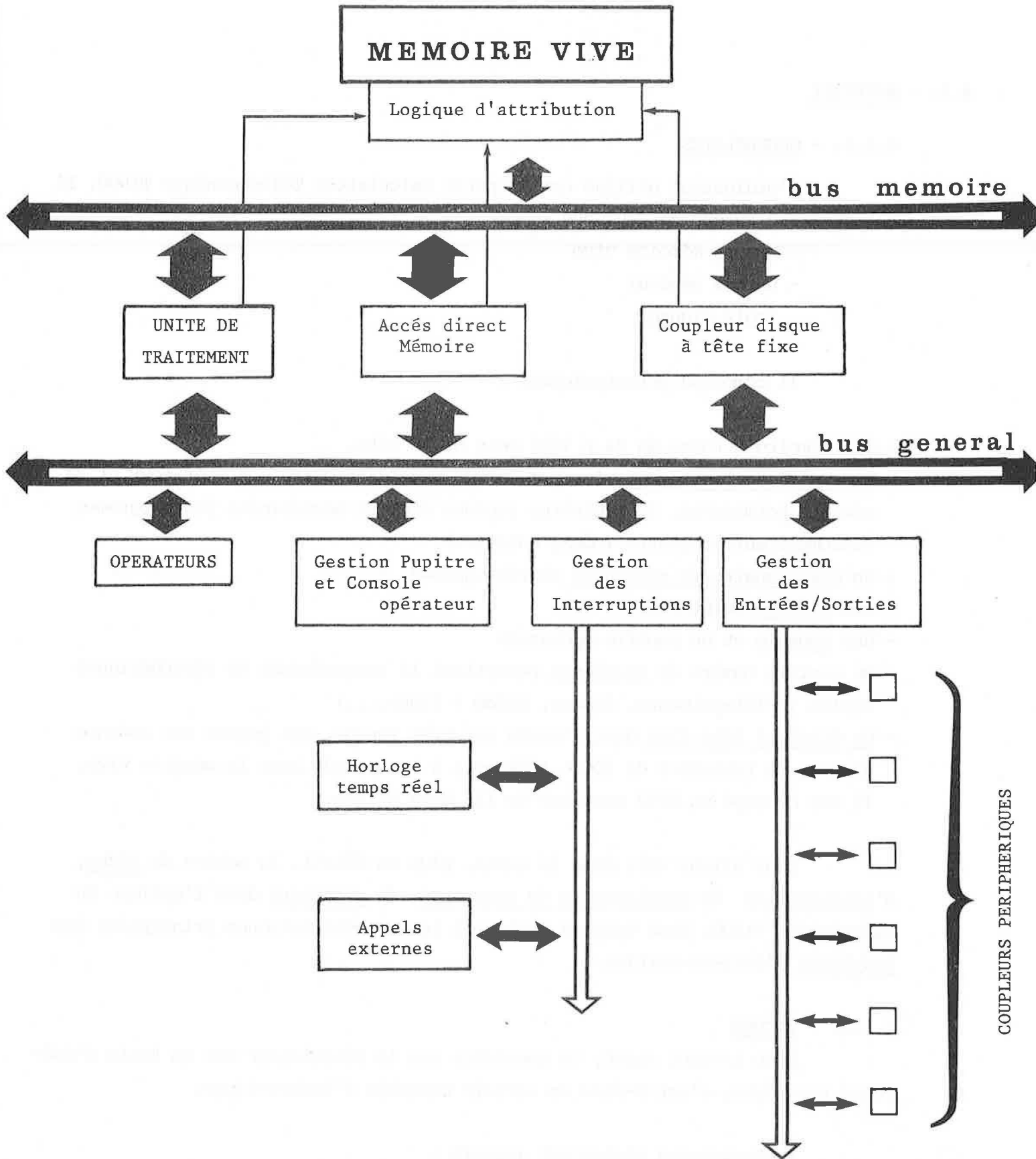
- Une mémoire à tores de 24 K 1024 mots de 16 bits.
- Une Unité de Traitement microprogrammée possédant outre son opérateur et sa mémoire permanente, 32 registres rapides dont 12 accessibles par programme (accumulateur, registre index, bases...)
- Un gestionnaire de ressource microprogrammé
- Une horloge temps réel
- Une console et un pupitre opérateur
- Un certain nombre de coupleurs permettant le raccordement de périphériques variés (téléimprimeurs, écrans, modem + ligne,...)
- Un disque à tête fixe dont l'accès est très rapide, qui permet une cadence globale de transfert de 200×1024 mots à la seconde avec la mémoire vive. Il est découpé en 8192 secteurs de 128 mots utiles.

Nous allons voir dans la suite, plus en détail, la notion de tâche, d'interruption de gestionnaire de ressource, de sémaphore dans l'optique du calculateur T1600. Nous verrons également les caractéristiques principales des coupleurs d'entrées-sorties.

6.1.2. - TACHES

A un instant donné, on considère que le calculateur est en train d'exécuter une tâche, c'est-à-dire un certain ensemble d'instructions.

A chacune des tâches est associé :



Organisation du T1600

- un contenu : programme et données
- un niveau de priorité : il ne peut exister deux tâches de même priorité.
- un contexte : c'est une zone mémoire qui lui est propre et qui sert à sauvegarder les 12 registres accessibles par programme, lorsque cette tâche n'est pas active.
- un état de disponibilité ou d'activation.

Niveau de priorité

Les tâches sont de deux types :

- les tâches de niveau matériel ou tâches à lancement immédiat. Elles sont déclenchées par une interruption. Il y a 8 niveaux de priorité possibles numérotés de 0 à 7 dans le sens des priorités décroissantes.
- Les tâches de niveau logiciel, ou tâches à lancement différé. Elles sont déclenchées par programme. Il existe 128 niveaux de priorité décroissante de 0 à 127.

Les tâches sont hiérarchisées par niveau de priorité. L'ensemble des tâches de niveau matériel a une priorité supérieure à l'ensemble des tâches de niveau logiciel.

Contexte

Le contexte d'une tâche est constitué par le contenu des 12 registres accessibles par programme.

A chaque fois que l'on passe d'une tâche à une autre, quelle que soit la tâche abandonnée, la tâche lancée, ou la raison du changement de tâche, le contenu des registres est sauvegardé dans le contexte de la tâche interrompue, puis les registres sont chargés à partir du contexte de la tâche lancée.

Etat

Une tâche peut se trouver dans un des trois états suivants :

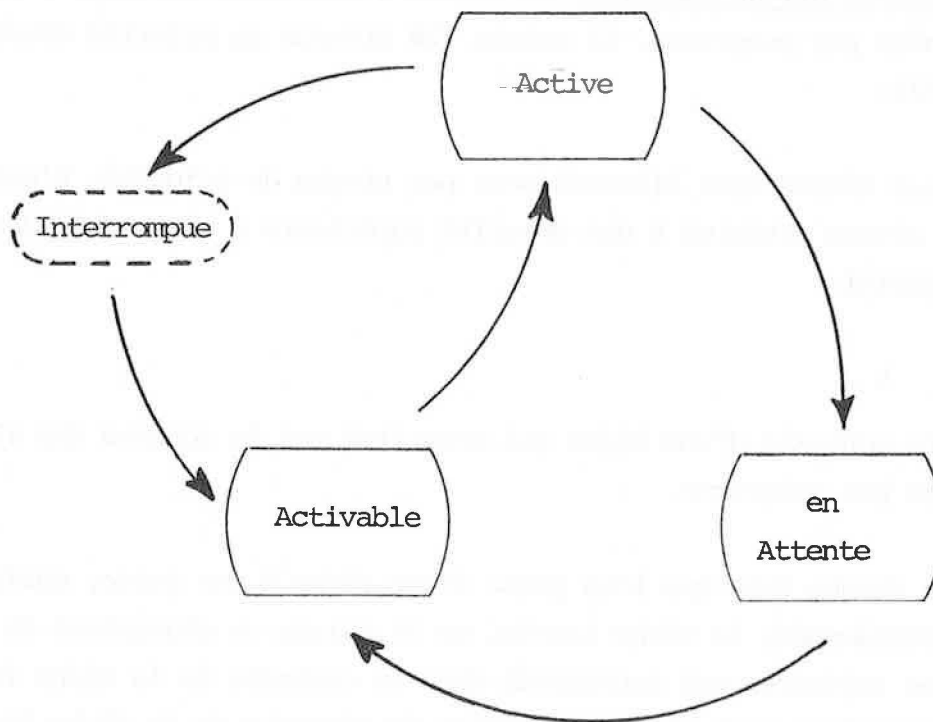
active : c'est elle qui se déroule.

activable : elle possède toutes les ressources, mais a une priorité inférieure à celle de la tâche active. (dans un contexte multiprocesseur, une tâche activable serait active).

en attente : elle ne possède pas toutes les ressources.

On peut considérer également un état intermédiaire :

interrompue : une tâche en attente qui était plus prioritaire que la tâche en cours vient d'obtenir ses ressources. La tâche active va donc passer dans l'état activable.



6.1.3. - INTERRUPTIONS

Les interruptions nécessaires à la prise en compte par le calculateur des évènements externes et internes sont réalisées par le lancement des tâches de niveau matériel.

A chacune de ces tâches est associé un niveau de priorité. A une tâche donnée, correspond soit un seul évènement, soit plusieurs évènements de même nature. Lorsque plusieurs évènements distincts déclenchent la même tâche, un sous-niveau est associé à chacun d'eux. Le sous-niveau attaché à un évènement permet à la tâche correspondante de reconnaître l'évènement qu'elle doit traiter.

La correspondance entre les tâches de niveau matériel et les évènements déclenchant les interruptions est la suivante, par ordre de priorité décroissante des niveaux et sous-niveaux :

NIVEAUX		SOUS-NIVEAUX	
0	Alarmes causées par programme	0	instruction privilégiée
		1	protection mémoire
		2 } 3 }	instructions spéciales
		4 }	
		5 }	
1	Inutilisé	-	
2	Horloge temps réel	-	
3	Appels internes	0 à 7	Un sous-niveau par appel possible (disque, console opérateur)
4	Inutilisé	-	
5	Inutilisé	-	
6	Inutilisé	-	
7	Entrée/Sorties	0 à 31	Un sous niveau par périphérique rattaché au système d'interruption.

Lorsqu'un évènement externe devant provoquer une interruption se produit, la demande d'interruption correspondante est mémorisée jusqu'à ce qu'elle soit prise en compte.

Une demande d'interruption et une seule peut être mémorisée pour chaque évènement distinct (c'est-à-dire pour chaque sous-niveau). Lorsqu'une demande d'interruption est prise en compte (lancement de la tâche correspondante et acquittement), la demande est annulée et une nouvelle demande peut être enregistrée.

6.1.4. - GESTIONNAIRE DE RESSOURCES

Il prend en charge le lancement et la suspension des tâches en fonction des demandes d'interruptions, du déroulement du programme et du niveau de priorité relatif des tâches. Il assure le choix de la tâche active et déclenche les changements de contexte nécessaires.

Il prend le contrôle soit sur apparition d'une demande d'interruption, soit sur la modification de l'état d'une tâche

En outre, il gère des outils permettant de régir les accès aux ressources et de faire communiquer les tâches entre elles : les sémaphores.

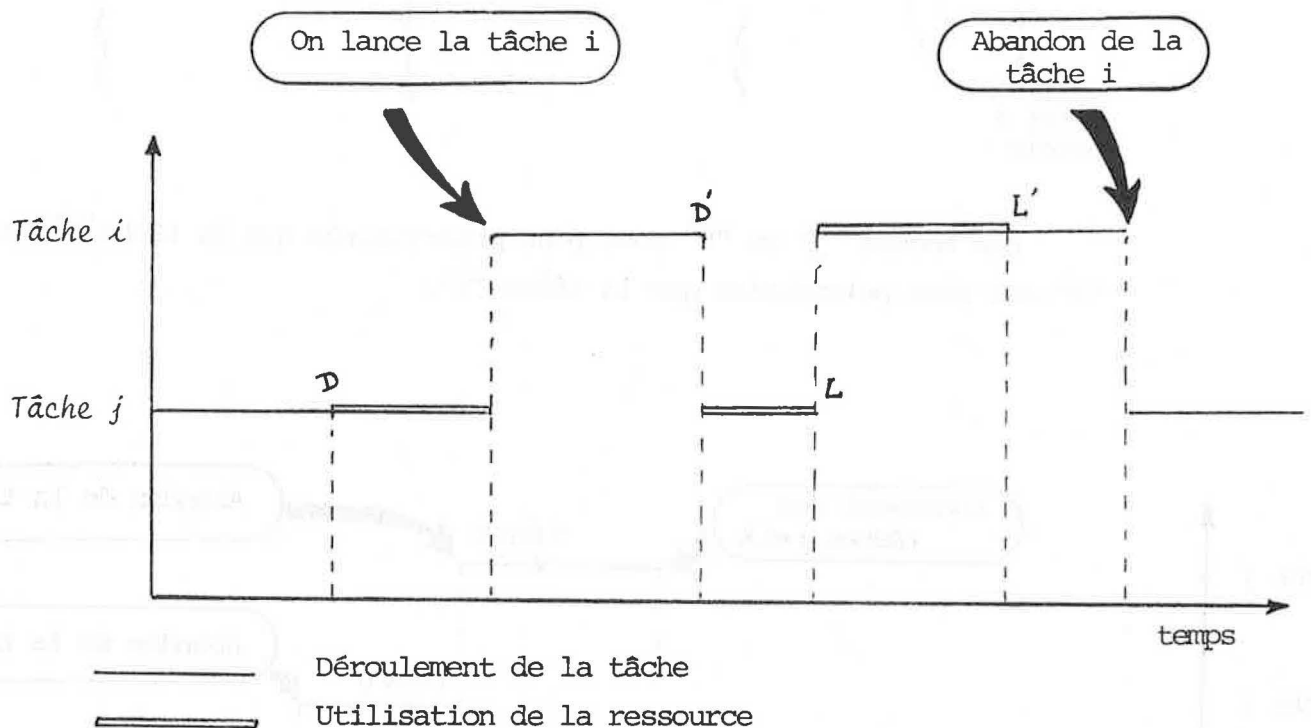
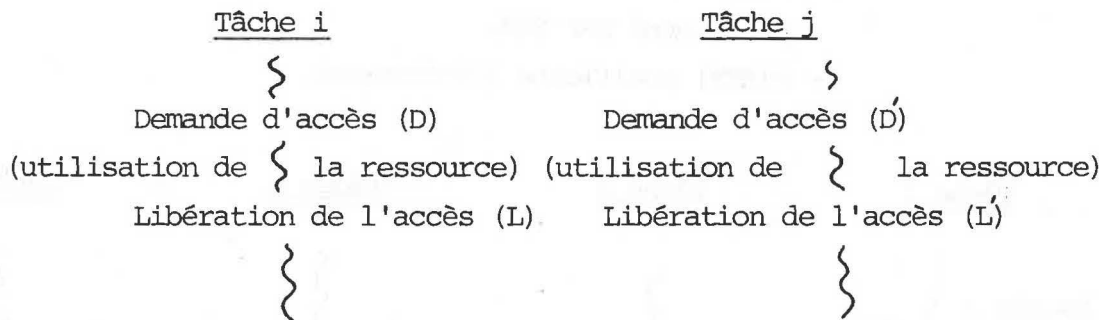
Les sémaphores sont de trois types :

1°) Sémaphores d'exclusion :

Il est attaché à une ressource et est constitué d'un compteur (initialisé avec le nombre d'accès possibles à la ressource), et d'une file d'attente des tâches requérant la ressource.

La gestion consiste à satisfaire les demandes d'accès tant que la ressource est disponible (décrément du compteur). Puis à mettre en attente les tâches demandeurs lorsqu'il n'y a plus d'accès (compteur nul ou négatif). Dès qu'une tâche libère un accès, le gestionnaire sélectionne la tâche en attente la plus prioritaire, restitue son contexte et lance son exécution.

Exemple : deux tâches : "i" et "j", la tâche i étant plus prioritaire.



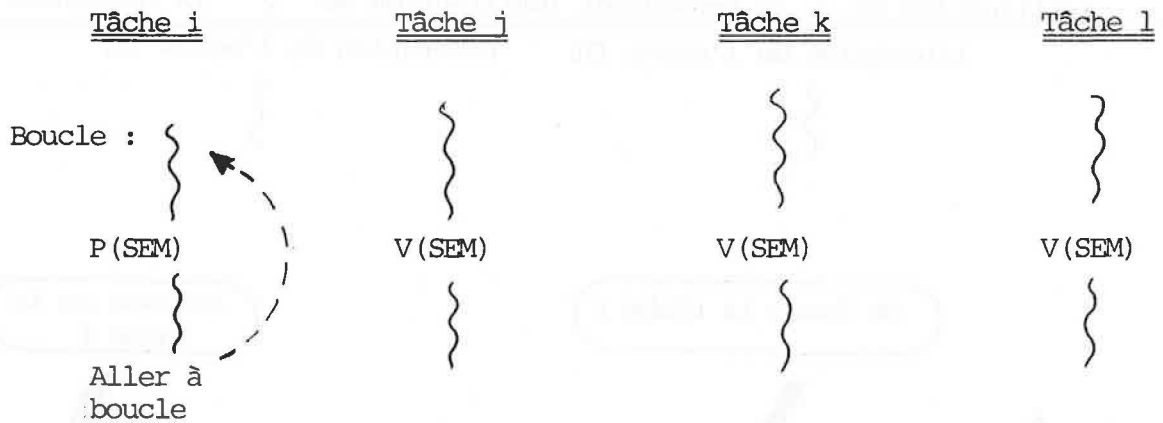
2°) Sémaphore privé simple

Il est lié à une tâche (qui se l'approprie) et permet de résoudre les problèmes de synchronisation entre tâches (une tâche se mettant en attente d'un évènement positionné par une autre tâche). On le nomme aussi sémaphore d'appel. (il permet à une tâche d'"appeler" une autre tâche).

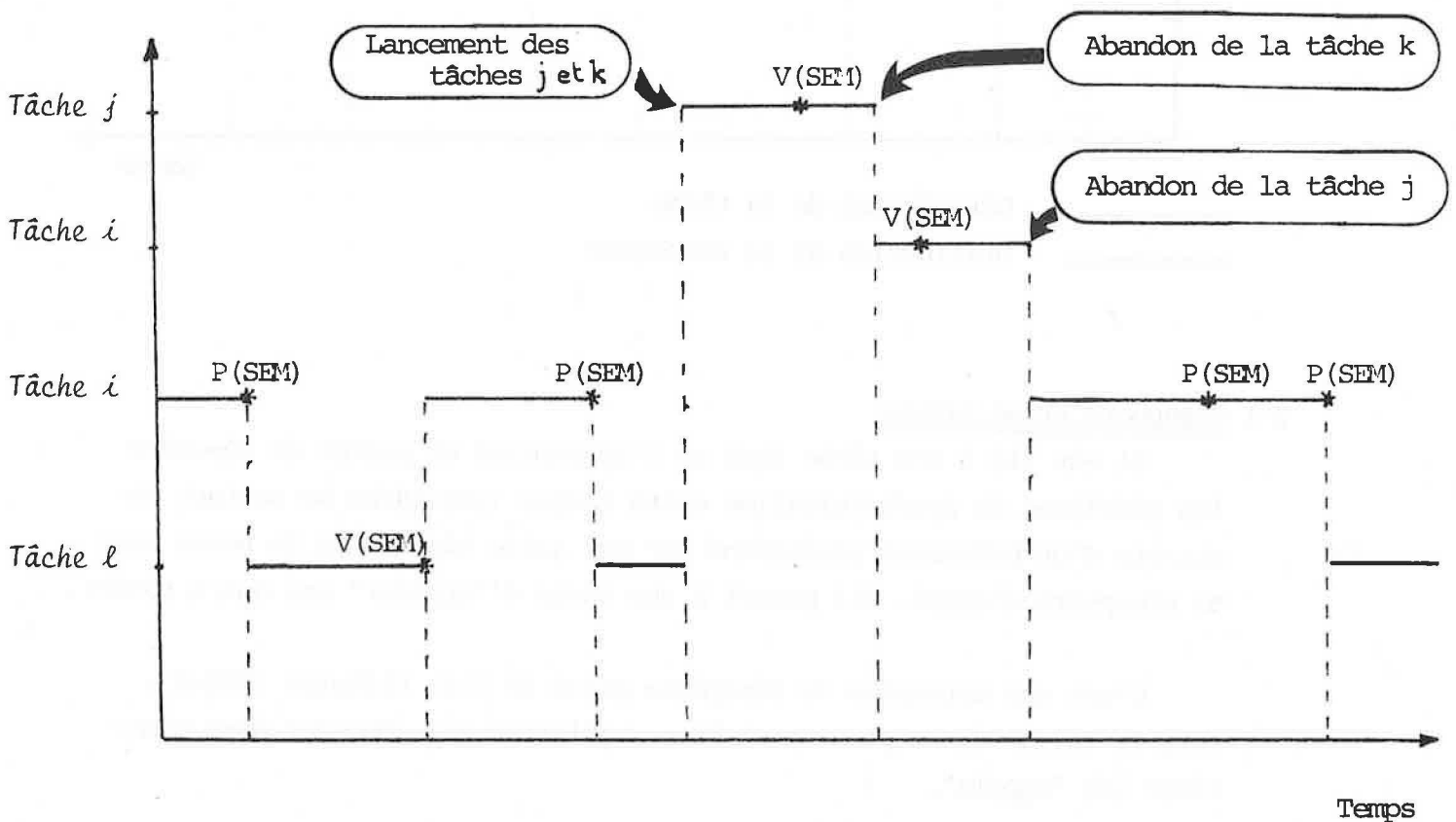
C'est une extension du sémaphore privé de E.W. Dijkstra [SYS-2]. Ici, la valeur du compteur peut être supérieure à 1. On peut donc mémoriser les "appels".

Exemple : On reprend les conventions de Dijkstra :

- P(SEM) permet de se mettre en attente sur l'évènement positionné par SEM.
- V(SEM) positionne l'évènement.



Les tâches "j" et "k" sont plus prioritaires que la tâche "i". La tâche "i" est plus prioritaire que la tâche "l".



3°) Sémaphore privé paramétré

Ce dernier type étend les possibilités du sémaphore privé simple : Il permet, lors du positionnement d'un évènement, d'enregistrer un numéro de paramètre. Outre la synchronisation, il rend possible la communication entre tâches.

Remarque : Dans l'exemple précédent, la tâche i ne pouvait connaître la tâche qui l'avait activée. Ceci aurait été possible avec l'utilisation d'un sémaphore privé paramétré en prenant comme paramètre le numéro de la tâche faisant un V (SEM)

6.1.5. - ENTREES-SORTIES - COUPLEURS

Nous laissons volontairement de côté le disque à tête fixe et la console opérateur dont la gestion est particulière.

L'ensemble des périphériques est rattaché au système d'interruption du calculateur par l'intermédiaire de coupleurs et d'un bac d'entrées-sorties (ou bus d'entrées-sorties). Le niveau de priorité correspondant est le niveau 7. C'est-à-dire que la tâche affectée à la gestion des entrées-sorties est celle qui a la priorité la plus faible parmi les tâches de niveau matériel.

Les coupleurs sont de types variés, suivant l'organe périphérique qui leur est attaché (terminal, écran de visualisation, modem + ligne téléphonique, etc...). Tous sont gérés en "mode caractère", c'est-à-dire que l'élément d'information émis ou reçu est le caractère. Cet évènement (émission ou réception d'un caractère) engendre une demande d'interruption dès qu'il intervient.

D'autre part, les coupleurs ne reconnaissent aucun caractère particulier, si ce n'est le caractère de synchronisation dans le cas des coupleurs gérant les transmissions en mode synchrone (cf CH I).

Ces caractéristiques, ou plutôt ces contraintes, ont entraîné certains choix dans la gestion des entrées-sorties. Nous détaillons ces choix dans la suite.

6.2. - POURQUOI PARTIR D'UN CALCULATEUR NU ?

Une étude du logiciel T1600 et son utilisation préliminaire nous ont montré son inadaptation au problème posé par un Frontal. Particulièrement, et c'est primordial dans notre cas, la gestion des entrées-sorties entraînerait de trop graves contraintes.

En effet, les systèmes standards existant sur le calculateur T1600 ont en commun le même moniteur d'entrées-sorties. Ce moniteur possède un certain nombre de caractéristiques qui nous paraissent peu appropriées à notre problème :

- Il est conçu pour une utilisation par des programmeurs dans le contexte d'un système d'exploitation. Il a donc été prévu pour assister l'utilisateur et fournir toutes facilités pour la composition des programmes. Il regroupe l'ensemble des programmes de gestion et d'exécution des échanges. De ce fait, une requête fait l'objet de tests nombreux et sophistiqués.
- Il n'autorise qu'un seul accès à une unité physique. C'est-à-dire que deux utilisateurs ne peuvent simultanément faire une demande d'entrée-sortie sur un même organe périphérique. De même, un utilisateur ne peut faire plus d'une demande si le précédent échange n'est pas entièrement terminé, pour un organe périphérique donné.

Ces deux points sont incompatibles avec un des critères recherchés : l'indépendance et la désynchronisation entre les utilisateurs du moniteur désirant réaliser des échanges et la cadence des entrées-sorties proprement dites.

Tous les traitements d'échanges sont effectués par la même tâche de niveau matériel 7 (tâche associée au niveau d'interruption du bac d'entrées-sorties).

Ce facteur, sans doute plus important encore que les deux premiers, limite considérablement le nombre de traitements simultanés possibles. Il faudrait alors réduire le nombre et la cadence des lignes synchrones (sous peine d'"erreur de cadence" au niveau des coupleurs) (cf. Ch. 3.4.2.).

Toutes ces considérations nous ont conduit à abandonner l'idée de construire un logiciel à partir d'un système d'exploitation standard. Nous avons donc réalisé entièrement un système spécialement adapté à partir d'un T1600 nu.

6.3. - CONTRAINTES ET CHOIX EN RESULTANT

6.3.1. - CONTRAINTES DUES AU MATERIEL UTILISE.

a) Gestion des demandes d'interruption

Les tâches de niveau matériel sont hiérarchisées et les coupleurs d'entrées-sorties sont associés à une même tâche du niveau matériel le plus bas.

D'autre part, les entrées-sorties sur lignes synchrones se font en mode caractère. Il est donc nécessaire de traiter tout caractère reçu ou transmis en un temps minimum afin de supporter le maximum de lignes simultanément sans erreur de cadence (cf. Ch. 3.4.2.)

* Traitement différé

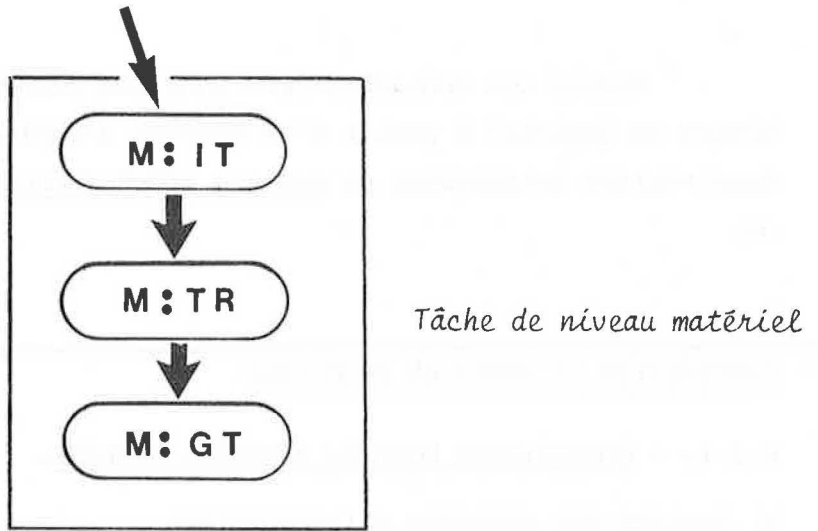
Une solution apportée est de rendre possible le traitement d'une demande d'interruption par une tâche lancée en différé (tâche de niveau logiciel).

De cette manière, on consomme le minimum de temps sous la tâche de niveau matériel. Sous cette tâche, seuls sont nécessaires l'identification de la ligne ayant provoqué la demande d'interruption (niveau et sous-niveau) et le lancement de la tâche de niveau logiciel qui traitera en différé, cette demande.

C'est le rôle de la machine M:IT, suivant le type de coupleur concerné, elle traite la demande d'interruption de deux façons possibles :

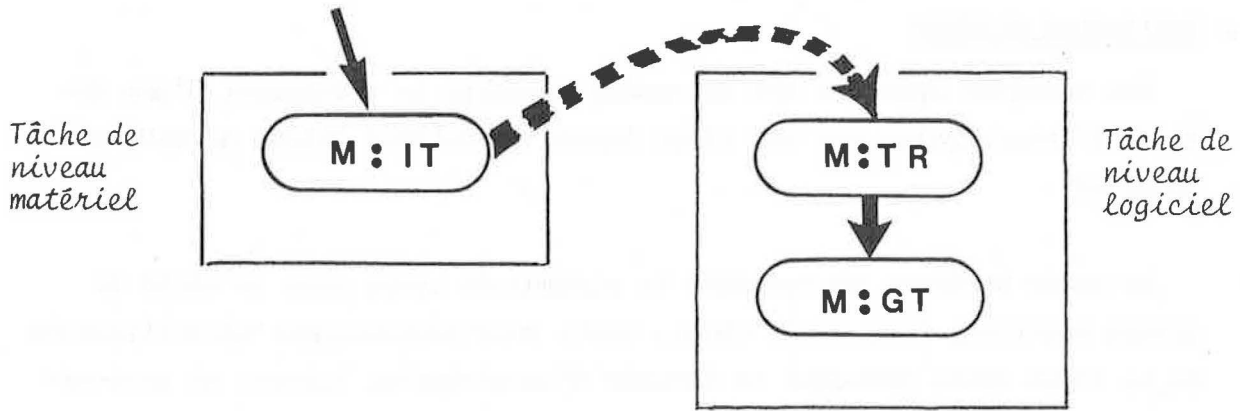
- en mode immédiat : l'activation de la machine M:TR gérant le coupleur, se fait par appel procédural au niveau de M:IT. Le traitement de l'interruption s'effectue alors entièrement sous le niveau de la tâche d'entrée-sortie (niveau matériel).

Interruption d'entrée/sortie



TRAITEMENT EN MODE IMMEDIAT

Interruption d'entrée/sortie



TRAITEMENT EN MODE DIFFERE

- en mode différé : La machine M:TR est lancée sous une tâche de niveau logiciel elle-même activée par la machine M:IT. La demande d'interruption est acquittée et la tâche de niveau matériel est ainsi rendue disponible.

Ainsi, on peut privilégier les lignes de transmission synchrone rapides qui seront traitées en mode immédiat. Les entrées-sorties lentes (telles que celles associées aux télétypes) seront traitées en différé.

Seront également traitées en différé d'une manière analogue, la gestion de l'horloge, ainsi que celle du disque. Tous les deux sont en effet associés à des tâches de niveaux matériels plus prioritaires (respectivement niveaux 2 et 3). Ce traitement en différé permet de ne pas retarder la prise en compte des appels sur les niveaux de priorité inférieure.

- * Afin de minimiser le temps d'activation d'une tâche de niveau matériel, on a eu recours à une astuce technique : lorsque le traitement à effectuer sur une demande d'interruption est particulièrement long, on répartit (lorsque cela est possible) ce traitement sur plusieurs des demandes d'interruptions suivantes; (voir Chapitre 3.4.2).

b) Place disponible en mémoire vive.

Le matériel considéré étant un minicalculateur, sa capacité mémoire est réduite. Nous avons donc été conduits à limiter la taille des programmes et aussi celle des tampons (c'est-à-dire les zones utilisées pour stocker temporairement des données).

Malgré tout, ce gain de place n'a pas toujours été notre préoccupation essentielle : en effet, les impératifs de rapidité des traitements et de modularité ont impliqué des compromis.

- Utilisation du disque comme mémoire secondaire

Une première solution a été de transporter sur disque les tampons à chaque fois que c'était possible.

C'est le cas des tampons d'entrées-sorties utilisés dans les échanges avec le réseau et ceux avec le (ou les) ordinateur(s) de traitement connecté(s) (particulièrement les échanges concernant le télétraitement par lots).

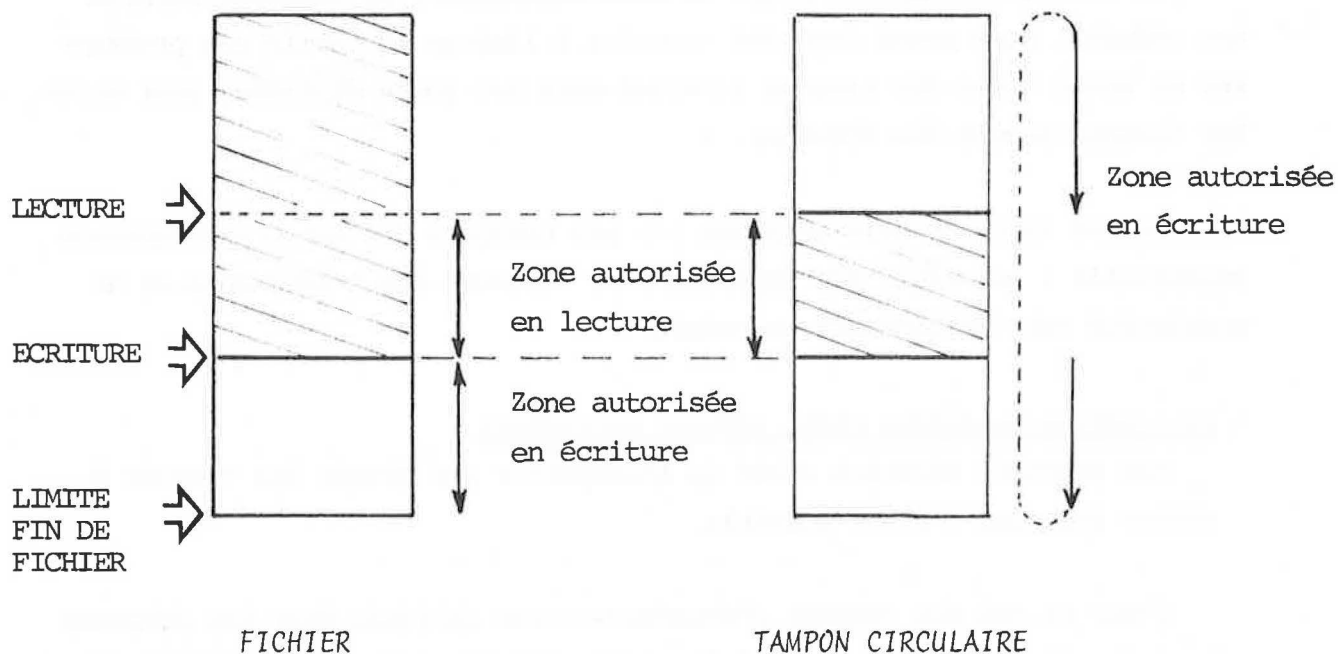
Dans le même esprit d'économie de mémoire, les messages du système et des différents processeurs sont stockés sur disque. On peut ainsi s'offrir le luxe d'avoir des messages clairs et facilement compréhensibles. Nous avons pour ce faire, réalisé une gestion de disque sophistiquée et très spécialisée. Le disque est divisé en zones qui peuvent être :

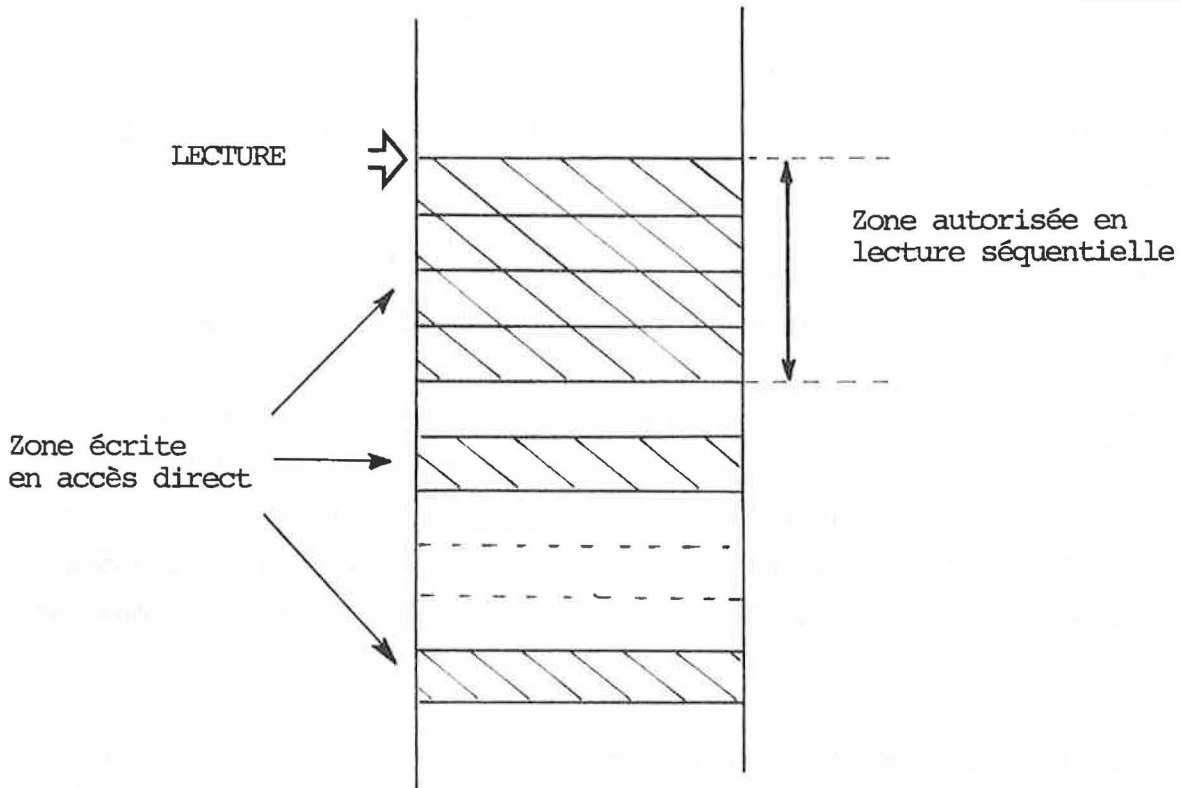
- des fichiers
- ou des tampons circulaires (voir figures)

L'accès à ces zones est :

- soit séquentiel
- soit direct

Il est possible d'accéder une zone, à la fois en lecture et en écriture. Dans ce cas, les deux accès sont synchronisés (on ne peut lire que ce qui a été écrit). Ceci s'applique particulièrement à la gestion des tampons d'entrées-sorties lors des échanges sur "voies-virtuelles" (cf chapitre IV).





ACCES DIRECT - ACCES SEQUENTIEL

Mécanisme spécialement adapté à la gestion des "voies virtuelles" : Les zones de communications sont écrites en accès direct (les tampons ne sont pas reçus dans l'ordre). Elles sont lues par l'abonné en accès séquentiel.

Remarque : Dans tous les cas de figure on peut échanger "lecture" et "écriture".

- Procédures réentrantes

Une deuxième solution est d'éviter de dupliquer des traitements identiques : ceci en faisant un découpage des traitements par modules distincts. Ces modules sont organisés en procédures réentrantes, chacune spécialisée dans un traitement particulier.

Nous rappelons qu'une procédure réentrante est constituée :
• d'un ensemble d'instructions et éventuellement de données communes, capables d'effectuer un certain traitement. Cet ensemble ne se modifie pas en cours d'exécution.

. d'une zone de travail comprenant les données propres à un traitement particulier.

Il y aura autant de zones de travail que de traitements particulier, mais il n'y aura qu'un ensemble d'instructions et de données communes.

Dans la réalisation, nous avons profité du fait que l'adressage, sur le T1600, est basé, c'est-à-dire que l'adressage s'effectue en relatif par rapport au contenu d'un registre de base. Le calculateur disposant de trois registres de base, l'un d'eux (RW) a été dédié pour servir à l'accès des zones de travail. Un changement de zone de travail se fait donc dynamiquement en changeant la base RW.

Exemple : La gestion d'une ligne de transmission est constituée de deux traitements logiquement distincts :

- la gestion d'un coupleur, commune à tous les coupleurs de ce type (Machine M:TR)
- la gestion de la procédure de ligne utilisée (Machine M:GT).

La réunion en un seul module de ces deux traitements aurait conduit à des répétitions facheuses : duplication du traitement du coupleur pour les différentes lignes.

De plus, l'expérience nous a montré que dans le cas d'une ligne, la décomposition en modules distincts conduisait à un gain de vitesse de traitement appréciable : le traitement moyen d'un caractère sur une ligne synchrone est 3 à 4 fois plus rapide par cette méthode qu'au moyen d'un module de gestion paramétré multi-procédure.

Dans certains cas, il a fallu choisir entre la modularité (entraînant une plus grande facilité de mise au point) et le gain de place.

En effet, le gain de place entraîné par la décomposition en modules distincts est moins évidents lorsque les traitements sont courts : ce découpage introduit des zones de travail et surtout certains dispositifs de gestion de liens (c'est-à-dire les différentes corrélations permettant l'enchaînement logique des modules entre eux).

On a toujours, dans la réalisation, donné la priorité à la modularité.

- Gestion de la mémoire libre.

Enfin, pour utiliser au mieux la place mémoire, nous avons créé un mécanisme d'allocation et de restitution de zones mémoire.

Pour ce faire, la mémoire libre est découpée en blocs de 32 octets. Lorsqu'à un niveau quelconque du système, un demandeur désire un tampon mémoire, il effectue une requête sous la forme d'un appel à un programme de service. Il sera alors alloué au demandeur le nombre de blocs demandé, si possible ; Sinon, le demandeur est mis en attente. De même, le demandeur peut restituer la zone mémoire allouée par une requête similaire.

6.3.2. - REALISATION DES MACHINES

Une machine est un ensemble fonctionnel constitué, suivant les commodités et les besoins, de procédures, de processus, ou des deux à la fois.

Dans ses relations avec l'extérieur, au niveau de son activation, une machine est vue globalement comme une procédure ou comme un processus.

- Dans le premier cas, la commande se fait par un appel procédural. La machine activée envoie en retour un compte rendu exprimant que la commande a été :
 - . acceptée
 - . impossible à exécuter avec les raisons du refus.
- Dans le deuxième cas, on active la machine par l'intermédiaire d'un séma-
phore. La machine se comporte alors globalement comme un processus et se déroule sous une tâche de niveau logiciel.

Une de nos préoccupations a été de réduire au minimum le nombre de tâches de niveau logiciel. En effet, outre la place mémoire prise par les contextes, il y a également les problèmes de perte de temps dues aux changements de contexte.

La règle que nous avons adoptée est de ne réaliser sous forme de processus que les machines répondant aux deux critères suivants :

- . nécessité d'asynchronisme avec la machine appelante (ou parallélisme entre différentes machines appelées par la même machine de niveau supérieur).
- . nécessité de traiter complètement la fonction demandée avant de le faire au profit d'une autre machine

Dans les autres cas, on préférera réaliser la machine sous forme de procédure. Ce sera particulièrement nécessaire dans les cas suivants :

- . nécessité d'une action immédiate dès l'appel.
- . lorsque l'action demandée influe sur le comportement de la machine appelante (attente) : le demandeur a alors besoin d'un compte rendu immédiat si la commande est refusée.

Les communications entre machines se font de deux manières, suivant que la machine est activée par un appel procédural, ou par positionnement d'un sémaphore :

- . passage de paramètres dans des registres. Ces paramètres étant la commande elle-même, l'adresse d'un bloc de commande, ou l'adresse d'une table.
- . file d'attente dont l'adresse est connue de la machine appelée, ou passage d'un paramètre au moyen du sémaphore.

6.4. - METHODOLOGIE

6.4.1. - PROGRAMMATION

LANGAGE

Pour la réalisation du logiciel, nous avons choisi un langage de plus haut niveau qu'un assembleur : PL1600. La décomposition du logiciel en "Machines" se prêtait particulièrement bien à l'utilisation d'un langage à structure de bloc.. C'est le cas de PL1600 qui est un langage du type ALGOL, possédant des instructions composées performantes :

```
IF... THEN...ELSE ;
DO WHILE .... ;
DO FOR... UNTIL... ;
CASE OF ... ;
etc...
```

Les avantages apportés par l'utilisation d'un tel langage sont la facilité de programmation, de modification, la clarté et la communicabilité des programmes produits.

Les réserves que l'on pourrait faire au sujet de ce langage concernent surtout les structures de données, trop rudimentaires, et l'absence de gestion automatique des registres, c'est-à-dire des caractéristiques qui rapprochent PL1600 d'un langage d'assemblage.

Dans un seul cas, nous avons été conduit à programmer en assembleur : pour les machines de très bas niveau, celles directement liées au matériel. En effet, la programmation en PL1600 de cette partie aurait perdu toute sa clarté du fait de l'utilisation abondante d'instructions particulières (en particulier : communication avec le BUS, avec le système d'entrées/sorties). La seconde contrainte est relative au temps de traitement : Toute cette partie se déroulant sous le niveau d'une tâche matérielle, il est impératif de minimiser le code exécuté.

PROGRAMMATION PROPREMENT DITE

Pour l'écriture de nos programmes, nous avons utilisé les techniques de programmation structurée chère à DIJKSTRA [SYS-6] et à WIRTH [SYS-7], [SYS-8]. La décomposition fonctionnelle en machines s'y prêtait particulièrement bien.

Nous entendons, par programmation structurée, l'observation d'un certain nombre de règles telles que :

- utiliser les mêmes mécanismes pour exécuter des actions de même nature.
- séparer nettement les sections de données des sections de programme.
- séparer les actions indépendantes en procédures, en définissant bien leurs interfaces.
- décomposer chaque procédure en blocs (éventuellement imbriqués) en utilisant intensivement des primitives de contrôles telles que IF-THEN-ELSE, des boucles (DO), des primitives de choix (CASE). Chaque action élémentaire est réalisée dans un bloc.

- 2°) "Patch" : permet de remplacer un mot mémoire par un autre. C'est un outil dangereux mais bien pratique lorsqu'on a localisé une erreur et que l'on désire faire un test dynamique immédiat.
- 3°) Trace : permet de visualiser (et éventuellement de stocker) toutes les informations qui transitent sur une ligne pendant un certain temps. Cet outil nous a permis une mise au point plus rapide des procédures de transmission. Il sert en outre dans la mise en oeuvre de nouvelles procédures ou les détections des erreurs d'ordinateurs nouvellement connectés.
- 4°) Etats : Une série de commandes permettent de visualiser des renseignements propres aux différentes machines :
- mot d'état d'un coupleur
 - zone de réentrance
 - tables
 - files d'attente
 - etc...
- avec la possibilité de les figer un instant donné.
- 5°) Compte-rendu : chaque machine rend un compte-rendu explicite qui peut être visualisé. Ceci est réalisé par la sortie d'un message (préalablement enregistré sur disque) indiquant la machine en cause et le motif de l'erreur.

Tous ces outils sont indépendants et peuvent être mis en oeuvre sans perturber le fonctionnement de l'ensemble du frontal.

D'autres outils de tests plus spécifiques ont été créés, particulièrement pour la Station de Transport CYCLADES (CF CH.4). Entre autre la possibilité de visualiser les paquets émis ou reçus sur le réseau de communication.

CHAPITRE VII

RESULTATS ET PERFORMANCES

7.1. POSSIBILITES PRATIQUES D'EXPLOITATION

7.1.1. *Etat actuel de réalisation et utilisation*

7.1.2. *Charge du frontal*

7.2. ORGANISATION ET IMPLANTATION DU LOGICIEL

7.2.1. *Place occupée*

7.2.2. *Transportabilité*

7.1. - POSSIBILITES PRATIQUES D'EXPLOITATION

7.1.1. - ETAT ACTUEL DE REALISATION ET UTILISATION

Le frontal T1600 est passé dans la phase d'exploitation depuis Janvier 1976.

Les utilisateurs disposent de trois serveurs qui donnent accès à un certain nombre de services :

a) Station de traitement par lots connectée à l'ordinateur Philips P1200 (CF Annexe I)

- Elle permet de rentrer des travaux à partir d'un périphérique quelconque connecté au T1600 (terminal, lecteur de ruban perforé,...). Ces travaux sont exécutés sur l'ordinateur de traitement.
- On dispose d'une console opérateur de la station permettant de suivre l'état d'avancement des travaux à distance.
- Les sorties des listes se font soit sur l'imprimante locale à l'ordinateur Philips, soit sur un périphérique connecté au frontal.

★ Ce service est utilisé régulièrement par le département de Métallurgie de l'Ecole pour l'exploitation de données. Les données sont obtenues, en sortie de bancs de mesure, sur ruban perforé. Elles sont traitées sur l'ordinateur Philips.

★ Un autre type d'utilisation est le transfert de programme sources entre le PDP 11/40 et l'ordinateur Philips (voir Annexe I) et inversement.

Dans ce cas, le PDP 11/40 est considéré par le frontal comme un couple : lecteur de cartes - Imprimante. Il n'y a pas à l'heure actuelle de protocole particulier (du genre "Protocole d'accès au frontal" (cf.4.5)

b) Concentrateur de terminaux sur le réseau CYCLADES

Il utilise la station de transport mise au point sur le frontal depuis 1975.

C'est un abonné "client" qui permet, à partir d'appareils locaux, l'accès aux services disponibles sur le réseau CYCLADES (traitement par lots, temps partagé).

Il se charge de la connexion et du dialogue entre les parties connectées à travers le réseau.

Il possède une liste de service existant sur le réseau avec leurs paramètres (numéro de l'abonné distant, nom du service, noms et types des entrées). Cette liste étant éventuellement mise à jour par l'opérateur du frontal.

Il suffit pour demander la connexion à un service réseau de préciser son mnémonique (CF Annexe II et III).

L'abonné concentrateur permet la connexion à la fois à :

- 3 services de traitement par lots distants
- 10 services de temps partagé.

Il gère toutes les voies virtuelles nécessaires à l'utilisation de ces services.

★ La mise en exploitation véritable du concentrateur est assez récente (Mars 1976). Nous n'avons donc pu à ce jour en tester les capacités et les limites. D'autre part, le nombre de services disponibles sur CYCLADES est assez limité.

4 services de temps partagé : $\left\{ \begin{array}{l} 1 \text{ sur IBM 360/67 (sous CP 67)} \\ 3 \text{ sur CII IRIS 80 (sous SIRIS 8)} \end{array} \right.$

2 services de traitement par lots : $\left\{ \begin{array}{l} 1 \text{ sur IBM 360/67 (sous OS)} \\ 1 \text{ sur CII IRIS 80} \end{array} \right.$

Le nombre d'utilisateurs simultanés a été jusqu'ici de 5.

Il faut donc attendre la rentrée 76/77, avec la mise à disposition en libre service du concentrateur aux élèves, pour faire de véritables tests en vraie grandeur.

- c) Terminal graphique interactif utilisable à partir d'un programme se déroulant sur le Philips P1175.

C'est un service particulier utilisé surtout par le département Métallurgie de l'Ecole pour visualiser des courbes et les faire évoluer en fonction de paramètres introduits.

7.1.2. - CHARGE DU FRONTAL

Les limitations réelles du Frontal sont dues principalement aux lignes de transmission synchrone (cf. 6.3.).

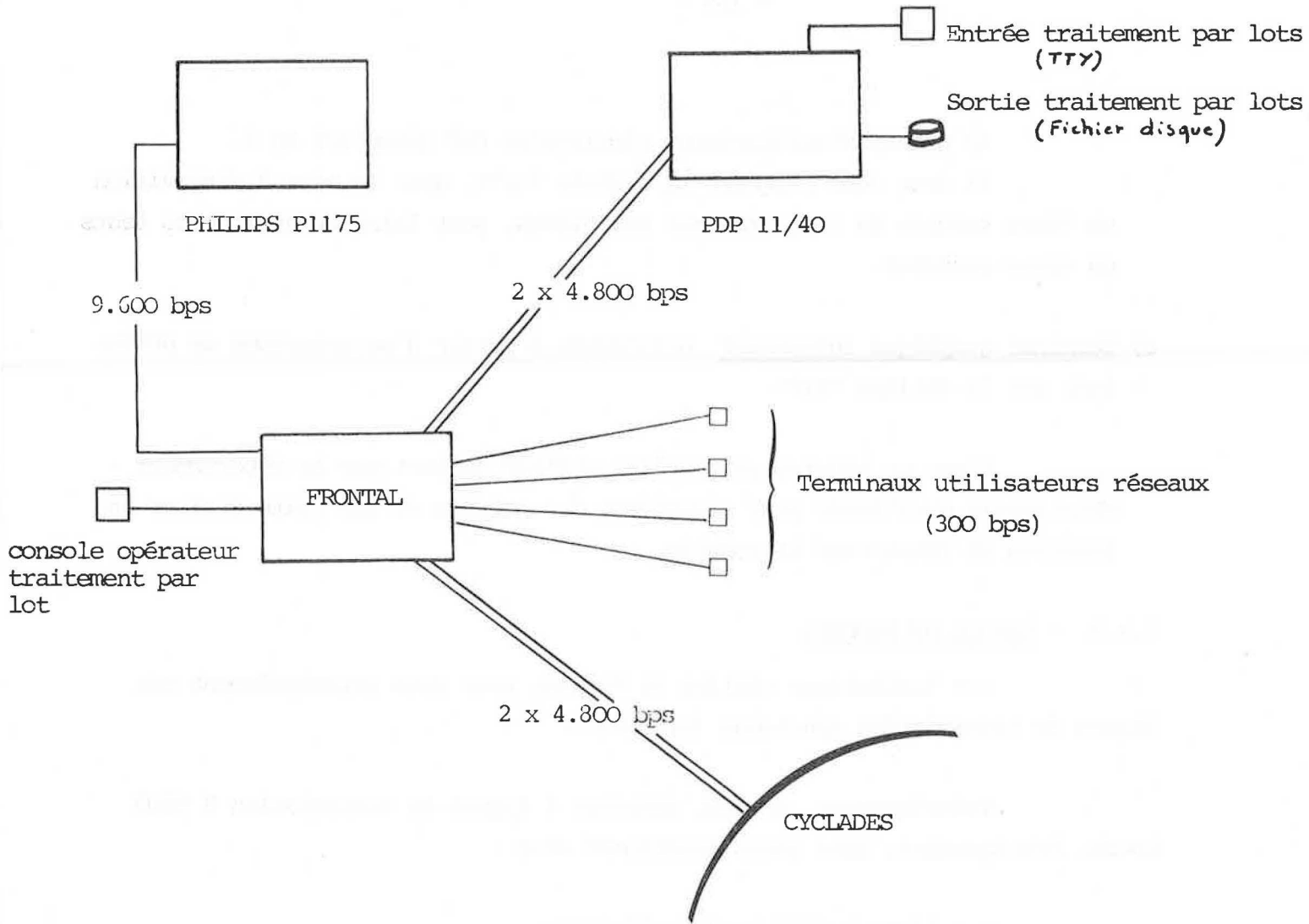
Théoriquement, on peut admettre 4 lignes de transmission à 9600 bauds. Pratiquement, nous avons fonctionné avec :

- 1 ligne à 9600 bauds half-duplex
- 2 lignes à 4800 bauds full-duplex

simultanément, sans pertes sensibles de synchronisation.

Les services utilisés étaient :

- concentrateur de terminaux sur le réseau avec 4 terminaux.
- traitement par lot sur le Philips P1175
 - avec { la console opérateur sur un terminal de frontal
 - le "lecteur de carte" et "l'imprimante" sur le PDP 11/40



Les temps de réponse pour les terminaux connectés aux services réseaux aussi bien que pour l'ensemble du traitement par lots est sans signification réelle pour le frontal. En effet, cela dépend beaucoup du nombre d'utilisateurs simultanés et de la charge locale des différents services concernés.

La seule expérience probante a été de connecter un terminal directement à l'un des services réseau (temps partagé sur IBM 360/67) via le réseau téléphonique commuté, et de comparer les temps de réponse avec ceux d'un terminal connecté via le frontal du même service. On obtenait sensiblement le même résultat.

7.2. - ORGANISATION ET IMPLANTATION DU LOGICIEL

7.2.1. - PLACE OCCUPEE

Pour des raisons de rapidité de traitement, tout le système se trouve en mémoire centrale d'une manière permanente. Un essai de recouvrement a été fait, mais pénalise beaucoup trop les temps d'exécution.

La place occupée par les logiciels des différentes machines est la suivante :

Le superviseur comprenant divers outils de maintenance et divers utilitaires (notamment un module de mise à jour de fichiers source à travers le réseau entre l'INSA de Rennes et l'Ecole des Mines)	6 K octets
La station de transport comprenant les services "lettres régulières", "voies virtuelles" et un abonné spécial permettant les dialogues inter-opérateurs.	6 K octets
Un abonné client "concentrateur de terminaux" et "terminal lourd" permettant l'accès simultané aux services réseaux distants (temps partagé et traitement par lots)	16 K octets
La gestion du protocole d' "appareil virtuel" CYCLADES	2 K octets
Un serveur local de traitement par lots	4 K octets
Le logiciel d'entrée sortie avec les méthodes d'accès supportant : <ul style="list-style-type: none">- 10 terminaux conversationnels- 1 lecteur de ruban rapide- 1 imprimante- 1 disque à tête fixe (et la gestion des fichiers)- la connexion au calculateur principal PHILIPS P1175- La connexion au calculateur PDP11/40- La connexion au réseau CIGALE	11 K octets
Les tampons en mémoire centrale (mémoire libre)	2 K octets

Par ailleurs, toutes les "zones de communication" de la station de transport et des abonnés sont situées sur disque à tête fixe (taille standard d'une zone de communication : 16 secteurs de 256 octets).

Enfin, le disque est également utilisé comme zone tampon entre le frontal et le calculateur principal pour les fichiers de traitement par lots.

7.2.2. - TRANSPORTABILITE

Le logiciel étant entièrement modulaire, on peut implanter un frontal "à la carte" sur n'importe quel ordinateur Télémécanique T1600, compte tenu du nombre et du type de connexions désirées et de la quantité de mémoire centrale disponible.

Outre les services généraux tels que le superviseur, les services réseaux ou locaux, le logiciel d'Entrée/Sortie est composé des machines nécessaires aux diverses connexions.

A l'heure actuelle, la bibliothèque des machines disponibles comprend :

- la gestion des coupleurs..... asynchrone standard
asynchrone multiplexé
synchrone
lecteur de ruban
imprimante
disque à tête fixe
disque à tête mobile
- la gestion des procédures de transmission TMM-UC full duplex
ECMA PHILIPS half duplex
BSC multileaving.
- La gestion des protocoles terminal TTY half duplex
terminal full duplex
CIGALE (avec la ST CYCLADES)
P.A.F. (cf CH 4.5.)
terminal lourd Philips P 850 T
Gestion de fichiers sur disque
à tête fixe.

★ Il existe un générateur de système en macrolangage, exécutable par le macroprocesseur du Philips P1175 qui permet, moyennant l'introduction des différents paramètres :

- services désirés
- types et implantation des coupleurs
- procédures de transmission désirées.

La génération du système complet en langage source . Un générateur utilisant le macro générateur du T1600 est en cours de réalisation. Il permettra d'éviter la phase de compilation de tous les modules et de ne générer en langage source que les tables nécessaires.

La phase finale étant l'appel de l'éditeur de lien avec la liste des modules en langage objet des machines nécessaires.

★ Le logiciel frontal a été implanté jusqu'ici dans deux sites : celui de l'Ecole des Mines où il a vu le jour, et celui de l'INSA de RENNES où il doit être utilisé pour l'accès aux services réseaux.

CHAPITRE VIII

C O N C L U S I O N S

Nous donnons d'abord notre point de vue sur l'intérêt pratique de raccorder des ordinateurs à un réseau à travers un mini-ordinateur frontal. Puis, mis à part l'aspect frontal, nous dégageons des idées pour intégrer au "système", dès la conception, un environnement télé-informatique. Enfin, après avoir suggéré diverses améliorations du produit, nous envisageons son futur cadre d'utilisation.

8.1. - INTERET ET LIMITES DU RACCORDEMENT A UN RESEAU A TRAVERS UN
FRONTAL

8.1.1. - Faisabilité

8.1.2. - Critiques

8.2. - L'INTEGRATION AU SYSTEME D'UN ENVIRONNEMENT TELE-INFORMATIQUE

8.2.1. - Nouveaux problèmes posés par la télé-informatique

8.2.2. - La solution apportée par le système

8.3. - LES PERSPECTIVES DU SYSTEME

8.3.1. - Utilisation actuelle

8.3.2. - Améliorations à envisager

8.3.3. - Nouvelles perspectives d'utilisation

8.1. - INTERET ET LIMITES DU RACCORDEMENT A UN RESEAU A TRAVERS UN FRONTAL

8.1.1. - FAISABILITE

L'expérience a montré [CYC-10] que la connexion d'un ordinateur à un réseau général tel que CYCLADES était une opération coûteuse, du fait de l'inertie des logiciels pré-existants qui n'ont pas été conçus pour prendre en compte une "dimension-réseau". Une telle connexion implique des modifications réparties de façon diffuse à travers le logiciel initial, y compris parfois les programmes d'application.

La solution consistant à déporter l'interface avec le réseau au niveau d'un mini-ordinateur frontal présente à nos yeux les avantages suivants :

- 1 - Les interventions dans le logiciel de l'ordinateur-hôte sont réduites au strict minimum. On évite ainsi de gros problèmes de mise à niveau à chaque nouvelle version du système livré par le constructeur.
- 2 - Le frontal se contente de simuler une série de portes extérieures bien connues de l'ordinateur-hôte comme points d'accès à ses services. L'évolution du système de l'ordinateur hôte n'apportera aucun bouleversement de cet interface.
- 3 - L'ordinateur hôte est totalement déchargé de la gestion du réseau, qui peut s'avérer coûteuse à la fois en place mémoire requise, en temps de traitement et en temps de mise au point.
- 4 - Le gain économique est important à long terme. En effet, la connexion de nouveaux ordinateurs-hôtes nécessite peu de développement supplémentaire de logiciel, car le frontal multiplexe sa gestion du réseau entre l'ensemble des ordinateurs hôtes qui lui sont connectés. Les efforts de programmation éventuelle sont centralisés dans le frontal.
- 5 - La flexibilité obtenue est plus grande dans la mesure où toute modification d'un niveau de protocole sur le réseau - ce qui est encore assez fréquent dans un stade d'expérimentation - limite les interventions au frontal.

Le coût initial de cette solution - programmation du frontal - n'est pas moindre que celui du raccordement direct d'un hôte. Il est même un peu plus élevé car un mini-ordinateur ne possède pas de facilités d'implémentation et de mise au point comparables à celles d'un gros ordinateur. Mais ce coût n'augmente pratiquement plus avec la connexion de nouveaux ordinateurs. D'autre part, comme dans notre cas, le mini-ordinateur peut être employé parallèlement à des fins purement locales où il est utilisé simultanément en frontal de chaque ordinateur du site (notion de frontal "orientable"). Il facilite ainsi le partage des ressources sur le site et permet une certaine banalisation des terminaux. Enfin, la baisse constante du coût des mini-ordinateurs accroît encore la faisabilité d'une telle solution.

8.1.2. - CRITIQUES

Tous les problèmes ne sont pas pour autant résolus par cette méthode. Elle peut même présenter certains nouveaux inconvénients :

- 1 - Elle entraîne des délais supplémentaires et une perte d'efficacité dans les transferts d'information. En effet, il faut tenir compte d'une liaison, souvent peu rapide, et d'une interprétation supplémentaire entre l'ordinateur hôte et le frontal.
- 2 - Les protocoles du réseau peuvent faire appel à des fonctions dont il n'existe pas d'équivalent dans les rapports ordinateur hôte-frontal.
- 3 - Les possibilités d'exploitation de la liaison ordinateur hôte-frontal sont étroitement dépendantes du "degré d'ouverture vers l'extérieur" du système hôte. Elles sont souvent limitées à des utilisations du type interaction avec un terminal, ou traitement par lots. Des problèmes se posent rapidement si l'on envisage des relations plus générales entre processus sur le réseau. Il faut alors recourir à des procédés ad-hoc plus ou moins efficaces. Par exemple un transfert de fichier vers un point distant du réseau pourra s'opérer en lançant à partir du frontal un programme utilisateur sur l'ordinateur hôte. Ce programme perforera le fichier voulu sur un perforateur de cartes virtuel déclaré au niveau du frontal. Encore faut-il que le système hôte soit capable de reconnaître un perforateur de cartes sur la ligne qui le relie au frontal ! Et de toutes manières, l' "overhead" d'initialisation du transfert est quasiment inacceptable.

- 4 - La connexion de plusieurs hôtes au frontal peut rendre crucial le problème de la congestion et de la fiabilité d'un unique point d'accès au réseau. Le doublement du frontal est alors à envisager.
- 5 - Cette méthode n'introduit aucun standard car chaque ordinateur connecté conserve ses propres protocoles et les raccordements sont calculés au coup par coup. Cela peut même aller à l'encontre de certains efforts de normalisation [GEN-8] vis à vis des constructeurs.

Nous ne prétendons pas avoir résolu le problème de la connexion à un réseau. Il ne pourra être convenablement résolu que par les constructeurs eux-mêmes. Eux seuls possèdent les moyens de développer de larges systèmes "orientés-réseau" et peuvent fournir des interfaces matérielles et logicielles standards en accord avec des normes établies au niveau international. Dans l'attente de cette heureuse époque, nous pensons avoir tout de même fourni un moyen pratique, économique et défendable de se connecter à un réseau.

8.2. - L'INTEGRATION AU SYSTEME D'UN ENVIRONNEMENT TELE-INFORMATIQUE

Pour réaliser ce frontal, nous avons été confrontés avec les divers problèmes de la télé-informatique, depuis la connexion d'un simple terminal jusqu'au raccordement à un vaste réseau. Cette expérience nous a permis de dégager des principes généraux pour la conception de systèmes en environnement télé-informatique. En effet, l'"approche-système" n'est pas fondamentalement différente à ce niveau, qu'il s'agisse d'un mini-ordinateur ou d'un plus gros ordinateur.

8.2.1. - NOUVEAUX PROBLEMES POSES PAR LA TELE-INFORMATIQUE

La plupart des systèmes d'exploitation actuellement commercialisés n'ont pas une structure adaptée à la télé-informatique. La gestion des appareils extérieurs (essentiellement des terminaux) est souvent répartie de façon diffuse dans divers niveaux de logiciels. Les programmes d'application eux-mêmes ne sont pas toujours indépendants des outils de transmission et imbriquent différents niveaux - logiques et même physiques - de gestion des télécommunications. Ce monolithisme limite forcément le partage des ressources, le raccordement de nouveaux appareils extérieurs, et à plus forte raison la participation à un réseau.

8.2.2. - LA SOLUTION APPOURTEE PAR LE SYSTEME

Le système NOS ("Network Operating System") a été développé pour le frontal sur machine nue, et entièrement conçu en fonction des besoins télé-informatiques. Son système d'Entrées/Sorties - ou machine de communication (cf. 5.3) - intègre l'ensemble des procédures de télécommunications et des protocoles de dialogue avec les appareils extérieurs. Il offre un interface standard constitué d'un ensemble de primitives donnant accès à des unités symboliques purement logicielles. Les applications - qui sont ici les "serveurs virtuels" du frontal (cf. 5.2.) - ne connaissent rien d'autre que ces unités symboliques et les primitives y donnant accès. Les échanges réels avec les appareils physiques sont totalement découplés des applications par des méthodes d'accès en cascade.

Le point essentiel concerne l'architecture en niveaux successifs de ce système d'E/S qui reflète bien la structure en couches d'un réseau. Il est découpé en plusieurs niveaux logiques puis physiques. Les niveaux physiques assurent la gestion des transmissions physiques de l'information sous forme de chaîne de caractères, en respectant une procédure de transmission. Les niveaux logiques prennent en charge les conventions de codage, de format et la régulation des flux d'échange imposées par les protocoles du réseau et des appareils locaux. Toutefois, nous n'avons pas assuré un découpage complet des applications par rapport aux appareils car le protocole d'appareil virtuel CYCLADES (PAV) n'est pas intégré au système d'Entrée/Sortie. Ce découplage est uniquement réalisé pour les appareils du site.

Cette structure par niveaux successifs (machines M:TR, M:GT, M:GP), avec spécialisation des machines à l'intérieur d'un même niveau, est particulièrement flexible et extensible (cf. 5.3.3.). Elle se prête bien à la prise en compte de nouveaux éléments extérieurs (terminal intelligent, nouvel ordinateur et même autre réseau), ou à la modification d'un niveau de protocole sans perturbation des niveaux environnants. Il nous semble que ce sera la voie générale d'évolution pour les logiciels à venir dont il faudra assurer une ouverture suffisante vers l'extérieur. Certains constructeurs se sont d'ailleurs déjà largement engagés dans cette voie pour la réalisation de réseaux complet [IBM-2 IBM-4, DEC-1, CDC-1].

8.3. - LES PERSPECTIVES DU SYSTEME

8.3.1. - UTILISATION ACTUELLE

Le produit développé est opérationnel à l'Ecole des Mines de Saint-Etienne depuis Janvier 1976 (cf. Chap VII). Il a été mis en exploitation durant le 1er semestre à raison d'environ 8 heures par semaine, le T1600 étant par ailleurs utilisé pour des sessions APL et des travaux de recherche. La part de cette exploitation doit encore être étendue à la rentrée scolaire.

- Il sert couramment sur le site à des utilisateurs non informaticiens (chimistes, métallurgistes). Ils l'utilisent en frontal du P1175 pour la soumission de trains de travaux nécessitant des organes périphériques spécialisés connectés au frontal (lecteur de ruban-papier, écran graphique). A la rentrée scolaire, une station complète d'accès à distance sera déportée physiquement chez ces utilisateurs. Cette station sera réalisée à partir d'un lecteur de cartes, d'une imprimante, d'une console opérateur, d'un lecteur de rubans et d'un écran graphique connectés au frontal. Elle pourra en partie être dupliquée à différents endroits. Des utilisateurs du PDP 11/40, récemment connecté au T1600, commencent à s'en servir également en frontal du P1175 pour des transferts de fichiers stockés sur des bibliothèques P1175.
- Concurrément, il est utilisé pour l'accès à CYCLADES par des étudiants désirant s'initier au maniement de systèmes de temps partagé (SIRIS 8 sur IRIS 80 et CP 67 sur IBM 360/67 au CIOG de Grenoble) et de langages non disponibles sur le site (PASCAL, ALGOL...). Une campagne de promotion a par ailleurs été lancée auprès des utilisateurs non informaticiens précédents pour l'utilisation des services offerts sur CYCLADES.
- De nouveaux services sont projetés. En particulier un serveur "éditeur de textes unique sur le site" est en cours de réalisation. L'éditeur conversationnel est déjà utilisé à partir des terminaux du P1175 et sera accessible en tout point du site via le frontal. A plus long terme (cf. 8.3.3) un serveur APL permettra d'accéder à partir du frontal à un interpréteur APL réalisé sur un SOLAR connecté au T1600 (cet interpréteur a été réalisé par une équipe de l'Ecole des Mines et est commercialisé par le constructeur). Cette opération est primordiale dans la mesure où l'interpréteur APL et le système frontal ne peuvent actuellement cohabiter sur le T1600 d'une part, par manque

de place en mémoire, d'autre part, en raison de l'incompatibilité entre notre système NOS et le système BOS du constructeur qui interface l'interpréteur APL.

Enfin, le système fonctionne également sur le T1600 de l'INSA de RENNES qui, ne disposant pas d'autre moyen informatique, voulait accéder aux services offerts par CYCLADES.

8.3.2. - AMELIORATIONS A ENVISAGER

Trois types de travaux peuvent être envisagés dans le futur :

1 - L'extension de la gamme des services - extérieurs - auxquels permet d'accéder le frontal. L'accès à chaque nouveau service nécessite le développement d'un nouveau serveur virtuel sur le frontal. L'architecture du logiciel (cf. 5.2) facilite ce genre de développement. Par contre, cela peut nécessiter le développement de nouvelles méthodes d'accès dans la machine de communication du frontal, particulièrement s'il faut se connecter à un nouvel ordinateur. Les problèmes peuvent-être plus ou moins complexes mais ont l'avantage d'être bien localisés par niveau.

2 - Les extensions précédentes requièrent une augmentation de puissance de la machine.

Cela concerne d'abord la capacité-mémoire actuelle qui est limitée (64 K. octets maximum). Or le T1600 ne possède pas de mécanisme d'adressage virtuel. D'autre part, nous n'avons pas prévu de structure de recouvrement des programmes à partir du disque. Les mécanismes seraient sans doute trop coûteux pour un produit dont la principale qualité doit être la rapidité. En outre, la nette décroissance du prix des mémoires semble plutôt plaider en faveur du passage à une machine de capacité-mémoire plus grande (ce qui devient maintenant courant, même pour un mini-ordinateur).

Parallèlement, le débit de transmission du T1600 ("ou throughput") est sérieusement limité par ses capacités d'E/S. Il a déjà atteint la limite de ses possibilités pour la configuration actuelle (cf. Annexe 4) avec un débit total dépassant les 19,2 kilobauds sur l'ensemble des lignes

synchrones. L'addition de nouvelles liaisons synchrones est difficilement envisageable si l'on désire un fonctionnement simultané de toutes les lignes. Une véritable amélioration ne peut être obtenue que par une plus grande décentralisation de la gestion des télécommunications avec introduction de processeurs périphériques spécialisés. Ces processeurs périphériques permettraient de décharger l'unité de traitement du frontal d'une grande partie de la gestion physique des transmissions (cf. 8.2.2.). Les procédures de transmission, actuellement traitées caractère par caractère par le logiciel, pourraient être micro-programmées sur ces processeurs. Cette approche conduirait à répartir la machine de communication du frontal entre l'unité de traitement et des processeurs périphériques prenant en charge les niveaux M:TR et M:GT. Cette évolution est d'ailleurs inévitable avec l'apparition des nouvelles procédures de transmission, type HDLC [PRO-2], qui nécessitent un traitement au niveau du bit.

Enfin, la capacité de commutation offerte avec une seule unité de traitement est très suffisante actuellement pour la rapidité des lignes de transmission utilisées. Par contre, on peut envisager une structure doublée pour le frontal afin d'accroître la disponibilité du produit et de réduire les possibilités de congestion. En effet, les extensions envisagées (cf. 8.3.3.) font du frontal un pivot essentiel de l'installation informatique du site.

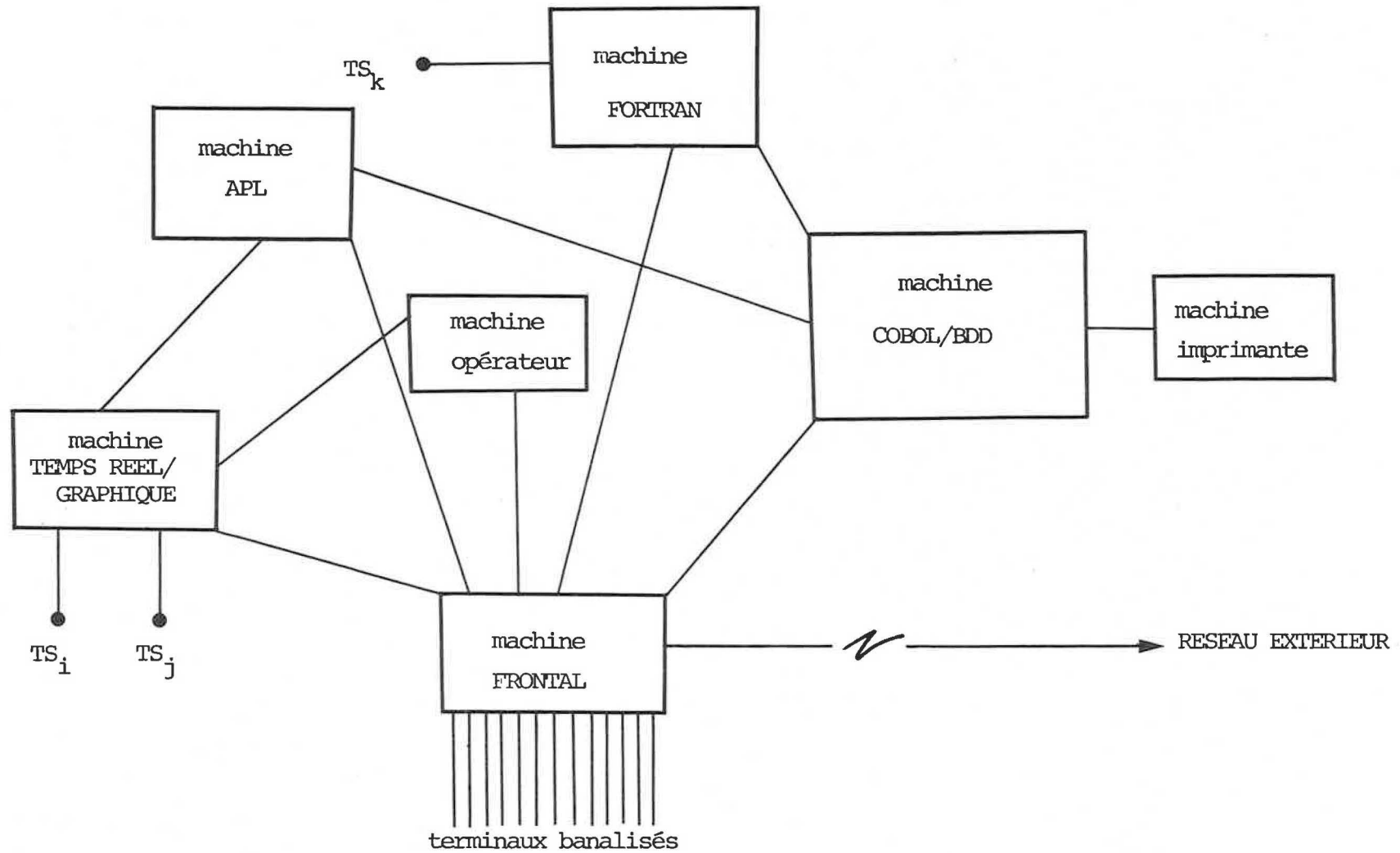
- 3 - Cette augmentation de puissance peut difficilement être obtenue à partir du T1600. Il faudrait donc envisager le passage à une autre machine satisfaisant les besoins énumérés : capacité-mémoire accrue - forte capacité de transmission et de commutation - grande disponibilité. Quelques études ont déjà été menées visant à la construction de machines adaptées à ces besoins télé-informatique, et particulièrement à la fonction de noeud de réseau. Ces machines ont généralement une structure multi-processeur/multi-bus, facilitant une augmentation de puissance graduée [ARP-6]. Le constructeur du T1600 a lui même développé la structure POLYBUS [TECH-7] pour servir de base à la construction des divers constituants matériels d'un réseau [TECH-8]. Cette structure, organisée à partir de processeurs SOLAR 16, permet un fort degré de parallélisme entre les traitements et les entrées/sorties. L'architecture du logiciel du frontal nous semble suffisamment flexible pour envisager son transport sur une telle machine, d'autant que le code d'ordre des unités de traitement SOLAR est compatible avec celui du T1600. Seule serait à reprendre la gestion physique des transmissions dans la machine de communication.

8.3.3. - NOUVELLES PERSPECTIVES D'UTILISATION

Les possibilités du frontal développé ont permis de ne pas lier son avenir à celui du réseau CYCLADES, puisqu'il est déjà largement utilisé pour des besoins locaux. Le seul intérêt est de conserver une ouverture vers un réseau extérieur (peut être le réseau public TRANSPAC à l'avenir). Ainsi tout nouveau terminal introduit sur le site est déjà automatiquement connecté au frontal. L'objectif est maintenant de développer encore son utilisation dans le cadre de la répartition des moyens de calcul de l'Ecole des Mines sur plusieurs mini-ordinateurs.

L'idée d'affecter à un ordinateur une (ou plusieurs) fonction(s) simple(s) n'est pas nouvelle [SYS-11] mais trouve maintenant une justification économique avec la baisse de prix enregistrée pour les mini-ordinateurs. D'autre part, un logiciel spécifique sur une petite machine est plus simple à développer et à maintenir - et souvent plus efficace pour l'application concernée - qu'un logiciel général sur une grosse machine. Partant de cette constatation on envisage de réaliser un centre de calcul réparti basé sur un réseau de mini-ordinateurs spécialisés chacun dans une fonction [SYS-12]. Le frontal regrouperait l'ensemble des terminaux des usagers et fournirait, à partir d'un langage de commande unique, les points d'accès aux services offerts à l'utilisateur (fig. 1).

Dans cette réalisation chaque mini-ordinateur est exploité avec un système spécialisé (traitement par lots, temps réel, APL en temps partagé etc...) fourni par le constructeur. Outre le développement d'un interpréteur (réparti ou non) du langage de commande, ce projet implique le développement d'un logiciel (réparti) capable d'appliquer le réseau des connexions logiques, entre services, sur le réseau des connexions physiques, entre ordinateurs. Cependant, les problèmes ne se posent pas du tout dans les mêmes termes que pour un réseau général type CYCLADES, dans la mesure où chaque ordinateur participant est spécialisé et que l'on peut en profiter pour particulariser certaines connexions. Le problème de l'augmentation de la puissance et de la disponibilité du frontal évoqué au cf. 8.2.2 devient par contre crucial.

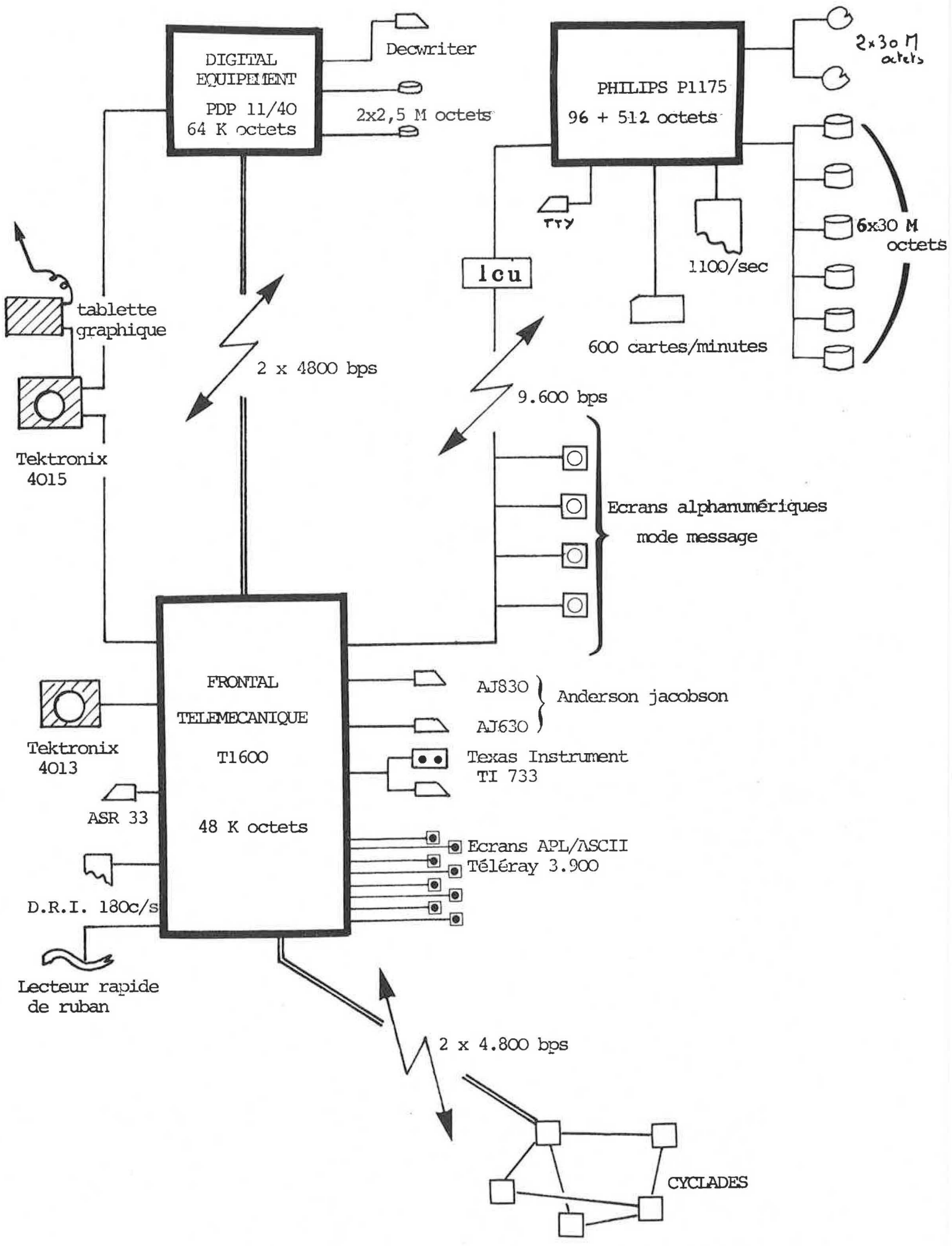


TS_i, TS_j, TS_k : terminaux spécialisés

Figure 1 - Centre de calcul réparti

ANNEXE I

CONFIGURATION DU CENTRE DE CALCUL



ANNEXE II

MANUEL OPERATEUR

- Généralités
- Comment opérer
 - 1 - Choix de la console opérateur
 - 2 - Utilisation de la console opérateur
- Liste des commandes

سَمَاءُ

سَمَاءُ

سَمَاءُ

GENERALITES

1 - SYSTEME NOS

Le Frontal T1600 offre à ses utilisateurs (terminal local ou à distance, programme...) la possibilité d'entrer en communication avec des services distants réalisés hors du Frontal (sur un ordinateur connecté, par exemple).

Le système NOS (Network operating system) n'offre lui-même aucun service réel. Il établit puis contrôle les communications grâce à un certain nombre de "serveurs virtuels spécialisés". Ce sont les représentants des services réels distants. Ils peuvent opérer concurremment et sont chargés de mettre en relation ces services réels et les utilisateurs.

2 - TERMINAL

a) vue du système

Un terminal connecté physiquement au T1600 est en relation avec le système par l'intermédiaire d'une voie logique et est connu sous un numéro : l'unité symbolique (SU).

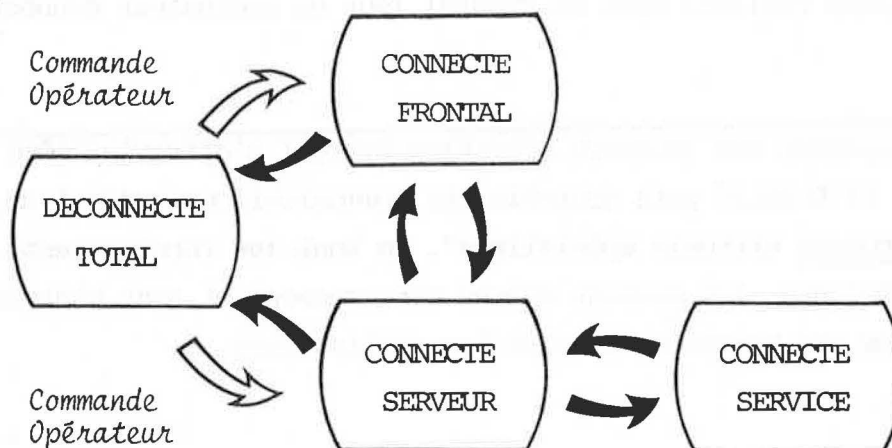
Ainsi, pour tout ce qui concerne ses relations avec le frontal, on nommera le terminal par son unité symbolique.

b) états

Un terminal du type console peut se trouver dans l'un des états suivants :

Déconnecté total	Il est ignoré de NOS. En particulier, l'utilisateur ne peut y entrer aucune commande
Connecté frontal	Il peut dialoguer avec NOS, et en particulier demander sa connexion à un serveur.
Connecté serveur	Il peut dialoguer avec le serveur virtuel et en particulier demander sa connexion à un service.
Connecté service	Il est directement en relation avec le service distant concerné.

On peut résumer ces états et le passage de l'un à l'autre par le diagramme d'états suivant.



Le passage de l'état "Déconnecté total" à un autre état nécessite l'intervention de l'opérateur.

COMMENT OPERER

1 - CHOIX DE LA CONSOLE OPERATEUR

On peut choisir la console opérateur soit :

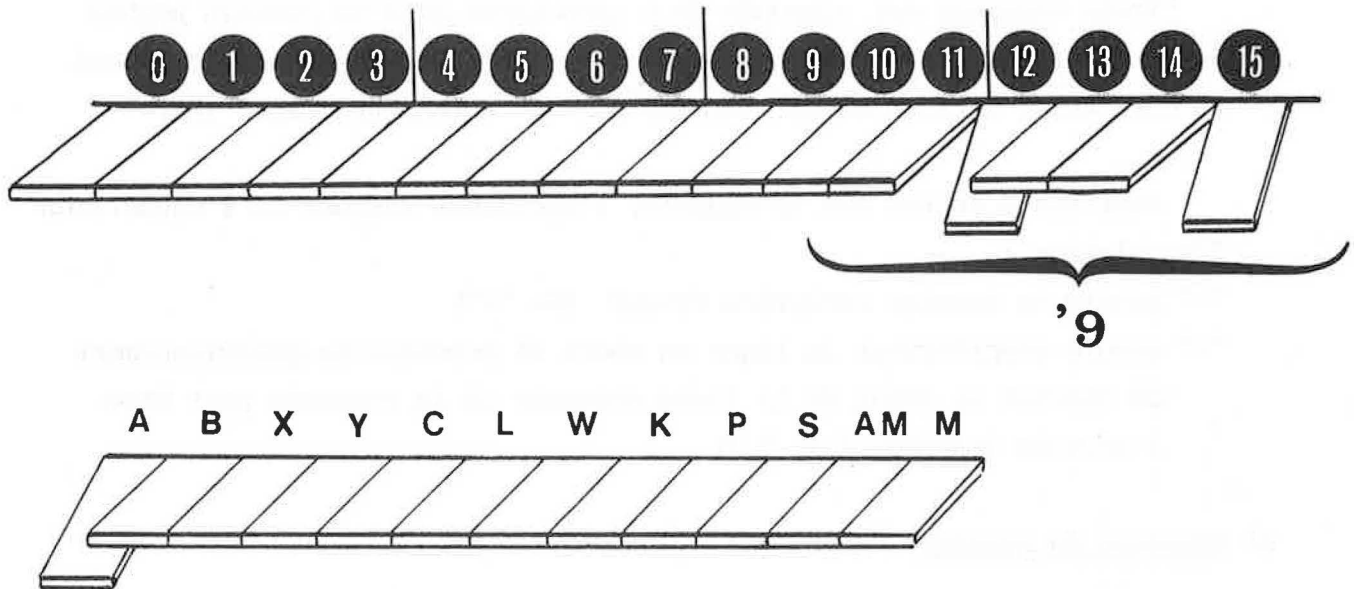
- avant le chargement du système en mémoire
- avant la réinitialisation et la relance du système
 - . par la commande "INOS" frappée sur l'ancienne console
 - . par l'action des clés du pupitre : ARRET, INI, MARCHE.

Pour cela, on positionne les clés du pupitre de commande :

- clé de fonction A abaissée
- configuration du numéro d'unité symbolique correspondant à la console choisie sur les clés de données (en hexadécimal) :

Exemple : on choisit la console correspondant à l'unité symbolique (SU)

N° 9



Remarque : si le n° configuré ne correspond pas à une console, tous les voyants s'allument. On recommence alors l'opération.

La clé A étant levée, c'est le Télétype de service qui est prise par défaut comme console opérateur.

2 - UTILISATION DE LA CONSOLE OPERATEUR.

Au démarrage du système, seuls sont connectés au frontal : la console opérateur et le disque à tête fixe.

Tous les terminaux et lignes sont dans l'état "déconnecté total". Les services sont inactifs.

L'opérateur peut lancer l'ensemble des commandes énumérées dans la suite.

a) Entrée d'une commande

Pour entrer une commande au système, l'opérateur doit appuyer sur la touche "ESCAPE" (Touche spéciale "APPEL" dans le cas de la télétype de service) de la console opérateur, et attendre l'impression du caractère "?". Le superviseur est alors en attente de la commande.

Toute commande est composée de 4 caractères et d'un certain nombre de paramètres séparés par le caractère "/" (slash). L'ensemble comprend 64 caractères maximum et est terminé par un "retour charriot" (RC)

Pour toute entrée sur la console, l'opérateur dispose de 2 caractères d'annulation :

"↑" annule le dernier caractère frappé (ou "^")

"←" annule complètement la ligne en cours et provoque le positionnement du chariot en début de la ligne suivante où la commande peut être à nouveau frappée. (ou "_")

b) Réponses du système

A la suite de chaque commande, le système envoie une réponse sur la console, précédée de l'heure.

Cette réponse peut être de trois types :

}	<u>hhH.mm DONE</u>	Commande exécutée
	<u>hhH.mm PASSED</u>	Commande prise en compte, initialisée et lancée.
	<u>hhH.mm ppp : xxx...</u>	Erreur détectée par le processeur ppp. La commande (ou partie de la commande) est refusée. La raison du refus : xxxxx ... est exprimée en clair.

Note : L'opérateur peut lancer, à partir de la console, toutes les commandes énumérées dans la suite.

Un terminal banal (console autre que la console opérateur) dans l'état "connecté frontal" ou l'état "connecté serveur" peut également lancer des commandes au système. Mais seulement certaines d'entre elles.

Ces commandes *non privilégiées* (ou partie de commande) sont figurées encadrées.

LISTE DES COMMANDES

- NOTE : : - les réponses du système sont soulignées
- les retours charriot sont notés (CR)
- les commandes (ou parties de commande) non privilégiées sont encadrées.

A - COMMANDES GENERALES

INOS (CR)	Réinitialisation de l'ensemble du logiciel du frontal
<u>DATE</u> [J] / [Mo] / [A] / [H] / [Mi] (CR)	Mise à jour et visualisation de la date et de l'heure. Les paramètres sont : jour, mois, année (2 chiffres), heure, minute.
OPEN <Su ₁ > [/ <SU ₂ > / <SU ₃ > ...] (CR)	Ouverture des voies logiques correspondant aux unités symboliques SU ₁ , SU ₂ , SU ₃ ... et attachement de ces unités ³ au frontal Cette commande fait passer les terminaux auxquels sont affectées les unités symboliques de l'état "déconnecté total" à l'état "connecté frontal"
<u>CLOS</u> <SU> (CR)	Fermeture de la voie logique correspondant à l'unité symbolique <SU> et détachement du frontal. Cette commande fait passer le terminal auquel est affectée l'unité symbolique <SU> à l'état "déconnecté total".
SRLD <ad> (CR)	lance l'écriture en mémoire d'une image mémoire sur disque à tête fixe, située à partir de l'adresse <ad> .

B - COMMANDES RELATIVES AUX SERVICES

a) Station de Transport Cyclades (MST)

<p>IMST (CR)</p>	<p>Initialisation et démarrage de la Station de Transport</p>
<p>Les commandes suivantes ne peuvent être acceptées qu'après l'initialisation de la Station de Transport.</p>	
<p>ECHO <i - ST> (CR)</p>	<p>Envoi d'un pli "écho" à la Station distante d'identification <i-ST> . Si cette station est active, elle renvoie le pli écho et le système le visualise sous l'une des formes suivantes :</p> <p>- <u>Station interne d'un noeud Cigale</u> :</p> <p>... STI <i - ST> ... <u><pli écho en hexadécimal></u></p> <p>- <u>Station de Transport d'un hôte</u> :</p> <p><u>R - ECHO <i-ST></u></p>
<p>CIGA <i-ST> (CR) <message> (CR)</p>	<p>Envoi du <message> à la télécopie opérateur du noeud CIGALE d'identification <i-ST></p>
<p>CTOP <i - ab>/<i - ST> (CR) ⌘ <message> (CR) ⌘ <message> (CR) ⌘ (CR)</p>	<p>Activation du processus de dialogue inter-opérateurs pour des messages destinés à l'abonné <i - ab> de la Station <i - ST>.</p> <p>Le processus est désactivé par un message vide (retour chariot).</p>

b) Abonné client concentrateur de terminaux (ABCT)

INCT (CR)	Initialisation de l'ensemble du logiciel du concentrateur de terminaux								
Les commandes suivantes ne peuvent être acceptées qu'après l'initialisation du concentrateur.									
NSER <nom du service> /<i-ab>/<i-service>/ <i-ent1>/<SU ₁ >[/<i-ent2>/<SU ₂ >/<i-ent3>/<SU ₃ >] (CR)	<p>Déclaration d'un nouveau service distant avec :</p> <p>- <u>Paramètres locaux</u> :</p> <p><nom du service> nom sous lequel les utilisateurs nommeront le service concerné.</p> <p>Il comprend :</p> <table border="0"><tr><td rowspan="2">}</td><td><type de service></td><td>TS (temps partagé ou</td></tr><tr><td></td><td>RB (traitement par lots)</td></tr><tr><td rowspan="2">}</td><td><nom></td><td>4 caractères quelconques</td></tr></table> <p><SU₁>, <SU₂>, <SU₃></p> <p>numéros d'unités symboliques auxquelles on fait correspondre les entrées dans le service dans le cas du traitement par lot :</p> <p><SU1> correspond à la console <SU2> correspond au lecteur de carte <SU3> correspond à l'imprimante</p> <p>Dans le cas de temps partagé, une seule entrée et <SU1> = 0.</p> <p>- <u>Paramètres réseaux</u></p> <p><i-ab> identification de l'abonné serveur distant.</p> <p><i-service> identification du service</p> <p><i-ent1>, (<i-ent2>, <i-ent3>) identification de l'entrée dans le cas du temps partagé (ou des entrées dans le cas de traitement par lots)</p>	}	<type de service>	TS (temps partagé ou		RB (traitement par lots)	}	<nom>	4 caractères quelconques
}	<type de service>		TS (temps partagé ou						
		RB (traitement par lots)							
}	<nom>	4 caractères quelconques							

<p><u>CXCT</u> <SU₁>[/<SU₂>/<SU₃>...] (CR)</p>	<p>Ouverture des voies logiques correspondant aux unités symboliques <SU₁>, <SU₂>, <SU₃>... et attachement de ces unités, au concentrateur. Cette commande fait passer les terminaux auxquels sont affectées les unités symboliques à l'état "connecté serveur" (Le serveur étant le concentrateur de terminaux).</p>
<p>MESS <SU> (CR) = <message> (CR)</p>	<p>Envoi d'un message sur la console à laquelle est affectée l'unité symbolique <SU>. Cette commande ne peut être acceptée que si le terminal est connecté au concentrateur.</p>

c) Traitements par lots hors CYCLADES (RB)

<p>INRB <SU₁>/<SU₂>/<SU₃> (CR)</p>	<ul style="list-style-type: none">- Initialisation du service.- Ouverture et connexion au service des périphériques auxquels sont affectées les unités symboliques <SU₁>, <SU₂>, <SU₃>. <p>Ces unités correspondent respectivement à :</p> <ul style="list-style-type: none"><SU₁> : console opérateur de traitements par lots.<SU₂> : lecteur de carte (entrée des données)<SU₃> : imprimante
<p>CLRB (CR)</p>	<ul style="list-style-type: none">- Désactivation du service.- Déconnexion des unités symboliques qui lui étaient attachées (et fermeture éventuelle si leur état antérieur n'était pas "connecté frontal").

d) Maintenance (MCE)

Toutes les commandes suivantes permettent de visualiser ou de modifier dynamiquement des éléments du système sans perturber son fonctionnement. (sauf modifications entraînant un fonctionnement anormal : en particulier avec la commande PTCH).

<p>DUMP <ad>/<n> (CR)</p>	<p>Vidage de <n> mots mémoire à partir de l'adresse absolue <ad>.</p>
<p>PTCH <ad>/<m> (CR)</p>	<p>Remplace le contenu du mot mémoire d'adresse <ad> par la valeur hexadécimale <m>.</p>
<p>MSGX <n> (CR) -> <message> (BELL)</p>	<p>Introduction du message de numéro <n> <hexadécimal> sur le disque. Le message est limité à 64 caractères, terminés par "Contrôle G" (Bell).</p>
<p>TEST <pu> (CR)</p>	<p>Mis en test de l'unité physique <pu> (correspondant à un coupleur de transmission). Trace dans un buffer circulaire tous les caractères émis et reçus sur cette unité physique.</p>
<p>ETAT <n> /<r> (CR)</p> <p><i>(voir le détail des tables et piles visualisées dans le Manuel de Maintenance).</i></p>	<p>visualisation d'une table ou d'une pile sous forme de vidage mémoire. C'est l'état à un instant donné de cette table ou de cette pile. <n> est le niveau de la "Machine" concernée. <r> est le rang de la machine dans le niveau.</p> <p>Le tableau suivant donne le détail des états suivants <n> et <r>.</p>

valeur de <n>	valeur de <r>	Etat associé
0	N° de P.U.	Table de l'unité physique PU (associée au coupleur d'entrée-sortie)
1	N° de P.U.	Zone de travail de la machine M:TR associée.
2	N° de P.U.	Zone de travail de la machine M:GT associée.
3	N° de F.U.	Table de l'unité fonctionnelle F.U. (associée au processus gérant les entrées/sorties d'une voie logique).
4	0	Files d'attente de la machine M;DM
	1	File des réveils.
	2	Table de gestion du disque.
	3	Liste des voies logiques connectées au frontal.
5	Diminutif de la voie virtuelle (référence locale)	Zone de travail de la voie virtuelle concernée

<p>DEBG <n>/<SU> (CR)</p>	<p><n> = 1 Trace sous forme de vidage mémoire de tous les plis émis et reçus de la Station de transport CYCLADES. Visualisation sur le terminal auquel est affectée l'unité symbolique <SU></p> <p><n>=0 Suppression de la trace</p>
---------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ANNEXE III

MANUEL UTILISATEUR

בְּחֵן הַיָּד

בְּחֵן הַיָּד וְהַלֵּב

בְּחֵן הַיָּד

AVERTISSEMENT.

Ce manuel est un complément du manuel opérateur. Pour toutes les commandes accessibles sous NOS, se reporter à ce Manuel.

On s'intéressera uniquement aux commandes accessibles dans l'état "connecté serveur", pour chacun des serveurs virtuels du Frontal. Cet état global peut être décomposé en états élémentaires suivant les serveurs.

Dans l'état "connecté service", on se reportera au manuel d'utilisation du service distant connecté.

ABONNE CLIENT CONCENTRATEUR DE TERMINAUX (ABCT)

A - SERVICES DISPONIBLES

- Temps partagé
 - Traitements par lots
- } sur le réseau CYCLADES

Voir la liste des services initialisés par l'opérateur et leur mnémoniques : <nom du service>.

Exemples :

TSCPDC	service documentation sous CP/67 (360/67)
TSCPTE	temps partagé sous CP/67 (360/67)
TSGREN	temps partagé sous SIRIS 8 (Grenoble)
TSLYON	temps partagé sous SIRIS 8 (Lyon)
TSIRIA	temps partagé sous SIRIS 8 (IRIA)
RBCP67	traitement par lots sous OS/MVT (360/67)
..etc..	

B - ENTREE D'UNE COMMANDE AU CONCENTRATEUR

Pour entrer une commande au concentrateur, l'utilisateur doit appuyer sur la touche "ESCAPE" de son terminal et attendre l'impression du caractère "/". Le concentrateur est alors en attente de la commande (Si le terminal était en attente d'entrée de données, frapper "retour chariot" après "ESCAPE").

Les caractères d'annulation de caractère ("↑" ou "∧") et de ligne ("←" ou "→") sont les mêmes que pour les commandes sous NOS.

C - TERMINAL CONNECTE A UN SERVICE DISTANT

L'état d'attente d'entrée d'un terminal connecté à un service distant est caractérisé par l'apparition du caractère "-" suivi de la "cloche". (C'est l'équivalent du déblocage du clavier sur certains terminaux). Dans le cas du secret (mot de passe par exemple), les caractères frappés n'apparaissent pas sur le terminal.

D - COMMANDES DISPONIBLES POUR TOUS LES TERMINAUX

(connectés au concentrateur)

ETAT (RC)	Visualisation sous forme de vidage mémoire de la table affectée au terminal (lorsque le terminal est déjà connecté au service). Pour la signification de chacun des mots de la table, se reporter au Manuel de Maintenance.
OPER <message> (RC)	Envoi d'un message sur la console opérateur du Frontal.
CSER<nom du service> (RC)	Demande de connexion au service distant dont le mnémonique est <nom du service>. Attendre l'établissement (ou le refus) de cette connexion.
ATTN (RC)	Envoi d'une "attention" au service distant connecté.
PURG (RC)	Suppression de la liste sur le terminal. Nota : l'arrêt de l'impression est local. Il faut donc attendre le déblocage du clavier caractérisé par "-" (BELL).
DSER (RC)	Demande d'annulation de la connexion établie (ou demandée) avec un service distant.
OUTP (RC)	Détachement du terminal du concentrateur. Retour à l'état "déconnecté total" (ou à l'état "connecté frontal" si le terminal était au préalable connecté à NOS.

E - COMMANDES OPERATEUR DE TRAITEMENT PAR LOTS (via CYCLADES)

Outre les commandes générales précédentes, l'opérateur d'un service de traitement par lots distant dispose des commandes suivantes :

DEXP <nom du service> (RC) = <message> (RC)	Envoi d'un message à l'opérateur distant du service connecté.
LECT <nom du service> (RC)	Relance du lecteur de cartes pour un service de traitement par lots distant connecté
RIMP (CR)	Relance de l'impression du listage en cours au début de celui-ci.

F - ETAT "CONNECTE SERVEUR" (voir diagramme page suivante)

Il se décompose en 3 états :

a) Déconnecté local

Le terminal n'est connecté à aucun service.

Les seules commandes possibles sont :

{ CSER
OPER
ETAT
OUTP

b) Connexion en cours

Une commande de connexion à un service a été

lancée. Les seules commandes possibles sont :

{ DSER
OPER
ETAT

c) Connecté local

Le terminal est connecté à un service.

Les commandes possibles sont :

{ DSER
ETAT
OPER
PURG
ATTN

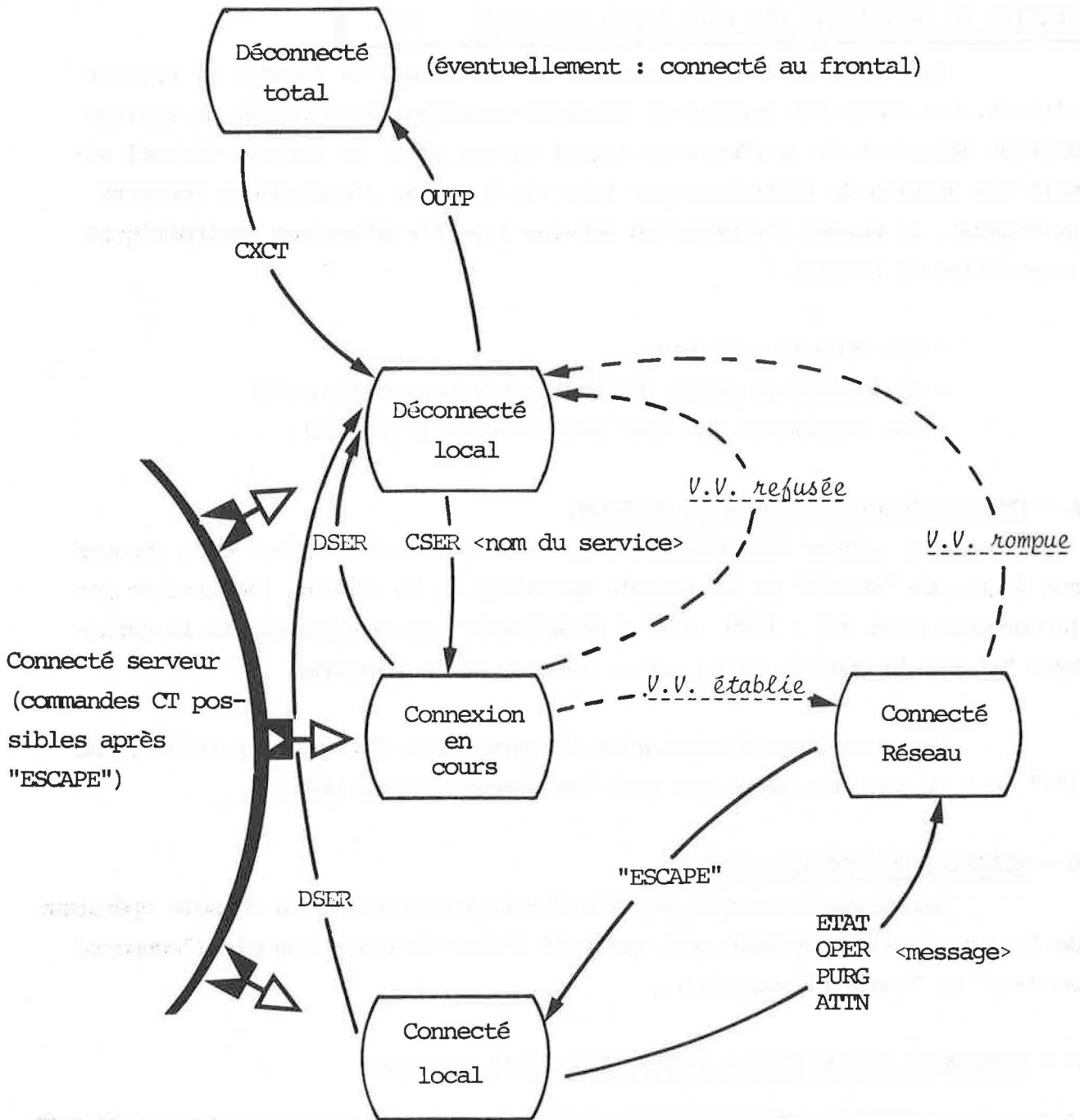


DIAGRAMME D'ETATS D'UN TERMINAL
VIS A VIS DU CONCENTRATEUR

SERVICE DE TRAITEMENT PAR LOTS (hors CYCLADES) (RB)

Ici, nous ne nous intéressons qu'aux commandes locales au serveur virtuel. Les commandes système et messages correspondants venant du service distant dépendent du système sous lequel il est géré. Le serveur virtuel simule une station de traitement par lots vis à vis de l'ordinateur connecté au Frontal. Il permet d'accéder au service à partir d'organes périphériques accessibles du frontal :

- une console opérateur.
- un lecteur de carte (ou tout autre organe d'entrée)
- une imprimante (ou tout autre organe de sortie).

A - ENTREE D'UNE COMMANDE A LA STATION.

Pour entrer une commande à la station, l'utilisateur doit appuyer sur la touche "ESCAPE" de la console opérateur de la station (déterminée par la commande sous NOS : INRB SUOP / SUCR/SUIMP). La station envoie le caractère "%" sur la console et se met en attente de la commande.

Les caractères d'annulation de caractères ("↑" ou "^") et de ligne ("←" ou "_") sont les mêmes que pour les commandes sous NOS.

B - MESSAGES DU SERVICE DISTANT

Les messages envoyés par l'ordinateur connecté à la console opérateur de la station sont imprimés quel que soit l'état de cette console ("connecté serveur" ou "connecté service").

C - COMMANDES DISPONIBLES A L'OPERATEUR DE LA STATION.

CORB (RC)	Provoque le passage de la console de l'état "connecté serveur" à l'état "connecté service". Au prochain "ESCAPE", le caractère "*" sera envoyé, et l'opérateur peut alors frapper une commande au système de l'ordinateur distant connecté. Pour sortir de cet état, il suffit d'envoyer une commande vide : "ESCAPE" (RC)
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

STRP (RC)	Arrête l'imprimante et provoque son passage à l'état "connecté serveur".
RERP (RC)	Relance l'impression du listage et provoque le passage de l'imprimante à l'état "connecté service".
STRC (RC)	Arrête le lecteur et provoque son passage à l'état "connecté serveur". (même action qu'une carte vide).
RERC (RC)	Relance la lecture des données et provoque le passage du lecteur à l'état "connecté service".
CLRB (RC)	Clôt la session de traitement par lot. Chaque terminal (console, lecteur, imprimante) repasse à son état initial ("déconnecté total" ou "connecté frontal").

D - UTILISATION DU "LECTEUR DE CARTES"

- ★ l'entrée des données peut se faire par l'intermédiaire :
 - . d'un lecteur de cartes
 - . d'un lecteur de ruban perforé
 - . d'un terminal interactif (télétype ou écran)Cette entrée est désignée par son unité symbolique (SU dans la commande INRB).

- ★ Une carte est constituée de 80 caractères ASCII au plus et terminée par un "retour chariot" (RC) (Eventuellement suivi ou précédé d'un "passage à la ligne" (LF)).

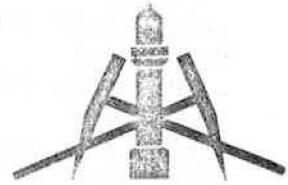
- ★ Dans un flot de cartes, la dernière lue est une carte vide (carte contenant (RC) comme premier caractère). La relance du lecteur nécessite l'intervention de l'opérateur de la Station.
Attention : l'arrêt du lecteur ne signifie rien vis à vis du service connecté : la fermeture du flot d'entrée du service s'obtient par une carte "fin de flot" ou une commande opérateur.

- ★ Dans le cas où l'entrée se fait à partir d'un terminal interactif, les cartes sont frappées sur le clavier et les caractères sont imprimés sur la feuille (ou visualisés sur l'écran).
Après chaque entrée de carte (caractérisé par "retour chariot"), le serveur positionne le chariot en début de ligne suivante indiquant ainsi qu'il est à nouveau en lecture.

A N N E X E I V

EXEMPLES DE SESSIONS

CONSOLE-OPERATEUR DU FRONTAL



ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

* NOSINI-V15 *

?DATE 18/10/76/19/35/
19H.35 18/10/76
19H.35 DONE
?IMST
19H.35 DONE
?ECHO 480
19H.36 DONE
?ECHO 480
19H.36 DONE
R-ECHO-ST:0480
?INCT

19H.36 **** CT-CYCL UP ****

?NSER TSCPDC/FFFF0480/CP67/DOC/0
?NSER TSCPIN/FFFF0480/CP67/INIT/0
?NSER TSCPTE/FFFF0480/CP67/TERM/0
?NSER TSIRIA/10282/SSTSIRIA/CT16/0
?NSER TSGREN/10482/SSTSGREN/CT16/0
?NSER TSLYON/10483/SSTSLYON/CT16/0
?OPEN 7/9/B/C

19H.43 DONE

AP 0B ATTACHE AU CT

AP 09 ATTACHE AU CT

POUR AP:0B

..CONNEXION TSGREN DEMANDEE

AP 07 ATTACHE AU CT

POUR AP:0B

..CONNEXION TSGREN ETABLIE

AP 0C ATTACHE AU CT

POUR AP:09

..CONNEXION TSCPDC DEMANDEE

POUR AP:07

..CONNEXION TSCPTE DEMANDEE

POUR AP:09

..CONNEXION TSCPDC ETABLIE

POUR AP:07

..CONNEXION TSCPTE ETABLIE

POUR AP:09

..CONNEXION TSCPDC ANNULEE

POUR AP:09

..CONNEXION TSIRIA DEMANDEE

?INRB F/A/11

19H.48

***** RB-P1175 UP *****



ÉCOLE NATIONALE SUPÉRIEURE
DES MINES DE SAINT-ÉTIENNE

POUR AP:09
..CONNEXION TSIRIA REFUSEE
POUR AP:09
..CONNEXION TSCPIN DEMANDEE
POUR AP:09
..CONNEXION TSCPIN ETABLIE
POUR AP:09
..CONNEXION TSCPIN ANNULEE
POUR AP:09
..CONNEXION TSCPTE DEMANDEE
POUR AP:0B
..CONNEXION TSGREN ROMPUE
POUR AP:09
..CONNEXION TSCPTE ETABLIE
19H.58

***** RB-P1175 DOWN *****

POUR AP:09
..CONNEXION TSCPTE ANNULEE
POUR AP:09
..CONNEXION TSGREN DEMANDEE
POUR AP:09
..CONNEXION TSGREN ETABLIE
POUR AP:07
..CONNEXION TSCPTE ANNULEE
AP 07 DETACHE DU CT

?MESS 9

-ATTENTION: SESSION "NOS-FRONTAL" TERMINEE DANS 5 MINUTES!!!

DE 09 BIEN RECU, JE DECONNECTE DANS 2 MINUTES

AP 0B DETACHE DU CT

POUR AP:09
..CONNEXION TSGREN ROMPUE
POUR AP:09
..CONNEXION TSLYON DEMANDEE
POUR AP:09
..CONNEXION TSLYON REFUSEE
AP 09 DETACHE DU CT

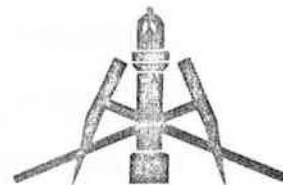
?ETAT 4/3

20H.31 ** VOIES LOGIQUES CONNECTEES **

DEVICES	SU	FU	PU	COUPLEURS
TTY DE SERVICE	1	1	8	SPEC.TTY
CYCLADES INPUT	6	6	A	SYN 2
CYCLADES OUTPUT	6	7	A	SYN 2
DISQUE	8	A	7	A.D.M.
TTY 7 (VOIE 7)	B	D	15	MUX 5
TTY 5 (VOIE 5)	C	F	13	MUX 5

20H.31 DONE

TERMINAL-USAGER EN SESSION CYCLADES



ÉCOLE NATIONALE SUPÉRIEURE
DES MINES DE SAINT-ÉTIENNE

19H.43 *** NOS: TERMINAL CONNECTE ***

?cxct

AP 09 ATTACHE AU CT

19H.44 **** CT-CYCL UP ****

/cser tscpd

..CONNEXION TSCPDC DEMANDEE



1

..CONNEXION TSCPDC ETABLIE

BIENVENUE SOUS LE SYSTEME DE DOCUMENTATION DU CIGG

CE SYSTEME MET A VOTRE DISPOSITION TROIS COMMANDES:

QUITTER POUR SORTIR DU SYSTEME DE DOCUMENTATION

IMPRIMER SITE NOM

SORT LE FICHIER DE NOM 'NOM' SUR L'IMPRIMANTE

DU SITE 'SITE'

SITE EST UN IDENTIFIACATEUR DE SITE DECLARE DANS

L'ANNUAIRE CYCLADES (EXEMPLE CP67 POUR L'IMAG)

LISTER NON

IMPRIME LE FICHIER SPECIFIE SUR VOTRE TERMINAL

/purg

-lister thermoda

LOAD PAPER; HIT RETURN.

-

T H E R M O D A T A

(Banque de donnees thermochimiques)

R; T=0.07/0.11 19:

46:46

-quitter

VOUS QUITTEZ LE SYSTEME DE DOCUMENTATION. SI VOUS N'AVEZ PAS
DE FICHIERS A SORTIR SUR IMPRIMANTE, VOUS POUVEZ ROMPRE LA
CONNEXION.

SINON, CONNECTEZ VOTRE IMPRIMANTE POUR RECUPERER VOS LISTES,
PUIS ROMPEZ LA CONNEXION.

0.2506 UNITE(S) DE FACTURATION DEPENSEE(S)

CONNECT= 00:01:57 VIRTCPU= 000:00.25 TOTCPU= 000:00.60

LOGOUT AT 19:47:28 ON 10/18/76

/dser tscpd

..CONNEXION TSCPDC ANNULEE



1



/cser tsiria

..CONNEXION TSIRIA DEMANDEE

← 2

..CONNEXION TSIRIA REFUSEE

← 2

19H.49 CONX: 5 EXPEDITIONS LT-RG "D-CONTACT" SANS AR

/cser tscpinit

19H.50 ABCT: PARAMETRE INCORRECT

/cser tscpin

..CONNEXION TSCPIN DEMANDEE

← 3

..CONNEXION TSCPIN ETABLIE
BIENVENUE SOUS CP/67 ...

SI VOUS AVEZ DES PROBLEMES, APPELEZ LE 54.61.45 POSTE 204

VOICI LA PROCEDURE A SUIVRE POUR OBTENIR LA CONNEXION
A LA MACHINE 'PUBLIC RELATION'.

SUR CETTE MACHINE, VOUS TROUVEREZ:

-UN MINI GUIDE CP/CMS

-LES INSTRUCTIONS POUR DECLARER VOTRE MACHINE DE TRAVAIL

/purg

/attn

/attn

-

/dser tscpin

..CONNEXION TSCPIN ANNULEE

← 3

/cser tscpte

..CONNEXION TSCPTE DEMANDEE

← 4

..CONNEXION TSCPTE ETABLIE
CP-V3.L1 a votre service

/attn

-login public
ENTER PASSWORD:

-

VEUILLEZ RECUPERER VOS ETATS DISQUES BANDES SERVICE DANS VOS CASIERS.
READY AT 19:55:50 ON MONDAY 10/18/76

CP

-q users!

-q nq^ames

09 USERS, 00 DIALED



```
-i cms
CMS..VERSION 3.0 octobre 74
-listf
FILENAME FILETYPE MODE NO.REC. DATE
DECLARE SYSIN P1 6 1/14
REPOSE SYSIN P1 6 1/14
DECLARE MODULE P1 1 1/14
IMPRIME MODULE P1 1 1/14
IMPRIME SYSIN P1 8 2/16
EXPLIQ SYSIN P1 6 1/14
```

```
/purg
-p declare sysin
```

```
DECLARE START 0
LR R12,R15
USING DECLARE,R12
TYPE 'ENTREZ VOS COORDONNEES'
```

```
/attn
SVC X'CA'
```

```
/etat
```

```
20H.01 ABCT: AP CONNECTE AU RESEAU
1F31: 0004 0000 0000 0000 0000 0900 FFFF 0480
1F39: 0054 0020 0000 21C8 B73A 000F 000F 0001
1F41: 0000 B7C1 001A 001A 0000 B7C1 0000 0000
1F49: 0000 0000 B7EE 0000 0000 0026 0000 0000
1F51: 0000 0000 0000 020A 0009 2237 0000 0000
1F59: 0000 0300 0000 0009 0000 0009 0000 0000
1F61: 0000 0001
```

```
/attn
DC AL4(ERLEC)
LH R2,COUNT+1
LTR R2,R2
BZ FIN
LA R1,WRITE
SVC X'CA'
DC AL4(EREGR)
B LECT
FIN MVC INPBUF(130),ETOILE
CP
-kt
```

```
R; T=0.05/0.09 20:02:48
```

```
/attn
CP
-logout
```

```
0.1064 UNITE(S) DE FACTURATION DEPENSEE(S)
CONNECT= 00:07:43 VIRTCPU= 000:00.09 TOTCPU= 000:00.25
LOGOUT AT 20:03:33 ON 10/18/76
```

```
/dser
```

```
..CONNEXION TSCPTE ANNULEE
```

/cser tsgren

..CONNEXION TSGREN DEMANDEE



..CONNEXION TSGREN ETABLIE

TS SYSTEM IS UP 18*10*76 20*06*54

!-l publ,public

LOGIN SUCCESSFUL. YOUR ID IS X001

!-command

*-keys

SYSTEM SECONDARY COMMANDS

ACCT:SYS	ALGOLRUN	ALGOLTS	ALGONLY	ALTERLMX	ANALM1	ANYTRA
BASIC	BASICI	BATCH	BIGANAL	CARTIMPR	CHANGE	COBOLTS
COMMAND	COMP	CONVLMEX	COP:CII	COPSTOK	COPYSR	COPY
COPYSTC0	COPYSTC1	COPYSTC2	COPYSTC3	COP0554	COUCOU	CRETABSP

/purg

-planconv

UNKNOWN COMMAND

*-

!-planconv

L ETABLISSEMENT DU PLANNING EST FAIT LE VENDREDI
VERS 14 HEURES
PLANCONV DONNE LES HEURES RESERVEES POUR
LA SEMAINE COURANTE, DU LUNDI AU VENDREDI A 14 H.

/purg

-lundi,mercredi

LUNDI 18

S8 - TRANSIRIS SAUVEGARDES

08H -22H 22H -23H

MARDI 19

BOGO/S8-TRANS S8 - TRANSIRIS SAUVEGARDES

08H -10H 10H -22H 22H -23H

MERCREDI 20

BOGO/S8-TRANS S8 - TRANSIRIS INSPECTION S8 - TRANSIRIS

08H -10H 10H -14H 14H -18H 18H -22H

SAUVEGARDES

22H -23H

TERMINE

TAPEZ VOTRE COMMANDE FORMAT : JOUR DEBUT[,JOUR FIN] [(SYSTEME)]
?-fin



```
!-basic
!FILE=?!-
V0401 C9CP00 00/000/201
>-let i=1
>-10 i=i+1
>-20 print 'i=',i
>-30 if i=10 then 50
>-40 goto 10
>-50 stop
>-goto 10
  I= 2
  I= 3
  I= 4
  I= 5
  I= 6
  I= 7
  I= 8
  I= 9
  I= 10
STOP AT LINE 50
>-end
!-clock
20:20 18 OCT 76
!-execute publicl,*
```

OPER:
ATTENTION: SESSION "NOS-FRONTAL" TERMINEE DANS 5 MINUTES!!!

```
JOB T940 ENTERED
!-status t940
T940 ACTIVE      CARDO,0000 LISTO,0001 RTIME,00*01*59 A HOLD
!-
```

/oper bien reçu,jedeconnecte dans 2 minutes

```
/attn
JOB T940 TERMINATED
!-edit
*-query
/J:LMPUBLIC          /T940          /:FDY          /J:COMI
/LMCRECUMU          /J:CRECUMU        /:CATLGCARDS   /:CTRLCARDS
/J:COMM             /J:CUMU           /J:CLIE
*-
*-end
!-charge
CPU TIME              0.07
TY TIME               21.75
CPU TIME*CORE         0.94
SWAP TIME*CORE        469.50
TEMPORARY TIME*QTM    0.05
IOS CALL              238
IOS BYTE              328118
TY CALL               219
TY BYTE               9641
!-logout
SAVE:(Y/N)?
!-n
  NB DE PTS COMPTABLES = 000000.99
LOGOUT DONE AT 20*29*46
```

..CONNEXION TSGREN ROMPUE

/cser tslyon

..CONNEXION TSLYON DEMANDEE



/etat

20H.28 ABCT: AP EN COURS DE CONNEXION A UN SERVEUR

..CONNEXION TSLYON REFUSEE



20H.28 CONX: 5 EXPEDITIONS LT-RG "D-CONTACT" SANS AR

/etat

20H.29 ABCT: AP CONNECTE AU CT

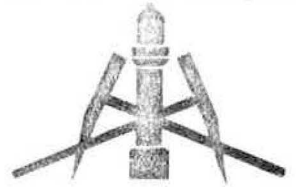
/outp

AP 09 DETACHE DU CT

?close

20H.30 *** NOS: TERMINAL DECONNECTE ***

CONSOLE-OPERATEUR D'UNE STATION "TERMINAL LOURD DU P1175"



ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

19H.48

***** RB-P1175 UP *****

%corb

*statsys 7

I99006 19.50 TN: 006,008,009,021

M00005 19.50 DONE

*statust 8

I99001 19.50 LN000,RC,HWA=018,DOWN,STOP=N,JOB=NONE,RPL=N,Q=N,SC=0
07,T=BI

M00005 19.50 DONE

*upt 8

M00005 19.50 DONE

*intn 8

M00005 19.50 DONE

001,A03016 19.50 IR165 READY FOR TN 008

*1,ok

*

%rerc

19H.50 RB: LECTEUR ACTIVE

I01102 19.52 JOB 0054-005,NAME=DC-NOS ,USER=21001

I03011 19.52 IR165 ENDED 001 JOBS,RB 005

%strc

19H.52 RB: LECTEUR DESACTIVE

%corb

*detail 54

M01004 19.52 0054-005 21001 DC-NOS 09 76-10-18,19.52 OUTPUT
QUEUE

*upt 9

M00005 19.53 DONE

I03024 19.54 TN009 OW166 FOR 0054-05 PROLOG

*

%rerp

19H.55 RB: IMPRIMANTE ACTIVEE

I03024 19.55 TN009 OW167 FOR 0054-05 01 FORTRAN

I03032 19.55 EOJ 0054-005

I03020 19.55 TN009 END OF SYSOUT

%corb

*downt 8

M00005 19.57 DONE

*downt 9

M00005 19.57 DONE

*

%clrb

19H.58

***** RB-P1175 DOWN *****

BIBLIOGRAPHIE

TELE-INFORMATIQUE ET RESEAUX : GENERALITES

- GEN-1 : "Computer Communication Networks"
Prentice-Hall Electrical Engineering Séries, Edit. by Franklin F. KUO (1973)
- GEN-2 : D.W. DAVIES & D.L.A. BARBER (NPL)
"Communication Networks for Computers"
John Wiley and Sons, New-York (1973)
- GEN-3 : L. POUZIN
"Architecture et variété de réseaux"
Congrès AFCET, Rennes (1973)
- GEN-4 : L. POUZIN
"Network Architectures and Components"
1st European Workshop on Computer Networks, Arles (avr. 73) pp. 153-174
Document CYCLADES SCH 516
- GEN-5 : R. MAHL
"Téléinformatique et réseaux d'ordinateurs"
Ecole d'été AFCET, Tarbes (Juil. 74)
- GEN-6 : Y. FEINROTH, E. FRANCESCHINI & M. GOLDSTEIN
"Télécommunications using a front-end Mini-computer"
Communications of the ACM 16,3 (mars 73) pp. 153-160
- GEN-7 : M. TOBIAS & G. BOOTH
"The future of remote information processing systems"
Proceedings of AFIPS - FJCC, vol.41(1972) pp. 1025-1035
- GEN-8 : H. ZIMMERMANN
"Réseaux informatiques et normalisation"
Congrès AFCET - GIF SUR YVETTE (nov. 76)

PROCEDURES DE TRANSMISSION ET PROTOCOLES

- PRO-1 : D. BJORNER
"Finite State Automation - Definition of data communication line control procedures"
Proceedings of AFIPS - FJCC, vol. 37 (1970) pp. 477-491
- PRO-2 : "High Level Data Link Control Procedure" (HDLC)
ECMA/TC9/73/76 (nov. 73)
- PRO-3 : L. POUZIN
"Basic elements of a Network Data Link Control Procedure" (NDLC)
Document Cyclades TRA 515 (Janv. 74)
- PRO-4 : L. POUZIN
"Structure d'une procédure de transmission point à point"
Document CYCLADES TRA 520 (déc. 1974)
- PRO-5 : L. POUZIN
"An Integrated approach to Network Protocols"
Document Cyclades NCP 500 (nov. 74)
- PRO-6 : L. POUZIN
"Network Protocols"
NATO International Advanced Study Institute, Brighton (sept. 73)
Document Cyclades SCH 517

RESEAU ARPA

- ARP-1 : F.E. HEART, R.E. KAHN, S.M. ORNSTEIN, W.R. CROWTHER & D.C. WALDEN
"The Interface Message Processor for the ARPA Computer Network"
 Proceedings of AFIPS-SJCC, vol 36 (1970) pp. 551-568
- ARP-2 : S. CARR, D. CROCKER & V. CERF
"Host-Host communication protocol in the ARPA network"
 Proceedings of AFIPS-SJCC, vol. 36 (1970) pp. 589-599
- ARP-3 : F.E. HEART, S.M. ORNSTEIN, W.R. CROWTHER, H.K. RISING, S.B. RUSSEL & A. MICHEL
"The terminal IMP for the ARPA computer network"
 Proceedings of AFIPS-SJCC, vol. 40 (1972) pp. 243-254
- ARP-4 : S. CROCKER, F. HEAFNER, R. METCALFE & J. POSTEL
"Function-oriented protocols for the ARPA computer network"
 Proceedings of AFIPS - SJCC, vol. 40 (1972) pp. 271-279
- ARP-5 : J.M. Mc QUILLAN, F.E. HEART, W.R. CROWTHER, B.P. COSSELL & D.C. WALDEN
"Improvements in the design and performance of the ARPA network"
 Proceedings of AFIPS - FJCC, vol. 41 (1972) pp. 741-753
- ARP-6 : F.E. HEART, S.M. ORNSTEIN, W.R. CROWTHER & W.B. BARKER
"A new minicomputer/multiprocessor for the ARPA network"
 Proceedings of AFIPS - NCCE, vol. 42 (1973) pp. 529-537
- ARP-7 : P.L. HIGGINSON, P.T. KIRSTEIN, A.V. STOKES
"The problem of connecting hosts into ARPANET via front-end computers"
 European Computer Workshop Series : "Distributed Computer Systems" Darmstadt
 (oct. 74)
- ARP-8 : R.G. ROBERTS & B.D. WESSLER
"Computer Network development to achieve ressource sharing"
 Proceedings of AFIPS - SJCC, vol 36 (1970) pp. 543-549

RESEAU CYCLADES

- CYC-1 : L. POUZIN
"Presentation and major aspects of the Cyclades computer network"
 3rd Data communication Symposium, TAMPA (nov. 73) pp. 80-87
- CYC-2 : L. POUZIN
"CIGALE, the packet switching machine of the Cyclades computer network"
 IFIP Congress (1974) Stockholm pp. 155-159
 Document CYCLADES MIT 556 (nov. 73)
- CYC-3 : H. ZIMMERMANN
"Protocoles de Communication"
 Journées ACM-AFCET sur les Télécommunications, PARIS (Juin 74)
 Document CYCLADES SCH 530
- CYC-4 : J.F. CHAMBON, M. ELIE, J. LE BIHAN, H. ZIMMERMANN
"Spécifications fonctionnelles des Stations de Transport du réseau CYCLADES"
 Document CYCLADES SCH 502.3 (Mai 73)
- CYC-5 : H. ZIMMERMANN
"Transport Protocol-Standard Host-Host protocol for heterogeneous computer network"
 IFIP WG 6.1., doc. 61 (avril 74)
 Document CYCLADES SCH 519.1
- CYC-6 : H. ZIMMERMANN
"Protocole client-serveur de terminaux"
 Document CYCLADES TER 502.2 (nov. 74)
- CYC-7 : H. ZIMMERMANN
"Virtual Terminal Protocol"
 Document CYCLADES TER 503.1 (nov. 74)
- CYC-8 : F. DENJEAN
"Connexion de terminaux à un réseau de commutation de paquets"
 Internal. meeting on mini-computers and data com. Liège (Janv. 75)
 Document CYCLADES TER 509.
- CYC-9 : J.F. CHAMBON & B. LE BIHAN
"Architecture d'un calculateur frontal en environnement télé-informatique"
 Séminaire IMAG - Grenoble (déc.75)
 Congrès AFCET - GIF SUR-YVETTE- (nov. 76) (sous le même titre)
- CYC-10: J.P. ANSART
"Connexion d'une machine à mémoire virtuelle "IBM 360/67" au réseau CYCLADES"
 Thèse de Docteur Ingénieur - USMG, GRENOBLE (Fév. 76)
- CYC-11: H. ZIMMERMANN
"Insertion d'une Station de Transport dans un système d'exploitation"
 Document CYCLADES SCH 546

AUTRES RESEAUX

- IBM-1 : J.H. Mc FAYDEN
"Systems Network Architecture : An overview"
IBM System Journal 15,1 (1976)
- IBM-2 P.G. CULLUM
"The transmission Subsystem in System Network Architecture"
IBM Systems Journal 15,1 (1976)
- IBM-3 W.S. HOBGOOD
"The role of the Network Control Program in System Network Architecture"
IBM Systems Journal 15,1 (1976)
- IBM-4 H.R. ALBRECHT & K.D. RYDER
"The virtual Telecommunications Access Method : A system Network Architecture perspective".
IBM Systems Journal 15,1 (1976)
- CDC-1 : D.J. MOSS
"The Control Data computer communication network"
European Computer Workshop series : "Distributed Computer Systems"
Darmstadt (oct. 74)
- DEC-1 : N.A. TEICHOLTZ
"Digital Network Architecture"
EUROCOMP, Brunel Univ. (sept 75), 13-24

CONCEPTION DE SYSTEMES

- SYS-1 : C.A.R. HOARE
"Monitors : An operating System Structuring Concept"
 Communications of the ACM 17, 10 (oct. 74) pp. 549-557
- SYS-2 : E.W. DIJKSTRA
"Cooperating Sequential Processes"
 Programming Languages, F. GENUYS ed., Academic Press Inc., New-york, 1968,
 pp 43-112
- SYS-3 : G.D. KNOTT
"A proposal for certain process management and inter-communication
 primitives"
 ACM, Operating Systems Review 9,1 (Janv. 75) pp. 19-41
- SYS-4 : D.L. PARNAS & D.P. SIEWIOREK
"Use of the Concept of Transparency in the Design of Hierarchically
 Structured Systems"
 Communication of the ACM 18,7 (Juil. 75) pp. 401-408
- SYS-5 : J.M. YOHE
"An Overview of Programming Practices"
 ACM, Computing Surveys 6,4 (déc. 74) pp. 221-246
- SYS-6 : E.W. DIJKSTRA
"Notes on Structured Programming"
 Technical University - Eindhoven - The Netherlands 1970.
- SYS-7 : N. WIRTH
"Program development by step wise refinement"
 Com. ACM 14,4 (Avril 70) p 221-227
- SYS-8 : W. WIRTH
"Systematic programming : an introduction"
 Prentice Hall Englewood Cliffs. New Jersey 1973.
- SYS-9 : KNUTH
"Structured Programming with GOTO Statement"
 Computing Surveys Vol.6 N° 4 (Dec 74) p 261-263
- SYS-10: DIJKSTRA
"GOTO Statement considered harmful"
 Comm. ACM 11,3 (Mars 68) 147-148, 538-541
- SYS-11: S. GUIBOUD-RIBAUD, C. OTRAGE
"Systèmes à la carte et multiprocesseur"
 Journées IRIA - Saint-Pierre de Chartreuse (nov. 73)
- SYS-12: S. GUIBOUD-RIBAUD
"Système de réseaux de mini-ordinateurs spécialisés"
 Note interne - Ecole des Mines de Saint-Etienne (déc. 75)
- SYS-13 : J.F. CHAMBON et B. LE BIHAN
"Logiciel d'entrée-sortie pour la téléinformatique"
 Séminaire Université Claude Bernard (Mai 1975)

REFERENCES TECHNIQUES

- TECH-1 : Manuel de présentation du calculateur T1600
TELEMECANIQUE
- TECH-2 : Data communications, séries P1000
PHILIPS
- TECH-3 : Procédures de transmission TMM
CII
- TECH-4 : Binary Synchronous Communications
IBM
- TECH-5 : Procédure de transmission multipoint ECMA-PHILIPS
PHILIPS
- TECH-6 : Implantation de la Station de Transport du réseau CYCLADES dans un
calculateur frontal.
Rapport final D.R.M.E. (Juil. 75) J.F. CHAMBON, B. LE BIHAN
- TECH-7 : POLYBUS 16
TELEMECANIQUE
- TECH-8 : Manuel de présentation de COMTEL
TELEMECANIQUE
- TECH-9 : Carnet de maintenance de NOS
Notes internes - Ecole des Mines de Saint.Etienne

**SOUTENANCE D'UNE THESE POUR L'OBTENTION D'UN
DIPLOME DE DOCTEUR - INGENIEUR**

NOM DES CANDIDATS : CHAMBON Jean-François
LE BIHAN Bernard

SUJET DE THESE : Architecture d'un frontal en environnement téléinformatique
(Application à CYCLADES)

SPECIALITE : Systèmes et Réseaux Informatiques

Vu le règlement intérieur relatif à la préparation des doctorats,

Vu les rapports de pré-soutenance établis par M. MAHL, Directeur de Recherche,
et M. OTRAGE, Ingénieur Télémechanique,

Vu les propositions de M. GUIBOUD-RIBAUD, responsable de la formation
pour la spécialité Systèmes et Réseaux Informatiques,

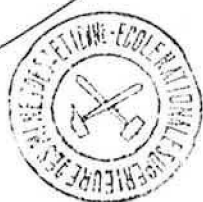
Le Directeur de l'Ecole Nationale Supérieure
des Mines de Saint-Etienne

1) autorise MM. CHAMBON Jean-François et LE BIHAN Bernard à soutenir
une thèse pour l'obtention d'un diplôme de docteur-ingénieur, sur
le sujet donné en référence,

2) désigne comme suit le jury de soutenance :

MM. BAZERQUE	Georges	Maître de conférences à l'Université des Sciences Sociales de Toulouse
BOLLIET	Louis	Professeur IUT Grenoble, Président
GUIBOUD-RIBAUD	Serge	Professeur E M S E
MAHL	Robert	Chargé de mission auprès du Préfet de la région Rhône-Alpes - Directeur de recherches
OTRAGE	Claude	Ingénieur Télémechanique Informatique

Ph. TRAYNARD
Président



Visa :



Je responsable
de la formation de
recherche
L. Bolliet

[Signature]

