

N° 324

Deux 75 Exempl. le 28-6-74

Sc. N. 74/51

CONNEXION ET COOPERATION

DE DEUX ORDINATEURS

CII.10070 - IBM.1800

THESE

POUR L'OBTENTION DU
GRADE DE DOCTEUR INGENIEUR
SOUTENUE LE 27 JUIN 1974
PAR

ANDRE SCHAFF



JURY

Président M. C. PAIR
Examineurs M. V. CORDONNIER
M. J.C. DERNIAME
M. R. HUSSON

CONNEXION ET COOPERATION

DE DEUX ORDINATEURS

CII.10070 - IBM.1800

THESE

POUR L'OBTENTION DU
GRADE DE DOCTEUR INGENIEUR
SOUTENUE LE 27 JUIN 1974
PAR

ANDRE SCHAFF



JURY

Président	M. C. PAIR
Examineurs	M. V. CORDONNIER
	M. J.C.DERNIAME
	M. R. HUSSON

Nous remercions vivement Monsieur le Professeur PAIR, Directeur de l'Institut Universitaire de Calcul Automatique de nous avoir proposé ce travail en commun et de nous faire l'honneur de présider ce jury.

Que Monsieur le Professeur HUSSON trouve ici l'expression de notre plus profonde gratitude pour l'accueil qu'il nous a réservé au laboratoire d'Automatique de l'E.N.S.E.M.

Nous remercions Messieurs les Professeurs DERNIAME et HUSSON pour les conseils et encouragements qu'ils n'ont cessé de nous prodiguer tout au long de ce travail.

Nous exprimons nos plus sincères remerciements à Monsieur le Professeur CORDONNIER pour l'attention qu'il a bien voulu porter à notre travail et pour l'honneur qu'il nous fait en participant à ce jury.

Nous remercions également Monsieur VILLARD pour l'intérêt qu'il a toujours porté à nos recherches, ainsi que Messieurs DENDIEN et DUCLOY.

Nous tenons à remercier Monsieur GADEL pour sa précieuse collaboration, Monsieur MUNIER pour l'aide qu'il nous a apportée dans les réalisations matérielles, ainsi que tous nos camarades et amis du laboratoire.

Que Madame HUSSON qui a assuré avec diligence la présentation de cet ouvrage trouve ici l'expression de notre sincère reconnaissance. Nous remercions aussi le personnel de l'Imprimerie du D.P.I.C. de l'Institut National Polytechnique de Lorraine qui a assuré la réalisation matérielle de cet ouvrage.

TABLE DES MATIERES

I N T R O D U C T I O N

CHAPITRE I - PROCEDURE DE TRANSMISSION

I - INTRODUCTION

II- COUPLEUR

II.1 Liaison Coupleur - IBM 1800

II.2 Emission

II.2.1 Fonctions dans le cas de la transmission ASCII

II.2.2 Fonctions dans le cas de la transmission mode binaire

II.2.3 Synchronisation

II.3 Réception

II.3.1 Fonctions dans le cas de la transmission ASCII

II.3.2 Fonctions dans le cas de la transmission mode binaire

II.3.3 Synchronisation

II.4 Hiérarchie des interruptions

III- MESSAGES

III.1 Vue d'ensemble des messages

III.2 Séquences de supervision

III.3 Blocs d'information

III.4 Accusés de réception

III.5 Demandes de suspension

IV - REPRESENTATION DE LA PROCEDURE DE TRANSMISSION

IV.1 Schéma simplifié

IV.1.1 Evènements extérieurs

IV.1.2 Evènements intérieurs

IV.2 Schéma réel

IV.2.1 Charges du terminal

IV.2.2 Demandes de suspension

IV.2.3 Reprises du central

IV.2.4 Test de la liaison

IV.2.5 Schéma complet

IV.3 Tables associées

IV.3.1 Table des états et des sorties

IV.3.2 Transitions état global → état partiel

CHAPITRE II - REALISATION

I - INTRODUCTION

II - ANALYSE GENERALE

III- PROGRAMME DIRECTEUR ET MODULE PASSIF COMMUN

III.1 Programme directeur EXAM

III.2 Module passif commun SPO

IV - POLLING

IV.1 Module SA₀

IV.2 Module SP₁

IV.3 Module SA₁

IV.4 Module SA₂

V - SELECTION

V.1 Module SA₃

V.2 Module SP₂

V.3 Module SA₄

V.4 Module SA₅

VI - TRAVAIL DE FOND

VI.1 En Polling

VI.2 En Sélection

VI.3 Initialisations

VII- ANALYSE DES MESSAGES

VII.1 Analyse des séquences de supervision

VII.2 Reconnaissance des messages d'acquiescement

VII.3 Reconnaissance de EØT

VII.4 Analyse des messages courants

VII.4.1 Elimination des caractères de synchronisation

VII.4.2 Analyse du préfixe de bloc

VII.4.3 Analyse du préfixe d'article

VII.4.4 Analyse des caractères courants

CHAPITRE III - PROCEDURE SYMETRIQUE

- I - INTRODUCTION
 - I.1 Phase initiale
 - I.2 Phase de transfert
- II - LES MESSAGES
 - II.1 Les séquences de supervision
 - II.2 Les blocs d'information
 - II.3 Les demandes de suspension
- III- REPRISES ET CONFLITS
 - III.1 Reprises
 - III.2 Conflits
- IV - ANALYSE ET REPRESENTATION DE LA PROCEDURE
 - IV.1 Niveau 1
 - IV.2 Niveau 2
 - IV.2.1 *Etat P_0*
 - IV.2.2 *Etat P_1*
 - IV.2.3 *Etats P_2 et P_3*
- V - PERSPECTIVES POUR LES PROCEDURES
 - V.1 Description
 - V.2 Implémentation
 - V.3 Conclusion

CHAPITRE IV - METHODE D'ACCES

- I - INTRODUCTION
- II - HYPOTHESES
- III- FONCTIONS
 - III.1 Idée directrice
 - III.2 Assignation
 - III.3 Ouverture
 - III.4 Emission - Réception
 - III.5 Tests
 - III.6 Fermeture
 - III.7 Communications avec opérateurs
- IV - SYNCHRONISATION
- V - GENERALISATION
- VI - APPLICATION A LA GESTION DU "REMOTE - BATCH"
- VII- REALISATION

CHAPITRE V - PROTOCOLE D'EXECUTION DE SOUS-PROGRAMMES A DISTANCE

- I INTRODUCTION
- II TABLE DES VALEURS ET ADAPTATION AUX FORMATS
 - II.1 Préfixes
 - II.2 Descripteur
 - II.3 Table des valeurs
 - II.4 Adaptation aux formats
 - II.5 Descripteurs de tableaux
- III Déclaration EXTERIEUR
 - III.1 Paramètres
 - III.2 Syntaxe
 - III.3 Compilation et table de correspondance
 - III.4 Exemple
- IV TACHES SUR LE SITE DEMANDEUR
- V LIAISON ENTRE SITE DEMANDEUR ET SITE FOURNISSEUR
- VI REMARQUES
 - VI.1 Cas de plusieurs appels du même sous-programme
 - VI.2 Cas d'appels de plusieurs sous-programmes
- VII CONCLUSION

CHAPITRE VI - TESTS ET MESURES

- I TESTS DU COUPLEUR
- II TESTS DES PROGRAMMES
 - II.1 Polling
 - II.2 Sélection
- III TESTS ET MESURES ENVISAGES
 - III.1 Tests
 - III.2 Mesures

C O N C L U S I O N

A N N E X E S

BIBLIOGRAPHIE

Ce travail est le résultat d'une collaboration étroite entre A. SCHAFF et J.P. THOMESSE, en liaison avec A. GADEL pour la partie technologique. Cette dernière est l'objet d'une thèse C.N.A.M. à paraître (9).

Nous nous sommes répartis les tâches de la façon suivante :

- La procédure de transmission et la méthode d'accès ont été analysées en commun.
- Pour la réalisation, A. SCHAFF s'occupa de la partie émission, J.P. THOMESSE de la partie réception. Néanmoins, un important travail commun a été nécessaire afin d'assurer l'enchaînement correct des divers modules et sous-programmes.
- Ensuite A. SCHAFF s'est plus particulièrement attaché à la schématisation et à la formalisation des procédures de transmission alors que J.P. THOMESSE s'occupait de la définition d'un protocole d'exécution de sous-programmes à distance.

Ce partage se retrouve dans la rédaction comme indiqué ci-dessous :

Chapitre I	§ I et II	J.P. THOMESSE
	§ III et IV	A. SCHAFF
Chapitre II	§ I à VI	A. SCHAFF
	§ VII	J.P. THOMESSE
Chapitre III	A. SCHAFF
Chapitre IV	J.P. THOMESSE
Chapitre V	J.P. THOMESSE
Chapitre VI	A. SCHAFF et J.P. THOMESSE

INTRODUCTION

Le travail proposé consiste à relier deux ordinateurs par ligne téléphonique : le C.I.I. 10070 de l'Institut Universitaire de Calcul Automatique de Nancy et l'I.B.M. 1800 de l'Ecole Nationale Supérieure d'Electricité et de Mécanique de Nancy.

Cette opération a deux buts :

- Le premier est d'employer l'I.B.M. 1800 comme terminal lourd pour l'entrée de travaux à distance (Remote-batch) sur le C.I.I. 10070.
- Le second est d'utiliser les deux configurations pour un même travail, c'est-à-dire que nous envisageons une coopération en temps réel des deux ordinateurs.

Comment concevoir et réaliser un système atteignant ces objectifs ?

Il faut d'abord construire un interface entre les deux ordinateurs et ensuite, en définir l'utilisation.

Le système se décompose en quatre parties :

- 1°) Transmission de données sur une ligne téléphonique
- 2°) Interface ordinateur - modem
- 3°) Méthode d'accès
- 4°) Protocole d'utilisation

1°) *Transmission de données*

Le support de transmission de données est constitué par une ligne téléphonique quatre fils fournie par les P.T.T. de qualité supérieure selon le gabarit M 102 du C.C.I.T.T. A chaque extrémité, les fonctions de modulation et démodulation sont assurées par deux modems identiques du type MØ 30 livrés par la C.I.I. (CII , {25}).

Le procédé de modulation est du type modulation de phase à codage quadrivalent, ce qui permet d'obtenir un débit binaire de 2.400 bits/seconde avec une rapidité de modulation de 1.200 bauds.

Le mode de transmission est du type asynchrone synchronisé : la transmission de tous les bits d'un message est synchrone, par contre les intervalles de temps variables séparant les messages introduisent un asynchronisme. Pendant ces intervalles de temps il n'y a pas de synchronisation.

Le modem récepteur se resynchronise sur le modem émetteur, à chaque message en détectant un préfixe constant. Cette détection assure l'asservissement de l'horloge réceptrice sur l'horloge émettrice.

2°) *Interface ordinateur - modem*

Cet interface est réalisé par software et par hardware sur chacun des deux ordinateurs. Sur C.I.I. 10070, nous utilisons le coupleur existant du type CTQM (CII , {25}) et la procédure de transmission TMM - RB (transmission en mode message - Remote batch).

Du côté I.B.M. 1800, le coupleur a été conçu en collaboration avec A. GADEL qui en a assuré la réalisation. Ses fonctions sont brièvement énoncées dans le chapitre I, § II et détaillées

dans le travail de A. Gadel (9).

La partie software est un ensemble de routines réalisant les fonctions qui ne le sont pas par hardware. Ces routines constituent la procédure de transmission qui est compatible avec celle de la C.I.I. , TMM - RB (CII , {23}).

Le chapitre I (§ III et suivants) est consacré à la description et à la schématisation de la procédure implémentée sur I.B.M. 1800. Sa réalisation est décrite dans le chapitre II.

Cette procédure présente quelques inconvénients pour réaliser les buts fixés : les transferts, en particulier, sont toujours initialisés par le même ordinateur (le central). Aussi, nous envisageons une autre, symétrique qui autorise chaque ordinateur à être tour à tour maître ou esclave.

Le chapitre III décrit cette procédure.

3°) Méthode d'accès

Nous voulons considérer l'ensemble coupleur, modem, ligne et procédure de transmission comme un simple périphérique en entrée et en sortie. Pour cela, nous définissons une méthode d'accès par un ensemble de fonctions qui devraient permettre à tout utilisateur de se servir de la liaison.

Le chapitre IV est consacré à l'analyse d'une telle méthode d'accès. En ce qui concerne sa réalisation, nous avons muni la procédure de transmission de quelques routines permettant l'entrée de travaux depuis l'I.B.M. 1800, sans avoir complètement implémenté la méthode d'accès.

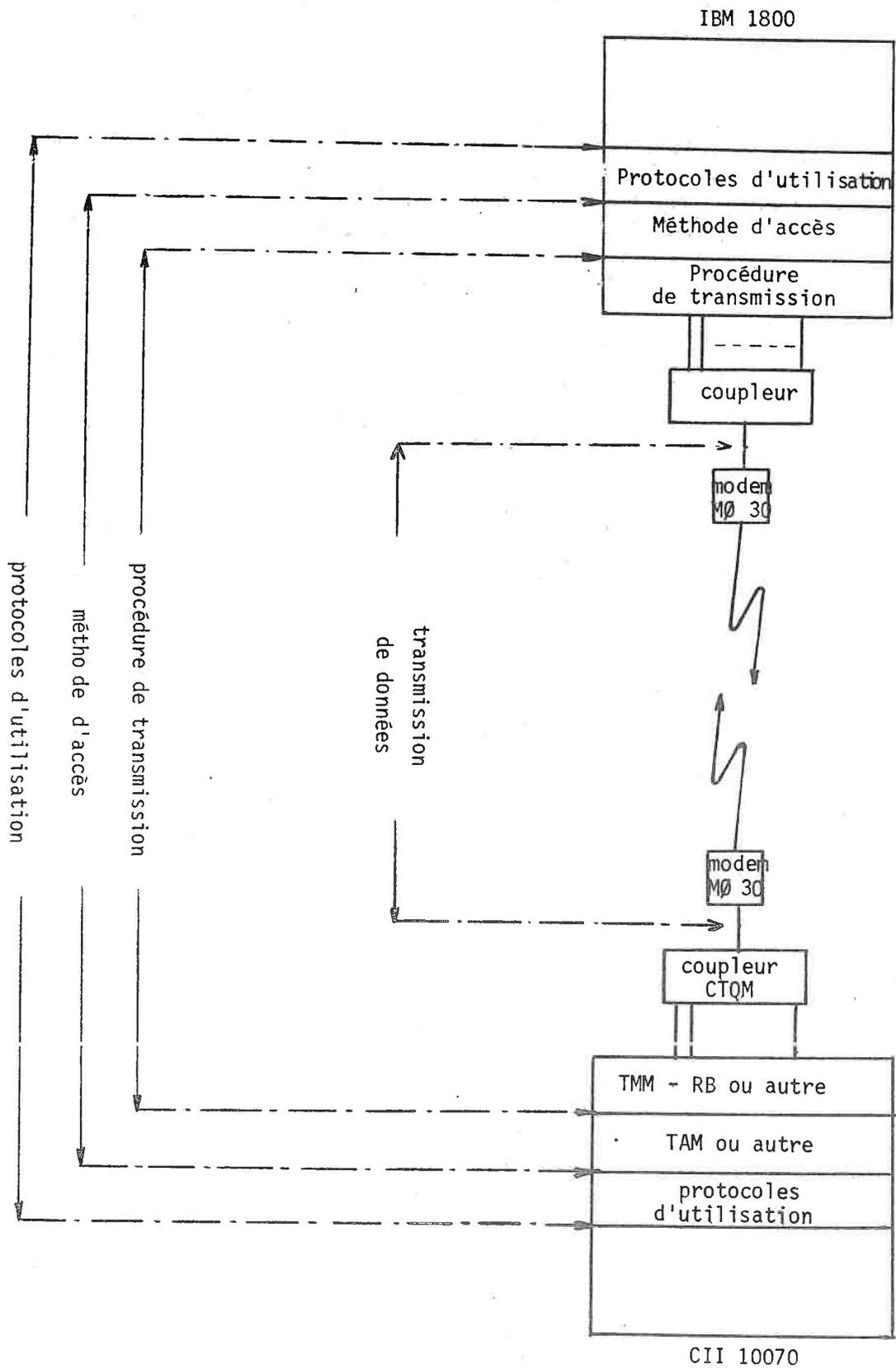
Dans le paragraphe V de ce chapitre, nous généralisons la méthode en définissant un protocole d'attribution de ressources à distances qui permet à deux processus distants de coopérer. Dans ce paragraphe, nous supposons que la procédure de transmission est symétrique, du type de celle qui est décrite dans le chapitre III.

4°) Protocole d'utilisation

En supposant que deux processus distants peuvent se voir allouer la liaison par l'intermédiaire de la méthode d'accès et du protocole d'attribution de ressources, nous envisageons une utilisation particulière de cette liaison : l'exécution de sous-programmes à distance. Le protocole d'exécution de sous-programmes à distance est décrit dans le chapitre V. Une application particulière serait la commande hiérarchisée d'un processus industriel (premier but de notre travail).

Dans un dernier chapitre, nous expliquons comment nous avons testé les réalisations hardware et software, comment nous envisageons les derniers tests lorsque la ligne téléphonique sera opérationnelle. Enfin, nous précisons comment nous pourrions effectuer quelques mesures élémentaires, relatives aux performances du sous système.

SCHEMA RECAPITULATIF



CHAPITRE I

PROCEDURE DE TRANSMISSION

I - INTRODUCTION

La procédure de transmission concerne la gestion de la ligne téléphonique. Cette gestion est réalisée par hardware et par software. Dans (11) J.P. GRAY énonce ces diverses fonctions.

L'étude et la réalisation de la partie hardware font l'objet des travaux de A. GADEL (9).

L'étude et la réalisation de la partie software comprennent deux volets :

- d'une part, les programmes d'entrées-sorties du coupleur et les programmes d'interruption ;
- d'autre part, le contrôle général de la transmission avec détection des erreurs, procédures de reprise, formatage des messages.

La plupart des fonctions réalisées par hardware auraient pu l'être par software. Si notre choix s'est porté sur une réalisation par hardware, c'est essentiellement pour optimiser le temps de traitement d'un message (cf. chapitre I - paragraphe II).

II - COUPLEUR

II.1 - Liaison coupleur-ordinateur

Notre première solution consistait à utiliser deux unités d'échanges simples (une en émission, l'autre en réception) spécifiques à la liaison téléphonique. Pour des raisons à la fois d'économie et de facilité de réalisation technologique, la solution retenue fut la suivante :

Nous utilisons les entrées sorties numériques de l'I.B.M. 1800. Les opérations d'entrées sorties se déroulent en mode canal avec synchronisation externe (I.B.M. 27).

Les informations en direction du (vers le) coupleur sont représentées par 16 bits en parallèle. Ces 16 bits correspondent exactement à un mot mémoire.

La synchronisation externe est un signal calibré en amplitude et en durée qui permet d'adapter les vitesses de fonctionnement différentes du coupleur (asservi à la ligne) et à l'unité d'échanges. Ce signal sera, dans notre cas, périodique, d'une période égale au temps d'émission ou de réception de deux octets ; il est émis par le coupleur, tous les 16 tops d'horloge émission ou réception du modem (C.I.I. 25) .

Les interruptions utiles sont transmises par l'intermédiaire des P.I.S.W. qui équipent l'I.B.M. 1800 (I.B.M. 27) , (I.B.M. 28).

II.2 - Emission

Sur une ligne téléphonique les informations circulent en série. A la sortie de l'ordinateur, elles sont présentes en parallèle. La première fonction du coupleur est d'effectuer la transformation parallèle série.

Cette transformation est réalisée par un registre à décalage de 8 bits donc caractère par caractère. Le registre à décalage est chargé alternativement avec le caractère se trouvant dans les poids forts du registre de sortie, puis avec le caractère se trouvant dans les poids faibles de ce registre de sortie.

Les informations échangées pouvant être codées de deux façons différentes (ASCII ou EBCDIC) les contrôles effectués par hardware sont différents dans chacun des cas, aussi nous les distinguons.

II.2.1 - Fonctions dans le cas d'une transmission ASCII

La détection d'erreur est faite par contrôles de parité transversale et longitudinale.

Le bit de parité transversale associé à chaque caractère est positionné par la procédure. En effet, cette opération réalisée par software ne coûte rien, ni en temps, ni en place mémoire.

La conversion en ASCII étant réalisée par consultation d'une table, il suffit que les caractères figurant dans cette table aient une parité transversale correcte.

Par contre, le calcul du caractère de parité longitudinale (BCC) est réalisé par le coupleur et émis automatiquement. Pour ce calcul, il est nécessaire de reconnaître certains caractères de

contrôle du message : cf. annexe 1

"SOH" pour déclencher le calcul

"ETB" ou "ETX" pour l'arrêter et lancer l'émission du caractère.

II.2.2 - Fonctions dans le cas de transmission mode binaire

La détection d'erreurs ne se fait plus par parité, mais par utilisation d'un code cyclique. A l'émission, le coupleur doit élaborer deux caractères de 8 bits qui représentent, concaténés, le reste de la division du message série, considéré comme un polynôme à coefficients dans $\{0, 1\}$ par le polynôme $X^{16} + X^{15} + X^2 + 1$ (C.I.I. 23).

Pour lancer et arrêter le calcul, le coupleur détecte certains caractères de contrôle comme en II.2.1.

En outre, comme on transmet en binaire pur, la configuration des caractères de contrôle peut être rencontrée dans le corps du message. Pour éviter toute confusion, on convient (19) de placer le caractère DLE avant tout caractère de contrôle qui l'est effectivement et de doubler le caractère DLE chaque fois qu'il est rencontré fortuitement.

Cette opération doit être réalisée par le coupleur.

II.2.3 - Synchronisation

Le coupleur est placé en tampon entre l'ordinateur et le modem. Toutes les opérations d'émission se déroulent de façon séquentielles synchrones, pilotées par l'horloge émission du modem.

A partir de cette horloge, le coupleur élabore le signal de synchronisation externe (cf. § II.1) attendu par l'unité d'échanges pour sortir les deux caractères suivants.

Toute anomalie ou fin d'émission est signalée à l'unité centrale par une interruption.

II.3 - Réception

La première fonction du coupleur est de restituer à l'ordinateur en parallèle (16 bits) les informations qu'il reçoit en série de la ligne téléphonique. La transformation série-parallèle se fait caractère par caractère (8 bits).

II.3.1 - Fonctions dans le cas de la transmission ASCII

La reconnaissance du caractère SØH implique le départ du calcul de la parité longitudinale du message en cours.

La reconnaissance de l'un des deux caractères ETB ou ETX implique l'arrêt de ce calcul et la comparaison entre le caractère calculé et le caractère reçu. De plus la parité transversale est vérifiée sur chacun des caractères compris entre SØH et ETB ou ETX inclus.

II.3.2 - Fonctions dans le cas de la transmission mode binaire

Les mêmes caractères déclenchent et arrêtent respectivement le calcul de deux caractères vérifiant le code cyclique employé et la comparaison entre les caractères reçus et calculés.

II.3.3 - Synchronisation

II.3.3.1 Synchronisations bit et caractère

Le coupleur est piloté par l'horloge réception du modem ; à chaque signal d'horloge correspond l'arrivée d'un bit, quand le modem est synchronisé sur l'horloge du modem du central.

A la reconnaissance de 8 bits consécutifs représentant la configuration du caractère SYN le coupleur initialise la synchronisation caractère qui lui permet de considérer 8 bits comme un caractère et non pas comme une partie d'un caractère et du caractère suivant.

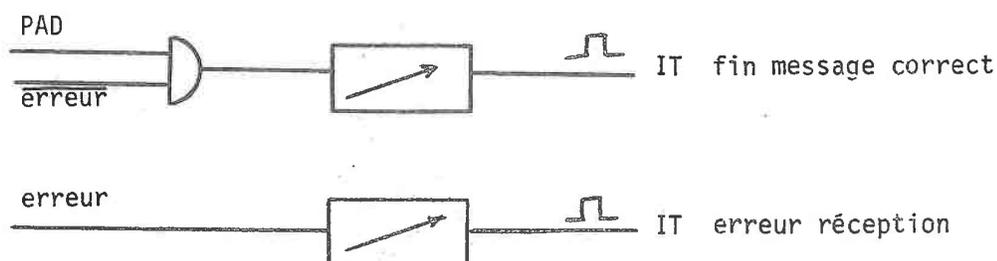
Quand cette synchronisation est initialisée, les signaux suivants sont le résultat d'une division par huit de la synchronisation bit.

II.3.3.2 Synchronisation coupleur-ordinateur

Les échanges se font par deux caractères à la fois.

A partir de la synchronisation caractère, le coupleur charge un buffer de 16 bits en deux temps, alternativement les poids forts puis les poids faibles. Quand ce buffer est chargé, un signal de synchronisation externe est envoyé à l'unité d'échange. La synchronisation externe est donc la synchronisation caractère divisée par 2.

Dès la reconnaissance du caractère SYN une interruption est envoyée de façon à commander la lecture sur le registre d'entrées numériques. La reconnaissance du caractère PAD, terminant tous les messages, cause une interruption qui est interprétée comme interruption fin de message et message correct. Par contre si une erreur a été détectée, une autre interruption est lancée à la place de la précédente.



L'unité d'échange est elle-même source d'interruption fin de table, erreur, etc...

II.4 - Hiérarchie des interruptions

Les interruptions "réception" sont prioritaires par rapport aux interruptions "émission" car les lectures sur le coupleur sont lancées sur interruption, d'où la nécessité de prendre celles-ci en compte le plus rapidement possible.

Les interruptions "émission" sont toutes sur le même niveau (niveau 7). Leur cause est détectée par software.

Les interruptions "réception" sont réparties sur plusieurs niveaux :

- interruption fin de table sur le niveau 2
- interruption SYN sur le niveau 6
- interruption PAD sur le niveau 6
- interruption erreur sur le niveau 6

Ces trois dernières sont elles-mêmes hiérarchisées dans le sens décroissant des priorités :

interruption "erreur"

interruption "PAD"

interruption "SYN" qui est la plus prioritaire

pour la prendre en compte rapidement.

La discrimination entre ces interruptions est entièrement faite par hardware. A chacune correspond un sous-programme spécifique.

Le choix fut soumis à d'importantes contraintes : nous ne disposons que de quatre niveaux libres, à savoir les niveaux 3, 6, 8, 11 qui se placent comme suit dans la hiérarchie des interruptions de la configuration. Le sens décroissant des priorités est 0 → 11.

<u>niveau</u>	<u>source</u>
0	horloges temps réel
1	disque
2	entrées numériques
3	P I S W libre
4	comparateur sur entrées analogiques
5	entrées analogiques
6	P I S W libre
7	sorties numériques
8	P I S W et machine à écrire
9	lecteur de cartes
10	clavier M A E
11	P I S W et interruption pupitre

III - LES MESSAGES

Dans la procédure TMM - RB, l'ensemble des messages possibles se répartit en quatre groupes distincts (toute suite de caractères circulant sur la ligne est appelée message, ou parfois séquence) :

- les séquences de supervision
- les blocs d'information utile
- les accusés de réception
- les demandes de suspension

La transmission s'effectue en mode synchrone d'octets (8 bits), en série, les poids faibles en tête. Tout message est précédé d'une séquence de caractères de synchronisation (SYN), dont le nombre est ajustable selon l'installation, et suivi d'un caractère spécial de "padding" indiquant la fin de message (PAD). Un contrôle de parité est élaboré sur les blocs d'information seulement.

Nous décrivons sommairement le contenu des divers types de messages ; les formats plus détaillés sont donnés en Annexe 1.

III.1 - Vue d'ensemble des messages (voir figure I.1)

III.2 - Les séquences de supervision

Elles permettent l'initialisation de la transmission d'informations utiles dans l'un ou l'autre sens :

- du terminal vers le central : polling
- du central vers le terminal : sélection

Sous TMM - RB, seule la station centrale a l'initiative de l'émission de telles séquences. Etant une interrogation, toute séquence

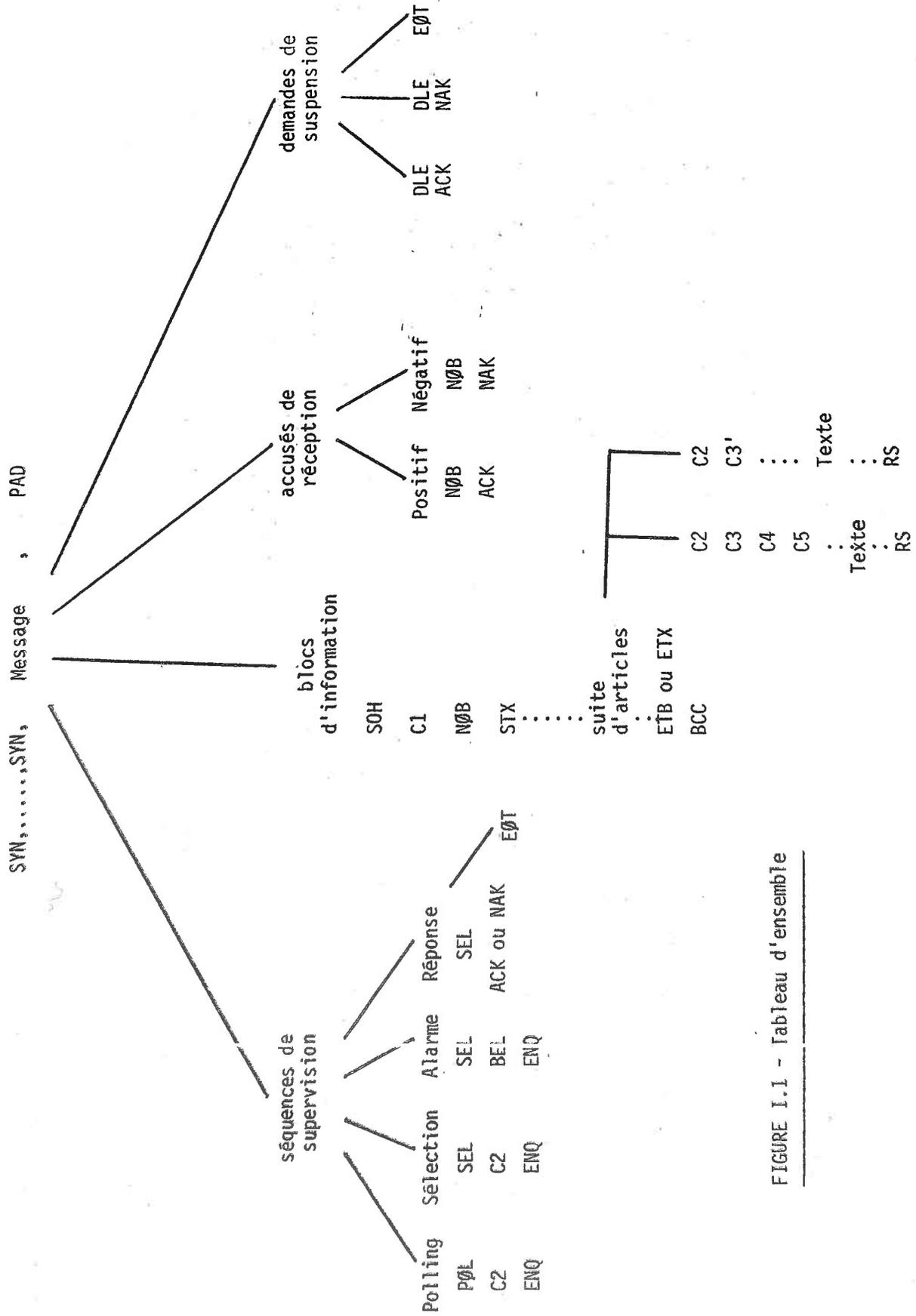


FIGURE I.1 - Tableau d'ensemble

de supervision exigera une réponse de la part de la station terminale désignée.

Ces messages sont formés par l'assemblage de trois caractères qui sont dans l'ordre :

- C1 indiquant la fonction polling (C1 = PØL) ou sélection (C1 = SEL) ainsi que l'adresse de la station terminale concernée.
- C2 contenant, outre la répétition de la fonction, l'identification du périphérique de la station appelée (cf. remarque dans le format des articles en annexe n° 1)
- Le caractère d'interrogation ENQ (Enquiry)

Les réponses du terminal seront :

- en polling, soit une séquence EØT (End of Transmission) s'il n'y a rien à transmettre, soit le premier bloc d'information utile
- en sélection, un message indiquant s'il est prêt ou non à recevoir.

La séquence EØT a pour but de réinitialiser la station qui la reçoit. La station en cours d'émission qui n'a pas ou plus de blocs à émettre transmet une séquence EØT pour suspendre la transmission. Il en est de même si elle reçoit une demande de suspension positive ou négative.

En cas de non réponse du terminal lors de la sélection initiale, la station centrale émet une séquence d'alarme pour attirer l'attention de l'opérateur du terminal au moyen d'alarmes lumineuses ou sonores. Elle requiert l'intervention de l'opérateur pour rendre opérationnel le terminal. Ce dernier doit répondre positivement

ou négativement à toute séquence de supervision consécutive à la séquence d'alarme. Une suite de non-réponses est interprétée par la station centrale comme un défaut de liaison.

III.3 - Les blocs d'information utile

Les données échangées entre terminal et central sont groupées en bloc. Un bloc est constitué d'un nombre entier d'articles encadrés de caractères spéciaux de procédure. Ces articles peuvent concerner des périphériques différents. Toutefois, on ne peut trouver qu'un seul article relatif au clavier de la machine à écrire, placé en tête du bloc. Chaque article possède son propre en-tête de procédure et est susceptible d'être compacté au premier niveau, c'est-à-dire que les blancs non significatifs de fin d'article ont été éliminés.

En transmission ASCII, aucun caractère de contrôle de transmission ne doit figurer dans la partie information du bloc.

Le format général d'un bloc d'information est le suivant :

En-tête	{	SØH marquant le début de l'en-tête
		C1 = SEL ou POL
		NØB indiquant le rang du bloc (modulo 8) dans le train des blocs transmis
	}	STX marquant le début du bloc
<hr/>		
L'information utile	{	Suites d'articles délimités par des séparateurs :
		2 formats d'articles sont possibles suivant que le nombre de caractères de l'article correspond ou non au format standard pour le périphérique concerné (cf. Annexe n° 1)
<hr/>		
Fin de bloc	{	ETB ou ETX pour marquer la fin du bloc
		BCC caractère de parité longitudinale pour tout le bloc

La séquence de fin ETB est employée pour les blocs courants d'un train, tandis que la séquence ETX n'est utilisée que pour le dernier bloc du train.

III.4 - Les accusés de réception

Un accusé de réception est émis par la station réceptrice après la réception d'un bloc d'information utile.

Il sera positif (ACK) si le bloc a été correctement reçu, négatif (NAK) dans le cas contraire.

III.5 - Les demandes de suspension

Elles se substituent aux accusés de réception lorsque, pour un motif quelconque, la station ne peut pas continuer sa fonction.

Une demande de suspension peut être positive, elle suppose alors que le dernier bloc reçu l'a été correctement.

Dans le cas d'une demande de suspension négative, on ne donne aucune indication quant à la prise en compte du bloc précédemment reçu. La station centrale peut émettre une demande de suspension quelconque. Par contre, pour simplifier la gestion de la procédure du côté central, la station terminale n'a le droit d'émettre que des demandes de suspension positives ; ceci accentue la dépendance du terminal.

R e m a r q u e

Certaines procédures sont normalisées. Leur description détaillée pourra être trouvée dans (STANDARD (19)) et (I.B.M. 33)

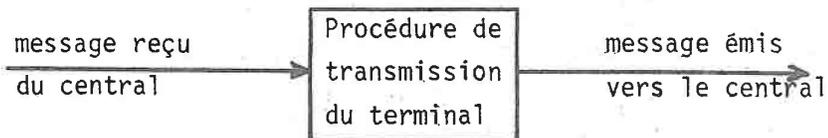
IV - REPRESENTATION DE LA PROCEDURE DE TRANSMISSION TMM - RB

Dans un premier temps, nous nous plaçons dans des conditions idéales où aucun message ne subit d'altération au cours de sa transmission, aucun compteur de temps n'atteint sa limite, aucune demande de suspension n'est faite. Alors tout message reçu au terminal est correct et correspond à un message attendu ; tout message émis arrive correctement au central.

Dans un second temps, nous essayerons de compléter le schéma en y ajoutant les problèmes de sécurité et de reprises.

IV.1 - Schéma simplifié

Sous les premières hypothèses la procédure du terminal peut être considérée comme un "automate" assez simple (LYNCH, (14)), (BARTLETT, (2)), (LYNCH, (15)).



IV.1.1 - Evènements extérieurs

Sur le diagramme d'état (figure I.2), nous notons P_i les divers états. P_0 sera l'état initial et l'état final.

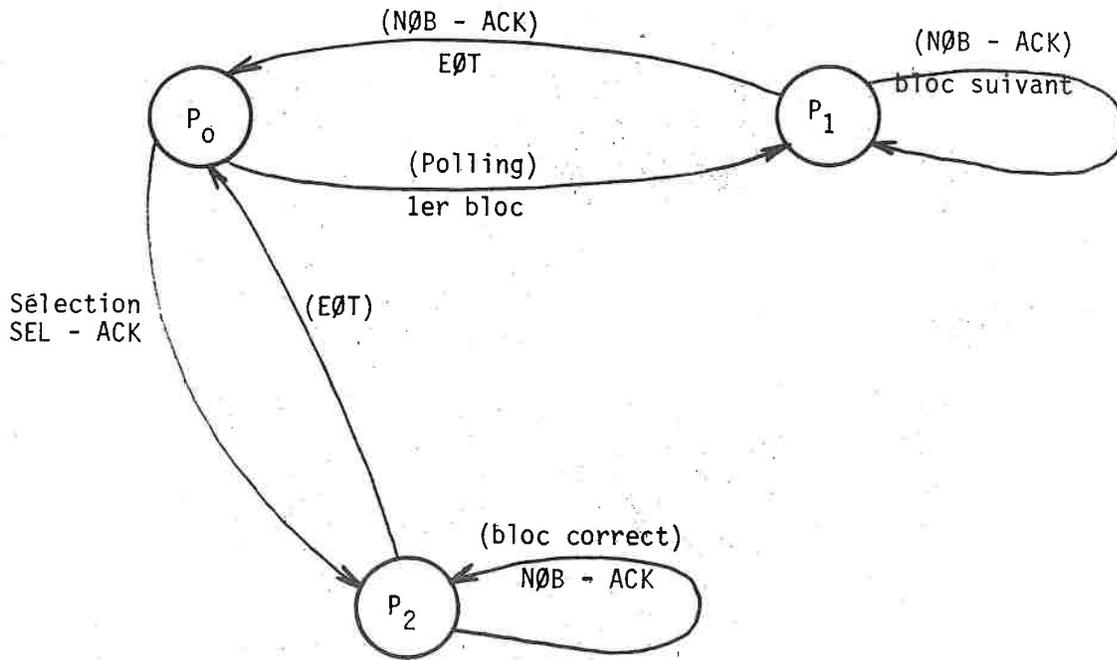
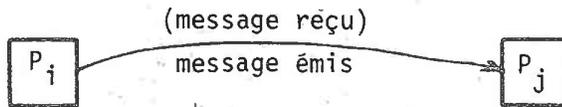
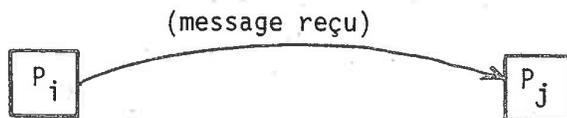


FIGURE I.2 - Schma global de dpart

Sur chaque arc, nous notons l'vnement (le message reu) ainsi que l'action (le message mis) provoque par cet vnement sous la forme suivante :



Si certains vnements ne donnent lieu  aucune action, ils seront reprsents par le message reu, not entre parenthses.



IV.1.2 - Evènements intérieurs

Remarquons alors que le schéma de la figure I.2 est déjà incorrect puisqu'en phase de "polling", étant dans l'état P_1 , la réception de (NØB - ACK) peut conduire à deux émissions distinctes : le bloc suivant s'il y a encore quelque chose à émettre, la séquence EØT s'il n'y a plus rien à émettre. Nous mettons ainsi en évidence une seconde espèce d'évènements dits "intérieurs". Par souci de symétrie, nous appellerons évènements "extérieurs" les messages reçus.

Par exemple, en polling, le fait qu'aucun bloc n'est prêt à être émis, que le dernier bloc a été émis, constitueront des "évènements intérieurs". Ces derniers seront matérialisés par des positionnements d'indicateurs. Nous appellerons "état global" un des états P_i précédents : c'est un ensemble d'états partiels P_{ij} , j désignant ici une configuration donnée des indicateurs associés à l'état global P_i .

En supposant toujours qu'aucun message n'est perturbé au cours de sa transmission, nous avons :

Pour P_0

	indicateur Prêt au Polling	indicateur Prêt à la sélection
P_{0_0}	0	0
P_{0_1}	0	1
P_{0_2}	1	0
P_{0_3}	1	1

Pour P_1

	le dernier bloc a été émis
P_{1_0}	0
P_{1_1}	1

Pour P_2

	le dernier bloc a été reçu correctement et NØB - ACK correspondant déjà émis
P_{2_0}	0
P_{2_1}	1

Un état global P_i et la configuration j de ses indicateurs associés, nous définissent à tout moment l'état partiel P_{ij} dans lequel se trouve la procédure de transmission du terminal.

Nous nous contenterons de représenter les événements extérieurs ; l'arc correspondant aura maintenant pour origine un état partiel P_{ij} et pour extrémité un état global $P_{i'}$.

On arrive ainsi au diagramme de la figure I.3.

En supposant que la transmission s'effectue dans des conditions idéales, nous avons évité l'important problème de la sécurité de la transmission, problème que nous allons aborder maintenant.

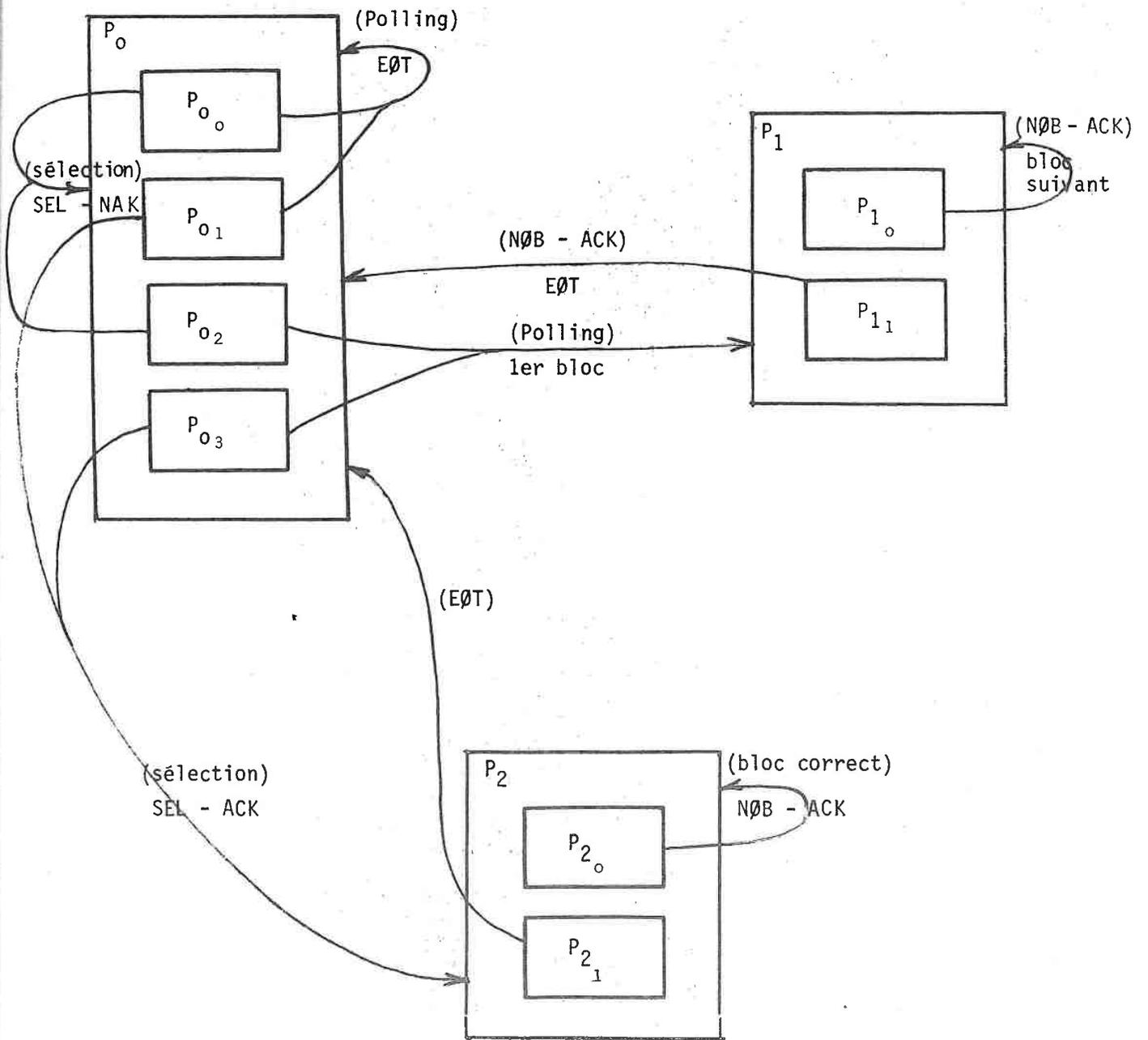


FIGURE I.3 - Schéma simplifié

IV.2 - Schéma réel

Sous TMM - RB, la station centrale règle entièrement les problèmes de reprise ; la station terminale reste passive de ce point de vue. La sécurité de la transmission est assurée par répétition des blocs d'information incorrectement reçus.

IV.2.1 - Les charges du terminal

Au niveau du terminal, nous appelons :

bloc invalide, tout bloc d'information dont les caractères d'encadrement (SOH, ETB ou ETX) sont incorrects ou n'ont pu être reconnus. La réception d'un tel bloc n'entraîne aucune réponse de la part du terminal.

bloc incorrect, tout bloc d'information présentant au moins une des caractéristiques suivantes :

- . un caractère BCC calculé à la réception non égal au caractère BCC reçu,
- . un caractère de service (autre que d'encadrement) à structure incorrecte,
- . une erreur de numérotation de bloc,
- . une erreur de comptage de caractères au niveau d'un des articles du bloc.

La réception d'un tel bloc entraîne l'émission d'un accusé de réception négatif.

bloc correct, tout bloc qui n'est ni invalide, ni incorrect. La réception d'un tel bloc entraîne l'émission d'un accusé de réception positif.

En dehors des blocs d'information, nous désignerons par "réponse invalide" de la station centrale, toute réponse présentant une non-conformité avec la structure de la réponse attendue. La réception d'une telle réponse n'entraîne également aucune réponse de la part du terminal.

IV.2.2 - Les demandes de suspension

Si, pour un motif quelconque, une station ne peut pas continuer sa fonction, une demande de suspension peut se substituer à l'accusé de réception.

POLLING : - par le terminal

Si momentanément il n'a plus rien à transmettre, le terminal envoie la séquence EØT à la station centrale en réponse à un accusé de réception positif. Cette dernière, reprenant alors l'initiative de la transmission, peut déclencher soit un nouveau polling, soit une sélection.

A la réception d'une séquence de polling, le bloc alors émis par le terminal, porte le numéro 1.

- par la station centrale

Si une demande de suspension positive (respectivement négative) est émise par la station centrale, le terminal doit émettre EØT et se réinitialiser dans l'attente d'une nouvelle séquence de supervision. A la réception d'une nouvelle séquence de polling, le terminal doit transmettre le bloc suivant (respectivement le dernier bloc émis avant la suspension). Ce bloc portera le numéro 1 dans la nouvelle phase du polling.

SELECTION : - par le terminal

Seulement après réception d'un bloc correct, le terminal peut faire une demande de suspension positive. La station centrale répondra EØT et le terminal doit se réinitialiser dans l'attente d'une nouvelle séquence de supervision.

- par le central

En émettant EØT à la place d'un bloc d'informations, la station centrale suspend la transmission. Le terminal doit alors se réinitialiser en vue d'une séquence de supervision quelconque.

Essayons déjà de compléter le schéma précédent avec les deux points examinés, sans encore faire intervenir les réactions de reprise de la station centrale. Ces deux points ont bien sûr tendance à augmenter le nombre d'états partiels des états globaux P_1 et P_2 . Ils n'ont aucune influence sur l'état global P_0 (fig. I.4).

- Pour P_1 le nouvel indicateur "momentanément plus rien à transmettre", suppose que le dernier bloc n'a pas encore été transmis. De plus, on ne peut faire intervenir cet indicateur que si le terminal reçoit un accusé de réception positif. Comme l'évènement attendu et l'éventuelle action entreprise sont identiques à ceux de l'état partiel P_{11} , nous sommes amenés à confondre les deux indicateurs attachés à l'état global P_1 en un seul : dernier bloc transmis ou momentanément plus rien à transmettre (DBMT).

	DBMT
P_{1_0}	0
P_{1_1}	1

- Pour P_2 un indicateur supplémentaire sera positionné dès que le terminal souhaitera une suspension. Une fois "DLE - ACK" émis, le terminal sera en attente d'une séquence EØT comme s'il avait déjà reçu correctement le dernier bloc. Remarquons alors qu'il est inutile de positionner l'indicateur de demande de suspension si le dernier bloc a été reçu correctement et NØB - ACK déjà émis, puisque, dans ce cas, nous sommes déjà en attente d'une séquence EØT. Comme pour P_1 , nous confondons les deux indicateurs en un seul : demande de suspension (DS) qui peut être positionné par l'un ou l'autre des deux événements intérieurs cités ci-dessus.

De plus, pour P_2 , la station centrale peut décider à tout moment de suspendre la transmission en envoyant la séquence EØT au lieu d'un bloc d'information.

	DS
P_{2_0}	0
P_{2_1}	1

Remarques :

- 1) *Le terminal dispose d'une temporisation de 1,5 seconde armée à chaque fin de réception d'une séquence quelconque et désarmée au début de chaque émission du terminal. L'émission du terminal, en réponse à une séquence reçue, doit avoir lieu avant épuisement de cette temporisation. Si le terminal n'est pas en mesure de respecter cette condition à un instant donné, il doit rester silencieux et attendre l'émission suivante de la station centrale, laquelle tiendra éventuellement compte de la non-réponse du terminal.*

Comment faire figurer cette temporisation sur le diagramme ?

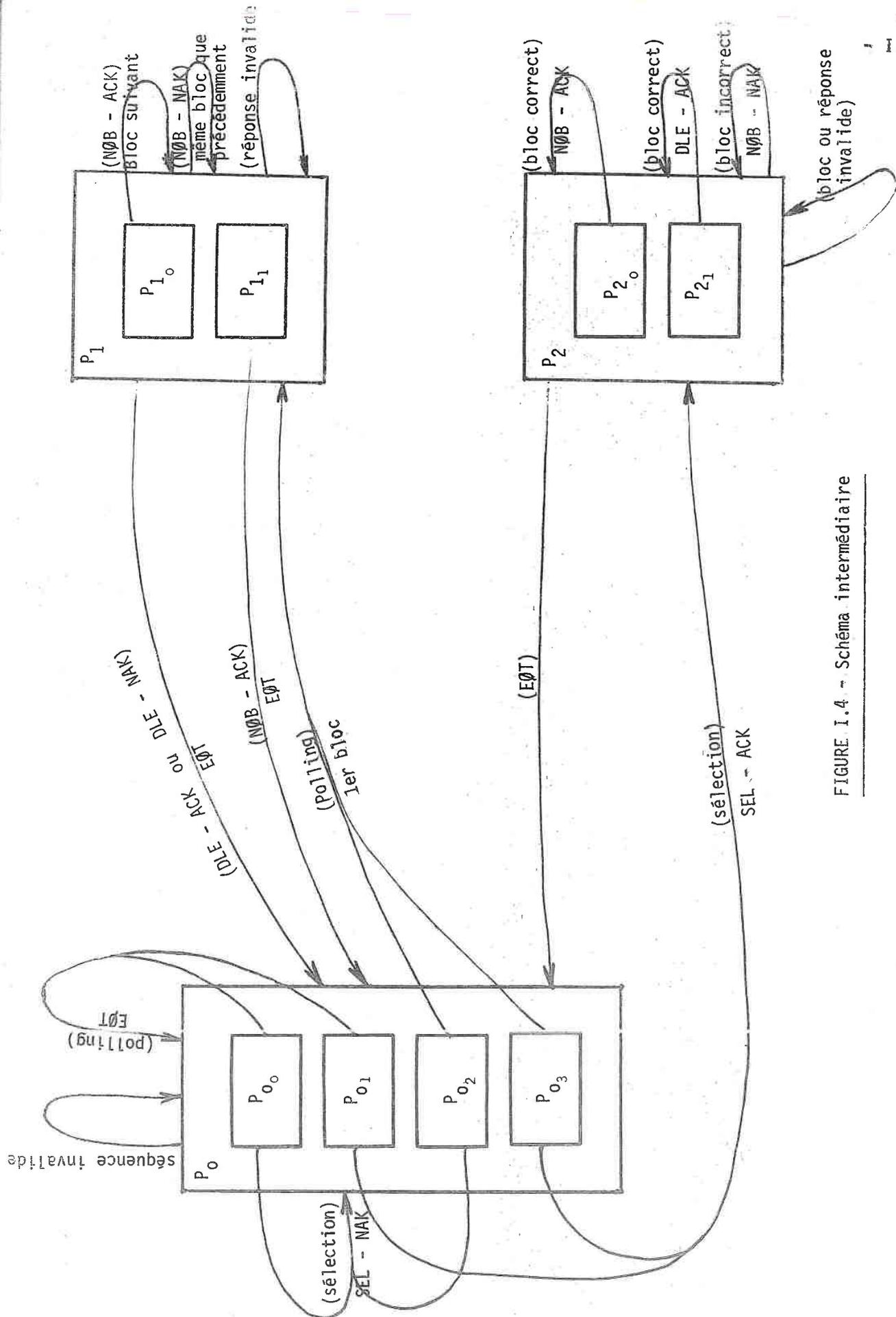


FIGURE I.4 - Schéma intermédiaire

Il suffit de considérer, en cas d'épuisement de la temporisation, que le message reçu est invalide, ce qui fait que l'on reste bien à l'état partiel où l'on était au moment de la réception du message.

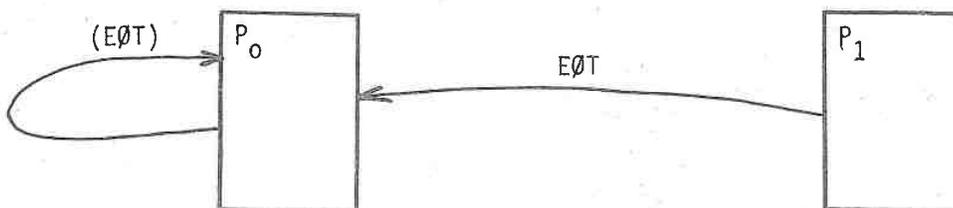
- 2) *Si le même couple (événement, action) se retrouve pour tous les états partiels, nous convenons que l'arc correspondant aura l'état global comme origine.*

IV.2.3 - Les reprises du central

Jusque là nous avons toujours supposé que tout message émis par le terminal était correctement reçu par le central. Cette hypothèse ne respecte pas la symétrie de la transmission et n'est donc pas très réaliste. Alors quelles sont les réactions de la station centrale quand le message reçu est incorrect ou invalide, ou quand il ne reçoit aucune réponse ?

1) séquences de supervision

Si, à une séquence de supervision, le central ne reçoit aucune réponse ou une réponse invalide, il envoie une séquence $E\emptyset T$ suivie par une nouvelle séquence de supervision. La séquence $E\emptyset T$ devant ramener le terminal à l'état initial P_0 , les phases de polling et de sélection restent, de ce fait, nettement séparées : il n'existe aucune liaison directe entre les ensembles P_1 et P_2 . Il faut donc compléter le diagramme précédent en y ajoutant les liaisons suivantes :



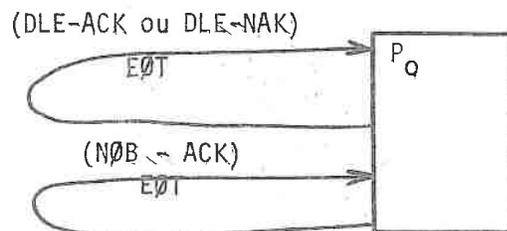
2) Polling

Le central dispose non seulement d'un nombre maximum p de déclenchements de la temporisation d'attente de réponse, mais encore d'un nombre maximum m de réceptions d'un bloc invalide ou incorrect. Les accusés de réception sont répétés tant que ces maxima ne sont pas atteints. Quand l'un ou l'autre est atteint, le central émet une demande de suspension négative. Tout caractère reçu en réponse à une demande de suspension négative ou positive est assimilé à EØT. Tant que le maximum p de non-réponse n'est pas atteint, la même demande de suspension est répétée. Quand p est atteint, une séquence EØT est émise par le central, éventuellement suivie par une nouvelle séquence de supervision.

Ces reprises en polling sont essentiellement consécutives à de mauvaises transitions entre les états globaux P_0 et P_1 . En effet, si le premier bloc émis par le central n'est pas reçu ou reçu de façon invalide par le central, ce dernier émet une séquence EØT suivie d'une nouvelle séquence de supervision ; comme indiqué au paragraphe précédent, ceci nous oblige à ajouter une transition (EØT) de P_1 vers P_0 pour nous permettre de revenir à l'état initial.

De même si la séquence EØT émise par le terminal n'est pas reçue par le central, celui-ci émet de nouveau le message précédent, c'est-à-dire : soit (DLE - ACK) ou (DLE - NAK), soit (NOB - ACK). Si le nombre de répétitions des demandes de suspension atteint p , alors une séquence EØT est émise par le central.

En conséquence, il faut rajouter les liaisons suivantes à P_0 .



3) Sélection

Le central dispose dans ce cas des mêmes maxima que ci-dessus. m correspond ici au nombre maximum d'accusés de réception négatifs ou incorrects. Tant que ni l'un, ni l'autre des deux maxima p et m n'est atteint, le central émet le même bloc ($p + m$ répétitions au maximum). Dans le cas contraire, le central émet une séquence EØT éventuellement suivie d'une nouvelle séquence de supervision. Ceci est déjà représenté sur le diagramme.

Ces considérations font que pour P_2 , il ne faut prendre en compte que l'indicateur de demande de suspension sans se soucier, en ce qui concerne la gestion de la transmission, du dernier bloc émis par le central.

Remarques

Signalons qu'en ce qui concerne la temporisation du central, il existe un autre moyen pour prendre en compte les messages réémis par le central. En effet, il suffit du côté terminal, de disposer d'une temporisation de même durée, armée quand le terminal émet un message d'une transition faisant changer d'état global, désarmée au début de réception d'une séquence quelconque.

Si aucune séquence n'est reçue avant épuisement de cette temporisation, le terminal doit se remettre dans l'état partiel d'où il est parti car il risque fort de recevoir une séquence identique à la précédente.

Le cas des messages invalides ne trouve pas une solution analogue et donc les modifications apportées en diagramme d'état restent valables.

Sommes-nous ainsi parés à toute éventualité ? Non sans doute, mais la sécurité atteinte semble être satisfaisante.

IV.2.4 - Test de la liaison

Il est assuré à l'aide du "périphérique fictif" et se déroule de la manière suivante :

La station centrale provoque une phase de sélection et transmet une suite de blocs adressés au périphérique fictif du terminal. Ce dernier se borne à recevoir et à contrôler les blocs. Ensuite la station centrale provoque une phase de polling avec le périphérique fictif du terminal. Celui-ci doit alors renvoyer à la station centrale le dernier bloc reçu lors de la phase de sélection précédente. Cet enchaînement ajoute à la difficulté de représentation de la procédure de transmission. (figure I.5).

Pour finir, signalons que les indicateurs utilisés ici peuvent être positionnés de diverses manières :

- par l'utilisateur côté terminal à travers la méthode d'accès ;
- par l'état des périphériques du terminal ;
- par le contenu des blocs d'information reçus.

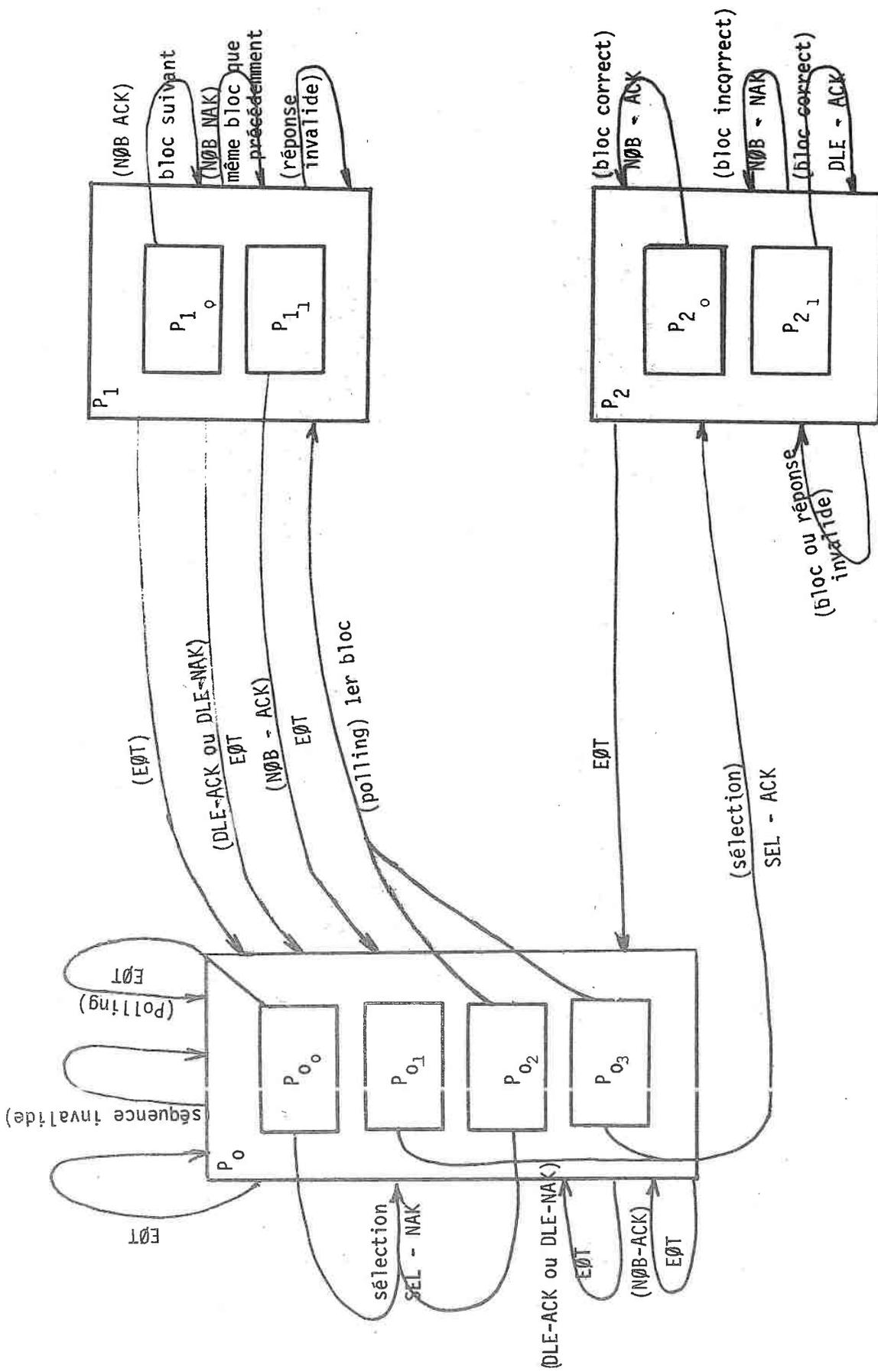
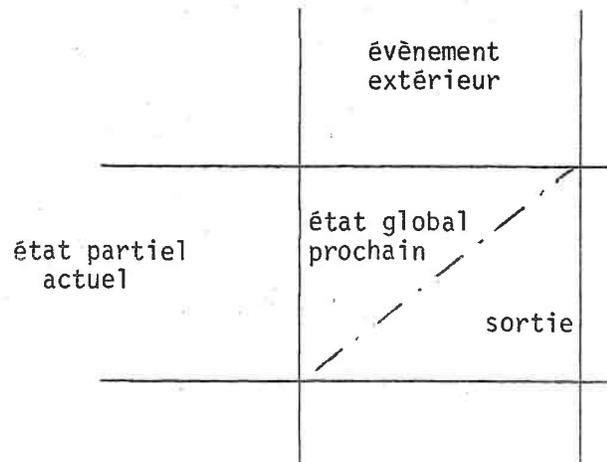


FIGURE I.5 - Schéma réel complet

IV-3 - Tables associées

La représentation de la procédure de transmission TMM - RB peut être faite à l'aide de deux tables, une table des états et une table des sorties, semblables à celles utilisées pour l'analyse des circuits séquentiels. Cette forme de représentation devrait permettre une étude plus systématique des procédures de transmission.

La représentation que nous donnons regroupe les deux tables en une seule sous la forme suivante :



IV.3.1 - Table des états et sorties (figure I.6)

Evénements extérieurs	Polling	Sélection	message invalide	NØB ACK	NØB NAK	DLE ACK ou NAK	bloc correct	bloc incorrect	EØT
états partiels	P ₀	P ₀ SEL NAK	-	P ₀ EØT	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
				P ₀ EØT	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
	P ₁	P ₀ SEL NAK	-	P ₀ EØT	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
				P ₁ 1er bloc	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
	P ₁	P ₂ SEL ACK	-	P ₀ EØT	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
				P ₁ 1er bloc	P ₀ -	P ₀ EØT	P ₀ -	P ₀ -	P ₀
	P ₁	P ₁ -	-	P ₁ bloc suivant	P ₁ même bloc	P ₀ EØT	P ₁ -	P ₁ -	P ₀
				P ₀ EØT	P ₁ même bloc	P ₀ EØT	P ₁ -	P ₁ -	P ₀
	P ₂	P ₂ -	-	P ₂ -	P ₂ -	P ₂ -	P ₂ NØB ACK	P ₂ NØB NAK	P ₀
				P ₂ -	P ₂ -	P ₂ -	P ₂ DLE ACK	P ₂ DLE NAK	P ₀

Comme indiqué au cours des paragraphes précédents, les transitions ont un état partiel P_{ij} comme origine et un état global P_i comme extrémité ; la concrétisation d'un état global en un état partiel se fait par l'intermédiaire des indicateurs associés à cet état global. Ces passages état global \longrightarrow état partiel sont rappelés ci-dessous.

IV.3.2 - Transitions état global \longrightarrow état partiel :

- Pour P_0 on a deux indicateurs :

- . prêt au polling PP
- . prêt à la sélection PS

PP , PS	00	01	11	10
P_0	P_{0_0}	P_{0_1}	P_{0_2}	P_{0_3}

- Pour P_1 un seul indicateur :

- . dernier bloc transmis ou momentanément plus rien à transmettre : DBMT

DBMT	0	1
P_1	P_{1_0}	P_{1_1}

- Pour P_2 un seul indicateur :

demande de suspension DS

DS	0	1
P_2	P_{2_0}	P_{2_1}

La procédure TMM - RB décrite ici n'accorde pratiquement aucune initiative au terminal, ni pour l'initialisation de la transmission, ni au cours du transfert. Dans le chapitre III, nous décrivons une procédure plus générale, inspirée par TMM - UC, où les deux extrémités ont des rôles plus symétriques et plus équilibrés. A cette occasion, nous essayerons de dégager quelques considérations plus générales en ce qui concerne la représentation des procédures de transmission.

Cette procédure plus générale n'est pas encore réalisée et le chapitre II de Réalisation ne concerne donc que ce que nous venons de décrire, c'est-à-dire la procédure du terminal compatible TMM - RB.

CHAPITRE II

REALISATION

I - INTRODUCTION

La réalisation de la procédure de transmission a été effectuée à partir de la représentation donnée pour la procédure TMM - RB (cf. chap.I § IV). Pour la rendre commode, il fallait que notre analyse nous conduise à des programmes de taille relativement faible, écrits en assembleur (IBM 27) , (IBM 29) , (IBM 32) . Si cette exigence peut être satisfaite, la compréhension, la programmation et la mise au point de toute la procédure sont ainsi considérablement facilitées.

Donc, tout en respectant la représentation donnée précédemment, nous devons aboutir à une schématisation nettement plus modulaire pour la procédure. Pour nous un module est un sous-programme essentiel qui, en faisant éventuellement appel à d'autres sous-programmes de moindre importance, est chargé de la gestion d'une partie de la procédure de transmission du terminal. L'analyse détaillée de chaque module doit être simple.

II - ANALYSE GENERALE

Pour obtenir une réalisation modulaire l'idée, d'ailleurs logique, est de séparer la reconnaissance des messages reçus de l'élaboration des réponses ou actions. Ainsi la procédure est-elle réalisée par un ensemble de modules les uns "passifs", les autres "actifs", ensemble géré par un programme directeur.

Les modules passifs sont chargés de la reconnaissance des messages reçus. Le terminal est donc en attente d'un message du central. Tant que le message reçu ne correspond à aucun des messages attendus pour ce module, la procédure du terminal reste en attente. Dans le cas contraire, le contrôle est donné soit à un module actif si un message ou une action doit être élaboré, soit à un module passif (éventuellement le même) si le message reçu ne donne lieu à aucune réponse (par exemple séquence EØT).

Les modules actifs sont chargés d'élaborer une réponse qui tient compte de la configuration des indicateurs concernés à ce point du déroulement de la procédure de transmission. Une fois la réponse envoyée, le contrôle est immédiatement rendu à un module passif. Par conséquent, le terminal n'est jamais en attente d'une séquence du central.

A chaque état global P_i de la représentation du chapitre précédent correspond un module passif noté $S P_i$ et, à priori, un module actif noté $S A_{ij}$ pour chaque séquence j acceptable. Mais nous avons pu regrouper les réponses élaborées : par exemple en polling ler bloc et bloc suivant. Du fait de ces regroupements, aucun lien ne doit être établi entre l'indice k d'un module actif $S A_k$ et celui i d'un état global P_i quelconque, contrairement au cas des modules passifs. La fonction essentielle d'un module actif est donc d'assurer la transition entre deux modules passifs. L'analyse générale des deux parties principales de la procédure pourrait être

décrite par des organigrammes comme dans (AYDELOTTE 1).
A cette méthode, nous avons préféré une schématisation comme ci-dessous (voir figures II.1 et II.2).

Les conventions de représentation sont toujours les mêmes à savoir : le message reçu est noté entre parenthèses, le message envoyé est indiqué sans signe particulier.



De plus sur ces deux schémas, nous ne représentons pas les conditions d'élaboration des différents messages qui seront décrites un peu plus loin.

D'après ce qui précède, la procédure du terminal est "réveillée" par l'arrivée d'une séquence quelconque émise par le central. L'interruption SYN déclenche la lecture de cette séquence, lecture avec synchronisation externe sur les entrées numériques du 1800 effectuée par le sous-programme *DINPT*. L'interruption fin de table ou PAD en indiquera la fin.

Cette dernière interruption déclenche l'exécution du programme directeur *EXAM* qui, par simple consultation de la variable ET, sait à quel module passif $S P_i$ il doit donner le contrôle en premier lieu. Tout module passif ou actif peut, au cours de son exécution, modifier la valeur de la variable ET, suivant la séquence reçue pour les modules passifs, suivant la réponse élaborée pour les modules actifs.

Tout module rendra le contrôle à *EXAM* qui déclenche l'exécution d'un module actif ou se remet en attente d'une séquence du central suivant que ET indique un module actif ou passif.

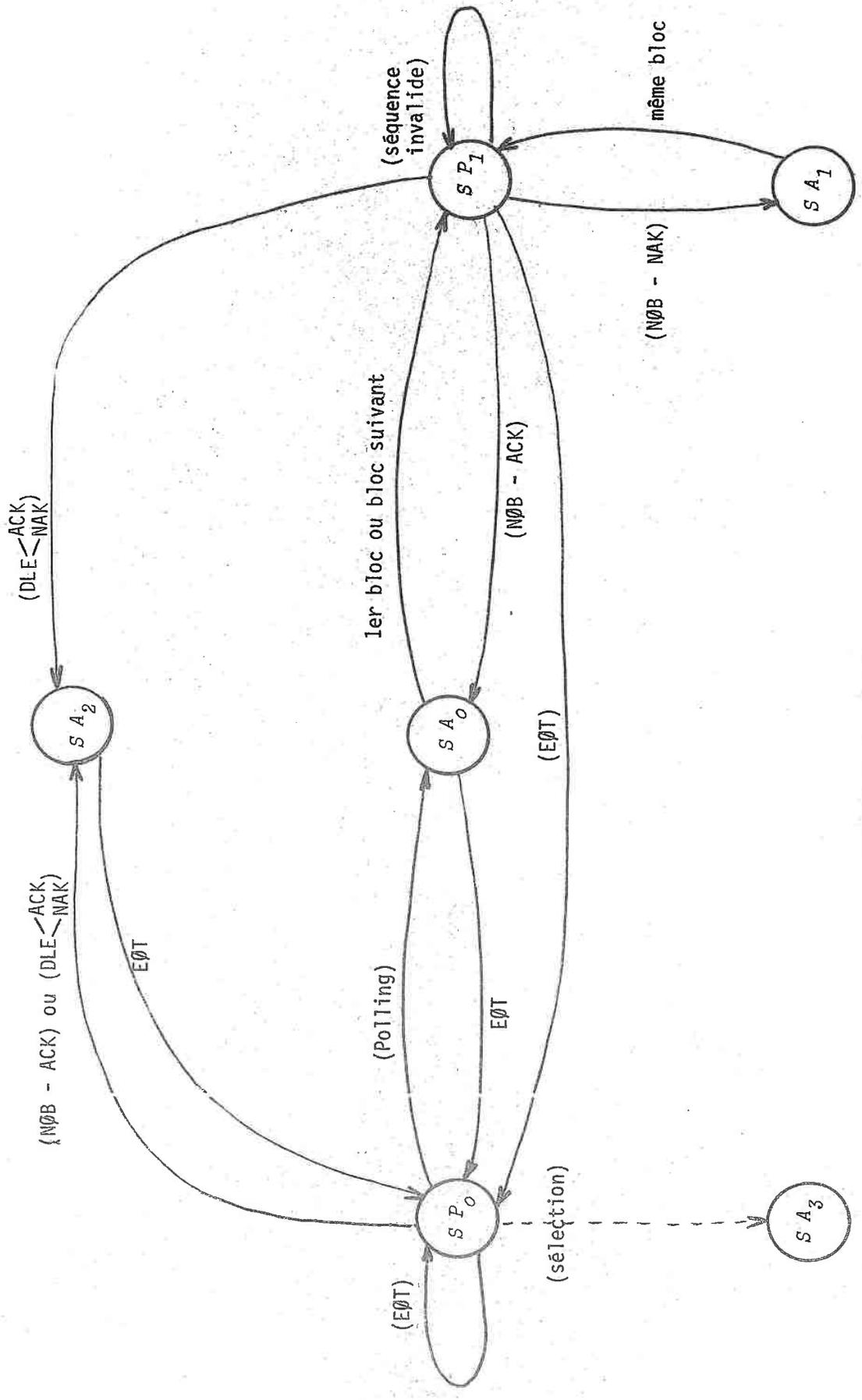


FIGURE II.1 - Polling

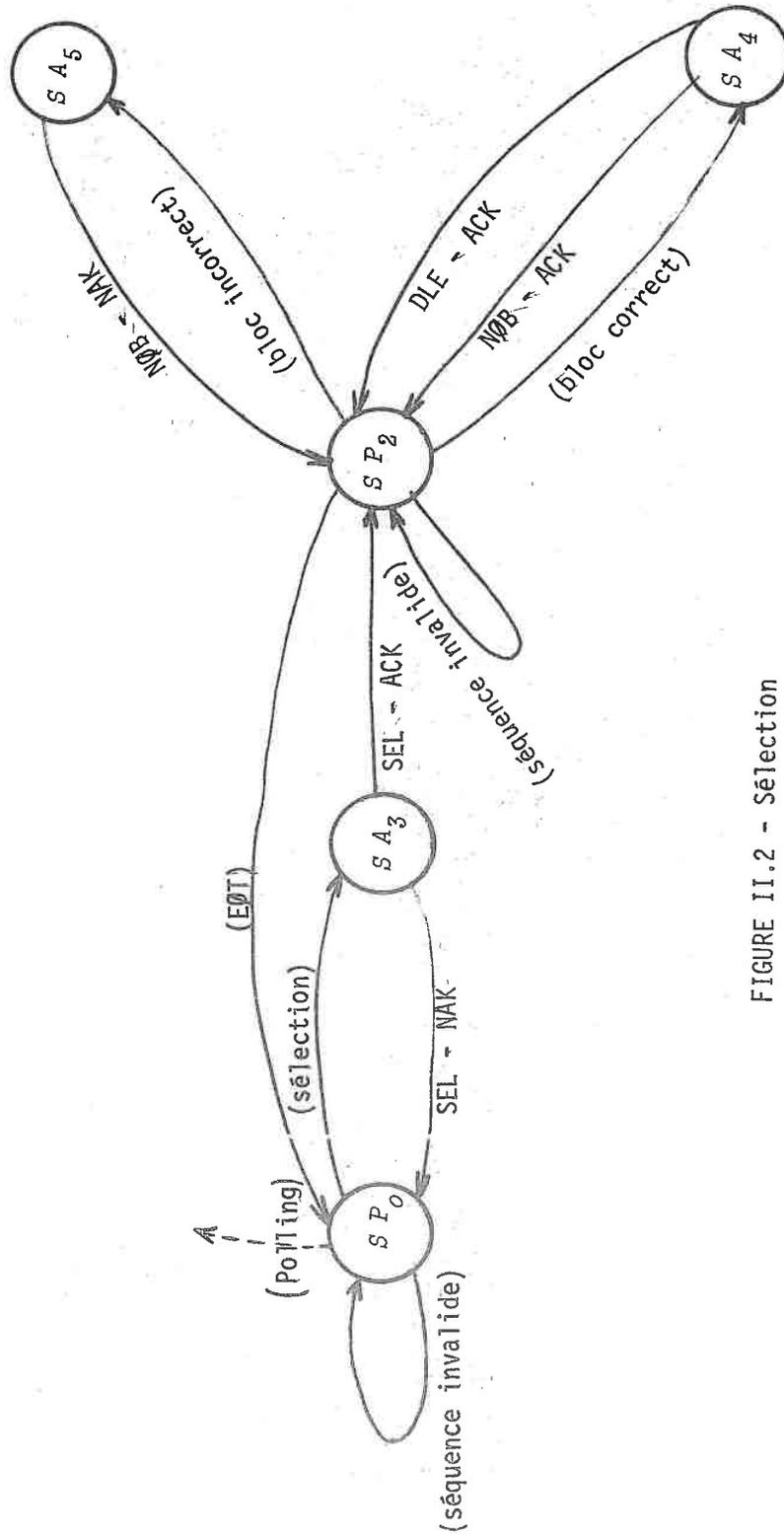


FIGURE II.2 - Sélection

Ici "attente" n'est pas synonyme de "repos" car cela correspond à l'exécution d'un *travail de fond* concernant l'information utile transmise : préparation des blocs à émettre en Polling, traitement des blocs reçus en Sélection.

Avant de décrire les différentes parties de ce chapitre, il nous faut signaler les trois points suivants, valables pour tout le chapitre :

1. Un certain nombre de variables, telle ET citée ci-dessus, sont utilisées par plusieurs modules et sous-programmes. Ces variables dites communes sont en "Inskel Common". La liste de ces variables sera établie au fur et à mesure ; à leur première apparition, nous les signalons par l'indication (v.c.) pour variable commune.
2. La réception de toute séquence, contrôle ou bloc d'information, est faite par l'intermédiaire du sous-programme de lecture *DINPT*. Quand la procédure est en attente d'une séquence, BREC (v.c.) doit contenir l'adresse de début de la zone de réception, WØRIN (v.c.) son nombre de mots. L'exécution de *DINPT* est déclenchée par l'interruption SYN ; il remplit la zone décrite par les deux variables BREC et WØRIN. Indiquons dès à présent que les séquences de contrôle sont reçues dans une zone T de 7 mots, les blocs d'informations dans deux tampons de 135 mots (voir les modules de Sélection). Nous ne détaillons pas plus l'utilisation de BREC et WØRIN pour les séquences de contrôle ; pour les blocs d'information, nous la signalons par l'indication "préparer lecture".
3. L'émission de toute séquence est faite par le sous-programme *DEXIT*. Il suffit de lui communiquer dans BEMI (v.c.) l'adresse de début de la zone à émettre, dans WØRDC (v.c.) le nombre de mots de cette zone, puis d'en faire l'appel CALL *DEXIT*. Dans la suite "émettre une séquence" signifie remplir BEMI et WØRDC

et faire l'appel du sous-programme. La zone d'émission peut être la zone B de 7 mots pour les séquences de contrôle ou l'un des deux tampons de 135 mots pour les blocs d'information (voir Polling).

Ce chapitre comprendra cinq parties décrivant l'essentiel de la réalisation :

- programme directeur *EXAM* et module passif commun *SPO*
- les modules du Polling
- les modules de la Sélection
- le *travail de fond*
- les sous-programmes d'analyse des séquences reçues

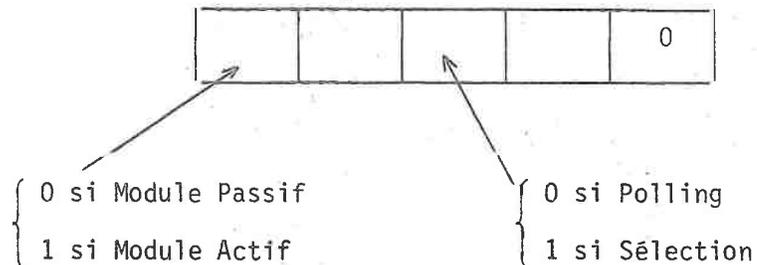
III - PROGRAMME DIRECTEUR ET MODULE PASSIF COMMUN

III.1 - Programme directeur EXAM

Les divers modules de la procédure sont bien sûr écrits sous forme de sous-programmes ; leur exécution est commandée par le programme directeur.

Puisque le choix se fait sur la valeur de la variable ET (v.c.) nous devons établir une correspondance biunivoque entre l'ensemble des modules passifs et actifs et l'ensemble des valeurs possibles pour la variable ET. Pour éviter la méthode des tests successifs, nous avons rangé les divers appels de sous-programmes (modules) dans un tableau T ; alors la valeur associée à un module est l'indice d'entrée dans ce tableau. Toutes ces valeurs sont paires car l'appel à un sous-programme (CALL *SP*) occupe ici 2 mots. Au cours de l'exécution, il suffira d'insérer l'élément choisi par la valeur de ET à l'endroit prévu pour l'appel.

La codification a été faite à l'aide de 5 positions binaires ; celle de poids le plus faible étant toujours nulle, n'a aucune signification.



les 2 bits restants complètent la numérotation des modules ce qui nous donne la correspondance suivante :

Module	Valeur décimale et identificateur associé	
$S P_0$	0	$m p_0$
$S P_1$	2	$m p_1$
$S P_2$	4	$m p_2$
$S A_0$	16	$m a_0$
$S A_1$	18	$m a_1$
$S A_2$	24	$m a_2$
$S A_3$	22	$m a_3$
$S A_4$	20	$m a_4$
$S A_5$	30	$m a_5$

Du fait que, partant d'un module passif, la procédure aboutit à un module passif (éventuellement celui de départ) en passant par un état actif au maximum (à la rigueur aucun), la description du module directeur se réduit à :

```
si ET < 16 alors début exécution T(ET) ;  
                si ET ≥ 16 alors exécution T(ET) ;  
                fin ;
```

III.2 - Module passif commun SP_0

Ce module se sert des sous-programmes suivants :

- *ELSYN* pour l'élimination des caractères de synchronisation
- *SEQSU* pour la reconnaissance des séquences de supervision
- *ACQUI* pour la reconnaissance des séquences contenant ACK ou NAK

On trouvera leur description un peu plus loin.

Notons ici que durant toute la procédure, nous aurons à tenir compte du test de liaison que le central peut déclencher. Rappelons que ce "test" est repéré par l'indication du périphérique fictif dans les séquences de supervision et qu'il se compose d'au moins une phase de sélection suivie d'une phase de polling où le terminal doit obligatoirement renvoyer le dernier bloc reçu (adresse conservée par BTL (v.c.)). Le positionnement de TL (v.c.) à la valeur 1 indiquera que nous sommes en test de liaison.

De plus le module SP_0 aura à réinitialiser correctement les numéros de blocs, en particulier en polling si la transmission a été suspendue ; ce traitement sera donné en annexe.

La structure générale du module SP_0 est alors simple : (voir ci-après).

Remarquons que quelle que soit la séquence reçue, non signalée ci-après, le terminal n'a aucune réaction et reste donc en SP_0 (c'est-à-dire aucune modification de ET). Ceci est également vrai pour la séquence EØT reçue ; par conséquent, nous avons jugé inutile de connaître cette séquence vu la dépendance dans laquelle se trouve le terminal par rapport au central. Dans une procédure plus générale (cf. chapitre III), il serait plus intéressant de reconnaître la séquence EØT afin de pouvoir en faire le compte en vue d'une utilisation ultérieure par la station esclave.

Si Séquence de Supervision alors faire A sinon faire B ;

A [début si Sélection alors faire C sinon faire D fin

C [début si Périphérique Fictif alors début TL := 1 ; ET := ma₃ fin

sinon début si TL ≠ 1 alors ET := ma₃ fin

fin C

D [début si Périphérique Fictif alors début si TL = 1 alors début préparer dernier
bloc reçu (adresse du
bloc dans BTL) ;
ET := ma₀

fin

fin

sinon début si TL ≠ 1 alors ET := ma₀ fin

fin D

fin A

B [début si { NOB - ACK }
ou
{ DLE - ACK }
ou
{ DLE - NAK } alors ET := ma₂

fin B

IV - POLLING

La partie polling de la procédure a besoin de la préparation des blocs réalisée en *travail de fond* (cf. II.VI). Ce dernier doit lui communiquer l'adresse du bloc préparé et donc prêt à être transmis.

Pour profiter de l'émission par canal des séquences (en particulier des blocs d'information), le bloc i est préparé pendant l'émission du bloc $i-1$ par le canal. Il faut donc disposer au minimum de 2 tampons T_1 et T_2 pour l'émission des blocs d'information. Leur gestion sera assurée par l'intermédiaire de deux variables de contrôle $BEMI_1$ (v.c.) et $BPRE$ (v.c.) :

- $BEMI_1$:
 - . contient l'adresse du bloc que l'on émet ou qui a déjà été émis une fois par le terminal
 - . sera remis à zéro à la réception du ACK correspondant ($NØB - ACK$ ou $DLE - ACK$)
 - . l'émission d'un bloc nouveau ne sera possible que si $BEMI_1 = 0$
- $BPRE$:
 - . contient l'adresse du bloc préparé pour l'émission
 - . sera remis à zéro dès que ce bloc sera pris en compte pour sa première émission
 - . une nouvelle préparation ne peut avoir lieu que si $BPRE = 0$

Puisque, pour émettre les blocs d'information, nous utilisons le sous-programme *DEXIT*, deux variables supplémentaires, $WBEMI$ et $WBPRE$, sont associés respectivement à $BEMI_1$ et $BPRE$.

Ces variables spécifient le nombre de mots des zones dont l'adresse est précisée dans les premières. Par conséquent, l'affectation $BEM11 := BPRE$ sous-entend l'affectation associée $WBEM1 := WBPRE$.

La partie Polling de la procédure se compose des modules SA_0 , SP_1 , SA_1 et SA_2 que nous allons décrire dans cet ordre.

Le traitement de $NØB$ a été détaillé en annexe 3.

IV.1 - Module SA_0

Pour que le Polling ait effectivement lieu, il faut que :

- d'une part le terminal soit "Prêt au Polling" (indicateur de l'état global P_0). Nous interdisons toute modification de cet indicateur au cours de la phase de Polling.
- d'autre part le terminal doit disposer d'un bloc prêt à être transmis, c'est-à-dire que $BPRE$ soit différent de zéro ou $BEM11$ différent de zéro dans le cas de suspension. Ceci correspond au test de l'indicateur de l'état global P_1 : dernier bloc transmis ou momentanément plus rien à transmettre (DBMT).

En cas de test de liaison, le module SA_0 peut savoir, par la valeur de BTL , si le contrôle lui a été donné par SP_0 ($BTL \neq 0$) ou par SP_1 ($BTL = 0$).

La description de SA_0 est alors la suivante :

Si Prêt au Polling alors faire A sinon début envoyer EØT ; ET := mp₀ fin ;

A. début si TL = 1 alors faire B sinon faire C fin A

B. début si BTL ≠ 0 alors début envoyer bloc d'adresse BTL ; ET := mp₁ fin
sinon début envoyer EØT ; TL := 0 ; ET := mp₀ fin

fin B

C. début si BEM11 ≠ 0 alors début envoyer bloc d'adresse BEM11 ;
ET := mp₁ ; initialiser NØB fin

sinon début si BPRE ≠ 0 alors début BEM11 := BPRE ;
envoyer bloc d'adresse
BEM11 ; BPRE := 0 ;
ET := mp₁ fin

sinon début émettre EØT ;

ET := mp₀ fin

fin

fin C

IV.2 - Module SP_1

SP_1 se sert des sous-programmes *ACQUI* et de *RCEØT* (reconnaissance de la séquence EØT). En cas de séquence invalide, la procédure reste en SP_1 . En cas d'acquiescement positif en cours de test de liaison, BTL sera remis à zéro.

Description de SP_1

Si Séquence EØT alors ET := mp₀ sinon début si NØB - ACK alors faire A sinon faire B fin ;

A [début si TL = 1 alors début BTL := 0 ; ET := ma₀ fin
 sinon début BEM1 := BPRE ; BPRE := 0 ; ET := ma₀ fin
fin A

B [début si NØB - NAK alors ET := ma₁
 sinon début si DLE - ACK alors début ET := ma₂ ;
 si TL=1 alors BTL = 0
 sinon début
 BEM1 := BPRE ;
 BPRE := 0
 fin
 fin
 sinon début si DLE-NAK alors ET := ma₂ fin
fin B

IV.3 - Module SA₁

Ce module ne se charge que de réémettre le bloc mal reçu par le central.

Sa description est donc très simple :

[Si TL = 1 alors émettre le bloc d'adresse BTL
sinon émettre le bloc d'adresse BEMII ;
ET := mp₁ ;

IV.4 - Module SA₂

Notons qu'en cas de test de liaison TL doit être remis à zéro si le terminal a reçu la séquence DLE - ACK, donc si BTL = 0.

La description de ce module est donc la suivante :

[Si BTL = 0 alors TL := 0 ;
ET := mp₀ ;
Envoyer séquence EØT ;

Remarques :

- 1) Dans la phase de polling seul le module actif SA_0 tient compte des indicateurs associés aux états globaux du Polling P_0 et P_1 . SA_0 correspond effectivement à la description d'un module actif comme indiqué dans l'introduction de ce chapitre. Nous gardons le qualificatif "actif" pour les modules SA_1 et SA_2 bien que l'élaboration de leur réponse n'ait pas à tenir compte d'indicateurs ; Ses deux modules améliorent la compréhension et leur existence simplifie énormément la programmation.

- 2) Pour que l'enchaînement de ces divers modules soit bien clair, il ne faut pas oublier la préparation des blocs, décrite par ailleurs, et se souvenir que dès que *EXAM* se remet en attente d'une nouvelle séquence, le *travail de fond* (donc la préparation des blocs) est repris au point où il avait été interrompu par l'interruption PAD ou fin de table.

V - SELECTION

La sélection comporte une partie traitement des blocs d'information réalisée en *travail de fond* (cf.II.VI). Des liens existent donc entre ce traitement et la partie sélection proprement dite, ce sont les trois variables communes que nous allons introduire maintenant : BLIB, BTRA et STØP.

Comme pour le polling et pour des raisons analogues, nous disposons de deux tampons T_3 et T_4 pour la réception des blocs d'information. Leur contrôle est assuré par les deux variables BLIB (v.c.) et BTRA (v.c.) :

- . BLIB contient l'adresse du tampon libre ou en cours de remplissage ou en attente de traitement
- . BTRA contient l'adresse du tampon en cours de traitement. Si BTRA = 0 aucun tampon n'est en cours de traitement.

Remarquons que, quand un bloc est susceptible d'arriver, le terminal doit être prêt à le recevoir. Ceci implique évidemment la disponibilité d'un tampon. En conséquence pour un bloc correctement reçu, l'acquiescement positif ne sera envoyé que quand le *travail de fond* aura fini de traiter le précédent bloc, c'est-à-dire BTRA remis à zéro.

Dans le cas contraire, le *travail de fond* a été suspendu par l'interruption de fin de ce bloc correctement reçu ; il faut donc effectuer un retour à ce travail afin d'achever le traitement du bloc précédemment reçu. Nous mettrons alors momentanément la procédure au repos ; aucune séquence ne sera prise en compte durant ce temps. L'indication de "procédure au repos" est donnée par la valeur 0 de la variable STØP (v.c.) ; pour qu'aucune séquence ne soit acceptée par le programme directeur EXAM, ET gardera la valeur ma_4 . Dès que le *travail de fond* aura fini de traiter le bloc précédent, il

enverra NØB - ACK, mettra STØP à 1, la procédure en attente (ET := mp₂) et commencera à traiter le bloc reçu.

Le test de liaison n'a ici presque aucune influence : aucun traitement n'est à effectuer sur les blocs d'information reçus à part, ranger dans BTL l'adresse du dernier bloc.

Le traitement de NØBS a été détaillé en annexe 4.

En ce qui concerne les indicateurs des états globaux, le module SA₃ prend en compte celui de P₀ (prêt à recevoir) et SA₄ celui de P₂ (demande de suspension).

Dans la description, "préparer la lecture", sous-entend lecture d'un bloc d'information et signifie BREC := BLIB et WØRIN := 135.

Les divers modules de la phase de sélection sont présentés dans l'ordre suivant : SA₃, SP₂, SA₄ et SA₅.

V.1 - Module SA₃

La description en est simple.

Si Prêt à recevoir *alors début* ET := mp₂ ; envoyer SEL - ACK ;
BLIB := ad. de T₃ ; initialiser NØBS ;
préparer la lère lecture *fin*

sinon début ET := mp₀ ; envoyer SEL - NAK *fin* ;

V.2 - Module SP₂

Ce module se sert de RCEØT (reconnaissance de EØT) et du sous-programme ANMES qui est chargé d'analyser les blocs d'information reçus. Ce dernier, après diverses vérifications, décide

entre autres choses si le bloc est correct ou non (cf.II.VII).

```
Si Séquence EØT alors ET := mp0
    sinon début si bloc correct alors ET := ma4
        sinon ET := ma5
    fin
```

V.3 - Module SA₄

Ce module ne peut rien faire si le traitement du bloc précédent n'est pas encore terminé ; la procédure est mise au repos dans ce cas.

Description

```
Si BTRA = 0 alors faire A sinon STØP := 0 ;
A début ET := mp2 ; BTRA := BLIB ;
    si BLIB = ad. de T3 alors BLIB := ad.T4
        sinon BLIB := ad.T3 ;
    préparer lecture suivante ;
    Si demande de suspension alors début envoyer DLE-ACK ;
        réinitialiser NØBS fin
    sinon début envoyer NØB-ACK ;
        incrémenter NØBS fin
    fin A
```

V.4 - Module SA₅

[envoyer NØB - NAK ;
ET := mp₂ ;

Remarquons tout de suite que la phase de sélection est la plus difficile à tester sans ligne, car il nous faut simuler toutes les réactions du central et en particulier, l'élaboration des blocs d'information. Nous y reviendrons dans le dernier chapitre.

VI - TRAVAIL DE FOND

Que ce soit en Polling ou en Sélection, la réalisation du *travail de fond* peut être conçue de deux façons différentes :

- sur un niveau d'interruption : ceci nécessite son lancement à l'intérieur des modules mêmes, mais permet d'exécuter des travaux "non-process" pendant les temps morts de la procédure.
- en travail "non process" sous forme d'une boucle dont on ne sort que si une indication de fin est donnée "attente active". Cette dernière conception a l'inconvénient d'accaparer entièrement l'unité centrale du terminal, elle a par contre l'avantage d'être plus simple à réaliser.

Dans un premier temps, nous avons choisi la deuxième solution et pensons la garder jusqu'à la fin des essais. Une fois la procédure complètement mise au point sous cette forme, nous effectuerons les transformations nécessaires pour passer à la première solution.

Nous présentons d'abord le *travail de fond* du polling, puis celui de la sélection et terminons en signalant les diverses initialisations que nécessite le démarrage de la procédure de transmission.

VI.1 - En Polling

Le *travail de fond* de Polling fait apparaître une variable IFIN signalant, si sa valeur est nulle, la fin de la phase de Polling : Un enregistrement vide nous signale qu'il n'y a plus d'informations à transmettre. La lecture de cet enregistrement provoque donc la remise à zéro de IFIN. D'autre part, la préparation d'un bloc d'information ne peut avoir lieu que si BPRE est différent de zéro.

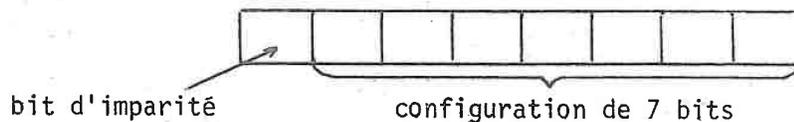
Description générale :

Tant que $IFIN \neq 0$ faire si $BPRE = 0$ alors Préparation d'un bloc ;

Préparation d'un bloc d'informations :

Elle est assurée par le sous-programme *PRBL* qui utilise lui-même les sous-programmes suivants :

- *MAE* : préparation d'un article à partir de la machine à écrire
- *LC* : préparation d'un article à partir du lecteur de cartes
- *PATR* : calcul du bit d'imparité pour une configuration de 7 bits :



- *ARTI* : rangement, en effectuant éventuellement un transcodage, d'un article donné dans la zone à préparer.
- *PALØ* : calcul du caractère BCC d'imparité longitudinale.

Ces cinq sous-programmes n'ayant qu'un caractère technique, ne seront pas décrits ici. Nous nous limiterons à la description sommaire de *PRBL*.

Les paramètres de *PRBL* sont :

- paramètres d'entrée :
 - . *MBL* (v.c.) contient l'adresse de début de la zone à préparer (T1 ou T2) ; le nombre de mots en est fixé : 135
 - . *MPER* (v.c.) spécifie indirectement l'identification du périphérique concerné.

- paramètres de sortie : . MNB (v.c.) contient l'adresse du mot où *PRBL* doit ranger le nombre de caractères de la zone préparée.
- . MIF (v.c.) contient l'adresse du mot qui doit être remis à zéro si ETX a été mis dans le bloc préparé.

La préparation d'un bloc comprend donc les opérations :

- affecter les valeurs qui conviennent aux variables énumérées ci-dessus
- CALL *PRBL* ;
- BPRE := MBL ;
- WBPRES := nombre de mots de la zone préparée ;

Les différentes étapes de *PRBL* sont :

- ① préparer le numéro du bloc à partir de NØB (appel de *PATR*) ;
- ② initialiser certaines variables et préparer certaines instructions suivant le périphérique concerné ;
- ③ si un article est déjà prêt alors aller en ⑥ ;
- ④ préparer un article (appel de *MAE* ou *LC*) ;
- ⑤ si enregistrement vide alors début mettre ETX ; mettre à zéro le mot d'adresse précisée par MIF ; aller en ⑦ fin ;
- ⑥ $\left. \begin{array}{l} \text{encore assez de} \\ \text{place dans la} \\ \text{zone} \end{array} \right\} \text{ alors début ajouter l'article (appel de } \textit{ARTI} \text{) ;}$
aller en ④ fin
sinon mettre ETB ;
- ⑦ ajouter les caractères BCC et PAD (appel de *PALØ*) ;
- ⑧ ranger le nombre de caractères à l'adresse précisée par MNB.

VI.2 - En Sélection

La variable IFSE signale la fin de la phase de sélection. IFSE est mis à zéro par l'apparition de ETX dans un bloc reçu.

```
Tant que IFSE U BTRA ≠ 0 faire
début si  $\exists$  ETX alors IFSE := 0 ;
      si TL = 1 alors BTL := BTRA
          sinon début traitement du bloc d'adresse BTRA ;
              si STØP = 0 alors faire A du module SA4 ;
          fin
      fin ;
```

Le traitement du bloc consiste à le ranger dans un fichier sur disque qui sera exploité par la méthode d'accès : suivant le préfixe de chaque article perforation ou impression du contenu de l'article.

VI.3 - Initialisations

Diverses initialisations sont nécessaires pour le démarrage de la procédure. Notamment :

```
STØP , IFIN , IFSE := 1 ;
ET , TL , BPRE , BEM11 , BTRA := 0 ;
```

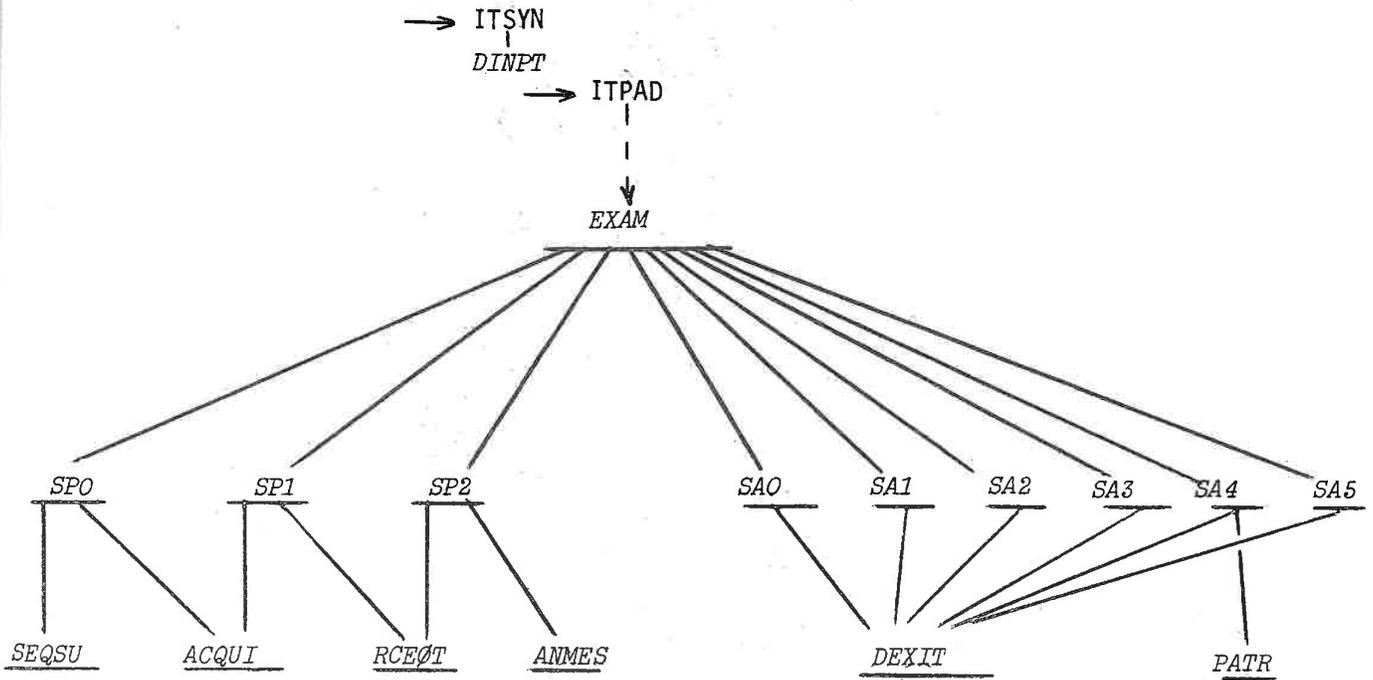
L'initialisation des indicateurs associés aux états globaux est faite par l'intermédiaire de la méthode d'accès.

Signalons de plus que la préparation des blocs peut être lancée par la méthode d'accès bien avant la réception d'une séquence de supervision de polling.

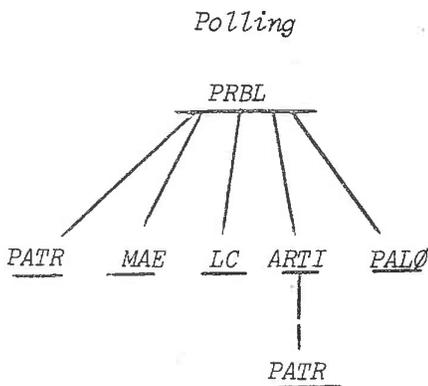
RECAPITULATIF

1) Enchaînement des modules et des sous-programmes

. Procédure



. Travail de fond



Sélection

Traitement

DEXIT

- > interruption externe
- - - - -> interruption programmée
- > appel de sous-programme CALL

2) variables communes essentielles

Variables	Modules ou Sous-Programmes les utilisant
ET	<i>EXAM, SPO, SP1, SA1, SA2</i> <i>SP2, SA3, SA4, SA5</i> Travail de fond Sélection
BREC, WØRIN	<i>DINPT</i> , passage $SA_i \longrightarrow SP_j$
BEMI, WØRDC	<i>DEXIT</i> , passage $SP_l \longrightarrow SA_k$
TL, BTL	<i>SPO, SA0, SP1, SA1, SA2</i>
BEMI1, WBEMI	<i>SA0, SP1, SA1</i> } Travail de fond Polling
BPRE, WBPRE	<i>SA0, SP1</i> }
NØB	<i>SPO, SA0, PRBL</i>
BLIB	<i>SA3, SA4</i> }
BTRA	<i>SA4</i> } Travail de fond Sélection
STØP	<i>SA4</i> }
NØBS	<i>SA3, SA4, SP2</i>
MBL, MPER, MNB, MIF	paramètres pour <i>PRBL</i>
ORDRC, ORDC2	paramètres supplémentaires pour l'analyse des messages

VII - Analyse des messages

VII.1 - Analyse des séquences de supervision (SEQSU)

Entrées de *SEQSU*

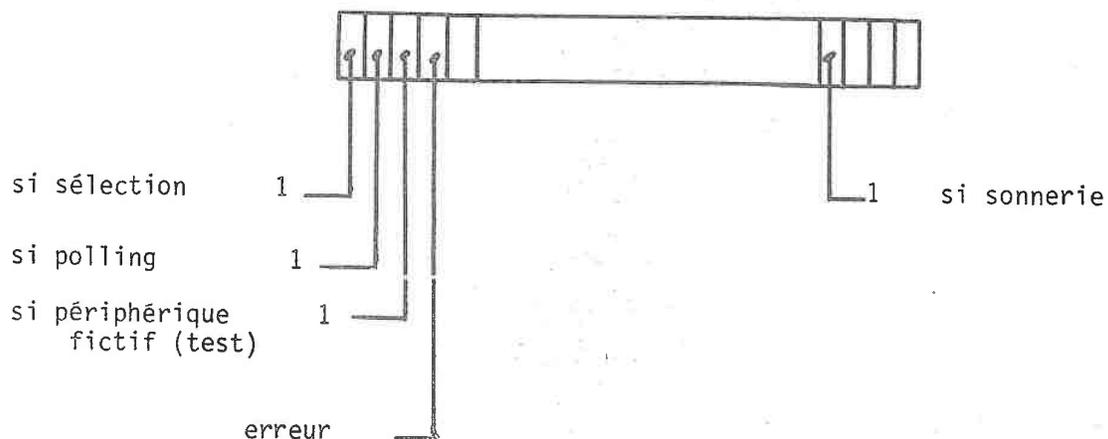
C(BREC) est l'adressé du mot contenant le 1er caractère de la séquence

C(ØRDRC) est l'adresse dans le mot du 1er caractère

Sorties de *SEQSU*

- ETAT contient des indicateurs représentant le résultat de l'analyse de la séquence de supervision si elle est correcte.
- Si la séquence est incorrecte, nous positionnons un mot ETAT1 à diverses valeurs suivant l'erreur détectée (ceci pour des facilités de mise au point).

ETAT



Ce sous-programme est écrit simplement.

Il est constitué d'une suite de tests.

```
Début si "le 1er caractère n'est pas ENQ" alors aller à SØR1 ;
      si "le 2nd caractère n'est pas l'adresse correcte de la
         station" alors aller à SØR1 ;
      si "le 2nd caractère n'est ni sélection, si scrutation"
         alors aller à SØR1 ;
      si "sélection" alors ETAT := /8000 * ;
      si "scrutation" alors ETAT := /4000 ;
      si "périphérique fictif" alors ETAT := ETAT . ØR./ 2000 ;
      si "sonnerie"      alors ETAT := ETAT . ØR./ 0008 ;
      aller à FIN ;

SOR1 : ETAT := /1000 ;
FIN   : fin ;
```

* /8000 représente $(8000)_{16}$

VII.2 - Reconnaissance des messages d'acquiescement (ACQUI)

Entrées de *ACQUI* : C(BREC) }
 C(ØRDRC) } comme pour *SEQSU*

 C(NØB) sur les trois bits de poids forts
 de NØB, le numéro du dernier bloc
 émis.

Sorties de *ACQUI*

ETAT2 contient des valeurs diverses suivant le message d'acquiescement reconnu.

Début

Si "DLE - ACK" *alors* ETAT2 : = /0100 ;
Si "DLE - NAK" *alors* ETAT2 : = /0200 ;
Si "NØB - ACK" *alors* ETAT2 : = /0400 ;
Si "NØB - NAK" *alors* ETAT3 : = /0800 ;
 RX2 : = /00FF ;

Fin

VII.3 - Reconnaissance de EØT (RCEØT)

entrées de *RCEØT* : C(BREC) et C(ØRDRC) comme dans *ACQUI*

sorties de *RCEØT* : *si* "EØT" *alors* RX1 : = /0010
 sinon RX1 : = /0020 ;

VII.4 - Analyse des messages courants

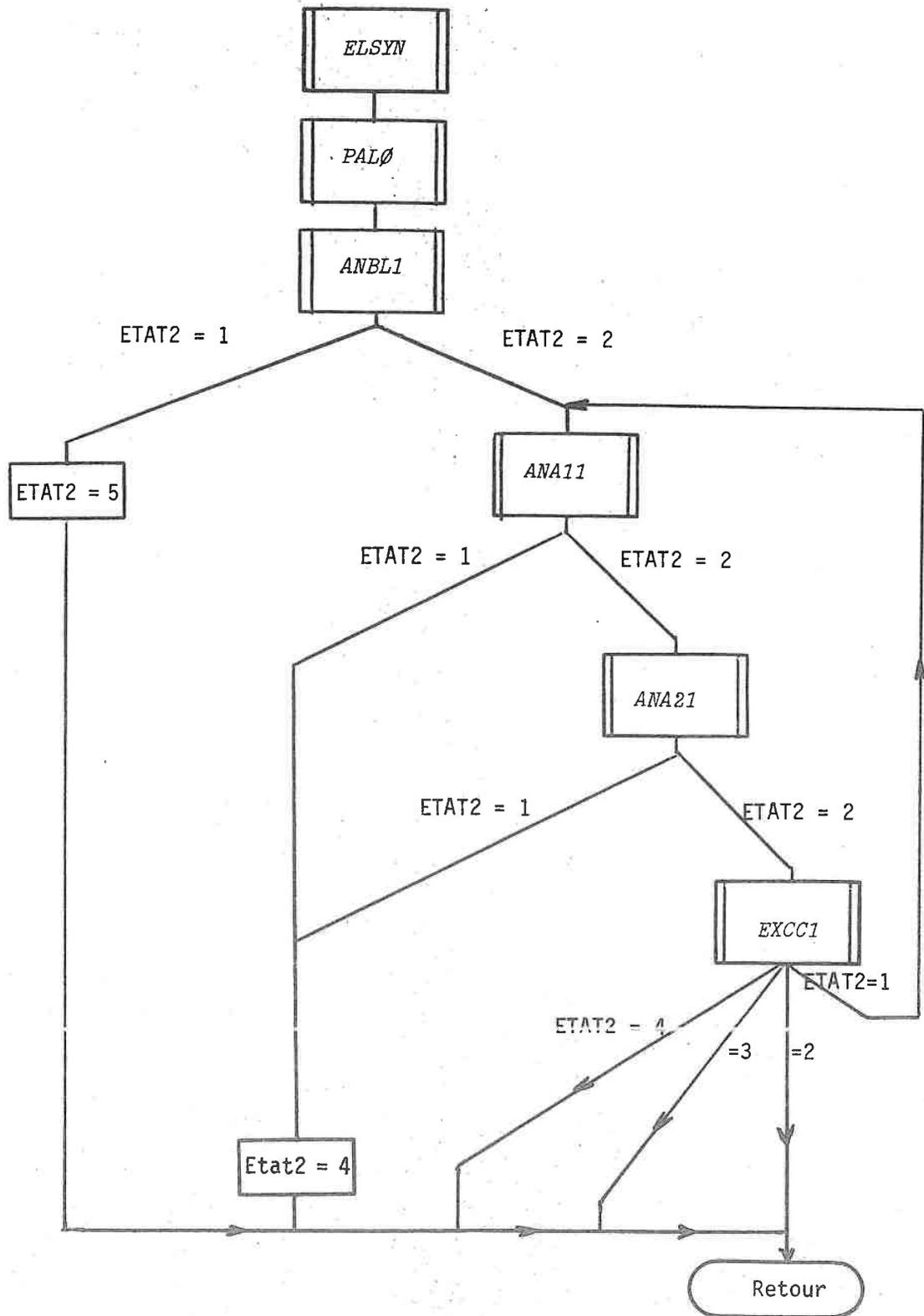
L'analyse des messages courants est réalisée par le sous-programme *ANMES*. Ce sous-programme est lui-même composé d'appels de plusieurs sous-programmes ayant chacun leur fonction propre.

- Elimination des caractères de synchronisation (*ELSYN*)
- Calcul de parité longitudinale (*PALØ*)
- Analyse des préfixes de blocs (*ANBL1*)
- Analyse des préfixes d'articles (*ANA11*) et (*ANA21*)
- Analyse d'un article courant (*EXCC1*)

A la sortie de *ANMES*

ETAT2 = 2 signifie fin de session (ETX rencontré)
ETAT2 = 3 signifie fin de message
ETAT2 = 4 signifie bloc erroné
ETAT2 = 5 signifie bloc invalide

Organigramme de ANMES



VII.4.1 - *Elimination des caractères de synchronisation (ELSYN)*

La synchronisation du coupleur récepteur n'est pas toujours réalisée après la réception d'un nombre constant de caractères de synchronisation.

Les caractères de synchronisation reçus après la synchronisation effective du coupleur ne devraient pas être pris en compte. Or le coupleur actuel ne réalise pas cette fonction. Donc nous la réalisons par software tant qu'elle ne le sera pas par hardware.

En entrée de *ELSYN*

C(BREC) = adresse du mot contenant les deux premiers caractères du message

En sortie

C(BREC) = adresse du mot contenant le premier caractère différent de "SYN"

C(ØRDRC) = adresse dans le mot chaîné par C(BREC) du 1er caractère différent de SYN C(ØRDRC) = 0 ou 1

C(ØRDRC2) = complément de C(ØRDRC)

si ØRDRC = 0 *alors* ØRDRC2 := 1

sinon ØRDRC2 := 0 ;

De plus, dans ce sous-programme, nous réinitialisons certains mots mémoire.

VII.4.2 - Analyse des préfixes de bloc ANBL1

Ce sous-programme analyse les quatre caractères d'en-tête d'un bloc de SØH à STX inclus.

Entrées de ANBL1

C(BREC) et C(ØRDRC) positionnés dans ELSYN

Sortie de ANBL1

ETAT2 = 2 si le préfixe de bloc est correct

ETAT2 = 1 si le préfixe de bloc est incorrect

début si le premier caractère n'est pas SØH alors aller à RETOUR 1 ;

si "le deuxième caractère n'est pas

le caractère de sélection correct alors aller à RETOUR 1 ;

si le numéro du bloc est erroné alors aller à RETOUR 1 ;

si le quatrième caractère n'est pas STX

alors aller à RETOUR 1

sinon aller à RETOUR 2 ;

RETOUR 1 : ETAT2 : = 1 ; *aller à* FIN ;

RETOUR 2 : ETAT2 : = 2 ;

FIN : *fin* ;

Dans le sous-programme ANMES, après l'appel de ANBL1 un aiguillage permet de suivre l'une des deux possibilités.

VII.4.3 - Analyse du préfixe d'article

L'analyse du préfixe d'article est réalisée par deux sous-programmes *ANA11* et *ANA21*.

ANA11 analyse le 1er caractère du préfixe

ANA21 analyse le 2ème et éventuellement les 3ème et 4ème caractères (taille de l'article).

Le premier caractère indique la destination de l'article (code périphérique). Ce caractère permettra d'aiguiller l'article dans un certain tampon propre à sa destination.

VII.4.4 - Analyse des caractères courants EXCC1

Ce sous-programme analyse tous les caractères compris entre le dernier caractère du préfixe d'article exclus et le dernier caractère d'un article (RS, ETB ou ETX).

Les sorties de *EXCC1* sont résumées en positionnant ETAT2 à certaines valeurs :

ETAT2 = 1 le dernier caractère est RS et *ANMES*
commandera l'analyse de l'article suivant.

ETAT2 = 2 le dernier caractère est ETX
le message reçu est le dernier d'une session

ETAT2 = 3 le dernier caractère est ETB
le message est terminé

ETAT2 = 4 il y a une erreur quelconque dans le message

CHAPITRE III

PROCEDURE SYMETRIQUE

I - INTRODUCTION

Sur 10070 la procédure de transmission actuellement implémentée est TMM - RB. Nous avons décrit (chap. I) et réalisé (chap. II) une procédure terminale 1800 compatible avec TMM - RB, seule procédure centrale dont nous disposons. Ceci nous permet d'utiliser l'I.B.M. 1800 comme terminal lourd, premier but défini dans l'Introduction, mais interdit du fait de sa non-symétrie, la coopération effective des deux ordinateurs en temps réel, second but de la liaison (cf. Introduction).

Pour pouvoir réaliser ce dernier point, nous souhaiterions disposer d'une procédure telle que l'ordinateur de l'E.N.S.E.M. puisse éventuellement avoir l'initiative des échanges (désir de l'utilisateur, exigence du processus contrôlé, ...) et soit, au cours du transfert, moins tributaire de celui de l'I.U.C.A.

I.1 - Phase initiale

Ces souhaits impliquent que les notions de central et de terminal ne préexistent pas à l'échange d'informations. Les deux unités centrales doivent avoir un rôle parfaitement symétrique dans le contrôle des transmissions ; chacune doit être libre d'envoyer des blocs d'information à l'autre lorsqu'elle le désire. La station qui émettra les blocs d'information sera dite "maîtresse" ; l'autre, ayant reconnu à la première ce statut maître, sera dite "esclave".

La conséquence la plus importante de ce qui précède est que les séquences de supervision, apanage du central dans la procédure TMM - RB, peuvent ici être émises par l'une ou l'autre des deux stations. Bien sûr, ces séquences de supervision sont destinées à attribuer le statut maître à l'une des deux, le statut esclave à l'autre ; à aucun moment, elles ne devront avoir le même statut.

- . Dans le cas où la liaison est en mode "Full Duplex", chaque station utilise deux voies :
 - une voie pour l'émission de ses messages
 - une voie pour la réception des messages venant de l'autre

La voie émission de l'une correspond évidemment à la voie réception de l'autre. Alors chacune des deux stations peut obtenir le statut maître sur sa voie émission et le statut esclave sur sa voie réception. Dans ce type de liaison, il ne peut donc y avoir de conflit en ce qui concerne l'attribution des statuts. Au lieu d'une procédure de transmission unique, il y en aura en fait deux : une pour l'émission et une pour la réception. Ces deux procédures, nettement séparées, pourront éventuellement fonctionner en "simultanéité".

- . Dans le cas de la liaison en mode "Half Duplex", la même voie est utilisée pour l'émission et la réception des blocs d'information. La station qui aura la première acquis le statut maître, pourra transmettre ses blocs d'information à l'autre. Un conflit peut alors se produire si les deux sites réclament en même temps le statut maître ; nous examinerons ce problème dans le paragraphe III de ce chapitre.

I.2 - Phase de transfert

Si une certaine symétrie au cours de la phase initiale a été ainsi obtenue, il est nécessaire de la compléter par des initiatives au cours de la phase de transfert.

En particulier, il est souhaitable que, sur le plan des reprises en cas d'erreur, la station esclave ne dépende pas autant

de la station maîtresse que le terminal du central dans la procédure TMM - RB. Cette souplesse peut être obtenue par quelques dispositions simples :

- chaque site, quel que soit son statut, tiendra compte de sa propre temporisation d'attente de réponse au bout de laquelle elle élabore une action : répétition du message précédent, demande de suspension, arrêt de la transmission...

Cet intervalle de temps, fixé d'avance, est unique pour la station concernée.

- chaque site comptera, pour une réponse attendue, le nombre p de non réponses ou de blocs invalides et le nombre n de séquences incorrectes. Pour chacun un maximum est fixé. En cas de dépassement d'un de ces deux maxima (P ou N) la transmission sera arrêtée et la liaison considérée comme non opérationnelle.
- la station esclave doit pouvoir demander une suspension à la réception d'un bloc quelconque, correct ou incorrect.

II - LES MESSAGES

La description des messages de la procédure TMM - RB, en EBCDIC mode binaire, reste largement valable ici. Nous nous contentons de signaler les quelques points de divergence.

II.1 - Les séquences de supervision sont ici :

- la demande de statut maître (DSM) émise par la station 1 qui veut initialiser un transfert d'informations.

SEL1 - SEL2 - ENQ

- l'attente du statut esclave (ASE) qui constitue la réponse de la station 2 à la demande précédente :

- . SEL1-ACK : si la station 2 est prête à recevoir le premier bloc d'information. Par l'émission de cette séquence, elle reconnaît son statut esclave et à la station 1, le statut maître ; la phase de transfert des blocs peut alors commencer.

- . SEL1-NAK : si la station 2 n'est pas prête à recevoir les blocs. La liaison est remise au repos par une séquence EØT émise par la station 1.

II.2 - Les blocs d'information et les accusés de réception ont des formats identiques à ceux de la procédure TMM - RB en EBCDIC mode binaire.

II.3 - Les demandes de suspension

- a) la station esclave a la possibilité d'interrompre la transmission après réception d'un bloc quelconque, correct ou incorrect, en demandant à la station maîtresse de suspendre l'émission de ses blocs :

DLE - ACK : si le dernier bloc a été correctement reçu et pris en compte

DLE - NAK : si le dernier bloc n'a pas été pris en compte

La réponse de la station maîtresse est impérativement l'envoi d'une séquence EØT qui met fin à la transmission et fait perdre à la station son statut maître.

- b) si c'est la station maîtresse qui veut interrompre la transmission de ses blocs d'information, elle ne peut le faire qu'à la réception d'un accusé de réception positif de la station esclave en envoyant une séquence EØT. Ceci afin d'être sûr que tous les blocs émis ont été correctement reçus par l'esclave, en particulier le dernier.

III - REPRISES ET CONFLITS

Les reprises sont les actions entreprises quand soit la séquence reçue est incorrecte ou invalide, soit la temporisation du site considéré est épuisée. Ces actions règlent les problèmes de la sécurité de la transmission. De plus l'ajustage convenable des temporisations apporte une solution acceptable aux rares conflits pouvant se produire en "Half Duplex" uniquement.

III.1 - Les reprises

De façon générale une station, quel que soit son statut, répètera le message précédemment émis, dans la limite des maxima P et N fixés d'avance, lorsque, après avoir émis une séquence, elle reçoit :

- une séquence incorrecte ou invalide
- aucune réponse après un intervalle de temps fixé (temporisation).

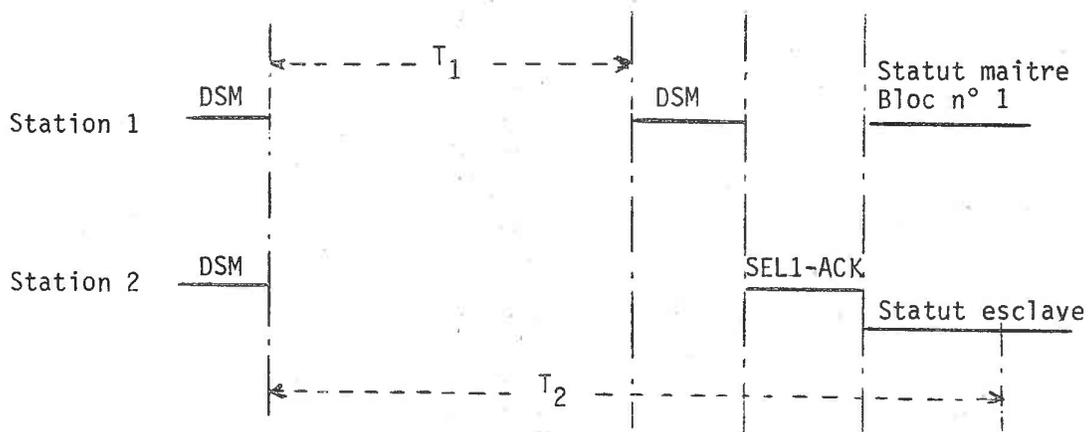
Si P ou N est atteint, la liaison sera considérée comme non opérationnelle et aucun message n'est plus émis, sauf la séquence EØT si cela se produit sur demande de statut maître ou en phase de transfert pour la station maîtresse ; cette séquence EØT remet la liaison au repos et fait perdre le statut maître à la station qui l'a émise.

III.2 - Les conflits

En "Half Duplex", coupleur et modem ne sont pas prévus pour travailler simultanément en réception et en émission : seul le mode en alternat est possible.

Or, dans notre procédure générale, chaque station peut initialiser la transmission ; un conflit est donc possible si les deux extrémités émettent simultanément une demande de statut maître. Les deux ordinateurs sont en émission et aucun n'est en réception à ce moment là ; la demande de statut maître ne sera donc reçue ni d'un côté, ni de l'autre.

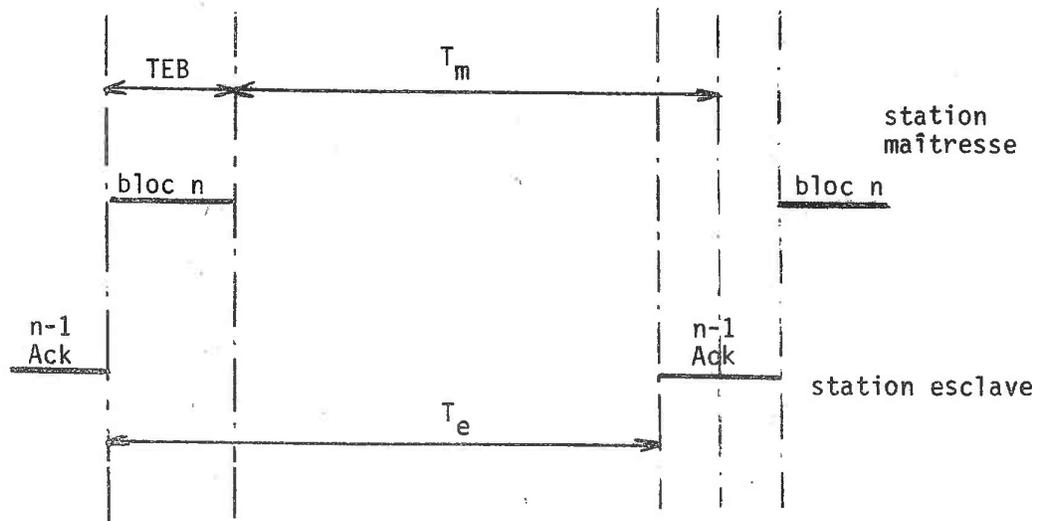
Le règlement de ce conflit s'opère à l'aide des reprises sur non-réponse. Il suffit en effet que les temporisations d'attente de réponse des deux stations soient de durées différentes. La C.I.I. estime que si des durées sont dans un rapport 2/3, le conflit trouvera sans doute une solution dès la première reprise : la station qui a la temporisation la plus brève, a bien sûr les plus grandes chances d'obtenir le statut maître.



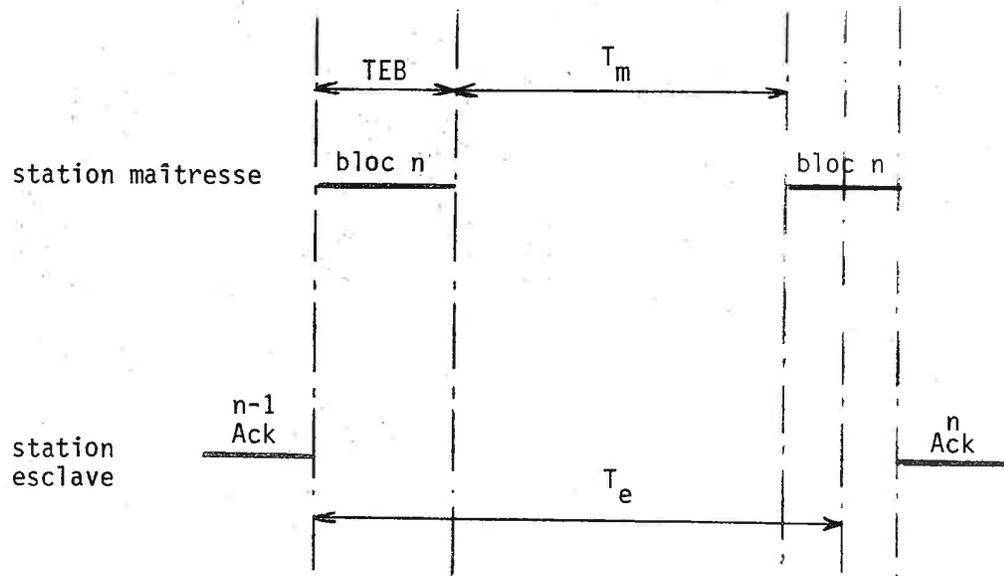
Sur le schéma ci-dessus la station 2 devra reconnaître la demande de statut maître de la station 1 et lui céder ce statut alors qu'elle a elle-même envoyé une demande analogue.

D'autre part, puisqu'au cours de la phase de transfert, la station maîtresse émet les blocs d'information utile, une légère priorité doit lui être accordée en ce qui concerne les reprises. Ceci implique qu'en cas de non réponse sur les deux stations, la priorité soit donné à la répétition du bloc d'information sur l'accusé de réception. Soient alors les temporisations T_m et T_e respectivement du maître et de l'esclave, la proposition précédente n'est satisfaite que si $T_e > T_m + TEB$ ($TEB = \text{Temps d'Emission d'un bloc}$).

Exemple : Supposons que, pour une raison quelconque, la station esclave n'ait pas reçu de réponse à son accusé de réception, alors si la relation $T_e > T_m + TEB$ n'est pas satisfaite, la répétition de l'accusé de réception précèdera celle du bloc.



Tandis que si cette relation est satisfaite, l'inverse se produira.



En résumé, la priorité est donnée à l'une quelconque des deux stations pour la phase initiale, à la station maîtresse pour toute la phase de transfert. Par conséquent, chacune des deux stations doit pouvoir a priori disposer des deux temporisations, l'une courte et l'autre longue $T\emptyset C$, satisfaisant les relations citées ci-dessus.

Néanmoins une seule sera active sur chaque site à un moment donné de la transmission :

- au cours de la phase initiale, $T\emptyset C$ pour la station 1 et $T\emptyset L$ pour la station 2
- au cours de la phase de transfert, $T\emptyset C$ pour le site ayant obtenu le statut maître, $T\emptyset L$ pour l'autre.

IV - ANALYSE ET REPRESENTATION DE LA PROCEDURE

Notons d'abord quelques différences essentielles avec TMM - RB :

- dans TMM - RB, seul un évènement extérieur pouvait provoquer une sortie. Or ceci est trop restrictif ici vu l'initiative laissée à chacune des extrémités. Donc quelques évènements intérieurs, par exemple le désir d'émettre certaines informations, doivent pouvoir provoquer une sortie d'information.
- l'évolution de certaines variables internes (les compteurs p et n,..) provoquent ou conditionnent quelques transitions. De ce fait, nous les représenterons en tant que sorties dans les différentes tables.
- les différentes phases de transfert de la station esclave étant très nettement séparées, nous avons augmenté le nombre d'états globaux de cette partie.

Nous essayerons de faire une "analyse descendante" de la procédure (BJORNER, (3)).

IV.1 niveau 1 :

Nous y distinguons essentiellement trois états :

- P_0 état initial de repos, mais non d'attente comme dans TMM - RB, car un évènement intérieur peut provoquer une demande de statut maître.
- P_1 où l'extrémité a acquis le statut maître
- P_2 où elle a acquis le statut esclave. Nous verrons plus tard comment P_2 éclatera en plusieurs états globaux.

A ces 3 états principaux que nous allons analyser plus en détail, il faut ajouter un quatrième état P_{-1} où la liaison est non opérationnelle et nécessite une intervention humaine (voir fig.III.1).

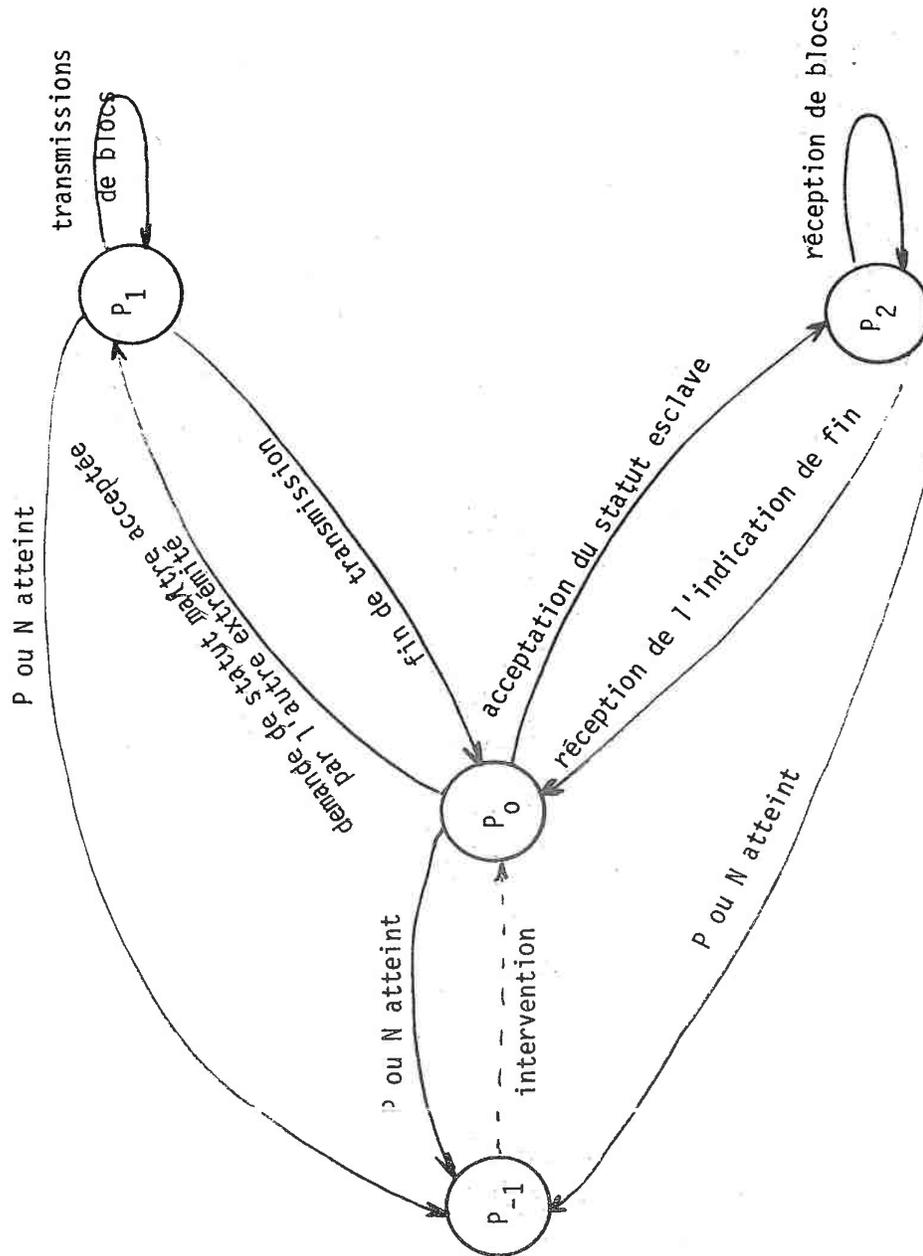


FIGURE III.1 - Schéma général

IV.2 niveau 2

La description de chacune des trois parties mettra en évidence les divers états partiels. Les conventions de représentation sont celles utilisées au chapitre II. Les compteurs p et n sont bien sûr uniques pour toute la procédure et ne sont donc pas attachés à un état quelconque.

2.1 - *Etat* P_0

Deux indications sont attachés à cet état global

- désir d'émettre D_e
- désir de recevoir D_r (et donc prêt à recevoir)

Signalons que le conflit sur demande de statut maître ne peut trouver de solution que si les deux indicateurs précédents sont positionnés.

• Etats partiels de P_0

$D_e D_r$	00	01	10	11	
P_{00}	P_{00}	P_{01}	P_{02}	P_{03}	← état partiel correspondant à la configuration
			DSM	DSM	← action extérieure
	$p,n:=0$	$p,n:=0$	$p,n:=0$	$p,n:=0$	← action intérieure

} entreprises lors du passage de P_{00} à P_{0i} ($i = 1,2,3$)

• Table des états et des sorties

état partiel \ évèn ^t ext.	DSM	SEL1 ACK	SEL1 NAK	NRP	SIC	EØT
P_{0_0}	P_0				P_0	P_0
	SEL1-NAK				-	-
					-	Raz
P_{0_1}	P_2			P_0	P_0	
	SEL1-ACK			-	-	
	* $p, n := 0$			$n := n + 1$	$p := p + 1$	
P_{0_2}	P_0	P_1	P_0	P_0	P_0	
	SEL1-NAK	1er bloc	EØT	DSM	DSM	
	Raz	$p, n := 0$	Raz	$n := n + 1$	$p := p + 1$	
P_{0_3}	P_2	P_1	P_0	P_0	P_0	
	SEL1-ACK	1er bloc	EØT	DSM	DSM	
	* $p, n := 0$	$p, n := 0$	Raz	$n := n + 1$	$p := p + 1$	

* Il faudra ajouter $SER := 0$ (cf. état P_2)

FIGURE III.2 = Table de P_0

Les cas non spécifiés reviennent à une séquence incorrecte (SIC). Notons encore une fois que la temporisation d'attente de réponse n'est bien sûr mise en route que si l'extrémité est en attente d'un message, ce qui n'est pas le cas pour l'état partiel P_{0_0} .

• Graphe des transitions partant de P_0

Dans la figure III.3, nous ne représenterons ni les transitions entre les états partiels de P_0 , ni les actions intérieures.

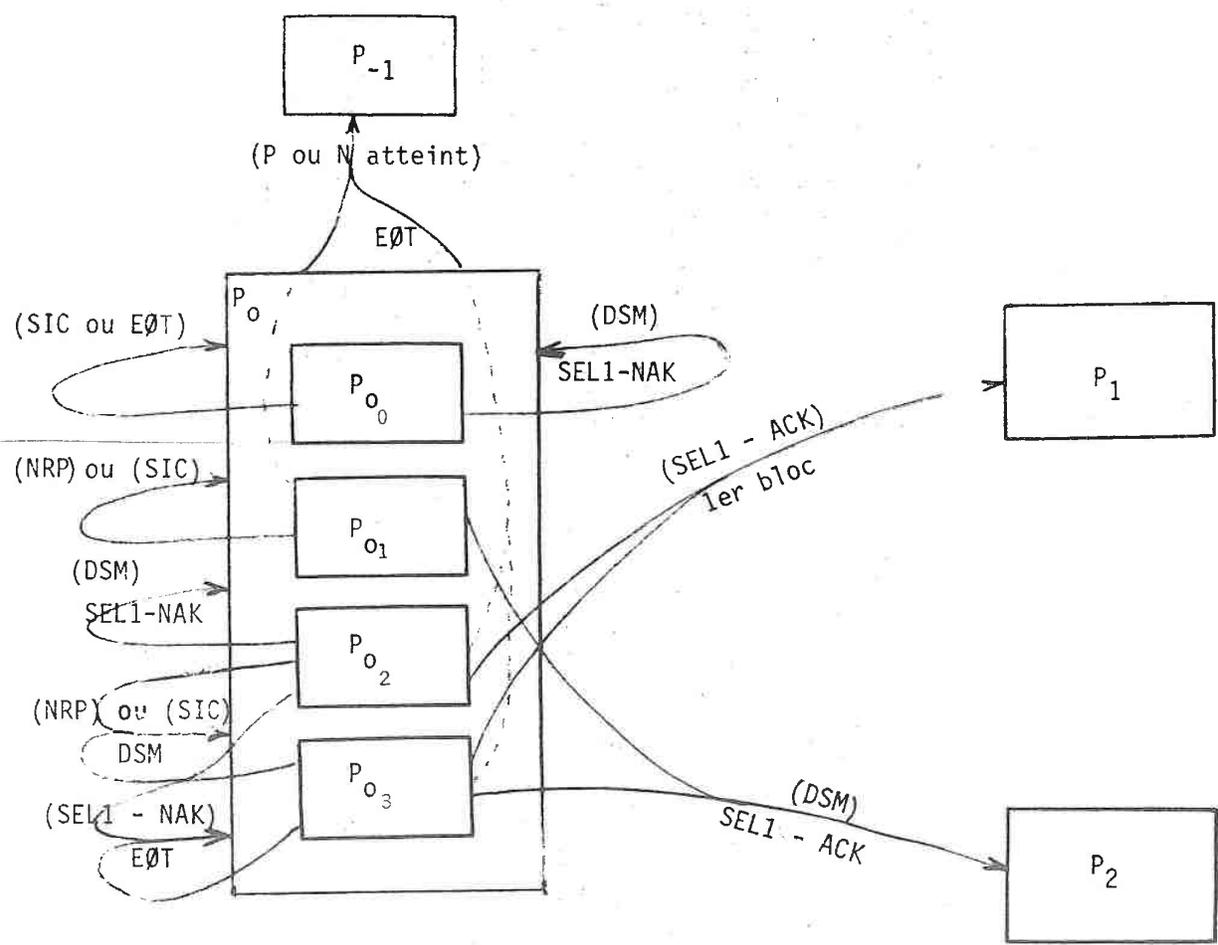


FIGURE III.3 - Schéma de P_0

2.2 - Etat P_1

L'état P_1 est celui où notre extrémité a acquis le statut maître. Deux indicateurs sont normalement attachés à cet état :

- indicateur dernier bloc émis
- demande de suspension

Mais puisque la station maîtresse ne peut suspendre la transmission qu'après avoir reçu un NØB - ACK, il est inutile de positionner l'indicateur de demande de suspension quand le dernier bloc a déjà été mis. Donc nous confondrons, du moins à l'analyse, les deux indicateurs en un seul : demande de suspension. La valeur de ce dernier indicateur constitue l'indice de P_1 .

. Table des états et des sorties

état partiel \ évèn ^t ext	NØB - ACK	NØB - NAK ou SIC	DLE ACK ou NAK	NRP
P_{1_0}	P_1	P_1	P_0	P_1
	bloc suivant	même bloc	EØT	même bloc
	$p, n := 0$	$p := p + 1$	Raz	$n := n + 1$
P_{1_i}	P_0	P_1	P_0	P_1
	EØT	même bloc	EØT	même bloc
	Raz	$p := p + 1$	Raz	$n := n + 1$

FIGURE III.4 - Table de P_1

. Graphe des transitions partant de P_1

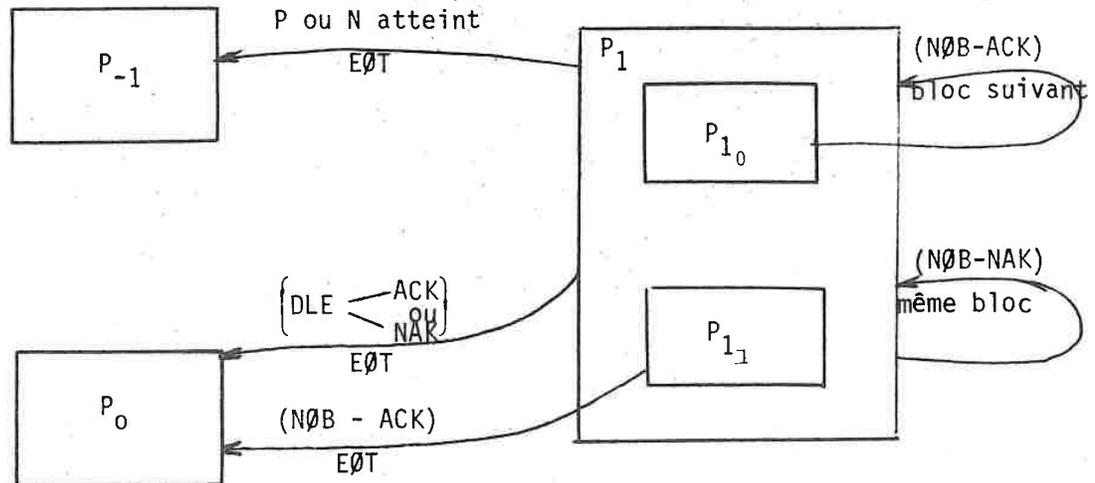


FIGURE III.5 - Schéma de P_1

Cette partie de la représentation ressemble à celle du terminal en phase de polling sous TMM - RB.

2.3 *Etats* P_2 et P_3

En statut esclave, nous distinguons trois étapes bien différentes :

- s'assurer que nous sommes bien en statut esclave. Seule la réception d'un bloc correct peut en effet nous confirmer ce statut ; tant que cet évènement ne s'est pas produit, la station reste en attente du statut esclave.

- phase de transfert effectif
- une demande de suspension a été faite dans une des deux étapes précédentes. Il faut alors savoir si le dernier bloc a été reçu incorrectement ou non : valeur de l'indicateur DB .

Nous remarquons que pour les deux premières étapes, l'ensemble des entrées possibles est identique. Ces deux étapes formeront donc un seul état, l'état P_2 : elles se distingueront par la valeur de l'indicateur de statut esclave reconnu, SER.

Nous aurons donc, pour P_2 , deux indicateurs associés : SER et demande de suspension. Notons alors que, lorsqu'une suspension a été souhaitée, les actions sont identiques quelle que soit la valeur de SER. Ceci nous conduit à l'existence de trois états partiels :

. Etats partiels

Ⓟ ₂ Demande de suspension	SER	00	01	1-
	P_2	P_{2_0}	P_{2_1}	P_{2_2}

Ⓟ₃ La valeur de DB indique l'état partiel.

. Table des états et des sorties

évt. ext. / état partiel	bloc correct	BIV ou NRP	SIC	EØT
P ₂₀	P ₂	P ₂	P ₂	P ₀
	NØB - ACK	SEL1 - ACK	SEL1 - ACK	-
	SER=1 p,n:=0	n:= n + 1	p:= p + 1	Raz
P ₂₁	P ₂	P ₂	P ₂	P ₀
	NØB - ACK	NØB - NAK	NØB - ACK	-
	p,n: = 0	n:= n + 1	p:= p + 1	Raz
P ₂₂	P ₃	P ₃	P ₃	P ₀
	DLE - ACK	DLE - NAK	DLE - NAK	-
	DB:=0 p,n:=0	DB:=1 p,n:=0	DB:=1 p,n:=0	Raz

FIGURE III.6 - Table de P₂

évt. ext. / état partiel	EØT	SIC	NRP
P ₃₀	P ₀	P ₃	P ₃
	-	DLE - ACK	DLE - ACK
	Raz	p:= p + 1	n:= n + 1
P ₃₁	P ₀	P ₃	P ₃
	-	DLE - NAK	DLE - NAK
	Raz	p:= p + 1	n:= n + 1

FIGURE III.7 - Table de P₃

. Graphe de transition

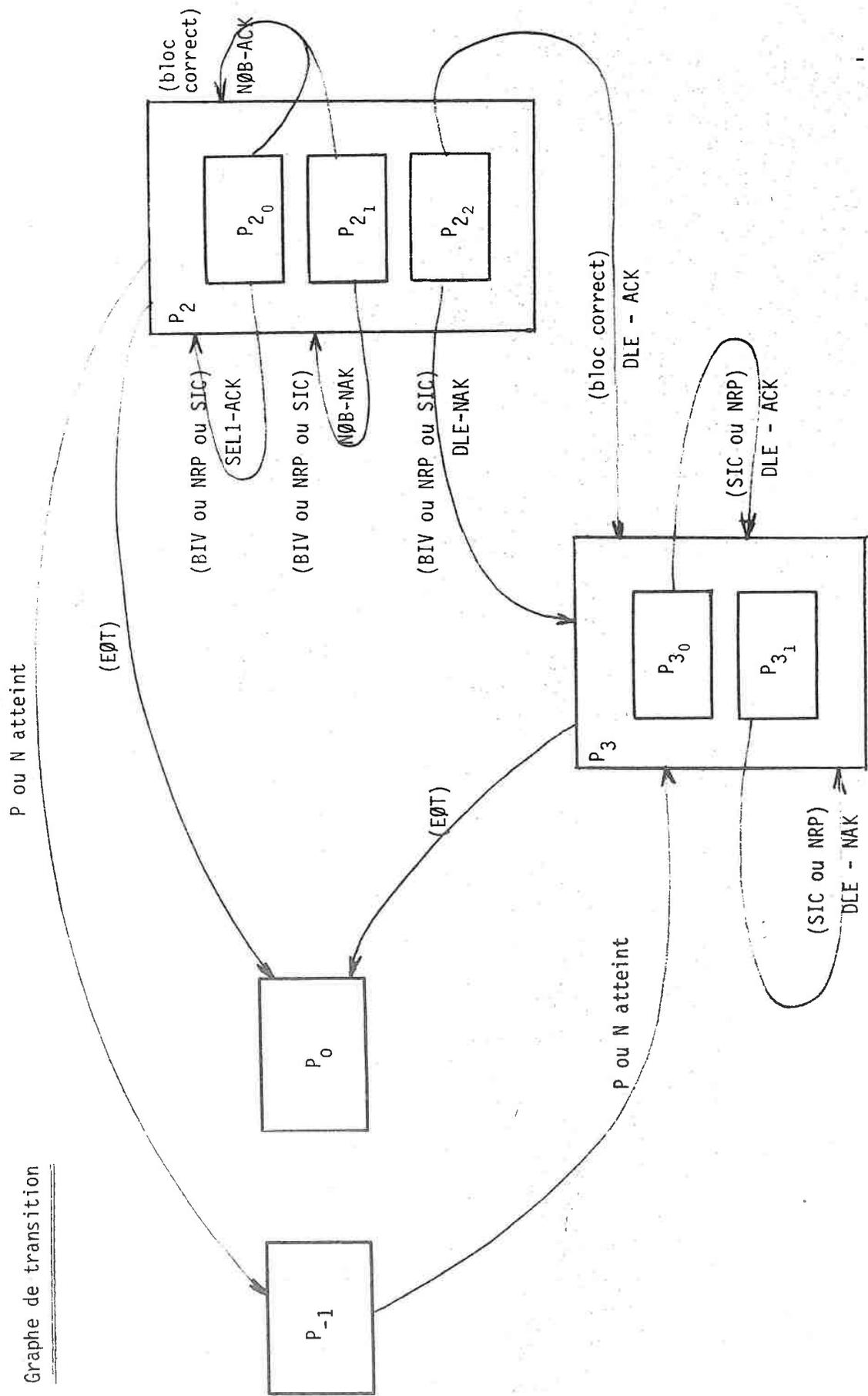


FIGURE III.8 - Schémas de P_2 et P_3

V - PERSPECTIVES POUR LES PROCEDURES

Dans (3) BJØRNER émet quelques idées intéressantes concernant la définition des procédures de transmission. Nées de la nécessité, elles souffrent d'un manque de formalisation dans leur représentation, conduisant ainsi à des réalisations tâtonnantes où les modifications ultérieures sont délicates et difficiles.

V.1 - Description

La description de la procédure souhaitée a été faite en deux niveaux. Nous ne nous sommes intéressés qu'aux séquences globales échangées entre les deux stations. Ces séquences se répartissent en deux groupes :

- les séquences x du langage X accepté par la procédure ; elles sont notées entre parenthèses dans nos représentations.
- les séquences y du langage Y généré par la procédure.

Ces deux langages X et Y sont construits à partir d'un alphabet A comprenant au plus 2^8 symboles en mode binaire, 2^7 en mode ASCII. Dans notre exemple, nous pouvons indiquer l'alphabet A utilisé :

$$A = \left\{ \text{syn, enq, soh, stx, etb, etx, dle, ack, nak, eot, pad} \right\} \cup A_{\alpha} \cup A_{\beta} \cup \text{bcc}$$

où A_{α} est l'ensemble des symboles α autorisés dans les en-têtes et les corps de blocs ;

A_{β} est l'ensemble des symboles non énumérés par ailleurs.

En transmission ASCII, A_{β} comprendra en particulier toutes les configurations de 8 bits ayant un bit d'imparité incorrect.

En transmission binaire, A_{β} est vide.

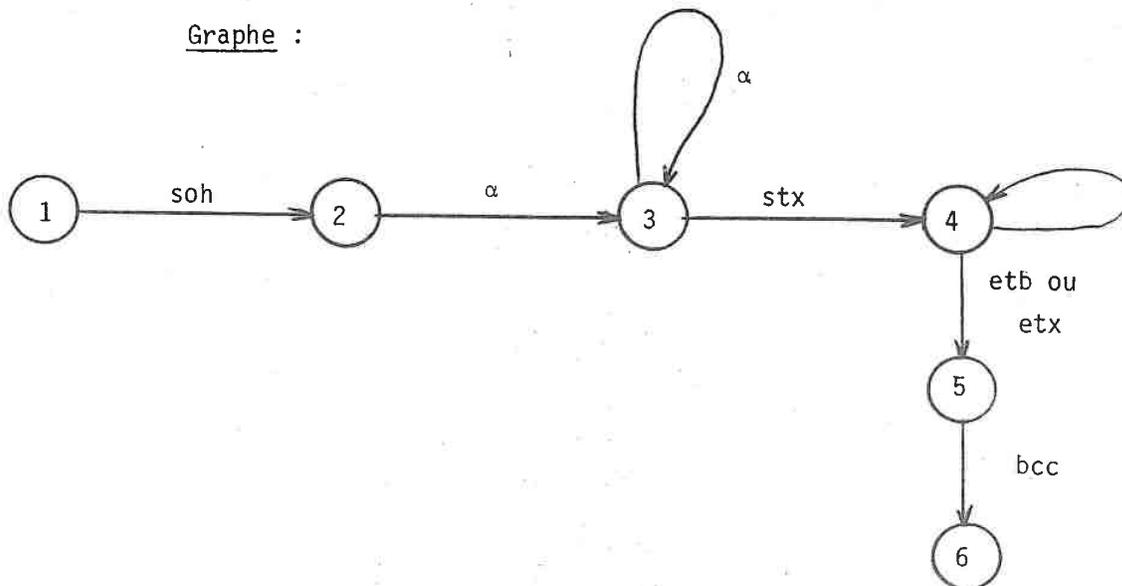
. bcc est, en transmission ASCII un caractère à configuration à priori quelconque, en transmission binaire une configuration de 16 bits.

Les langages X et Y sont des sous-ensembles du monoïde libre A^* .

Nous sommes alors tout naturellement amenés à développer l'analyse de la procédure en y ajoutant un troisième niveau de description : l'examen détaillé du format des messages.

. Limitons nous ici à donner un exemple : le bloc d'information le plus simple possible (soh, en-tête, stx, texte, etb ou etx, bcc).

Graphe :



Règles de production

$$\left\{ \begin{array}{l} E ::= \text{soh } F \\ F ::= \alpha H \\ H ::= \alpha H \mid \text{stx } I \\ I ::= \alpha I \mid \text{etb } J \mid \text{etx } J \\ J ::= \text{bcc} \end{array} \right.$$

ou bien l'expression régulière équivalente

$$\text{soit } \alpha^+ \text{ stx } \alpha^* \text{ (etb } \bigcup \text{ etx) bcc}$$

α^+ désignant une suite finie non vide d'éléments de A

$$\alpha^* = \alpha^+ \bigcup \{ \text{l'élément vide } \Lambda \}$$

Ce troisième niveau de description peut nous donner un aperçu des grammaires engendrant les langages X et Y.

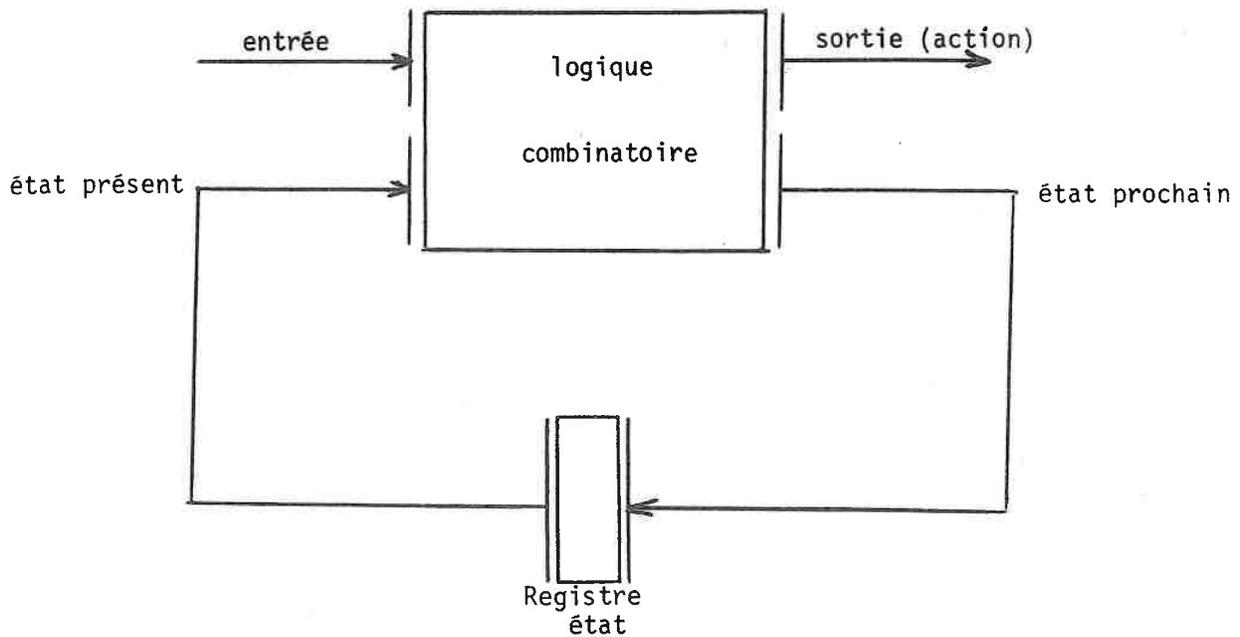
Un quatrième niveau de description est possible en prenant le bit comme information élémentaire. Toutefois, ce niveau est inutile ici pour deux raisons :

- la procédure de transmission considère le caractère comme information élémentaire ;
- la transformation (ensemble de bits) \longrightarrow caractère se fait très simplement à l'aide d'un registre à décalage.

V.2 - Implémentation

Une description complète de la procédure de transmission, jusqu'au troisième niveau indiqué ci-dessus, est d'une grande utilité, quelle que soit l'implémentation choisie, de par son caractère complet et non ambigu. Cette représentation peut servir à l'implémentation d'une telle procédure sur les deux stations car elle décrit à la fois le générateur et le transducteur des transactions. Signalons de plus que l'implémentation peut être faite par hardware ou par software :

- sous forme de circuits séquentiels du modèle ci-après :



- sous forme de procédures software :

Soit S l'ensemble des états et A l'alphabet de l'automate.

Sa programmation se réduit à la consultation d'un tableau à 3 dimensions

$$T (1 : |S| , 1 : |A| , 1 : 2)$$

Etant dans l'état "s" et ayant le caractère "a" en entrée, l'état prochain sera placé en $T(s, a, 1)$ (on y mettra 0 s'il n'existe pas d'état prochain), l'éventuelle action à entreprendre sera mise en $T(s, a, 2)$. L'état initial est 1. L'ensemble final est donné dans un tableau $f(1 : |S|)$, $f(s)$ étant vrai si et seulement si "s" appartient à l'ensemble final.

La consultation du tableau T se réduit alors au programme ci-après :

```
s:=1 ;  
lec: lire (a) ;  
  si a = pad alors écrire (f (s) )  
      sinon début ep:= T (s, a, 1) ;  
          si ep = 0 alors écrire (erreur)  
              sinon début call action (T (s,a,2)) ;  
                  s:= ep ;  
                  aller à lec  
          fin  
      fin  
  fin
```

- sous forme de contrôle microprogrammé :

Le schéma général est alors, pour l'essentiel, identique à celui du circuit séquentiel en substituant au circuit combinatoire l'ensemble décodeur, mémoire de contrôle. Cette dernière contiendra les tables de transitions et des sorties-actions.

V.3 - Conclusion

Cet outil de description a été rarement utilisé jusqu'à présent. (BJORNER, {3}), (LYNCH, (14)). Pourtant le fait de rapprocher les procédures de transmission de la théorie des langages ne peut être que profitable pour leur formalisation. Néanmoins, il reste un problème important, celui créé par l'insécurité des transmissions ; nous l'avons abordé en présentant les reprises consécutives aux séquences incorrectes ou aux non-réponses au bout d'un quantum de temps, temporisation fixée à l'avance pour chaque station. Nous avons résolu ce problème par l'adjonction de transitions supplémentaires entre les états (cf. chap. I surtout), permettant ainsi de prendre en compte certaines répétitions de séquences.

CHAPITRE IV
METHODE D'ACCES

I - INTRODUCTION

Une procédure de transmission gère les échanges des messages définis (au point de vue format, taille, code, détection d'erreurs) sur une ligne de transmission physique.

De plus la procédure de transmission implémentée autorise un accès à distance à quelques ressources définies et inamovibles sauf à grands frais de modifications.

Il est intéressant pour un utilisateur d'être libéré de ces contraintes physiques d'une part et d'avoir accès à partir d'un ordinateur à toutes les ressources hardware et software disponibles sur un autre site d'autre part.

La procédure de transmission ne doit être qu'une fonction de transport (Elie, (8)).

Nous introduisons donc entre l'utilisateur et la procédure de transmission, une méthode d'accès dont le but est de permettre à tout travail de considérer la ligne de transmission comme un simple périphérique en entrée et en sortie.

Ce périphérique est un ensemble constitué de tous les matériels (coupleur, ligne de transmission, modem) et de la procédure de transmission.

II - HYPOTHESES

Actuellement, le système SGT (système de gestion de transmission de la C.I.I. autorise l'accès à certains périphériques du terminal (C.I.I. 21)

Les périphériques sont :

- 1) dans le cas du central émetteur :
 - . une imprimante
 - . un perforateur de cartes
 - . une machine à écrire

- 2) dans le cas du central récepteur :
 - . un lecteur de cartes
 - . un clavier de machine à écrire

Le nombre des accès est limité.

L'idée directrice de nos travaux était de rendre quasiment illimité le nombre des accès possibles d'un ordinateur à un autre sans préoccupation des notions de Central et Terminal. Les accès peuvent être des accès à des périphériques ou à des processus. Dans la suite, nous considérons qu'un périphérique muni de ses routines de gestion, est lui-même un processus.

Mais une méthode d'accès est tributaire de la procédure de transmission existante. En effet, si un ordinateur est toujours le maître, l'autre l'esclave, il est évident que le terminal ne pourra jamais initialiser un transfert.

C'est à cause de cette contrainte que nous fixons d'abord certaines hypothèses.

Hypothèse n° 1 : la procédure de transmission est compatible avec la procédure TMM - RB (procédure réalisée cf. chapitres 1 et 2).

Hypothèse n° 2 : nous nous restreignons au cas d'une ligne bipoint.
La ligne de transmission entre les deux sites est unique.

Hypothèse n° 3 : (se déduit de l'hypothèse n° 1)
Un site est le central, l'autre est le terminal.
Le central est toujours maître, le terminal toujours esclave.

Hypothèse n° 4 : les transmissions se déroulent en half-duplex ou bidirectionnel à l'alternat.

Sous ces hypothèses, en supposant que l'accès d'un processus sur le central, à une quelconque ressource du terminal, soit possible, alors, avec une procédure de transmission élargie où chacun des sites peut être alternativement maître et esclave, nous aurons résolu le problème en implémentant le système sur chacun des sites.

Dans un dernier paragraphe, nous donnons un essai d'extension et de généralisation en utilisant une procédure de transmission du type de celle décrite dans le chapitre III.

III - FONCTIONS D'UNE METHODE D'ACCES

III.1 - Idée directrice

Les ressources accessibles par SGT sont des périphériques séquentiels :

Exemple : quand un travail sur 10070 demande à distance une lecture de carte, le terminal lance la lecture de la première carte présente dans le lecteur et émet l'information vers le central.

Pour cela, la demande de lecture ne contient que l'adresse du périphérique lecteur de cartes ; c'est le caractère C2 de la séquence de supervision qui précise cette adresse périphérique (cf. chapitre I, § III). Aucun autre paramètre n'est nécessaire pour réaliser correctement l'entrée-sortie à distance.

Par contre, si on désire accéder à un fichier disque, d'autres informations sont nécessaires en plus de l'adresse du périphérique. Ceci n'est pas possible avec SGT.

Un moyen d'obtenir au choix l'un et l'autre de ces accès est le suivant :

Tout échange d'informations se fera à partir de ou vers une zone tampon en mémoire centrale. Nous décomposons ici l'accès à distance à un périphérique en deux accès successifs :

- un premier consiste à accéder à une zone tampon en mémoire centrale de l'autre ordinateur ;
- le second est propre au site et consiste en l'un des multiples accès pouvant exister sur le site même.

Une méthode d'accès doit permettre ces deux accès. Ainsi tout message émis par le central sera reçu par le terminal dans une zone précisé par le processus récepteur et l'utilisation de l'infor-

mation sera fonction des processus émetteur et récepteur.

Exemple : à la réception d'un message destiné à être imprimé (caractère C2 = code imprimante) ce n'est pas la procédure de transmission qui devrait lancer l'impression, mais le message devrait être reçu par un processus qui serait ici un processus d'impression du message.

Dans l'état actuel des liaisons possibles avec le CII 10070, cinq processus peuvent être désignés (cf. § II). En particulier, la fonction test de liaison est un processus particulier qui est dans le réseau Cyclades dénommé ECHØ.

Pour augmenter ce nombre de processus et pour donner plus de souplesse à l'utilisation des périphériques du terminal, nous désignerons ceux-ci par une adresse symbolique que nous appellerons étiquette opérationnelle, plutôt que par leur adresse réelle. A chaque étiquette opérationnelle, sera associé un processus de service (impression, lecture de cartes) chargé essentiellement de gérer un périphérique. Par assignation, une de ces étiquettes opérationnelles pourra désigner un processus quelconque d'utilisateur. Dans ce cas, les informations émises ou reçues avec l'adresse réelle associée à cette étiquette opérationnelle doivent être prises ou rangées dans une zone tampon de l'utilisateur en mémoire centrale. Ceci revient à dire qu'une étiquette opérationnelle désigne la mémoire centrale du terminal. La zone précise de la mémoire sera donnée en paramètres des ordres d'entrée-sortie sur la ligne.

III.2 - Assignation

Cette fonction établit la correspondance entre adresse symbolique et adresse réelle. L'association est réalisée par l'ordre :

ASSIGNER {étiquette opérationnelle , code périphérique}

même quand nous recevons une séquence de supervision différente de celle attendue à la suite d'une ouverture.

Dès qu'une demande d'ouverture a eu lieu, aucune autre demande ne peut avoir lieu.

Un autre paramètre de cet ordre est l'indication du mode de restitution du contrôle dans le cas d'anomalie ou d'erreur.

$$\text{OUVRIR} , \left\{ \begin{array}{l} \text{EMISSION} \\ \text{RECEPTION} \end{array} \right\} , \left\{ \begin{array}{l} (\text{ERREUR, BRANCH, ETIQ}) \\ (\text{ERREUR, INTERRUPT, NØ}) \end{array} \right\} \\ \left\{ \begin{array}{l} (\text{ANØRM, BRANCH, ETI2}) \\ (\text{ANØRM, INTERRUPT, NØ}) \end{array} \right\} , \text{ETAT}$$

Mode de restitution du contrôle en cas d'erreur ou d'anomalie

Par ce paramètre, l'utilisateur a le choix de la restitution du contrôle lorsque l'opération n'est pas correctement terminée.

Nous considérons que l'opération est correctement terminée lorsque le message d'acquiescement positif correspondant au message envoyé, ou reçu, a été reçu ou émis respectivement.

Si une suspension intervient ou si après N répétitions demandées par le récepteur, aucun message d'acquiescement positif n'est parvenu à l'émetteur, alors l'opération est considérée comme incorrectement terminée.

N est un paramètre de la procédure de transmission ajustable par l'opérateur pour des facilités de mise au point et des mesures sur le système (cf. chapitre VI). Il aurait pu être paramètre des instructions d'émission ou de réception avec une limite supérieure.

Le contrôle peut être restitué de deux façons différentes :
adresse de branchement ou : interruption.

1) adresse de branchement

L'utilisateur précise une adresse symbolique de retour. Dans une séquence d'instructions, il peut alors analyser le code d'erreur qui est transmis par le quatrième paramètre et agir en conséquence.

2) interruption

L'utilisateur précise un niveau d'interruption sur lequel il a prévu un sous programme de traitement de l'anomalie toujours par l'intermédiaire du quatrième paramètre.

Ce mode est à utiliser avec beaucoup de précautions. En effet, tous les niveaux présents sur le site ne peuvent être utilisés. Le contrôle de la validité du niveau doit être réalisé par la méthode d'accès.

III.4 - Emission et Réception

Ces deux fonctions sont symétriques : à une émission sur un site correspond une réception sur l'autre et réciproquement.

Ces fonctions ont pour rôle d'émettre (respectivement recevoir) un bloc d'informations à partir de (respectivement dans,) une zone tampon de mémoire centrale allouée au processus utilisateur de la ligne.

Les ordres sont "LIRE" et "ECRIRE"

Les paramètres de ces deux fonctions sont identiques à un près.

Paramètres

- 1 - adresse en mémoire centrale de la zone tampon
- 2 - taille de la zone
- 3 - attente de fin d'entrée-sortie ou non
- 4 - mode de restitution du contrôle au processus utilisateur
- 5 - type du prochain ordre
- 6 - dernier bloc (dans le cas d'émission seulement)

III.4.1 - adresse en mémoire centrale de la zone tampon

Ce paramètre permet de faire connaître à la procédure de transmission l'adresse de début de la zone contenant les informations à transmettre. La zone doit contenir l'ensemble des informations d'au moins un bloc. Les préfixe et suffixe du bloc sont placés par le système. L'utilisateur n'a pas à s'en préoccuper.

III.4.2 - taille de la zone

Ce paramètre indique combien il y a d'octets à transférer. Dans le système SGT chaque article d'un message correspond à une ligne d'imprimante ou à une carte perforée. Les messages contiennent un nombre entier d'enregistrements physiques.

Pour compatibilité, la taille d'un article ne devra pas dépasser 80 caractères en émission et 133 caractères en réception.

Ces tailles maximales sont une contrainte pour l'utilisateur.

Pour une plus grande souplesse d'utilisation, nous devrions rendre ces contraintes transparentes. A un ordre d'entrée ou de sortie correspond un ensemble logique d'informations. Cet ensemble peut

constituer un ou plusieurs messages physiques ; le découpage étant réalisé à l'émission, le réassemblage l'étant à la réception. Pour cela, il serait nécessaire de préciser le nombre total de messages physiques qui composent le message logique. Ceci est une application de la commutation de paquets au cas particulier de la liaison entre deux ordinateurs seulement.

III.4.3 - attente de fin d'entrée-sortie

L'utilisateur a la liberté de choisir l'un des deux modes : attente ou non.

Dans le premier cas, le contrôle ne lui sera rendu que lorsque l'entrée-sortie sera terminée, correctement ou non. Dans le second cas, le contrôle lui est rendu dès que l'entrée-sortie est lancée. Ce contrôle lui sera repris par le jeu des interruptions en provenance du coupleur ou de l'unité d'échange de façon à assurer correctement les échanges d'informations.

III.4.4 - mode de restitution de contrôle

Normalement le contrôle est restitué en séquence. Par contre, si l'utilisateur le désire, il peut lui être rendu sur interruption programmée, quand l'entrée-sortie est terminée. Cette option a été prise pour similitude et compatibilité avec d'autres opérations d'entrée-sortie propres à l'I.B.M. 1800.

III.4.5 - type du prochain ordre

Ce paramètre indique si le prochain ordre de transfert sera une émission ou une réception. Il est nécessaire dans la majeure partie des cas en fonction de l'hypothèse n° 3. Il permet d'attendre la séquence de supervision suivante et d'y répondre. Dans le cas où le prochain ordre est une réception, il permet au terminal d'accepter la réception de messages avant même que l'ordre de lecture soit exécuté.

III.4.6 - Fin d'émission

Ce paramètre permet de préciser s'il s'agit du dernier bloc à émettre afin que la procédure émette le caractère ETX au lieu de ETB comme caractère de début de suffixe.

III.5 - Test

Cette fonction permet à l'utilisateur de tester :

- 1) si la liaison est effectivement ouverte ou non
- 2) si l'échange a eu lieu et de quelle manière il s'est terminé.

III.5.1 - Test d'ouverture de la liaison

Tant que la liaison n'est pas effectivement ouverte, aucun message, autre que les messages de contrôle ne peut transiter sur la ligne.

Aucun ordre d'entrée-sortie ne peut être exécuté avant l'ouverture effective.

Ce test est l'équivalent du test "périphérique libre" avant le lancement d'une entrée-sortie sur un périphérique classique.

Cet ordre s'écrit :

TEST , OUVERTURE , ETAT

Si l'ouverture est effective, ETAT prend la valeur 2 sinon la valeur 1.

III.5.2 - Test sur échange terminé

Cette option permet à l'utilisateur de tester la fin d'un échange qu'il a demandé. Elle est importante particulièrement dans le cas où l'échange est sans attente.

L'instruction s'écrit :

TEST , FIN , ETAT

Si l'échange est terminé, ETAT prend la valeur 2 sinon la valeur 1.

Nous n'avons pas besoin d'autres paramètres tels que adresse de ligne car elle est supposée unique, ou ordre d'échange concerné par le test car la transmission a lieu en half - duplex.

Si la transmission avait lieu en full - duplex, il faudrait ajouter un paramètre "adresse de ligne" car nous considérons une liaison full duplex comme deux liaisons half duplex.

III.6 - Fermeture

La fermeture consiste essentiellement à désallouer la ligne à l'utilisateur. Elle ne peut avoir lieu qu'à la fin de la dernière session. L'utilisateur doit le prévoir.

En fait, il existe deux types de fermeture : fermeture temporaire et fermeture définitive. Seule cette dernière est explicitement ordonnée par l'ordre FERMER.

La fermeture temporaire est implicitement donnée par le paramètre "Fin d'émission" de l'ordre d'écriture ou par la réception du caractère ETX au lieu de ETB dans un ordre de lecture. Ce caractère ETX marque la fin d'une session. La ligne reste allouée mais toute nouvelle session devra être réinitialisée par une séquence de supervision.

C'est pourquoi, nous plaçons dans les ordres d'entrée sortie le paramètre "type du prochain ordre".

Quand le type de cet ordre n'est pas connu à l'écriture du programme (par exemple à la suite d'une instruction conditionnelle, ou peut avoir à exécuter tantôt une entrée, tantôt une sortie) il

est conseillé à l'utilisateur de l'indiquer dès qu'il en a connaissance par un ordre REOUVRIER, RECEPTION .

Nous verrons dans le paragraphe "synchronisation" que cet ordre est important. Il permet en effet de recevoir des informations avant même que l'ordre de lecture n'ait été exécuté.

III.7 - Communication avec l'opérateur et entre opérateurs

Cette fonction présente une particularité par rapport aux autres : elle n'est pas accessible à l'utilisateur. Toutefois elle est importante d'une part pour le dialogue entre l'opérateur et la méthode d'accès locaux, d'autre part entre les deux opérateurs.

Nous distinguons donc ces deux types de communication.

1) Communication locale

Pour la mise au point du système, certains paramètres peuvent être ajustés par l'intermédiaire du clavier de l'opérateur.

De plus, le lancement de certains processus se fera également par l'intermédiaire du clavier, par exemple : lancement de l'entrée de travaux à distance.

D'autre part, la méthode d'accès édite certains messages permettant de tenir l'opérateur au courant des divers événements susceptibles de se produire.

2) Communication entre opérateurs

Cette communication est importante pour que les deux opérateurs soient avertis des divers événements intéressant les deux parties, en particulier dans le cas d'une utilisation en Remote batch, ou dans le cas d'utilisation de fichiers à distance (demande de montage de bande magnétique par exemple).

Cette communication se fera par insertion des messages opérateur - opérateur, dans le flux des messages utilisateurs. A cause de l'hypothèse n° 3, nous attendrons la séquence de supervision correspondant au sens de l'échange. Cette insertion est entièrement transparente à l'utilisateur.

Les messages opérateurs - terminal, ou opérateur - opérateur sont donnés en annexe 2.

IV - SYNCHRONISATION

A un ordre d'écriture (respectivement, de lecture) sur le central correspond un ordre de lecture (respectivement, d'écriture) sur le terminal.

En fonction de l'hypothèse n° 3, les ordres d'entrée-sortie sur le terminal ne seront exécutés que lorsque les ordres symétriques sur le central, le seront.

Pour chaque sens d'échange, quatre cas peuvent se produire :

- a) le terminal exécute un ordre de lecture sur la ligne avant que le central exécute l'ordre d'écriture correspondant.
- b) le terminal exécute un ordre de lecture sur la ligne après que le central ait exécuté l'ordre d'écriture.
- c) le terminal exécute une écriture sur la ligne avant que le central exécute la lecture.
- d) le terminal exécute une écriture sur la ligne après que le central exécute la lecture.

Nous prenons comme hypothèse que la liaison est ouverte sur chaque site. L'ouverture sur chacun des sites pourra être indépendante ou liée. Par indépendante, nous entendons une communication entre opérateurs, nécessaire pour activer de part et d'autre les processus utilisateurs de la ligne. Par liée, nous entendons une activation d'un processus à distance par l'autre processus.

Ce problème sera étudié plus loin (§ V - chapitre IV).

- IV.1 - Le terminal exécute un ordre de lecture sur la ligne avant que le central exécute l'ordre d'écriture correspondant.

Le processus actif sur le terminal sera mis en attente s'il l'a demandé. Sinon par l'instruction TEST il pourra attendre effectivement que l'opération soit terminée.

L'opération sera initialisée par une séquence de supervision de sélection. Après une réponse affirmative, le terminal recevra les blocs utiles et les restituera au processus récepteur.

- IV.2 - Le terminal exécute un ordre de lecture sur la ligne après que le central exécute l'ordre d'écriture correspondant.

L'ouverture étant demandée, le terminal est en attente d'une séquence de supervision de sélection. Cette séquence est reçue avant que l'instruction de lecture soit exécutée.

La réponse sera affirmative et les messages seront reçus, stockés et restitués au processus récepteur lors de l'exécution de ses ordres de lecture.

Une autre solution consisterait à ne pas accepter la réception demandée par la séquence de supervision. Ceci aurait pour effet de mettre le processus émetteur sur le central, en attente, et d'abandonner la transmission si au bout d'un quantum de temps défini dans la procédure, aucune réponse affirmative en provenance du terminal n'est reçue par le central. Cette solution a par contre, l'avantage d'une réalisation plus simple.

En effet, la première solution consiste à construire un symbiont d'entrée sur la ligne qui remplirait un fichier disque à partir des messages reçus, placés en file d'attente.

Si lors d'une session, il y a une grande masse d'informations transférées dans le sens central - terminal, le terminal doit pouvoir emmagasiner le tout. A la limite, ceci est impossible. Aussi, lorsque la file d'attente est pleine, le terminal le signale au central en envoyant une séquence de suspension.

Alors la transmission ne reprendra que lorsqu'il y aura de la place en file d'attente. Tant que cette condition ne sera pas réalisée, toute séquence de supervision en sélection sera refusée.

- IV.3 - Le terminal exécute une écriture sur la ligne avant que le central exécute la lecture.

Ce cas est présent lorsque, à la suite d'une lecture, le central émet une séquence de polling. Si le terminal n'a encore rien à émettre, c'est-à-dire que l'ordre d'écriture n'a pas encore été exécuté, la réponse est négative. La réponse négative a pour effet de placer le central en attente et éventuellement d'abandonner la liaison pour la même raison qu'en IV.2. C'est la seule solution.

- IV.4 - Le terminal exécute une écriture sur la ligne après que le central exécute la lecture.

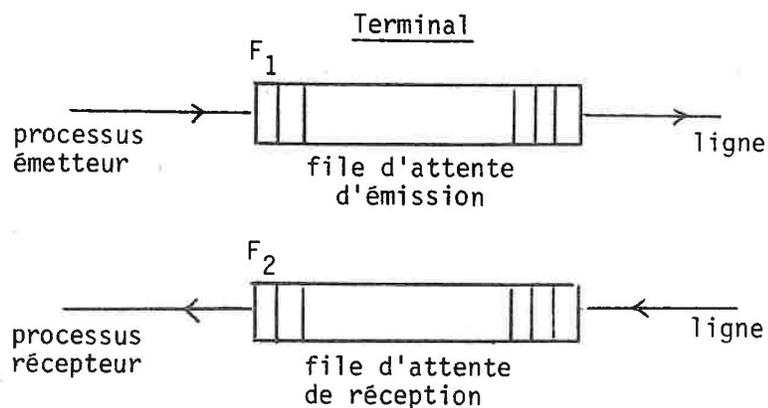
L'émission du terminal vers le central ne peut avoir lieu tant que la séquence de supervision de polling n'a pas été reçue ; par contre cette attente peut être utilisée pour préparer les messages à émettre. Une solution est donc de construire une file d'attente des messages à émettre, au rythme du processus émetteur. Cette file d'attente sera vidée sur demande du processus récepteur (central) après réception d'une séquence de supervision.

- IV.5 - Les quatre cas envisagés représentent ce qui se passe à l'initialisation de la liaison.

De plus, à cause des vitesses de fonctionnement différentes, il peut se produire un déphasage entre les deux processus, au cours de la session. Un déphasage sera détecté par une absence de messages pendant un quantum de temps (time - out). Les procédés

de reprise dans ces cas, sont prévus dans la procédure de transmission (cf. chapitre I § 4-2-3).

En conclusion, la synchronisation entre processus émetteur et processus récepteur est réalisée par des files d'attente dans les cas où le terminal est en avance en émission et en retard en réception. Dans les autres cas, il y a attente du processus récepteur.



A cause des initialisations, toujours faites par le central, nous sommes obligés d'asservir le terminal dans tous les cas, sauf en IV.3.

Pour une symétrie parfaite entre les deux ordinateurs, il serait plus simple de placer F₁ comme file F₂ sur le central et autoriser le terminal à émettre dès qu'il rencontre un ordre de sortie.

V - GENERALISATION

Dans ce paragraphe, nous prenons les hypothèses suivantes :

- Hypothèse V.1 - les deux ordinateurs sont reliés par une seule ligne de transmission ;*
- Hypothèse V.2 - Sur chaque site, la procédure de transmission est du type de celle présentée dans le chapitre précédent (ch. III).*
- Hypothèse V.3 - Chaque site peut être extrémité d'autres lignes de transmission (déjà vrai pour le 10070).*

Ce qui suit est valable pour l'un quelconque des deux sites.

V.1 - Assignment

L'assignment consiste d'abord à demander la ressource constituée par la ligne de transmission, à cause de l'hypothèse V.3.

Comme nous voulons réaliser un accès à un processus particulier, à distance, allouer une ligne à un utilisateur est insuffisant. Nous devons considérer le processus distant comme étant lui-même une ressource nécessaire du processus demandeur. Pour cela, l'utilisateur P_1 du site O_1 doit pouvoir désigner le processus P_i du site O_2 qui doit lui être alloué.

Nous généralisons la notion "d'adresse du périphérique" par la notion d'identificateur de processus afin de permettre une désignation plus souple et plus large des ressources accessibles à distance.

Donc l'utilisateur demandera non seulement la ligne en la désignant par une étiquette opérationnelle propre à chaque site, mais

aussi un processus P_i . La demande et l'attribution de cette ressource P_i donneront lieu à un échange de messages. Ces messages sont générés et reçus par un protocole que nous nommons Protocole d'attribution des ressources. Il fait partie de la méthode d'accès car il initialisera tous les transferts programmés dans les processus connectés. Nous trouvons un protocole du même type dans le projet SØC (10), (18) et dans le réseau ARPA (5), (13).

A la rencontre d'une commande du type :

ASSIGNER , étiquette opérationnelle , identificateur
de la ligne du processus P_i

sur 01*, le protocole d'attribution de ressources (P.A.R.) devra dès que possible (la ligne peut être déjà allouée) faire la demande de connexion avec P_i par l'intermédiaire du PAR de 02*. Pour cela, la ligne sera toujours allouée, soit aux deux protocoles PAR, soit à deux processus P_j et P_i . Les protocoles PAR décident eux-mêmes de se désallouer la ligne pour l'allouer aux processus qui l'ont demandée. Quand ces processus restitueront la ligne, automatiquement elle sera réallouée aux P.A.R.

Exemple de message de demande de ressources :

" CONNECTER identificateur , identificateur "
de P_j de P_i

On suppose que d'un site, on sait désigner un processus de l'autre site.

Messages d'accord ou de désaccord suite à la demande :

" NON CONNECTE identificateur , identificateur "
de P_j de P_i

" OUI CONNECTE id P_j , id P_i "

* 01, 02 : ordinateur 1 , ordinateur 2

Chaque site doit avoir émis et reçu ce message pour que la liaison soit ouverte. Ce message est émis à l'exécution de l'ordre

"OUVRIR , $\left\{ \begin{array}{c} \text{EMISSION} \\ \text{ou} \\ \text{RECEPTION} \end{array} \right\}$, ident. P_i , sur le site de P_j

et "OUVRIR , $\left\{ \begin{array}{c} \text{RECEPTION} \\ \text{ou} \\ \text{EMISSION} \end{array} \right\}$, ident. P_j , sur le site de P_i

L'option émission ou réception n'est utile qu'après l'ouverture effective de la liaison. Elle n'est là que pour faciliter la synchronisation des deux processus P_i et P_j dans le cas où le premier à émettre le fait avant que l'autre n'ait lancé l'ordre de lecture.

Remarque

Il se peut que plusieurs processus P_j^1 ($j = 1, 2 \dots$) sur le site 1 demandent à utiliser la ligne pour établir une liaison avec des processus P_i^2 ($i = 1, 2 \dots$).

Or, nous n'allouons la ligne qu'à un seul processus à la fois. Nous proposons la solution suivante à ce problème.

Le protocole PAR tient à jour une table des couples de processus (sur chaque site).

Site 1	Site 2	indicateurs
P_{j1}^1	P_{i1}^2	
P_{j2}^1	P_{i2}^2	

Le couple est rangé dans la table à la commande ASSIGNER et les indicateurs refléteront les divers états possibles de la liaison.

- 1°) Réponse affirmative reçue à l'assignation
- 2°) Ouverture émise
- 3°) Ouverture reçue

La réponse affirmative suite à la demande d'assignation permet de décider si on peut lancer le processus.

A un instant donné, plusieurs couples de processus peuvent avoir reçu une réponse affirmative.

Au moment du lancement de P_{j1}^1 par exemple, P_{i1}^2 devra être lancé sur le site 2.

Un conflit peut se produire : à savoir que deux processus P_{j1}^1 sur site 1 et P_{i2}^2 sur site 2 peuvent être lancés. A ce moment, il y a deux émissions de demande d'ouverture non correspondantes. Le conflit est analogue au conflit de demande de statut maître (cf. chapitre III) et sera réglé de la même façon.

V.3 - Synchronisation

La synchronisation est ici plus simple à réaliser. Nous n'avons en fait que deux cas à envisager : le processus récepteur est en avance ou en retard sur le processus émetteur, en vertu de l'hypothèse n° V.1.

S'il est en avance (i.e., rien n'a encore été envoyé) il est placé en attente.

S'il est en retard, une file d'attente sert de tampon et les messages seront trouvés dans la file d'attente.

Donc à l'exécution de l'ordre "LIRE", deux cas sont à envisager :

- les informations sont dans la file auquel cas, la méthode d'accès les délivre au processus utilisateur en fonction des paramètres de l'ordre LIRE ;
- les informations ne sont pas dans la file et le processus récepteur sera mis en attente de fin d'entrée.

VI - APPLICATION A LA GESTION DU "REMOTE BATCH"

L'entrée de travaux à distance étant une des premières fonctions à avoir été implémentée sur les réseaux de terminaux ou d'ordinateurs, beaucoup d'articles traitent de ce problème. Nous en indiquons un qui analyse et résume toutes ces fonctions : "Remote job entry" de H.P. Daendliker (6).

Nous donnons ici les programmes (émission et réception) de gestion de l'entrée de travaux à distance tels qu'ils pourraient être écrits dans le langage de la méthode d'accès.

Emission

```
Début          ouvrir , émission
                i:=1
lect : lecture d'une carte ;
                préparer l'article correspondant (formatage) ;

test : si "bloc i non rempli"
        début ranger l'article dans le bloc i ;
        si dernière carte alors
                début écrire le bloc i avec fin émission ;
                fermer ;
                aller à finem
                fin
        sinon aller à lect ;
        fin
        sinon début          écrire le bloc i , émission ;
                            i:= i + 1
                            aller à test
        fin ;

finem : fin
```

Réception

```
Début      ouverture , réception ;  
            i:=1 ;  
            lect : lire bloc i avec attente ;  
            si quelque chose à imprimer alors imprimer  
                                     sinon perforer ;  
  
            si "non fin" alors aller à lect ;
```

Fin

VII - REALISATION

Comme il a été dit dans le paragraphe "synchronisation", nous supposons ici que la méthode d'accès est munie de files d'attente.

Elle comprend donc deux symbionts, un en entrée, l'autre en sortie, associés à la ligne. Ce qui explique que dans l'écriture du "remote batch", nous ne nous soucions pas de savoir, en lançant une écriture, si la ligne est ouverte ou non.

Ceci est intéressant car ainsi nous optimisons l'utilisation du terminal : en effet, si nous ne disposons pas de symbionts en sortie sur la ligne, le lecteur de cartes serait une ressource bloquée pour le remote batch même dans l'attente d'une séquence de supervision.

Pour des facilités de réalisation sur IBM 1800, les ordres définis dans ce chapitre s'écriront sous forme d'appels de sous-programmes. A chaque ordre sera associé un sous-programme : ouverture, lecture, écriture, tests et fermeture, analogues aux procédures méta symbol sur 10070 M : OPEN , M : READ , M : WRITE , M : CHECK et M : CLOSE.

Les sous-programmes de la méthode d'accès sont liés au processus utilisateur par leurs paramètres et par une table de contrôle analogue au DCB du type siris 7 sur CII 10070.

Dans cette table, sur IBM 1800, on trouvera les renseignements suivants :

retour en cas d'anomalie
retour en cas d'erreur
ouverture demandée en émission ou réception
adresse de retour quand l'attente est terminée
attente ou non

Cette méthode d'accès n'est pas encore implémentée. Par contre, l'entrée de travaux est possible à partir de l'IBM 1800. Pour cela, nous avons ajouté des routines à la procédure de transmission. Ces routines constituent le *travail de fond* décrit dans le chapitre II : Réalisation.

CHAPITRE V

PROTOCOLE D'EXECUTION DE
SOUS PROGRAMMES A DISTANCE

I - INTRODUCTION

Disposant d'un protocole d'attribution de ressource sur chaque site, un processus A du site 1 peut communiquer avec un processus B du site 2.

Exemple :

Un travail ne peut être exécuté sur un seul site. Nous le décomposons en deux processus A et B, chaque processus pouvant être exécuté sur son site propre. Nous supposons que chaque processus constitue ici un travail pour le système d'exploitation de son site. Nous supposons de même que les tâches compilation et édition de liens sont terminées. Donc chaque travail se présente sous la forme d'un module de chargement exécutable.

Au moment de la programmation, il aura fallu prévoir les ordres d'entrée-sortie permettant une communication entre eux :

- assignation
- ouverture
- lecture écriture
- fermeture

Pour certaines utilisations, il peut être intéressant d'éviter à l'utilisateur la programmation de ses échanges. en particulier si on sait reconnaître dans le programme certains ordres qui nécessitent un échange, en particulier :

- l'exécution de certains sous-programmes
- l'accès à un fichier distant

Comme C. Chabrol, D. Comte et O. Gelly (4), nous nous intéressons ici à un protocole d'exécution de sous-programmes à distance.

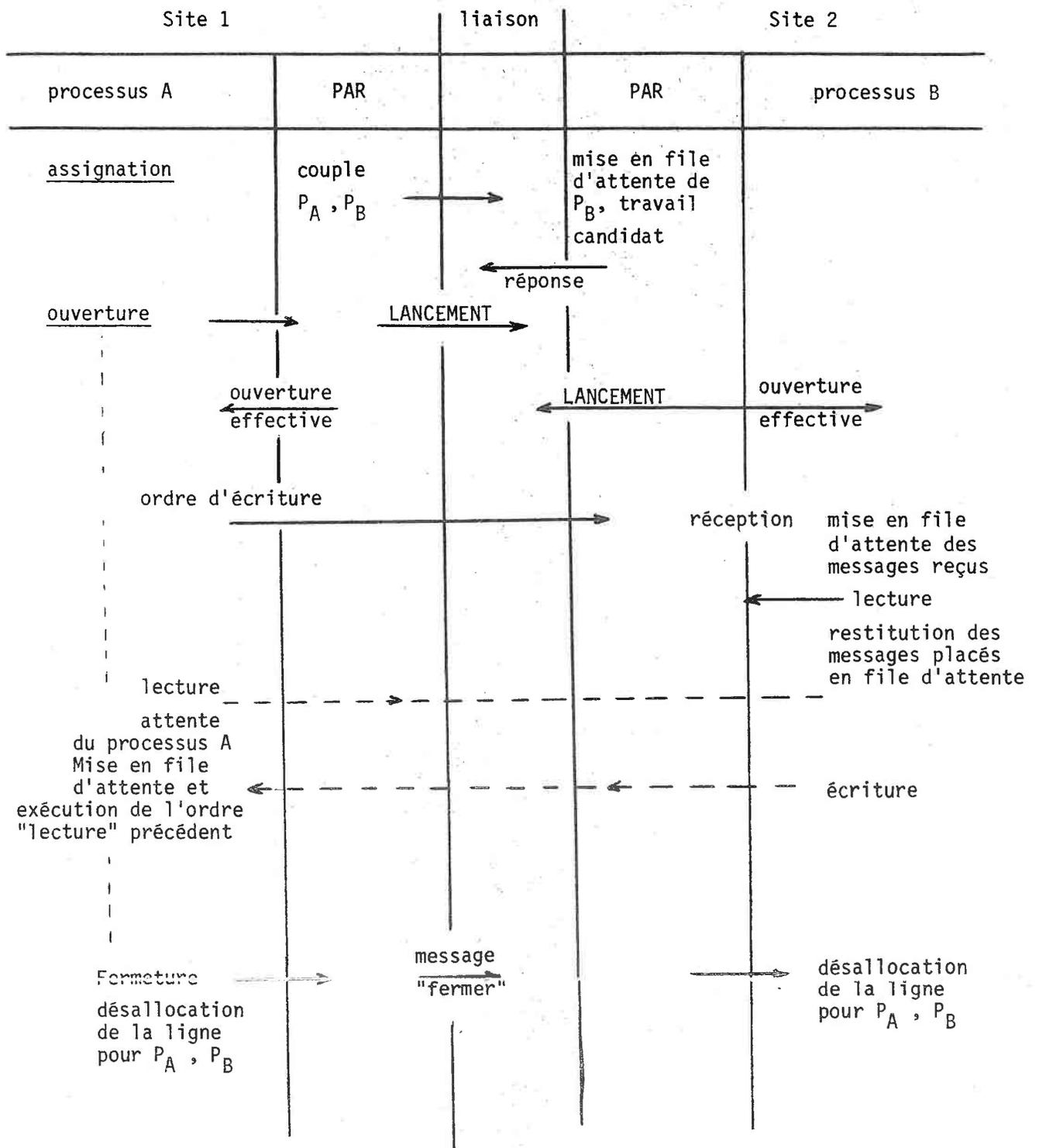
Ce protocole s'inspire de la notion de partage de ressources (12) et (20) dans un réseau, en particulier la coopération de deux ou plusieurs ordinateurs hétérogènes, complémentaires (17).

Exemple :

Dans le processus P_A , on fait appel à un sous-programme SP devant être exécuté sur le site 2. Au moment de l'exécution, l'appel de SP doit appartenir à un processus P_B du site 2. Ce processus P_B doit être actif au moment de l'appel de SP sur le site 1.

A l'exécution sur le site 1, ceci doit se réaliser par une ouverture d'une liaison (P_A , P_B) au moment du lancement de P_A , puis par une écriture sur la ligne vers le site 2 au moment de l'appel de SP, et une lecture sur la ligne en provenance du site 1 pour la restitution des résultats.

Sur le site 2, l'ouverture de la liaison (P_A , P_B) sera acceptée au lancement de P_B , une lecture sera lancée pour acquérir les paramètres effectifs et à l'ordre RETURN, une écriture renverra les paramètres au processus P_A .



II - TABLE DES VALEURS ET ADAPTATION AUX FORMATS

Au moment de l'exécution, les valeurs à transférer doivent être regroupées dans une zone mémoire de façon à ce qu'elles soient consécutives. Cette zone est appelée table des valeurs. De plus, les mêmes objets n'ont pas la même représentation sur les deux sites (10070 et 1800). Donc une adaptation aux formats du site où se déroulera l'exécution est nécessaire. Pour ce nous transmettons à la fois le type de l'objet et sa valeur.

II.1 - Préfixe

A chaque objet transmis est associé un préfixe de quatre bits qui indique son type :

b_3 b_2 b_1 b_0	Type
0 0 0 0	Entier simple précision (EN)
0 0 0 1	Entier double précision (DE)
0 0 1 0	Réel simple précision (RE)
0 0 1 1	Réel double précision (DR)

b_3 positionné à 1 indique un tableau dont les éléments sont du type signalé par b_0 et b_1 .

Nous n'avons pas défini d'autres types car les langages utilisés sur I.B.M. 1800 n'en considèrent pas d'autres.

On pourrait utiliser b_2 pour décrire d'autres objets, tels que complexes simple et double précision, booléens, chaînes de caractères.

II.2 - Descripteur

Pour la transmission des préfixes et des valeurs, nous regroupons plusieurs préfixes entre eux sous forme de descripteur.

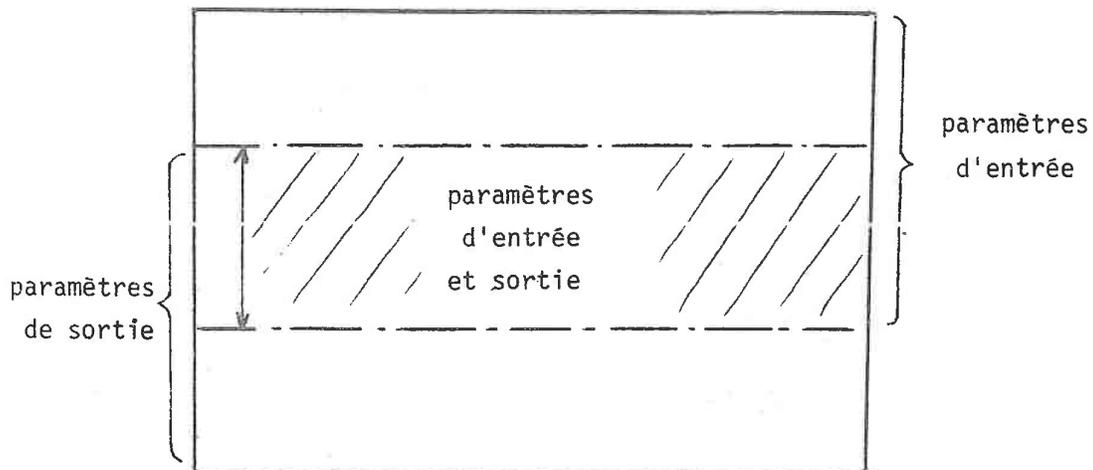
Un descripteur est un ensemble d'au plus 8 préfixes. Il peut en comporter moins, aussi nous avons besoin d'utiliser certaines combinaisons de b_3 , b_2 , b_1 , b_0 pour indiquer la fin d'un descripteur.

II.3 - Table des valeurs

Pour optimiser la place occupée en mémoire centrale, nous ne mettons dans la table des valeurs que les valeurs d'objets simples. Le cas des tableaux sera envisagé plus loin.

D'autre part, certains paramètres peuvent être à la fois paramètres d'entrée et sortie du sous-programme. Ceux-ci ne figureront qu'une fois dans la table des valeurs.

Table des valeurs T.V.



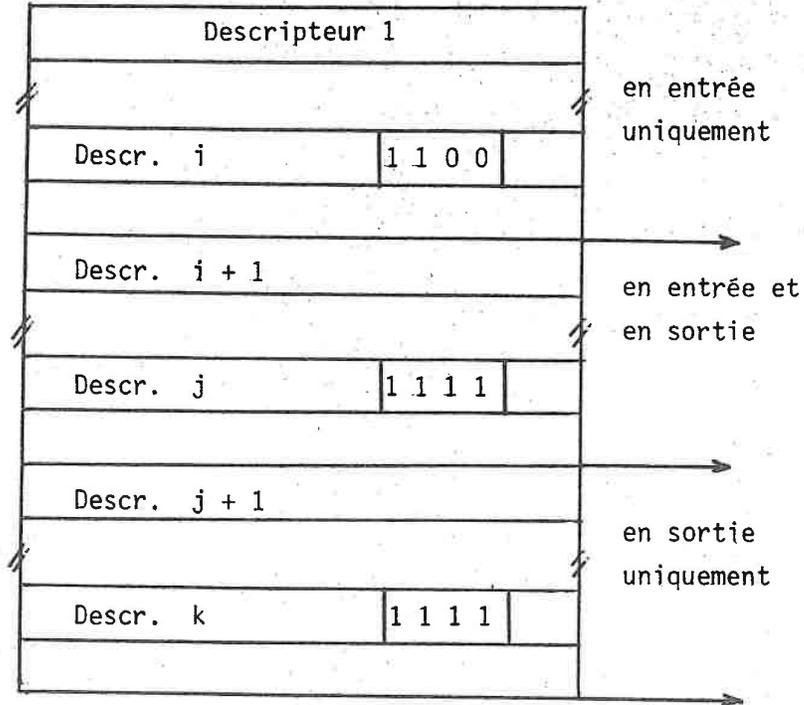
Avec une telle structure, nous sommes amenés à distinguer plusieurs préfixes pour marquer les fins de descripteur :

- un descripteur peut ne pas être complet. Un préfixe marquera la fin du descripteur (1100) ;
- un descripteur marquant la fin de tous les paramètres d'entrée (y compris éventuellement certains paramètres de sortie) ou la fin de tous les paramètres de sortie (1111).

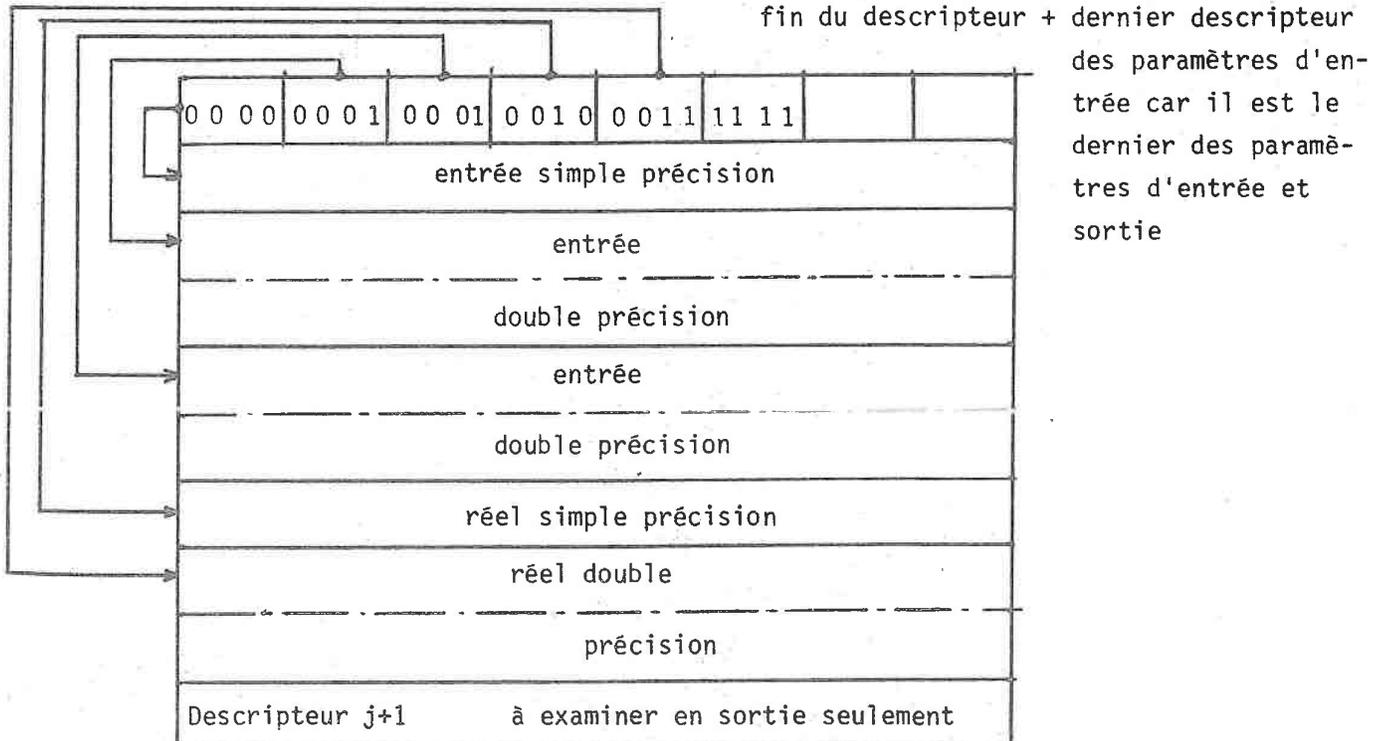
R e m a r q u e

En général, les objets en double précision rangés sur deux mots, le sont à partir d'une adresse paire. Nous pouvons optimiser l'ordre des valeurs dans la table des valeurs pour respecter cette contrainte.

Table des valeurs



Détail d'une zone
zone décrite par le descripteur j



II.4 - Adaptation aux formats

Connaissant le type d'un objet, son format sur un site, il est facile de l'adapter au format de l'autre site. Toutefois, l'utilisateur devra veiller aux possibilités d'adaptation.

Ex e m p l e :

Une entrée simple précision sur 1800 est représentée sur 16 bits ; on peut l'étendre sur 32 bits pour compatibilité avec le C.I.I. 10070. Par contre, la réciproque n'est pas toujours vraie.

Le problème se pose de savoir qui fera l'adaptation. Nous avons deux solutions :

- 1°) l'adaptation est centralisée sur un ordinateur. Elle est faite avant l'émission et après la réception.
- 2°) l'adaptation est décentralisée :
 - 2 a) - elle est réalisée par l'émetteur
 - 2 b) - elle est réalisée par le récepteur

La première solution présente l'avantage de ne réaliser qu'une routine de transcription. Par contre, elle n'est plus viable lorsque plusieurs ordinateurs sont connectés entre eux.

Nous préférons la solution 2 b) car elle reste dans le cadre général de la procédure de transmission et de la méthode d'accès où le processus récepteur est asservi au processus émetteur.

II.5 - Descripteurs dans le cas des tableaux

Il serait inconcevable de constituer une table spéciale des valeurs des éléments des tableaux paramètres. La table est le tableau lui-même.

D'autre part, tous les éléments d'un tableau étant de même type, il est inutile de prévoir un préfixe pour chaque élément.

Un descripteur seul décrira le tableau. Les quatre premiers bits de ce descripteur préciseront qu'il s'agit d'un tableau d'un certain type (ex : 1000 : tableau d'entier simple précision). Les quatre bits suivants indiqueront le nombre de dimensions. Puis autant de groupes de 16 bits qu'il y a de dimensions, indiqueront les bornes supérieures des indices associés aux dimensions.

Nous ne considérons ici que le cas des tableaux manipulés en Fortran dans lesquels la borne inférieure est toujours égale à 1.

Exemple d'un tableau à une dimension dont les éléments sont des entrées simple précision

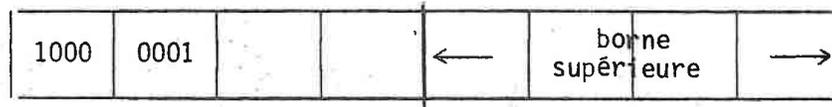
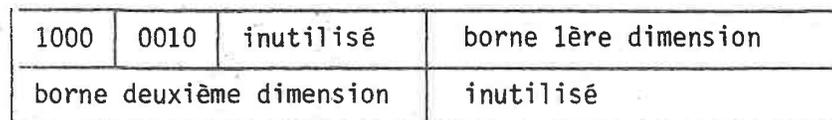


Tableau à deux dimensions



Nous supposons que sur les deux sites, les tableaux sont rangés de la même façon de manière à ce que le calcul de l'adresse effective d'un élément à partir de ses indices donne le même résultat sur chacun des deux sites.

III - Déclaration EXTERIEUR

De façon à ce que l'utilisateur n'ait pas à modifier son programme source lorsqu'il désire faire exécuter un sous-programme sur un autre site, nous regroupons tous les renseignements nécessaires à la bonne communication, en une seule commande

EXTERIEUR , nom du SP , paramètres

Dans la table des valeurs, les paramètres sont regroupés suivant les critères d'entrée et / ou sortie. Ces critères ne sont pas en général pris en compte pour l'écriture d'un sous-programme. Il est donc nécessaire d'établir une correspondance entre la table des valeurs et les paramètres effectifs d'appel du sous-programme.

III.1 - Paramètres de la déclaration EXTERIEUR

Les paramètres de cette déclaration sont tout d'abord le nom d'un sous-programme puis certains renseignements. Ce sont ces derniers renseignements que nous nous proposons d'examiner ici.

Nous précisons la liste des paramètres du sous-programme. Ces paramètres sont formels.

Pour chacun d'eux, nous indiquons son type et s'il est en entrée ou en sortie.

III.2 - Syntaxe

$$\text{PARAM} \left(\left(X , AA , \left[\begin{array}{c} \{E\} \\ \{S\} \end{array} \right] \right) , \left(YY , AA , \left[\begin{array}{c} \{E\} \\ \{S\} \end{array} \right] \right) \dots \right)$$

AA est une suite de deux caractères. Ils constituent un code mnémorique pour désigner le type du paramètre formel X ou YY, etc...

E précise que le paramètre est un paramètre d'entrée, S qu'il s'agit d'un paramètre de sortie.

Si ni E, ni S ne figure à la suite d'un nom, alors ce paramètre est à la fois en entrée et en sortie.

Les tableaux seront déclarés également dans la déclaration extérieure sous la forme suivante :

$$\left(\text{TAB } (n_1, n_2, n_3, \dots) \text{ (AA, } \left\{ \begin{array}{l} E \\ S \end{array} \right\} \text{) } \right)$$

n_1, n_2, n_3, \dots sont les bornes supérieures des indices du tableau de nom formel TAB, de type AA.

Les paramètres de la déclaration extérieure doivent apparaître dans l'ordre des paramètres déclarés à la définition du sous-programme.

Exemple

Un sous-programme de nom SSP est défini par :

```
SUBROUTINE    SSP (X , Y , Z, N)          X et N en entrée
INTEGER N                                           Y et Z en sortie
REAL X , Y
DIMENSION    Z (10 , 10)
```

La déclaration "extérieur" s'écrirait :

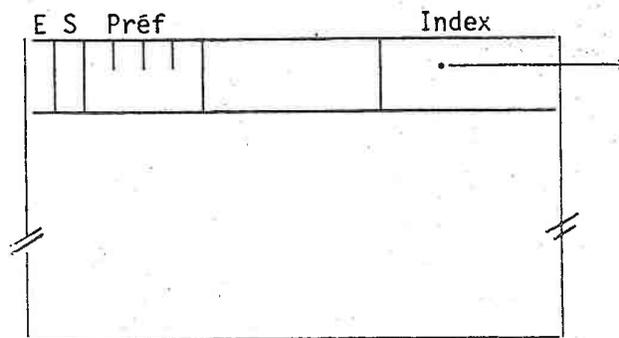
```
EXTERIEUR    SSP , PARAM { (X, RE, E) , (Y , RE , S) ,
                          (Z (10 , 10) , RE , S) , (N , EN , E) }
```

III.3 - Compilation de la déclaration EXTERIEUR

A la rencontre de cette déclaration, nous préparons la table des valeurs nécessaire à l'exécution, en parallèle avec une table de correspondance permettant de faire le rapport entre l'ordre des paramètres effectifs dans l'instruction d'appel et l'ordre du rangement des valeurs dans la table des valeurs.

Table de correspondance

Cette table contient une entrée par paramètre formel déclaré dans EXTERIEUR et pour chaque entrée, nous y trouvons les renseignements suivants :



E = 1 si le paramètre est d'entrée
S = 1 si le paramètre est de sortie

Préf : ici est le préfixe correspondant au type du paramètre.

Index : est la valeur d'un pointeur désignant l'adresse, relative au début de la table des valeurs, de l'emplacement réservée à la valeur effective de ce paramètre dans la T.V.

L'index n'est utilisé que dans le cas d'objets simples.

III.4 - Exemple

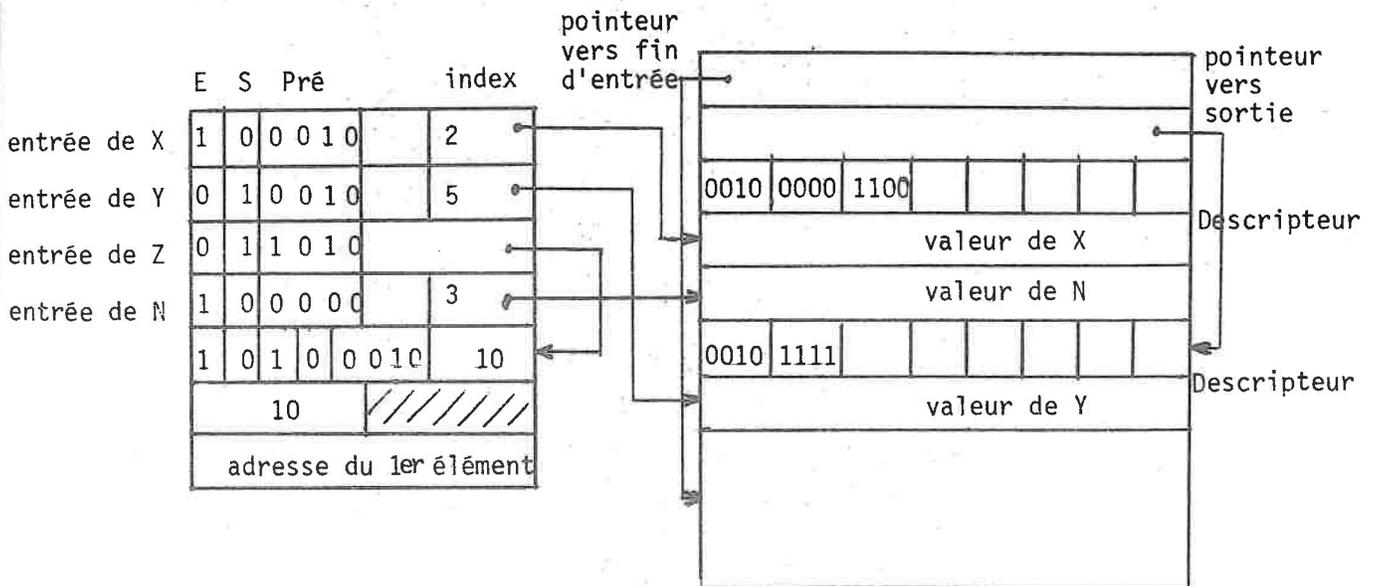
Nous reprenons l'exemple précédent.

Sa table des valeurs et la table de correspondance sont comme suit :

EXTERIEUR SSP , PARAM $\left\{ (X , RE , E) , (Y , RE , S) , \right.$
 $\left. (Z (10 , 10) , RE , S) , (N , EN , E) \right\}$

Table de correspondance

Table des valeurs



Dans le cas de tableaux, le descripteur est rangé lui-même dans la table de correspondance. A l'entrée correspondante au tableau paramètre, nous plaçons un pointeur vers le descripteur de façon à garder la similitude : 1 entrée = 1 mot = 1 paramètre. Les descripteurs de tableaux sont rangés en fin de table de correspondance.

IV - TÂCHES SUR LE SITE DEMANDEUR

IV.1 - Compilation sur site demandeur

A la rencontre de la déclaration "EXTERIEUR", nous devons construire les deux tables (correspondance et valeurs), la seconde devant être remplie à l'exécution.

Puis il y a génération de l'ordre d'assignation, de l'ordre d'ouverture de la liaison ;

A la rencontre de l'appel du sous-programme déclaré en EXTERIEUR, il y a génération d'un appel de sous-programme chargé de la préparation des transferts ; Ce sous-programme remplira la table des valeurs en particulier.

Puis génération des ordres d'écriture pour la table des valeurs, puis pour les tableaux notés "entrée" ;

Puis génération des ordres de lecture avec attente pour remplissage de la partie de la table des valeurs notée en entrée, et génération des ordres de lecture des tableaux notés en entrée.

IV.2 - Edition sur site demandeur

L'édition de liens se passe normalement entre les divers modules objets du travail utilisateur et les modules objets correspondant à la gestion de la liaison, en particulier les divers modules de la méthode d'accès.

IV.3 - Génération de l'ordre d'assignation et de l'ordre d'ouverture

Ces ordres doivent préciser le nom du processus P_3 . Or ce processus n'existe pas encore. Il doit donc être dénommé et son nom doit être connu des deux sites. Une solution simple consiste à fixer

et à réserver sur chaque site un format fixe pour les noms de tels processus du type

"APPEL _____"
 nom nom
 du sous- du
 programme site

Ce format doit être adapté aux différents systèmes d'exploitation de façon à fixer une taille en caractères qui soit la plus petite des tailles maximales autorisées sur l'ensemble des deux sites.

Une autre solution consisterait à laisser au site 2 la liberté de créer un nom moyennant certains critères à préciser, ce nom devant être transmis en retour au site 1.

Quand on sait désigner les deux processus, il n'y a plus de problème pour générer les deux ordres assignation et ouverture.

V - LIAISON ENTRE SITE DEMANDEUR ET SITE FOURNISSEUR

Tout d'abord sur ordre du site demandeur, il doit y avoir création du travail relatif au processus P_B .

Ce travail doit être généré, édité sur le site fournisseur avant d'y être exécuté.

Le travail est décrit sur le site 1 et envoyé sur le site 2. Nous définissons un processeur appelé GENERATEUR sur le site 2. C'est ce processeur qui générera le travail. Il reconnaît un langage qui est uniquement constitué de la déclaration "EXTERIEUR".

Le travail est donc constitué de :

```
! JØB
! GENERATEUR , nom du processus  $P_B$ 
EXTERIEUR .....
! LØAD .....
```

Le texte source est écrit sur le site 1, envoyé sur le site 2 pour y être traité. Il est écrit dans le langage de commande du site 2.

Cette phase nécessite l'utilisation du processus d'envoi de travaux à distance. Cette demande peut être automatique ou non.

A la rencontre de EXTERIEUR sur le site 1, par l'intermédiaire des protocoles P.A.R. nous demandons l'activation du processus "Remote batch" de façon à ranger sur le site 2 le texte du travail dans la file d'attente des travaux.

La demande systématique de construction d'un processus P_B sur le site 2 pose un problème.

Doit-on toujours construire P_B chaque fois qu'on a besoin de l'exécuter ? Pour certaines applications destinées à être souvent mises en oeuvre, ceci serait une importante contrainte. Il est donc

souhaitable d'autoriser le catalogage permanent de tels processus sous forme d'un module de chargement. Mais alors, il apparaît nécessaire d'exercer un contrôle sur les noms des processus créés. En effet, si le processus P_B est déjà créé sous le même nom, il ne doit pas l'être une deuxième fois. Mais rien ne prouve qu'ils sont identiques parce qu'ils ont le même nom. L'utilisateur demandeur doit être averti d'une telle situation. Il est seul à pouvoir régler le problème. Pour cela le site 1 (demandeur) sera prévenu et un message devra être communiqué à l'utilisateur.

La demande de construction ne peut donc pas être systématiquement émise du site demandeur vers le site fournisseur.

Un paramètre supplémentaire de la déclaration EXTERIEUR doit préciser si le processus P_B est à générer ou non.

Une autre solution plus simple, mais moins automatique, est de dissocier la génération du processus P_B , de la compilation du processus P_A , c'est-à-dire que l'utilisateur écrit lui-même le travail destiné à être compilé, édité et exécuté sur le site 2.

Par exemple :

```
! JØB
! GENERATEUR , nom du processus
EXTERIEUR  SSP .....
! LØAD avec catalogage du module de chargement
```

Ce travail est envoyé sur le site 2 par l'intermédiaire du processus "entrée de travaux à distance".

Au besoin, si le texte du sous-programme n'est pas connu du site fournisseur, il peut être envoyé dans le même travail :

```
! JØB
! FORTRAN
      ⋮  texte du sous-programme
! GENERATEUR , nom du processus
EXTERIEUR , .....
! LØAD avec catalogage
```

On a décrit ici le travail dans le langage d'exploitation du site fournisseur supposé être le C.I.I. 10070. Une meilleure solution serait de définir un langage de commande commun aux deux sites de façon à éviter à tout utilisateur d'avoir à connaître les deux langages.

Générateur

Ce compilateur construit sur le site 2 les ordres symétriques de ceux qui sont construits sur le site 1 lors de la compilation de la déclaration "EXTERIEUR" et de l'appel du sous-programme.

La seule différence est que l'appel du sous-programme est généré systématiquement après le ou les ordres de lecture (s'ils existent).

Le module objet résultant doit être compatible avec les autres modules objets générés sur ce site pour l'édition de liens.

Le passage des paramètres objets simples se fait par l'adresse de leur zone en table des valeurs.

Demande automatique de génération sur le site fournisseur

Processus A
site 1

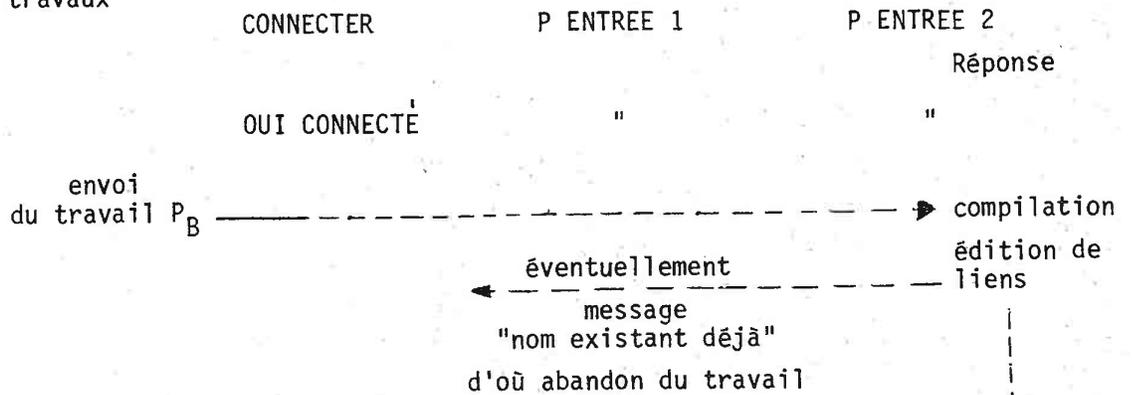
PAR
site 1

PAR
site 2

entrée d'un travail
contenant EXTERIEUR

compilation et
construction du travail P_B
contenant ! GENERATEUR
destiné à être exécuté
sur le site 2.

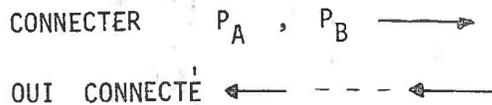
Demande de connexion
des deux processus
d'entrée de travaux
à distance



les deux travaux P_A et P_B sont à l'état de modules
de chargement exécutables.

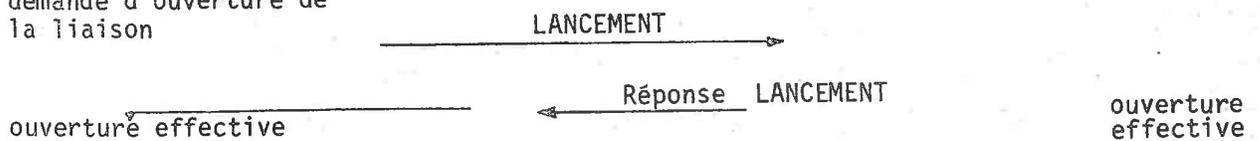
avant lancement de P_A

demande connexion P_A et P_B



lancement de P_A

demande d'ouverture de
la liaison



VI - REMARQUE

V.I.1 - Cas de plusieurs appels du même sous-programme

Il est bien évident qu'on ne générera pas plusieurs fois le même processus P_B sur le site fournisseur. Le problème se pose de savoir si ce processus P_B restera actif, en attente de l'exécution suivante ou s'il devra être activé par P_A autant de fois qu'il est nécessaire.

Première solution

La liaison est désallouée après chaque exécution. La liaison est donc fermée après chaque retour des résultats (éventuellement) ou après chaque entrée (éventuellement aussi).

A chaque nouvel appel du sous-programme sur le site 1, une nouvelle ouverture de la liaison doit avoir lieu d'où une attente de disponibilité de la ligne qui se répercute sur les temps d'attente de P_A .

Deuxième solution

La ligne reste allouée au couple (P_A , P_B) tant que P_A n'a pas demandé de fermeture, et chaque fois que P_B a terminé une exécution il redevient actif à son début. Il perdra son état actif quand le P.A.R. aura reçu la demande de fermeture de la liaison.

La première solution optimise l'utilisation de la ligne. Par contre, la seconde optimise le service rendu à l'utilisateur en diminuant les temps de réponse d'où les temps d'attente de ses deux processus.

La première solution est plus rationnelle. Dans tous les

cas, l'attente d'un processus ne doit pas être active. L'unité centrale doit être libérée pour d'autres travaux. Le contrôle n'est redonné au processus qui était en attente que lorsque l'entrée-sortie est terminée.

VI.2 - Cas d'appels de plusieurs sous-programmes

Pour chaque sous-programme, nous construisons un processus P_{B_i} particulier et il y aura autant de liaisons entre couples (P_A, P_{B_i}) qu'il y a de sous-programmes SP_i à exécuter sur le site 2.

Si cette possibilité est autorisée, il est évident que la deuxième solution du paragraphe précédent n'est plus viable.

■ Nous devrions désallouer la ligne au couple (P_A, P_{B_i}) après chaque exécution de P_{B_i} (quitte à allouer un peu plus tard la ligne au même couple) pour permettre en particulier une éventuelle liaison $(P_A, P_{B_{i'}})$ avec $i \neq i'$.

VII - CONCLUSION

Dans ce chapitre, nous avons montré comment un certain processus sur un site pouvait demander la génération d'un autre processus sur un site distant et comment, lors de l'exécution, ils pouvaient coopérer.

Nous n'avons pas détaillé les problèmes d'insertion dans les systèmes d'exploitations.

L'ensemble procédure de transmission et méthode d'accès doit être un sous-système du système d'exploitation analogue à un système de gestion de fichiers.

Le protocole d'attribution de ressources, est un programme qui doit être résidant en mémoire centrale car c'est par lui que passent toutes les demandes de connexion entre processus distants. Il est le noyau des protocoles que l'on pourrait définir.

En ce qui concerne le protocole d'exécution de sous-programmes à distance, il resterait à le réaliser en cherchant à augmenter ses possibilités (sous-programmes eux-mêmes paramètres).

Une application intéressante de ce protocole serait la commande optimale en temps réel de grands systèmes industriels par les deux ordinateurs C.I.I. 10070 et I.B.M. 1800.

Une commande optimale exige deux types de travaux exécutés par des ordinateurs :

- 1°) *la commande proprement dite* : un calculateur tel que l'I.B.M. 1800 est apte à la réaliser. Il gère les sécurités, fait des acquisitions de données, calcule les commandes à l'aide d'un algorithme pré-élaboré. Pour que ces commandes soient aisément et rapidement calculables, seuls les algorithmes linéaires sont en général envisagés.

Cette simplicité de la loi de commande exige des mises à jour régulières pour trois raisons essentielles :

- . lorsque la production est modifiée
- . à cause de la variation de paramètres
- . à cause de l'évolution du processus

2°) *le calcul, très important en place mémoire et en temps, des algorithmes de commande* : il doit être réalisé sur un calculateur puissant (ex. 10070). En général un calculateur de commande de procédés (1800) ne dispose pas de la puissance nécessaire. A partir des mesures effectuées par ce dernier, le C.I.I. 10070 pourrait réajuster le modèle mathématique de l'installation industrielle. Quand ce modèle a été élaboré, il calcule alors la nouvelle loi de commande à transmettre au calculateur de procédés. Il s'agit donc de réactualiser le modèle par une identification en temps réel.

Bien que la ligne téléphonique ne permette pas des débits binaires très importants (2.400 b/s), le temps de réponse est suffisant pour certaines catégories de procédés industriels. Pour pouvoir appliquer un tel système, il suffit que le temps d'élaboration d'une nouvelle commande, à compter de l'instant d'acquisition, y compris les temps de transfert et de calcul, soit tel que la nouvelle loi de commande ne soit pas vieillie avant d'avoir pu être mise en oeuvre. Ceci dépend des procédés à commander.

CHAPITRE VI

TESTS ET MESURES

I - INTRODUCTION

Dans ce chapitre, nous décrivons d'une part les tests déjà réalisés, d'autre part nous signalons les tests et mesures qui restent à effectuer.

Pour les premiers essais de notre réalisation, coupleur et programmes de la procédure de transmission, il était bien sûr hors de question de disposer de la ligne.

D'une part, ayant défini les diverses fonctions du coupleur, nous avons écrit quelques petits programmes qui permettent de le mettre au point, de tester et de vérifier son bon fonctionnement par affichage sur des voyants et par impression sur la machine à écrire.

D'autre part, les essais de nos programmes se sont déroulés en deux étapes fort naturelles, le polling et la sélection, que nous décrivons dans cet ordre.

De plus, il nous faut prévoir un plan de tests et de mesures à effectuer lors du raccordement effectif au central.

II - TESTS DU COUPLEUR

Les fonctions du coupleur actuel ont été testées d'une part par hardware, d'autre part par software :

- par hardware

Sur simulateur logique, les fonctions ont été testées l'une après l'autre, puis ensemble (GADEL, (9)).

D'autre part, un ensemble logique appelé "séquenceur" capable de commander le coupleur indépendamment de l'ordinateur, a été mis au point.

Cet ensemble génère des messages et visualise par des voyants lumineux l'état des divers modules du coupleur. Il a été conçu pour une aide à la maintenance de l'appareil.

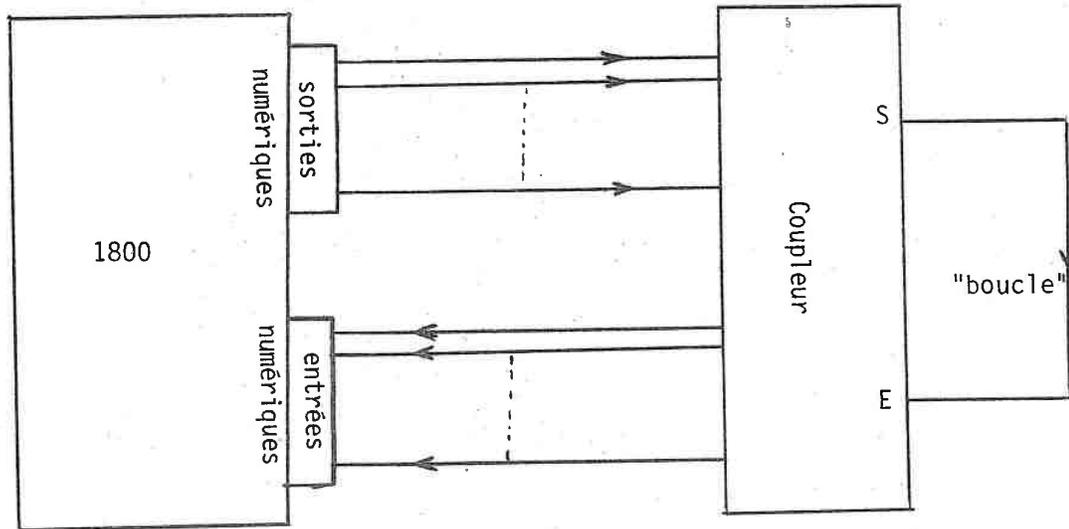
- par software

Pour chaque fonction du coupleur, nous avons rédigé un programme de test. Ces programmes consistent essentiellement à émettre un message particulier et à le recevoir en bouclant la sortie série du coupleur sur son entrée série. Quand le modem nous sera livré, la première série de tests consistera à exécuter ces mêmes programmes sur l'ensemble coupleur - modem. Ils permettront en particulier de vérifier si les informations émises par le coupleur vers le modem sont correctes au point de vue physique (amplitude des signaux, niveaux de tension, etc...).

III - TESTS DES PROGRAMMES

Tous les modules et sous-programmes ont d'abord été mis au point chacun à part. Puis les essais ont été faits par petits groupes : préparation des blocs d'information, analyse des séquences... Les transitions entre les modules du schéma de réalisation ont été vérifiées et intégrées pas à pas pour le polling, puis pour la sélection.

Ne disposant pas de ligne pour réaliser nos tests, une idée fort simple est de se servir simultanément de deux fonctions principales du coupleur en bouclant l'émission sur la réception.



D'une façon ou d'une autre, l'ordinateur 1800 doit, en plus de sa situation de terminal, jouer le rôle de central car les réactions du central sont indispensables pour l'essai et la mise au point de notre procédure. Renonçant à écrire la procédure complète du central, nous en avons programmé une petite partie (préparation des messages en phase de sélection notamment). Les autres réactions ont été "simulées" de façon manuelle aux clés de l'ordinateur (séquences de supervision et d'acquiescement en phase de polling). Seuls, les blocs d'information transitent par la boucle ; les autres séquences sont prises en compte directement par nous et par la procédure.

III.1 - Polling

Les interruptions SYN et PAD du coupleur ne sont dans ce cas, utilisées que pour visualiser l'information transmise : elles ne déclenchent pas l'exécution de *EXAM*. Par contre les séquences lues aux clés de l'ordinateur entraînent l'exécution et la poursuite normale de la procédure.

Nous avons donc besoin d'un sous-programme de lecture de données aux clés *LEC* et de deux sous-programmes d'interruption *P13* (*SYN*) et *P12* (*PAD*) spéciaux.

- . *LEC* déclenché manuellement (bouton poussoir au pupitre), lit une configuration quelconque de 16 bits (2 caractères) affichée aux clés de données du pupitre de commande. Il place la configuration lue dans un mot d'une zone (adresse du mot dans un *IØCC*).

A la rencontre du caractère *PAD* dans l'une ou l'autre des 2 positions (configuration *xxFF* ou *FFxx* en hexadécimal), il lance, par interruption programmée l'exécution du module directeur *EXAM*.

- . *P13* déclenché par l'interruption *SYN*, remplit *BREC* et *WØRIN* (*BREC*:= adresse de *T3* ; *WØRIN*:= *WBEMI*) et lance l'entrée d'informations en appelant *DINPT*.
- . *P12* déclenché par l'interruption *PAD*, se contente d'initialiser l'écriture du contenu de la zone *T3* sur la machine à écrire. Cette visualisation nous permet de répondre par un accusé de réception ou demande de suspension. Notons qu'en procédant ainsi nous vérifions une fois de plus le bon fonctionnement du coupleur.

Nous avons ainsi essayé et mis au point l'enchaînement des modules de la phase de polling ainsi que la préparation des blocs.

III.2 - Sélection

La "simulation" de la préparation des blocs d'information du central a été réalisée en se servant au maximum des sous-programmes de préparation de blocs déjà écrits pour le polling.

Le sous-programme *PRBL1* est obtenu à partir de *PRBL* en effectuant quelques modifications mineures : caractère *PØL* → *SEL*,

numéro du bloc, format d'identification du périphérique dans l'en-tête d'un article ...

Le sous-programme d'interruption *SEL*, déclenché manuellement (niveau 11), appelle *PRBL1* puis lance l'émission par *DEXIT*.

L'interruption *SYN* (*P13S*) initialise la lecture de ce bloc d'information, l'interruption *PAD* (*P12S*) démarre son examen par la procédure en tant que bloc d'informations reçu du central. Les réponses du terminal sont écrites sur la machine à écrire ; à nous de déclencher une nouvelle préparation si *NØB - ACK* est écrit...

Ceci nous permet de mettre au point l'enchaînement des modules de la phase de sélection et l'analyse des séquences.

Remarque : Pour voir la progression de la procédure, nous avons, au cours des essais, visualiser un certain nombre de variables communes, en particulier ET.

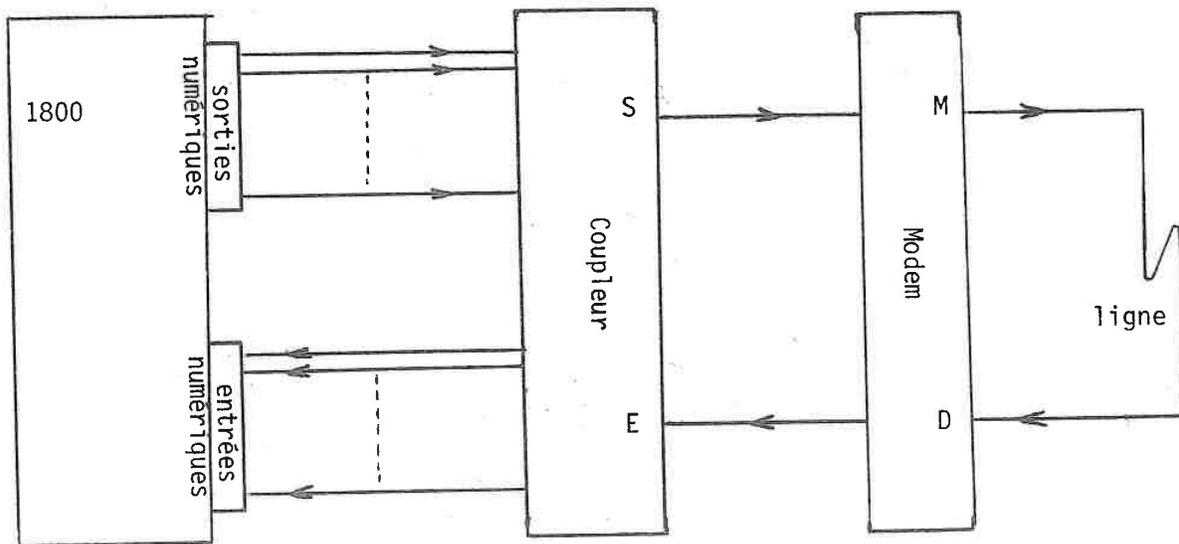
IV - TESTS ET MESURES ENVISAGEES

Les essais que nous avons effectué jusque là sur le coupleur et sur nos divers programmes, nous ont montré que la mise au point d'un tel ensemble est délicate et nécessite l'écriture de nombreux programmes supplémentaires. Ces essais en local, sans modem, du coupleur, du polling et de la sélection se sont terminés sur des résultats satisfaisants et encourageants par la suite.

Il nous faut maintenant prévoir et planifier autant que possible les essais et mesures à venir.

IV.1 - Tests

A la réception du modem, la jonction coupleur-modem sera testée en local par hardware puis par quelques petits programmes simples d'émission et de réception d'un message spécifique.



Quant aux essais réels des programmes, ils ne pourront être effectués qu'à la livraison de la ligne. Le déroulement de ces tests aura la même allure que les tests locaux : essai et mise au point progressive de toute la procédure de transmission du terminal.

Ces tests porteront essentiellement sur trois points : formatage des blocs d'information, analyse des séquences et l'enchaînement des divers modules décrits au chapitre II.

Nos programmes ont été écrits en s'appuyant sur des documents de base C.I.I. mais depuis leur publication, des modifications mineures

sont peut-être intervenues dans la réalisation de la procédure de transmission.

Enfin nous avons signalé en leur temps certaines bizarreries dans les formats des séquences de TMM - RB, telle que l'identification du périphérique dans l'en-tête du bloc et dans l'en-tête de chaque article.

Ces remarques font qu'il nous est difficile de prévoir un plan de tests dans ses moindres détails. Toutefois, pour pouvoir assurer la mise au point de nos programmes, il nous faudra stocker sur chacun des deux sites, toutes les séquences émises et reçues ainsi que la progression de la procédure par l'intermédiaire de certaines variables de contrôle, ET en particulier.

Ces tests nécessitent l'écriture de quelques programmes sur les deux sites. Ceux du 1800 seront mis au point à l'avance ; ceux du 10070 pourront l'être avec un autre terminal déjà opérationnel.

IV.2 - Les mesures

La mesure des qualités d'une transmission concerne la détection des erreurs.

Pour cela, on mesure :

- 1) l'efficacité e d'une transmission comme étant le rapport du nombre des messages reconnus faux (N_{ef}) sur le nombre total de messages faux (N_f)

$$e = \frac{N_{rf}}{N_f}$$

- 2) le taux d'erreurs brut τ est égal au nombre de messages faux (N_f) sur le nombre de messages reçus (N_t)

$$\tau = \frac{N_f}{N_t}$$

3) le taux d'erreur global q est le nombre de messages faux non reconnus sur le nombre de messages reçus :

$$q = \frac{N_f - N_{rf}}{N_t}$$

On sait uniquement mesurer N_{rf} et N_t automatiquement lors de la transmission. Pour mesurer N_f ou $N_f - N_{rf}$, il faut disposer à la fois des messages émis et des messages reçus. Ce n'est que par comparaison entre un message émis et un message reçu que l'on peut se rendre compte si le message reçu reconnu juste, est en réalité faux.

Pour réaliser cette comparaison de façon automatique, nous devrions procéder comme suit pour des échantillons de messages :

- Emettre des messages du site 1 vers le site 2
- Les stocker par exemple en fichier disque sur le site 1
- Sur le site 2 stockage des messages reconnus justes
- Emission des messages reconnus justes du site 2 vers le site 1
- Comparaison sur le site 1 des messages reconnus justes 2 fois (sur site 2 et site 1) avec les véritables messages émis.

Cette méthode laisse échapper les messages reconnus justes une première fois sur le site 2 et faux en réalité, dont les erreurs seraient corrigées par un heureux hasard lors de la retransmission vers le site 1. Ce cas étant au premier abord très improbable, il est raisonnable de le négliger.

Ces mesures permettront d'apprécier la qualité de la transmission uniquement.

Il serait également intéressant de mesurer les performances du système. Une mesure simple à réaliser en ce qui concerne l'ensemble du système, consiste à compter le temps d'Unité Centrale effectivement consacré à la gestion de la transmission.

C'est pour cela que nous avons prévu d'enregistrer les heures de début de réception d'un message, de début de traitement de ce message et de fin de traitement, celle-ci correspond en fait à l'heure d'émission du message d'acquiescement.

Ces mesures sont toutefois sommaires, mais facilement réalisables lorsque le système fonctionnera et permettront de classer le système par rapport à d'autres systèmes.

De plus, les résultats de mesure de performance permettront de détecter quelles seraient les fonctions actuellement réalisées par software qui pourraient l'être par hardware de façon à diminuer considérablement les temps de traitement des informations échangées (émises ou reçues).

Les diverses informations mesurées sont stockées au fur et à mesure des échanges dans une table en mémoire centrale, puis sur disque, pour exploitation ultérieure.

Dans une table, on trouve une première partie qui contient des paramètres de la liaison (ajustables par l'opérateur). Ce sont d'ailleurs des variables communes pour la procédure :

- nombre maximum d'échéances de time-out
- valeur du time-out
- nombre maximum de NAK émis pour un message.

Les deux premiers paramètres devraient être présents autant de fois qu'il y a de time-out, dans le cas de la procédure définie dans le chapitre III.

Dans une deuxième partie, on trouve les événements qui surviennent dans la liaison :

En émission

- . nombre de messages différents émis
- . nombre effectif de messages émis
- . nombre de NAK reçus
- . nombre de messages acquittés au premier envoi

N fois {
 heure d'émission d'un message
 heure de réception du message d'acquiescement correspondant

Cette zone est stockée périodiquement dans un fichier disque.

- . nombre de caractères SYN effectivement éliminés par message d'acquiescement.

En réception :

- . nombre de messages différents reçus
- . nombre effectif de messages reçus
- . nombre de NAK envoyés
- . nombre de messages acquittés à la première réception

N fois {
 heure de réception d'un message
 heure de début de traitement
 heure d'émission du message d'acquiescement

- . nombre de caractères SYN effectivement éliminés par message
- . nombre de tests de liaison

Ces renseignements permettront une évaluation de la liaison, en coût et en performance. Il faudrait aussi stocker la taille des messages émis ou reçus de façon à trouver la taille optimale d'un message.

Table des mesures

	Nombre maximum d'échéances de time-out	
	Valeur du time-out	
	Nombre maximum de NAK	
	Nombre de messages différents émis	(ou reçus)
	Nombre effectif de messages émis	(ou reçus)
	Nombre de NAK reçus	(ou émis)
	Nombre de messages acquittés au 1er envoi	A
N fois	Heure de préparation d'un message	(d'analyse)
	Heure d'émission	(de réception)
	Heure de réception de l'acquittement correspondant	(d'émission)

En supposant que nous transmettons en half-duplex, la partie A de la table sera la même en émission ou en réception.

CONCLUSION

Dans ce travail, nous avons donc étudié les différents niveaux d'un système de transmission entre deux ordinateurs : depuis le niveau hardware jusqu'au niveau utilisateur pour lequel toutes les contraintes et moyens de transmission doivent être transparents.

En plus de la réalisation d'une procédure et de son utilisation pour l'entrée de travaux à distance, nous avons développé deux points principaux : la schématisation des procédures et le protocole d'exécution de sous-programmes à distance.

Ces deux points nous permettent d'envisager quelques extensions :

A partir de la définition d'une procédure comme automate fini, il serait souhaitable de concevoir une logique cablée ou microprogrammée qui accepte et génère le langage de la transmission.

Par rapport à la solution cablée, la microprogrammation présente l'avantage d'une plus grande souplesse. A condition de recenser les fonctions des diverses procédures de transmission déjà développées ou en cours de développement, il sera sans doute possible d'élaborer un processeur en acceptant plusieurs, donc utilisable sur divers matériels, en attendant une standardisation de ce moyen de communication.

En ce qui concerne la méthode d'accès et le protocole d'exécution de sous-programmes à distance, une suite logique serait leur implémentation.

Deux extensions peuvent être envisagées :

- . augmenter les possibilités de ce protocole en autorisant d'autres paramètres (par exemple, les noms de sous-programmes)
- . étendre ce protocole au cas d'un réseau de N ordinateurs ($N > 2$)

ANNEXES

I - ASCII

On désigne par P le bit d'imparité additionnel.

I.1 - séquences de supervision

a) Polling

{	C1 = PØL	P	1 0	X X X X X	(1)
	C2	P	1 0	A B C D E	(2)
	ENQ		1 0 0	0 0 1 0 1	

Le caractère C2 est, en notation hexadécimale :

4 0	pour le périphérique fictif
C 1	pour le lecteur de cartes
C 2	pour le clavier de la machine à écrire

- (1) XXXXX *spécifie l'adresse ligne du terminal*
 (2) ABCDE *identifie le périphérique appelé (voir remarque à la fin de l'annexe)*

b) Sélection

{	C1 = SEL	P	1 1	X X X X X	(1)
	C2	P	1 1	A B C D E	(2)
	ENQ		1 0 0	0 0 1 0 1	

En notation hexadécimale, le caractère C2 est :

E0 si périphérique fictif
61 si imprimante
62 si machine à écrire
E3 si perforateur de cartes

La réponse du terminal sera :

. si prêt à recevoir

{ SEL P 1 1 X X X X X (1)
ACK 1 0 0 0 0 1 1 0

. si non prêt à recevoir

{ SEL P 1 1 X X X X X (1)
NAK 0 0 0 1 0 1 0 1

c) Séquence d'alarme et séquence EØT

• { SEL P 1 1 X X X X X (1) • EØT 0 0 0 0 0 1 0 0 0
 • { BEL 0 0 0 0 0 1 1 1
 ENQ 1 0 0 0 0 1 0 1

I.2 - Blocs d'information

a) Format d'un article

- si une suppression des blancs non significatifs en fin d'article
a eu lieu :

{ C2 F F F A B C D E (1)
 C3 1 0 1 0 0 0 0 1
 C4 P 1 0 0 Z Z Z Z }
 C5 P 1 0 0 W W W W } (2)
 ⋮
 texte de l'article
 ⋮
 RS 1 0 0 1 1 1 1 0

- si on utilise le format standard, c'est-à-dire :

lecteur de cartes : 80 caractères
imprimante : 133 caractères
machine à écrire : 72 caractères

}	C2	FFF	ABCDE	(1)
	C3	001	0000	
	:			
	:			
	:	texte de l'article		
	:			
	RS	100	11110	

(1) FFF = { P10 en polling ABCDE identifie le périphérique appelé
 { P11 en sélection

(2) ZZZZ WWWW indique en binaire la longueur de l'article

Remarque

Le caractère C2 du format d'un article rend la présence de ce même caractère inutile dans les séquences de supervision et dans l'en-tête du bloc.

b) Format d'un bloc

}	SØH	0000	0001		
	C1	FFF	XXXX	(1)	
	NØB	TTT	0VVV	(2)	
	STX	0000	0010		
	:				
	:	suite d'articles			
	:				
	:	ETB	1001	0111	
	:	ou			
	:	ETX	1000	0011	
	BCC	YYYY	YYYY	(3)	

La longueur maximum d'un bloc est de 256 caractères de SØH inclus à ETB ou ETX inclus.

La séquence de fin ETB ou ETX tient lieu de caractère séparateur RS pour le dernier article du bloc

- (1) C1 = SEL ou PØL
- (2) TTT = $\begin{cases} \text{P10} & \text{en Polling} \\ \text{P11} & \text{en Sélection} \end{cases}$

VVV désigne en binaire le rang du bloc (modulo 8) dans le train de bloc transmis. Le premier bloc d'un train porte le numéro 1. Ce rang est réinitialisé à chaque séquence de supervision.

- (3) Le caractère de parité bloc BCC porte sur tous les caractères du bloc à partir du premier caractère qui suit SØH jusqu'au caractère ETB ou ETX compris.

I.3 - Les accusés de réception

<u>Positif</u>	<u>Négatif</u>
$\begin{cases} \text{NØB} & \text{F F F O O N N N} & (1) \\ \text{ACK} & \text{1 0 0 0 0 1 1 0} \end{cases}$	$\begin{cases} \text{NØB} & \text{F F F O O N N N} & (1) \\ \text{NAK} & \text{0 0 0 1 0 1 0 1} \end{cases}$

- (1) FFF = $\begin{cases} \text{P10} & \text{en Polling} \\ \text{P11} & \text{en Sélection} \end{cases}$

NNN désigne le rang (modulo 8) du dernier bloc correctement reçu.

I.4 - Les demandes de suspension

. pour la station émettrice, la séquence EOT tient lieu de demande de suspension.

. pour la station réceptrice des messages spéciaux sont prévus :

- demande de suspension positive	DLE	0 0 0 1	0 0 0 0
	ACK	1 0 0 0	0 1 1 0
- demande de suspension négative	DLE	0 0 0 1	0 0 0 0
	NAK	0 0 0 1	0 1 0 1

Cette dernière ne peut pas être utilisée par la station terminale.

I.5 - Caractères de synchronisation et de padding

. SYN 0 0 0 1 0 1 1 0

. PAD 1 1 1 1 1 1 1 1

I.6 - Remarque

Le caractère C2 est utilisé dans le format d'un article et désigne alors effectivement le périphérique appelé : Dans les séquences de supervision, la valeur C1/61 du caractère C2 doit être considérée comme adressage général de tous les périphériques du terminal pouvant être concerné, soit par le polling, soit par la sélection.

II - EBCDIC

Nous signalons ici les différences essentielles avec les formats en ASCII.

- 1) les caractères STX, ETX, ETB sont précédés du caractère DLE.
- 2) Le caractère BCC est remplacé par deux caractères BCS1 et BCS2 qui représentent respectivement les octets de poids fort et de poids faible constituant le reste de la division du bloc (du 1er caractère qui suit SØH jusqu'au caractère ETB ou ETX compris) par le polynôme

$$x^{16} + x^{15} + x^2 + 1$$

- 3) Tout caractère SYN introduit dans le bloc pour des raisons physiques de transmission doit être obligatoirement précédé de DLE. Un tel couple DLE - SYN doit être éliminé à la réception et ne participe pas au calcul de BCS.
- 4) Tout caractère présentant dans le corps du bloc la configuration d'un caractère de service SØH, STX, ETX, ETB, EØT devra être transmis tel quel.
- 5) Le caractère DLE de la séquence de fin ne participe pas au calcul du BCS, mais le caractère DLE qui précède BTX y participe.
- 6) La longueur maximum d'un bloc est de 256 caractères, de SØH inclus à ETB ou ETX inclus, sans compter les caractères DLE et SYN additionnels.

7) Format des caractères

HEXADECIMAL

SØH	01
STX	02
ETX	03
EØT	37
ACK	2E
BEL	2F
DLE	10
NAK	3D
SYN	32
ETB	26
ENQ	2D
RS	1E

BINAIRE

PØL	0 1 0 X X X X X
SEL	1 1 1 X X X X X
C2	Y Y Y A B C D E
C3	1 0 1 0 0 0 0 0 absence C4, C5
C3'	1 0 1 0 0 0 0 1 présence C4, C5
C4	1 1 0 0 Z Z Z Z
C5	1 1 0 0 W W W W
NØB	1 1 0 0 0 V V V

Annexe 2
COMMUNICATION ENTRE OPERATEURS ET ORDINATEURS

I - COMMUNICATION OPERATEUR TERMINAL - ORDINATEUR TERMINAL

# RB ↵ EM	Demande d'émission
# RB ↵ RE	Demande de réception
# RB ↵ SU	Demande de suspension
# RB ↵ RP	Demande de reprise

La demande d'émission a pour objet de lancer la partie émission de la procédure de remote-batch.

La demande de réception a pour objet de lancer la partie réception de la procédure de remote-batch.

La demande de suspension a pour effet d'interrompre la transmission en cours. Quand la transmission peut être reprise, l'opérateur l'indique par # RB ↵ RP.

D'autres messages ont été créés pour la mise au point du système. Ils consistent essentiellement à ajuster la valeur de certains paramètres tels que, compteurs de time-out, période des times-out, compteur de répétitions en cas de messages erronés, etc...

II - COMMUNICATION OPERATEUR TERMINAL ↔ CENTRAL

Elle est compatible avec le fonctionnement du télésymbiont sous siris 7 (22).

II.1 - si l'opérateur terminal désire envoyer un message à l'opérateur central, le message a le format :

/ MESSAGE < texte >

- Quand un travail est entré sur le central à partir du terminal, le télésymbiont envoie le message :

```
sys-id    JØB IN    job-id    compte    date    heure
```

sys-id est l'identificateur du travail donné par le télésymbiont
job-id est l'identificateur du travail
compte est le numéro de compte de l'utilisateur.

II.2 - L'opérateur terminal peut demander l'état d'un travail pour lequel il a déjà reçu son identificateur "sys-id".

Pour cela, il frappe la commande :

```
/STATUS    sys-id
```

Le télésymbiont répond alors d'une des manières suivantes :

1°) SYS-id IN CARDI , nombre (HOLD)

nombre est le nombre de cartes entrées. Le travail est sur disque, mais pas encore en file d'attente.

HOLD indique que les résultats de ce travail seront stockés sur disque en attendant que l'opérateur terminal demande leur retransmission.

Ce message signifie que le travail n'est pas dans la file d'attente des travaux.

2°) SYS-id QUEUED PRT , chiffre { W } (HOLD)
D

indique :

- a) que le travail a été mis en file d'attente avec la priorité indiquée par "PRT , chiffre"

b) qu'il est actuellement à l'état dormant (D) ou en attente de ressources (W).

3°) SYS-id ACTIVE CARDO , nombre1 LISTO , nombre2 RTIME, durée $\left. \begin{matrix} W \\ A \end{matrix} \right\}$

indique :

- a) le travail est actif
- b) il y a "nombre 1" cartes à perforer
"nombre 2" lignes d'imprimante à écrire
- c) le temps résiduel qui lui est alloué est de "durée"
- d) il est en attente de ressources, ou d'évènement

4°) SYS-id REPORTING CARDO , nombre LISTO , nombre 2

Le travail est terminé, mais tout n'est pas encore envoyé vers le terminal.

Les renseignements ont la même signification que dans (3).

5°) SYS-id UNKNOWN

Le travail est inconnu du système.

II.3 - L'opérateur terminal peut décider dans une certaine mesure du moment où il recevra les résultats

- / HOLD demande que tous les fichiers en sortie soient maintenus sur le central

- / RELEASE demande que ces fichiers maintenus soient maintenant expédiés.

/ SWITCH (nom de station) $\begin{cases} \text{LP} \\ \text{CP} \end{cases}$

demande que la sortie des résultats se fasse sur une certaine station (le central si on le désire, ou une autre station terminale)

/ RESTORE rétablit la situation précédant la commande /SWITCH

Annexe 5 - BIT D'IMPARIÉTÉ de BCC

La question que nous nous sommes posée était la suivante : le coupleur doit-il vérifier le bit d'imparité du caractère BCC ?

Autrement dit : nombre de 1 du caractère BCC doit-il toujours être impair ? (dans le cas de parité, le nombre de 1 de BCC est toujours pair).

Soit un bloc de n caractères :

Appelons b_{ji} ($i = 1$ à 8) les bits du caractère j de ce bloc :

le bit d'imparité est $b_{j8} = 1 + \sum_{i=1}^7 b_{ji} \pmod{2}$

Appelons B_i ($i = 1$ à 8) les bits du caractère BCC

$$B_i = 1 + \sum_{j=1}^n b_{ji} \pmod{2}$$

La question est alors : est-ce que la relation $B_8 = 1 + \sum_{i=1}^7 B_i \pmod{2}$ est toujours vérifiée ?

$$B_8 = 1 + \sum_{j=1}^n b_{j8} \pmod{2}$$

$$B_8 = 1 + \sum_{j=1}^n 1 + \sum_{j=1}^n \sum_{i=1}^7 b_{ji} \pmod{2}$$

$$B_8 = \sum_{j=1}^{n+8} 1 + \sum_{i=1}^7 (1 + \sum_{j=1}^n b_{ji}) \pmod{2}$$

$$B_8 = \sum_{j=1}^n 1 + \sum_{i=1}^7 B_i \pmod{2}$$

Donc le caractère d'imparité longitudinale est vérifié transversalement si $\sum_{j=1}^n 1 = 1 \pmod{2}$, c'est-à-dire si le nombre de caractères n du bloc est impair et seulement dans ce cas. Le coupleur ne doit donc pas vérifier le bit d'imparité du caractère BCC.

BIBLIOGRAPHIE

R E F E R E N C E S

- { 1 } W.M. AYDELOTTE
Communications message switching an analysis
Computer and automation, july 1971, p. 8-13

- { 2 } K.A. BARTLETT - R.A. SCANTLEBURY and P.T. WILKINSON
A note on reliable full duplex transmission over half
duplex links
C. ACM, vol. 12, may 1969, p. 260-261

- { 3 } D. BJORNER
Finite state automation
Définition of data communication line control procedures
Fall joint Computer Conference - AFIPS, 1970

- { 4 } C. CHABROL - D. COMTE - O. GELLY
Moniteur de communication entre programmes répartis sur
le réseau
Congrès AFCET, nov. 1973, p. 307-317

- { 5 } S.D. CROCKER - J.F. HEAFNER - R.M. METCALFE - J.B. POSTEL
Function oriented protocols for the ARPA computer Networks
Spring Joint Computer Conference, 1972, p. 271-279

- { 6 } H.P. DAENDLIKER
Remote job entry
Bull. SEV60, juin 1969, p. 577-579

- { 7 } J. DU MASLE
Quelques aspects des langages de commande pour réseaux
d'ordinateurs
Workshop ACM - IRIA, mars 1972, p. 277-287

- { 8 } M. ELIE
Décomposition et représentation de la fonction de transport
de l'information dans un réseau.
Workshop ACM - IRIA, mars 1972, p. 63 - 78

- (9) A. GADEL
Conception et réalisation d'un coupleur pour transmission
synchrone
(à paraître)
- (10) GIRARDI
L'interface de communication d'un réseau d'ordinateurs (SØC)
Workshop ACM - IRIA, mars 1972, p. 265-273
- (11) J.P. GRAY
Line Control procedures
Proc of the IEEE on computer communications
Vol. 60, n° 11, p. 1301-1312, nov. 1972
- (12) R.E. KAHN
Resource sharing computer communications Networks
Proc. of the IEEE, vol. 60, n° 11, p. 1397-1407, nov. 1972
- (13) G. LOUIT
Les protocoles de ARPANET
Workshop ACM - IRIA, mars 1972, p. 59-61
- (14) W.C. LYNCH
Reliable full duplex file transmission over half duplex
telephone lines
C. ACM, vol. 11, p. 407-410, june 1968
- (15) W.C. LYNCH
Commentary on the foregoing note
(cf. référence n° 2)
C. ACM. vol. 12, p. 261, may 1969
- (16) D.L. MILLS
Communication software
Proc. of the IEEE, vol. 60, n° 11, p. 1333-1341, nov. 1972

- { 17 } L. POUZIN
Network architectures and components
1st european workshop on computer networks
Arles 1973, p. 227-265
- { 18 } M. SOMIA
Le projet SOC
Workshop ACM - IRIA, mars 1972, p. 35-45
- { 19 } Standards ACM
- Data communication control procedures for the USA
standard code for information interchange
C.ACM, vol. 12, n° 3, march 1969, p. 166-178
 - Heading format for data transmission
C.ACM, vol. 11, june 1968
 - Description of systems used for data transmission
C.ACM, vol. 9, october 1966
- { 20 } D.C. WALDEN
A system for interprocess communication in a ressource
sharing computer networks
C.ACM, april 1972, vol. 15, n° 4, p. 221-230

ARTICLES ET OUVRAGES NON REFERENCÉS

M.D. BACON and G.M. BULL

Data transmission

Ed. Mac Donald and American Elsevier Inc. New-York

J. CLAVIER - M. NIQUIL - G. COFFINET - F. BEHR

Théorie et technique de la transmission des données

Tomes I et II, éd. Masson, 1972

N. CULLMANN

Codes détecteurs et correcteurs d'erreurs

Dunod, 1967

D.W. DAVIES

Packet switching in a public data Network

IFIP proc. Ljubljana 1971, p. 622-627

D.W. DAVIES

Principles of packet switching

1st european Workshop on computer networks

Arles, 1973, p. 175-190

R. DESPRES

Procédure de transmission full duplex et point à point

Workshop ACM - IRIA, mars 1972, p. 59-61

J.M. GOETHALS

Le contrôle des erreurs dans les transmissions digitales

Congrès AFCET 1973, exposé de synthèse p. III - XIII, tome 1

L. POUZIN

Les choix du réseau de communication Cigale

Congrès AFCET, nov. 1973, p. 265

A. SPATARU

Théorie de la transmission d'information

Tome 2, Masson 1973

R.D. WADE - G.P. CAWSEY - R.A.K. WEBER

A teleprocessing approach using standard equipment

IBM syst journal, n° 1, 1969, p. 28-47

COMPTE-RENDUS DE COLLOQUES

Workshop ACM - IRIA

Réseaux d'ordinateurs

23-24 mars 1972

Congrès AFCET

Rennes 7-9 novembre 1973

Tomes 1 et 2

1st European workshop

Computer networks

Arles, 24 avril - 4 mai 1973

NOM DE L'ETUDIANT : SCHAFF André

NATURE DE LA THESE : DOCTEUR INGENIEUR

VU, APPROUVE

& PERMIS D'IMPRIMER

NANCY, le 12 Juin 1974

LE PRESIDENT DE L'UNIVERSITE DE NANCY I



J.R. HELLOUT