



**HAL**  
open science

# Mécanisme de communication par bus série pour des réseaux informatiques locaux

Marian Marinescu

► **To cite this version:**

Marian Marinescu. Mécanisme de communication par bus série pour des réseaux informatiques locaux. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1978. Français. NNT: . tel-00288289

**HAL Id: tel-00288289**

**<https://theses.hal.science/tel-00288289>**

Submitted on 16 Jun 2008

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

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

# THESE

*présentée à*

**Institut National Polytechnique de Grenoble**

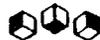
*pour obtenir le grade de*

**DOCTEUR DE 3<sup>ème</sup> CYCLE**

**Génie Informatique**

*par*

**Marian MARINESCU**



**MECANISME DE COMMUNICATION PAR BUS SERIE  
POUR DES RESEAUX INFORMATIQUES LOCAUX**



Thèse soutenue le 12 septembre 1978 devant la Commission d'Examen .

Président : L. BOLLIET

Examineurs : F. ANCEAU

G. NOGUEZ

J.M. PITIE



# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1978-1977

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET

M. Georges LESPINARD

---

## PROFESSEURS TITULAIRES

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

## PROFESSEURS SANS CHAIRE

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

.../...

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

#### MAITRES DE CONFERENCES

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

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

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

#### Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique) E.N.S.E.E.G.

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

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

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

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

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

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



*à ma femme*

*à mes parents*



Je tiens à remercier,

Monsieur L. BOLLIET, Professeur à l'Université de Grenoble,  
pour l'honneur qu'il me fait en acceptant de présider le  
jury de cette thèse ;

Monsieur F. ANCEAU, Maître de Conférences à l'Institut  
National Polytechnique de Grenoble, pour ses conseils et  
encouragements tout au long de cette étude ;

Monsieur G. NOGUEZ, Professeur à l'Université de Paris VI,  
dont les remarques pertinentes m'ont aidé à améliorer la  
rédaction de ce document ;

Monsieur J.M. PITIE, Directeur du Groupe Architecture et  
Fonction de Commande au CNET à Lannion, pour l'attention  
et le soutien permanents qu'il a accordés à ce travail ;

les membres de l'Equipe d'Architecture des Ordinateurs de  
l'ENSIMAG, avec qui j'ai pu discuter nombre de points de  
cette étude ;

Madame CHALAND pour sa dactylographie soignée et rapide ;

et enfin le service de reprographie de l'IMAG qui a assuré  
le tirage de ce document.

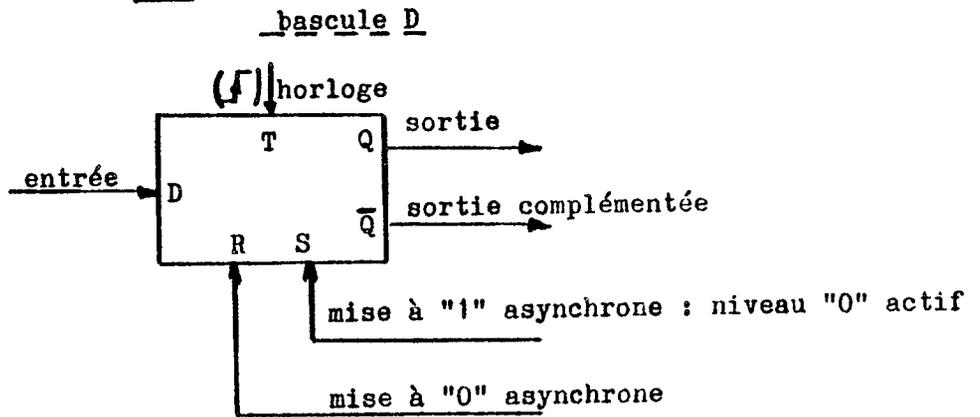
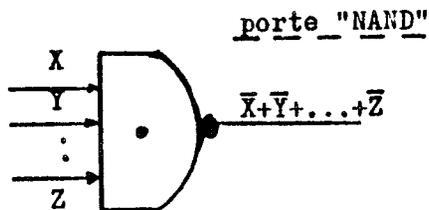
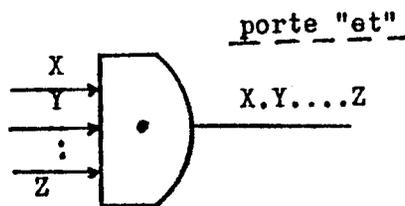
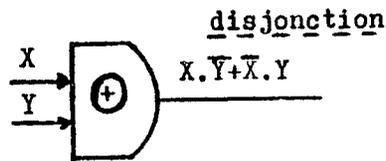


## S O M M A I R E

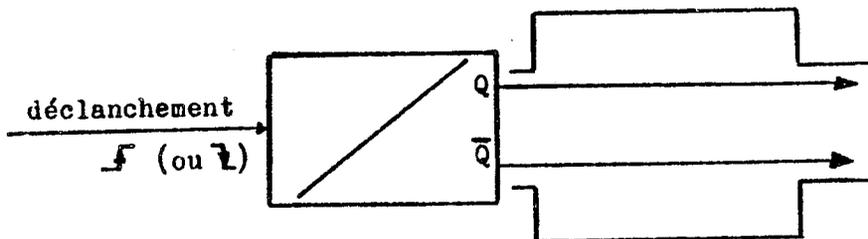
INTRODUCTION	p. 1
CHAPITRE 1 : RÉSEAUX LOCAUX, QUELQUES STRUCTURES	
1.1. Types de réseaux	p. 5
1.2. Réseau centralisé. Procédure SDLC	p. 9
1.3. Réseau décentralisé	p. 12
1.3.1. Connexion en boucle	p. 12
1.3.2. Connexion par bus. Auto-allocation	p. 14
1.4. Conclusion	p. 18
CHAPITRE 2 : PRINCIPES DU MÉCANISME DE COMMUNICATION	
2.1. Généralités	p. 19
2.1.1. Démarche	p. 19
2.1.2. But de l'étude	p. 23
2.2. Mécanisme d'auto-allocation	p. 25
2.2.1. Principes de l'auto-allocation	p. 26
2.2.1.1. Détection de ligne libre	p. 28
2.2.1.2. Détection de conflit	p. 31
2.2.1.3. Algorithme de l'auto-allocation	p. 33
2.2.2. Types d'auto-allocation	p. 36
2.2.3. Comportement idéal de l'auto-allocation	p. 38
2.2.4. Comportement réel de l'auto-allocation. Calcul du débit	p. 41
2.2.5. Généralisation de l'auto-allocation	p. 50

2.2.6. Mécanisme élémentaire d'échange	p. 53
2.3. Protocole de base	p. 56
2.3.1. Format des messages	p. 57
2.3.2. Accusé de réception	p. 60
2.4. Performances du mécanisme de communication	p. 65
2.4.1. Efficacité du mécanisme de communication	p. 65
2.4.2. Comportement des files d'attente	p. 68
2.5. Conclusion	p. 72
CHAPITRE 3 : IMPLÉMENTATION DU MÉCANISME DE COMMUNICATION	
3.1. Architecture du réseau	p. 74
3.2. Structure d'une station	p. 77
3.3. Application. Autocommutateur téléphonique	p. 82
CHAPITRE 4 : RÉALISATION DU COUPLEUR	
	p. 93
CONCLUSIONS	
	p.118
ANNEXES	
Annexe 1. Caractéristiques électriques des lignes longues	p.120
Annexe 2. Calcul de la dispersion introduite par l'auto-allocation	p.129
Annexe 3. Reconnaissance de l'information élémentaire. Calcul du débit	p.137
Annexe 4. Etude du modèle : un serveur et plusieurs files d'attente	p.141
BIBLIOGRAPHIE	
	p.147

Conventions pour la représentation des circuits



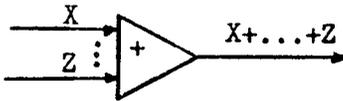
monostable



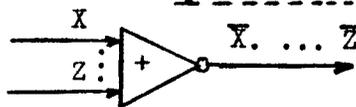
inverseur



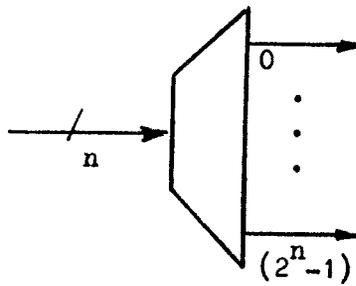
porte "ou"



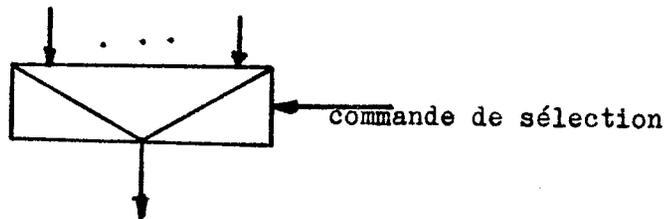
porte "Ni"



décodeur



sélecteur (multiplexeur)



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

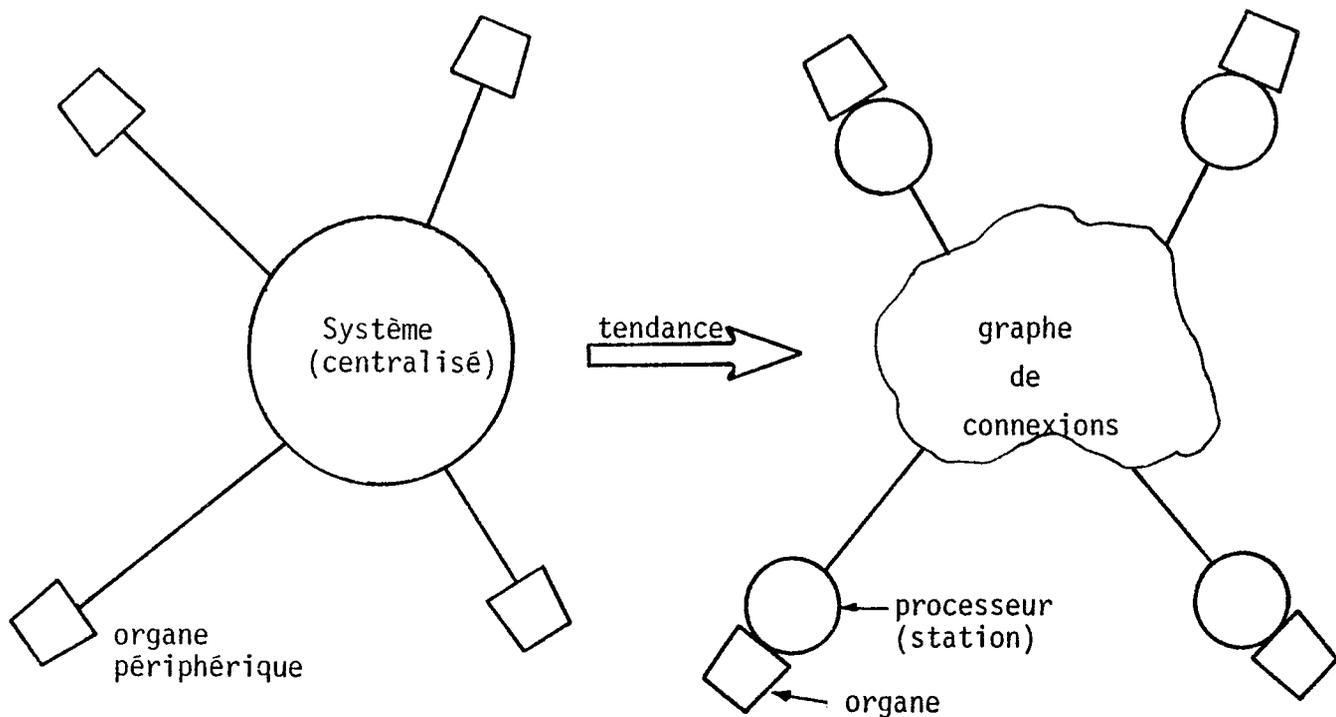
## 1. INTÉRÊT DES RÉSEAUX SÉRIE POUR LES SYSTÈMES RÉPARTIS

Le début des années 1970 a vu l'apparition des circuits intégrés de grande complexité (LSI) et plus particulièrement celle du microprocesseur (unité centrale intégrée). La tendance s'est développée en s'amplifiant.

On assiste actuellement à une augmentation vertigineuse de la complexité, à une diminution du prix, et les constructeurs ne voient pas encore de limite technologique à cette tendance. Ceci fait que l'on doit s'attendre à l'apparition prochaine d'un ordinateur intégré sur une seule pastille de silicium. Cette puissance de traitement bon marché a une influence considérable sur l'architecture des systèmes informatiques par les perspectives nouvelles qu'elle ouvre aux systèmes répartis.

Le très grand rapport performance/coût des microprocesseurs rend possibles et souhaitables des solutions "décentralisées" par la répartition des fonctions là où on en a besoin, c'est-à-dire généralement à la frontière du système avec l'environnement. Ceci fait qu'actuellement, on assiste dans la conception des systèmes à un passage accéléré des systèmes centralisés aux systèmes répartis géographiquement.

Les systèmes répartis fonctionnellement offrent des avantages non négligeables du point de vue de la qualité des services rendus, sûreté de fonctionnement et facilité de conception (le passage de l'application réelle à la réalisation informatique est plus aisé et plus rapide).



La présence d'un grand nombre de processeurs (stations) spécialisés répartis dans l'espace, fait que les systèmes sont de plus en plus conçus en tant que "réseaux de processeurs" et non en tant que "machines multiprocesseurs" (autour d'une mémoire commune). Le dialogue entre les stations implique évidemment la nécessité des mécanismes de communication : protocole d'échange + liaisons physiques dans le réseau.

Pour des questions de fiabilité et de coût, on cherche à minimiser le nombre de lignes (support de l'information) dans le réseau. Cette minimisation va dans le bon sens (si l'on peut dire), du fait que chaque processeur se suffit relativement à lui-même et fait rarement appel au reste du réseau. Autrement dit, le débit d'informations induit par le dialogue dans le réseau entre des modules intelligents n'est pas très élevé.

Ces considérations nous amènent à choisir la liaison série comme élément de base du mécanisme de communication.

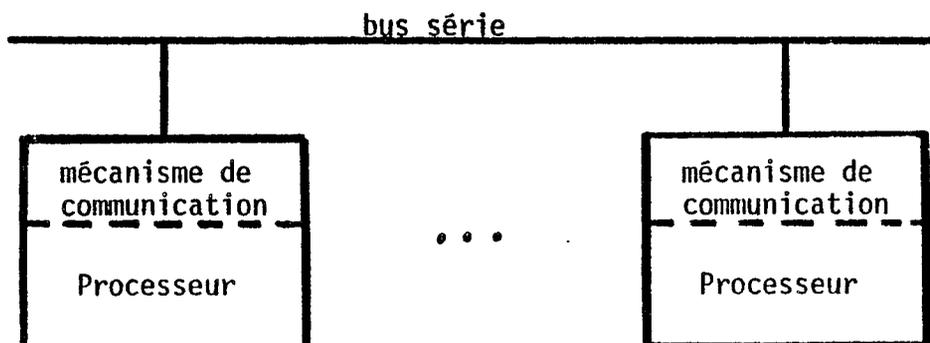
## 2. APERÇU GÉNÉRAL DE L'ÉTUDE

Notre étude concernera les réseaux locaux, c'est-à-dire associés à des applications localisées dans un domaine géographique restreint (la centaine de mètres donne l'ordre de grandeur de cette localisation).

Le travail exposé dans ce document consiste en la définition/réalisation d'un mécanisme de communication adapté aux réseaux géographiquement locaux. L'étude de ce mécanisme a été guidée par deux axes principaux :

- . architecture répartie,
- . sûreté de fonctionnement.

Le système envisagé se présentera comme un ensemble de stations dialoguant à travers une voie (bus série) d'échange unique et ayant les mêmes prérogatives.



Après avoir présenté dans le chapitre 1 quelques structures importantes de réseaux locaux, nous exposerons au chapitre 2 les principes du mécanisme de communication proposé. Nous y présentons la communication au niveau de l'échange élémentaire (principalement le mécanisme d'auto-allocation) et du protocole de base, ainsi qu'une estimation des performances. Le chapitre 3 montre l'application du mécanisme de communication à un autocommutateur téléphonique réalisé sous la forme d'un réseau local. Enfin, le chapitre 4 expose la réalisation du coupleur (mécanisme de communication) dans la technologie actuelle.



# CHAPITRE 1

## RÉSEAUX LOCAUX

### QUELQUES STRUCTURES

## 1.1. TYPES DE RÉSEAUX

Les critères de classification des réseaux sont multiples suivant l'aspect sous lequel ils sont regardés. Un des aspects essentiels est le mécanisme de communication, puisque par définition, la structure "réseau" est introduite en vue de la communication.

Un premier critère considéré est celui de la *localité* et on distingue :

- . réseau local : courtes distances,
- . réseau global : longues distances.

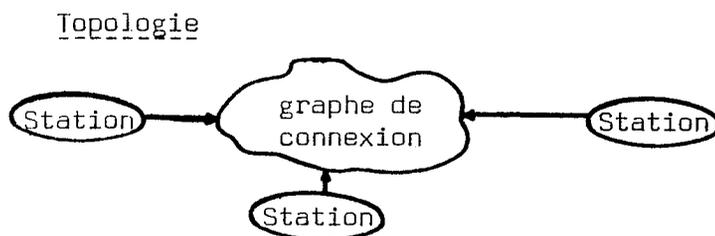
Ce critère peu rigoureux (la notion de courte-longue distance est assez subjective), concerne le périmètre géographique à l'intérieur duquel l'application est définie.

Notre étude portera sur les réseaux locaux. Ce choix est fait en tenant compte de l'existence d'une large gamme d'applications, pas nécessairement informatiques (commutation téléphonique, contrôle de processus industriels), localisées dans un espace géographique restreint et dont la complexité nécessite des moyens informatiques.

La complexité croissante de ces applications localisées fait que les réseaux locaux associés doivent avoir des mécanismes de communication élaborés, comparables à ceux de réseaux télé-informatiques classiques.

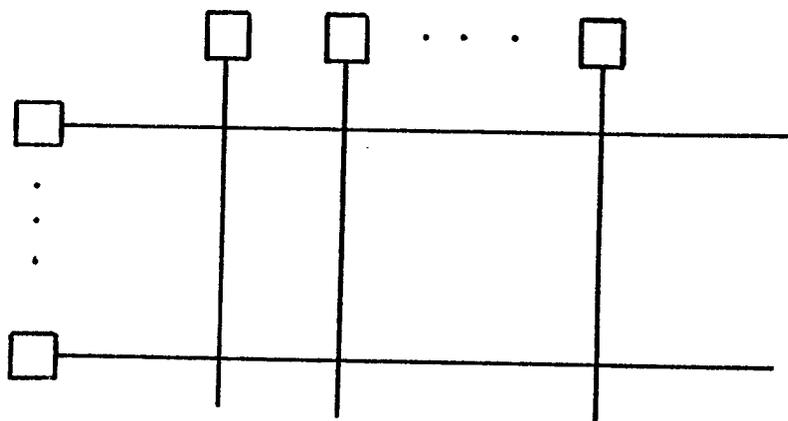
La spécificité de ces applications fait que l'architecte doit chercher des structures adaptées qui exploitent cette propriété de localisation, en trouvant le bon compromis entre les différentes contraintes (qualité de service, rapport performance/coût, sûreté de fonctionnement, ..). On doit dire que la propriété de localité est mise en évidence au niveau de la réalisation matérielle, mais elle est transparente au niveau fonctionnel du système.

Ceci étant dit, on passera en revue les autres aspects concernant la communication.



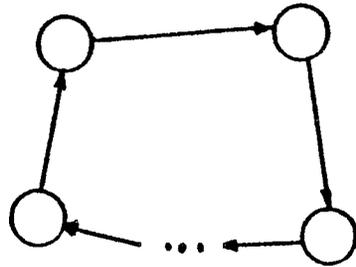
Un cas intéressant est celui des structures régulières qui permettent la recherche de mécanismes homogènes dans le réseau.

.connexion matricielle



De par le grand nombre de liaisons nécessaires, cette structure est employée pour de très courtes distances et le système est plutôt une machine (ordinateur) réalisée sous forme de réseau, comme par exemple la machine CORAIL [POU], le système PROCOP [DEG].

.connexion en boucle



.connexion par bus (ligne multipoint ou encore équipotentielle)



Centralisation / décentralisation

- . Réseau centralisé : il existe un organe privilégié (maître, arbitre) qui impose la discipline du dialogue.
- . Réseau décentralisé : pas d'organe privilégié (allocateur réparti) ; toutes les stations ont les mêmes prérogatives.

Capacité des liaisons

Ceci concerne le nombre de lignes nécessaires pour échanger l'unité d'information choisie pour le réseau.

- . Liaisons parallèles : utilisées pour de très courtes distances, comme par exemple la machine CORAIL [POU].
- . Liaisons série : une seule ligne (l'information élémentaire est le bit). C'est la liaison généralement adoptée pour le réseau.

### Modes de transmission

- . Mode asynchrone : cette transmission s'effectue caractère par caractère (caractère = chaîne binaire de longueur fixe), l'intervalle de temps entre deux caractères étant quelconque. L'intervalle de temps entre deux bits consécutifs d'un même caractère est fixe (constante du système).
- . Mode synchrone : cette transmission s'effectue par blocs (bloc = suite de longueur variable de caractères). Chaque bit du bloc est transmis à intervalles de temps constants. Des caractères spéciaux (de synchronisation) permettent au récepteur de cadrer la chaîne binaire pour restituer la suite initiale de caractères.

## 1.2. RÉSEAU CENTRALISÉ. PROCÉDURE SDLC

Un dialogue typique dans un réseau centralisé est celui régi par la procédure SDLC [DON]. Cette procédure permet de véhiculer des chaînes de bits de longueur variable. C'est une procédure synchrone.

Le format des messages est le suivant :

<u>flag</u>	<u>adresse</u>	<u>commande</u>	<u>informations</u>	<u>code cyclique</u>	[flag]
8	8	8	variable	16	

N.B. Il est important de distinguer la procédure SDLC (procédure centralisée) du format SDLC, ce dernier pouvant être très bien utilisé pour une procédure décentralisée.

Le flag a la configuration 0 1 1 1 1 1 1 0 et permet la délimitation du message. La présence accidentelle du flag dans le corps du message est supprimée par l'insertion d'un "0" (éliminé à la réception) après toute séquence de cinq "1" consécutifs.

Le champ "adresse" spécifie l'adresse de l'émetteur ou du destinataire (c'est toujours l'adresse d'une station secondaire).

Le champ "commande" spécifie la sémantique du message.

Le champ "code cyclique" permet la vérification de la transmission.

La procédure SDLC suppose l'existence d'une station principale et de plusieurs stations secondaires.

Il existe dans le réseau un sens "logique" de parcours, qui part de la station principale, passe par les stations secondaires et revient à la station principale (figure 1).

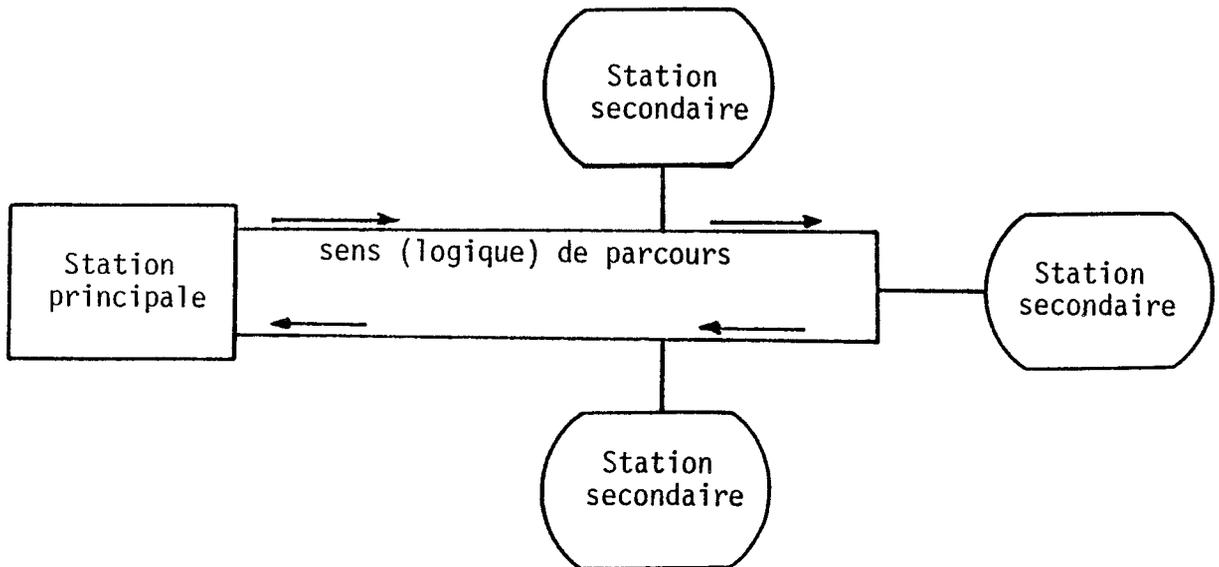


FIG.1. Réseau régi par la procédure SDLC

La procédure peut être utilisée dans des configurations point à point, multipoint, ou en boucle. Dans tous les cas, le dialogue se fait entre la station principale et les stations secondaires.

La procédure permet deux types de dialogue :

. Arbitre centralisé :

La station principale dialogue individuellement (auscultation) avec chacune des stations secondaires.

. Passage de relais :

La station principale lance un "relais" (configuration 0 1 1 1 1 1 1 1) vers la première station secondaire (dans le sens du parcours). La station qui a reçu le relais émet des messages (si elle a quelque chose à communiquer) et à la fin elle passe le relais vers la station suivante qui, à son tour, émet des messages, après quoi elle passe le relais, ... Quand le relais revient à la station principale, celle-ci peut continuer en mode "passage de relais" (relance de relais) ou commuter en mode "arbitre centralisé".

Le contrôle de la bonne réception d'un message par le destinataire n'est pas traité au niveau de la procédure, mais reporté au niveau supérieur et doit être pris en charge par la station principale.

La procédure SDLC (et son format), par sa simplicité, la tentative de normalisation dont elle fait l'objet et par le développement de circuits spéciaux pour sa mise en oeuvre (SDLC 8273 d'INTEL, ADLC 6854 de Motorola, COM 5025 de SMC, SIO de Zilog), semble être bien adaptée à l'interconnexion de processeurs spécialisés.

## 1.3. RÉSEAU DÉCENTRALISÉ

### 1.3.1. Connexion en boucle [REA]

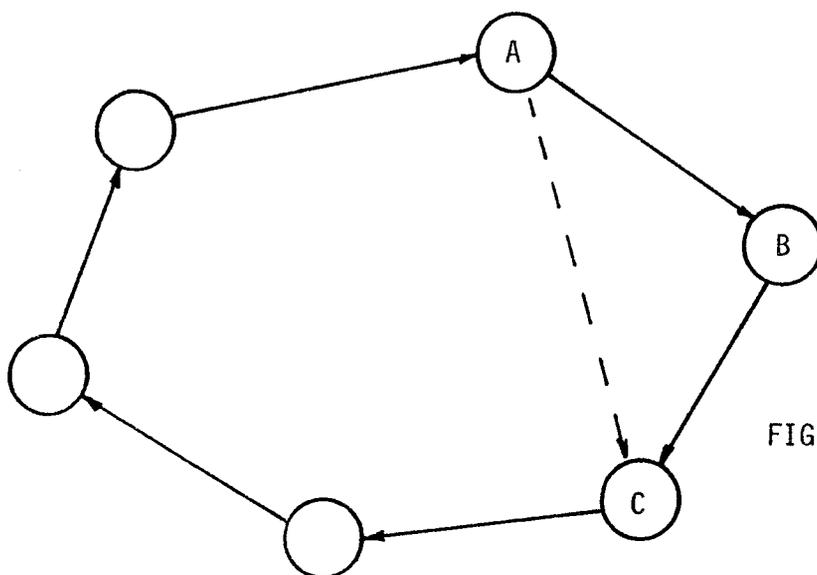


FIG.2. Connexion en boucle

Tout message est émis dans le sens de la boucle et "traverse" toutes les stations qui se trouvent entre l'expéditeur et le destinataire du message. Toute station émet sur sa ligne aval d'une part ses propres messages (générés par elle) et d'autre part les messages reçus sur la ligne amont et qui ne la concernent pas (ne lui sont pas adressés).

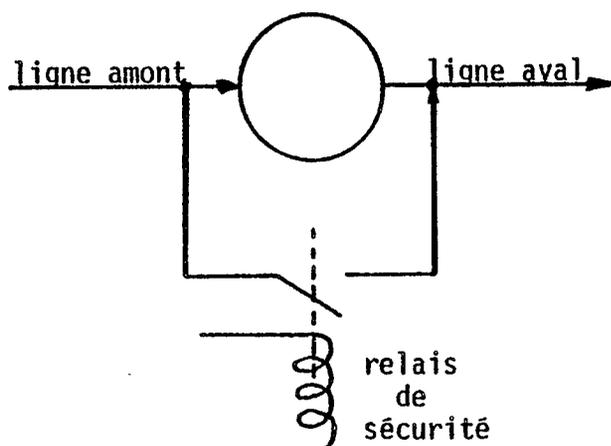
Supposons un message  $m$  envoyé par A dont le destinataire est C (figure 2). Ce message  $m$  est reçu (dans le sens de la boucle) tout d'abord par B. Si B est en repos (n'émet rien), il fait passer le message vers C, sinon il le stocke temporairement dans une file d'attente. Cette file sera vidée (vers C) quand la voie vers C deviendra libre. Le message  $m$  finit par arriver à C qui, reconnaissant que c'est pour lui, ne le retransmet pas plus loin.

On voit que les messages circulant sur la boucle sont retardés par les stockages temporaires. Le délai de transmission ainsi introduit croît avec la charge du système et avec le nombre de stations.

Un premier avantage de la connexion en boucle est de permettre un débit global d'échange élevé (en particulier plus élevé que la connexion par bus), puisque des messages différents peuvent circuler *en même temps* sur les différentes branches de la boucle. Un deuxième avantage de la boucle est l'utilisation de simples liaisons point à point entre les stations.

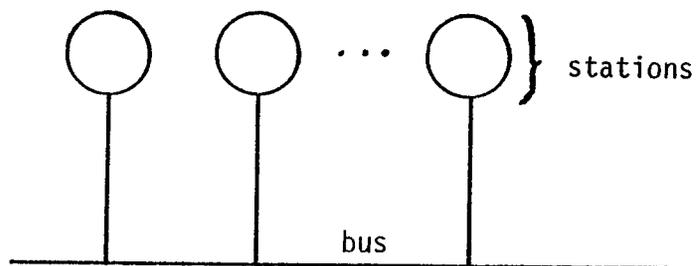
La structure en boucle est désavantageuse du point de vue de la sécurité. Comme une station quelconque manipule des messages qui ne la concernent pas (mais qu'elle stocke temporairement), en cas de panne d'une station, le dialogue dans le système est compromis presque totalement (ne serait-ce que par la perte des messages stockés temporairement par la station fautive).

N.B. Pour que le système puisse continuer à fonctionner, il faut que toute station défaillante soit court-circuitée pour être éliminée de la boucle.



Un autre problème est constitué par l'accusé de réception. Du fait des stockages temporaires, on ne peut pas définir un accusé de réception au niveau de la procédure de base, accusé qui arrive dans une courte tranche de temps après l'émission du message. L'accusé de réception est donc reporté aux niveaux supérieurs, ce qui fait que le temps d'attente de cet accusé de réception influe sensiblement sur les performances. Pendant ce temps, l'expéditeur doit garder une copie du message émis pour une éventuelle ré-émission (si l'accusé de réception manque), ce qui constitue un encombrement supplémentaire de mémoire. Cette attente dépend du temps mis par le message pour parcourir la boucle entre l'expéditeur et le destinataire et du temps mis par l'accusé de réception pour parcourir le reste de la boucle entre le destinataire et l'émetteur, et elle est d'autant plus grande que le système est chargé.

### 1.3.2. Connexion par bus. Auto-allocation



Il y a dans le réseau une seule voie de communication (le bus), partagée entre les stations, ce qui fait que toute station travaille en half-duplex.

L'allocation du bus est réalisée d'une manière complètement décentralisée (distribuée) par le mécanisme que l'on appellera (à défaut d'autre nom consacré) *mécanisme d'auto-allocation*, et qui repose essentiellement sur l'algorithme de Nisnevitch [NIS1].

Le principe de base de l'auto-allocation est la détection d'incohérence (ou de conflit) : si la valeur du bus est égale à la valeur émise, on continue à émettre, sinon cela veut dire que quelqu'un d'autre est concurrent et on cesse d'émettre (on ré-essaie plus tard).

On présentera plus en détail dans le chapitre 2 les conditions de bon fonctionnement de l'auto-allocation.

Deux types d'auto-allocation peuvent être dégagés, types qui *dépendent de la réalisation de la ligne* :

**a) auto-allocation "déterministe" :**

Si plusieurs stations émettent en même temps, il y a toujours une station dont le message est véhiculé par le bus et ceci fait que le nombre de stations n'a pas d'influence sur les performances de la communication (pour une quantité de travail donnée). L'efficacité du mécanisme est fixe.

N.B. Efficacité = durée utile / (durée utile + temps du mécanisme).

**• Système à priorité décentralisée (Algorithme de Nisnevitch [NIS2])**  
.....

Dans ce système, le bus est composé de deux parties : bus message,  
bus allocation.

Les stations ont des adresses (priorités) distinctes.

Le bus d'allocation est composé de n lignes, le nombre n étant choisi pour que les adresses distinctes des stations puissent être écrites sur n bits. Il permet d'allouer le bus message à une des stations.

On prend l'hypothèse : une ligne réalise le "ou" logique des bits que les stations émettent sur elle.

Une station qui veut émettre sur bus message met (quand le bus est libre) sur le bus allocation ses bits d'adresse, ce qui a pour effet d'occuper le bus. Pour chaque ligne de bus allocation, la station a un détecteur de conflit (incohérence) et un conflit détecté sur une ligne a pour effet d'inhiber l'envoi des bits adresse de poids inférieur. Si aucun conflit n'est détecté (donc la station a capturé le bus), la station émet sur le bus message, sinon elle attend que le conflit disparaisse. A la fin du message, on enlève l'adresse du bus allocation.

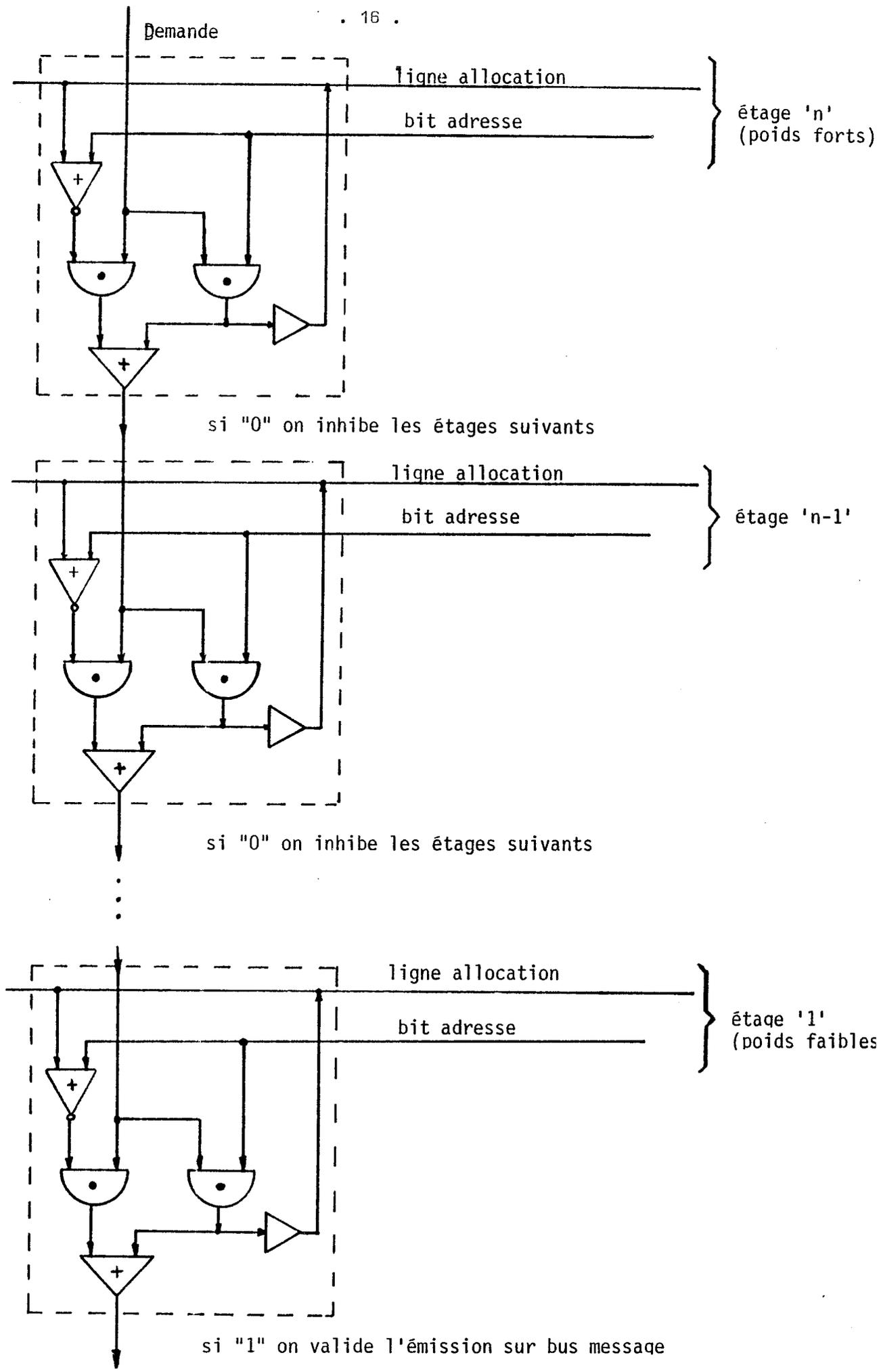


FIG.3. Schéma de l'auto-allocation pour bus parallèle (d'après Nisnevitch)

• Systeme Cobus [SOM]  
.....

C'est un reseau serie base sur l'auto-allocation.

Le bus (ligne serie) est realise avec des portes en collecteur ouvert.

La procedure de communication est asynchrone et comprend l'accuse de reception.

b) Auto-allocation "probabiliste" :

Quand plusieurs stations emettent en meme temps, l'information vehiculee par la ligne est erronee (aucun message correct n'est emis), le taux de conflits ayant ainsi une influence sur les performances.

L'augmentation de la charge ou du nombre de stations entraine une diminution de l'efficacite du mecanisme.

Ce type d'auto-allocation a ete utilise dans le reseau ETHERNET [MET].

Son avantage est de pouvoir s'appliquer meme quand le support de l'information est constitue par des ondes dans l'espace.

## 1.4. CONCLUSION

Nous chercherons à exploiter la propriété de localisation géographique qui est caractéristique pour une large gamme de systèmes répartis.

Pour des questions de sûreté de fonctionnement, standardisation, coût, facilité de conception, des solutions modulaires par l'itération d'un nombre restreint de structures élémentaires, sont à rechercher.

La nécessité d'utiliser partout des mécanismes homogènes, nous a amenés à étudier le réseau décentralisé et plus précisément la connexion par bus régie par l'auto-allocation qui présente une meilleure sécurité que la connexion en boucle.



## CHAPITRE 2

### PRINCIPES DU MÉCANISME DE COMMUNICATION

## 2.1. GÉNÉRALITÉS

### 2.1.1. Démarche

Du point de vue de la méthode notre principe directeur est le fait que tout système informatique présente une structure hiérarchique de niveaux (couches) d'interprétation [ANC1]. Chaque niveau ( $i$ ) est interprété par le niveau inférieur ( $i-1$ ). Le dernier niveau, le niveau 0, est le niveau matériel.

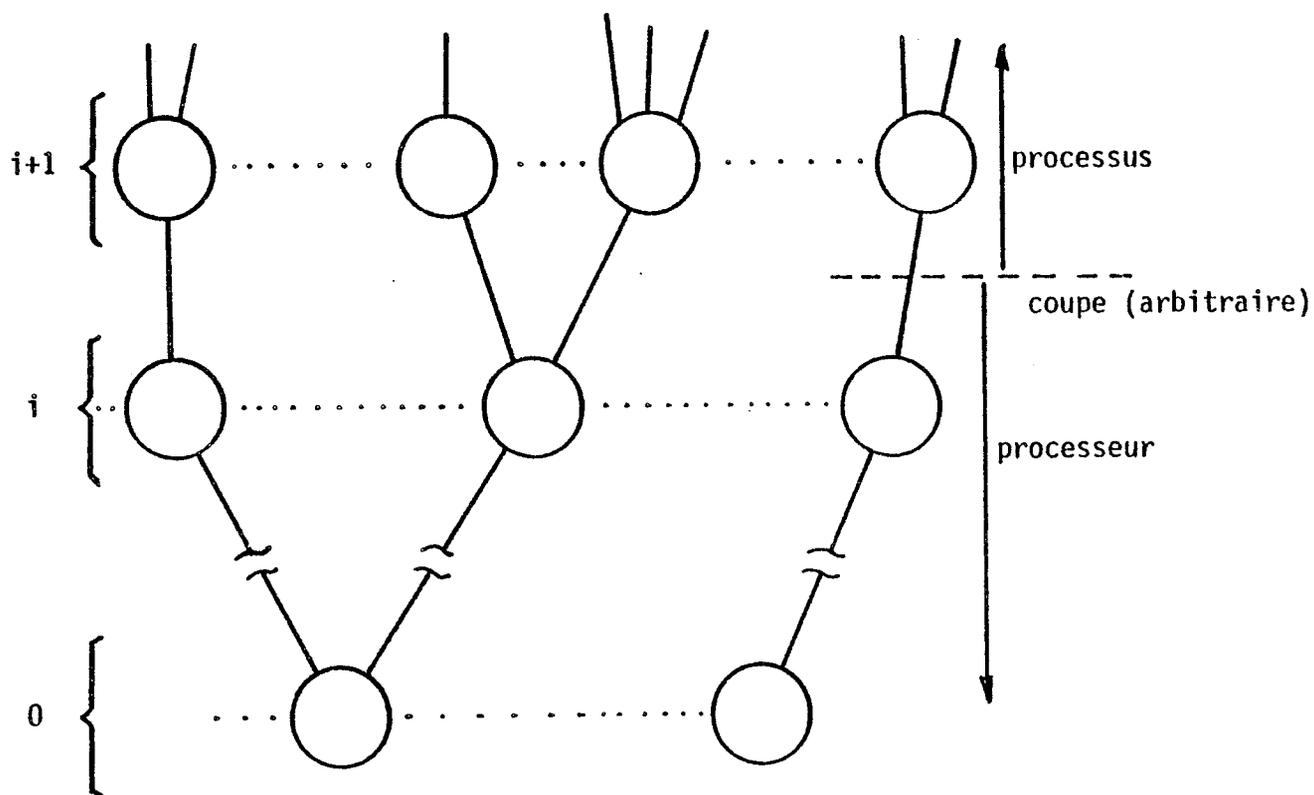


FIG.4. Structure hiérarchique

Cette hiérarchie se manifeste aussi pour les mécanismes de communication : le dialogue à un niveau est réalisé par les mécanismes de communication des niveaux inférieurs. Chaque niveau de la hiérarchie est composé de plusieurs processus qui dialoguent entre eux.

Nous entendrons par dialogue seulement l'échange d'information qui se fait entre des processus d'un même niveau. Autrement dit, si deux processus dialoguent entre eux, ils sont nécessairement au même niveau dans la hiérarchie. Chaque niveau est muni d'un mécanisme de communication (réalisé par les niveaux inférieurs). Il y a donc dans le système, dialogue entre les processus du même niveau et induction du dialogue entre les niveaux.

N.B. induction = propagation de l'activité.

L'induction se propage jusqu'au niveau matériel 0 où a lieu le dialogue effectif sur le support physique de l'information. A l'émission de l'information une induction a lieu dans le sens niveau supérieur vers niveau inférieur. A la réception, l'induction a lieu dans le sens inverse.

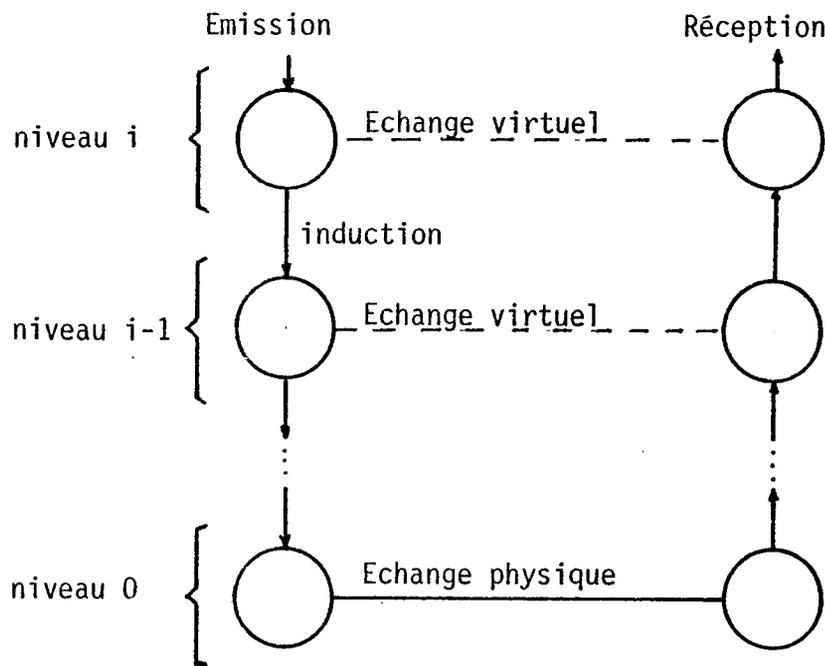
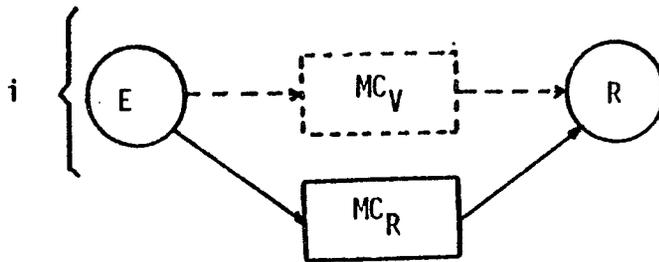


FIG.5. Hiérarchie des dialogues

Cette structure hiérarchique entraîne la transparence du moyen de communication. A tout niveau, les informations échangées sont structurées en *messages*. Le message est défini comme étant la structure élémentaire d'information que les processus d'un même niveau peuvent échanger entre eux. A un niveau  $i$  donné, un échange d'information est schématisé comme suit :



où : E processus émetteur  
R processus récepteur  
 $MC_V$  mécanisme de communication associé au niveau  $i$  (assure l'échange virtuel de l'information)  
 $MC_R$  mécanisme de communication réalisant l'échange réel.

A un niveau  $i$ , les processus disposent explicitement du mécanisme de communication  $MC_V$  associé à ce niveau. Ce mécanisme est défini par le format des messages et par les primitives de dialogue du niveau  $i$ .

L'échange réel est réalisé par le mécanisme  $MC_R$  constitué par les niveaux inférieurs et qui est transparent pour le niveau  $i$  considéré.

N.B. Au niveau matériel 0 il y a identité entre  $MC_V$  et  $MC_R$ .

A un niveau donné, les contraintes sur  $MC_R$  sont imposées par les spécifications de  $MC_V$ .

Dans le contexte d'une architecture répartie, on cherchera à avoir les qualités suivantes pour un mécanisme de communication MC :

- . extensibilité : le nombre de processus dialoguant par l'intermédiaire de MC peut varier ;
- . flexibilité : tout processus "connecté" à MC peut communiquer avec tout autre processus "connecté" à MC. Autrement dit, tous les processus d'un niveau ont la possibilité de dialoguer entre eux.

N.B. Pour un processus, les restrictions de dialogue se feront seulement en jouant sur l'ensemble des voisins (du même niveau) que ce processus connaît.

- . asynchronisme : dû à l'asynchronisme des processus ;
- . uniformité : les processus sont équivalents par rapport à MC ;  
(ont les mêmes prérogatives)
- . modularité
- . sécurité : ceci implique la possibilité de déterminer si l'échange s'est déroulé correctement ou non ;
- . fonctionnement en mode dégradé : en cas de panne, on assure le fonctionnement du système avec des performances diminuées.

N.B. Cette qualité est reliée à l'extensibilité, la flexibilité, l'uniformité, la modularité, la sécurité des mécanismes de communication.

- . utilisation optimale des ressources (files d'attente, lignes physiques, ..) concernées par le dialogue, dans le but d'atteindre les performances désirées.

Nous allons détailler un peu l'aspect sécurité du mécanisme de communication. On a généralement deux moyens :

- . information redondante incorporée dans le message pour permettre au processus récepteur de déterminer si le message reçu est correct ou non ;
- . accusé de réception envoyé par le récepteur à réception du message, pour permettre au processus émetteur de déterminer s'il y a eu ou non bonne réception.

On s'aperçoit que pour l'information redondante, on a une induction dans le sens niveau supérieur vers niveau inférieur et une induction dans le sens contraire pour l'accusé de réception.

### 2.1.2. But de l'étude

Dans notre étude, on s'intéresse aux systèmes répartis réalisés sous forme de réseaux locaux (quelques centaines de mètres).

On aura un ensemble de processus implémentés sur les stations du réseau. En ce qui concerne le dialogue, on considérera les niveaux suivants :

- . protocole fonctionnel : c'est le dialogue entre les processus logiques du système,
- . protocole de base : c'est le dialogue entre les coupleurs d'entrée-sortie des stations,
- . échange physique : transmission effective des informations sur les liaisons physiques du réseau.

Exemples :

protocole fonctionnel : échange de blocs de données,

protocole de base : échange de messages en format SDLC,

échange physique : émission des bits sur une ligne multipoint auto-allouée.

Vues les distances assez importantes entre les stations, pour des questions de coût et de sûreté de fonctionnement, les liaisons dans le réseau seront des liaisons série binaires.

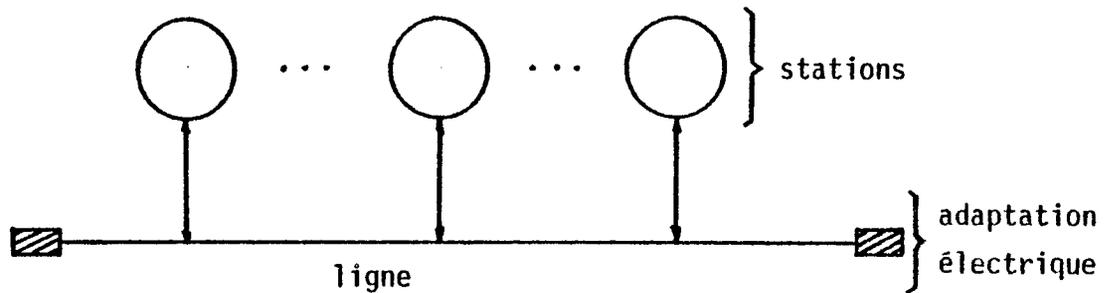
Notre étude portera sur les deux derniers niveaux. Le mécanisme de communication formé par ces deux derniers niveaux doit satisfaire aux conditions énoncées ci-dessus au § 2.1.1. Il doit fournir aux niveaux supérieurs des outils suffisamment puissants pour que la conception du protocole fonctionnel soit facilitée en la rendant le plus possible indépendante de la réalisation matérielle.

Un intérêt particulier sera accordé à l'aspect modularité.

On s'efforcera de définir un mécanisme de communication entièrement réparti sur les stations du réseau.

## 2.2. MÉCANISME D'AUTO-ALLOCATION

Dans ce qui suit on s'intéresse aux réseaux locaux où le dialogue se réalise sur une ligne multipoint partagée par les stations du réseau.



Au niveau matériel, le mécanisme de communication présente deux parties :

- . allocation de la ligne,
- . échange effectif de l'information.

Comme on se place dans l'optique d'une architecture répartie, on cherchera des solutions modulaires où les stations ont toutes les mêmes moyens de dialogue.

Au niveau matériel, un mécanisme d'allocation complètement distribué entre les stations du réseau est le mécanisme dit *d'auto-allocation*. Sa principale caractéristique est de réaliser (sous certaines conditions) en même temps et sur la même ligne, les fonctions d'allocation et d'échange effectif de l'information.

Dans tout ce qui suit on considère une ligne série.

### 2.2.1. Principes de l'auto-allocation

Considérons une ressource partagée par plusieurs processus utilisateurs. Si la ressource est utilisée par plusieurs processus en même temps, il se peut qu'elle ne rende pas le service qu'un processus particulier a demandé. Il faudrait donc que chaque processus dispose d'un mécanisme de vérification de la cohérence entre le service rendu et le service demandé. Si un processus détecte une incohérence, il doit évidemment redemander le service.

Il est clair qu'avec ce seul mécanisme de vérification, les performances du système sont faibles, puisque rien n'empêche un processus 1 de commencer l'utilisation de la ressource quand un processus 2 est déjà en train de l'utiliser (ce qui peut entraîner une incohérence au niveau du processus 2, d'où la nécessité pour lui de recommencer, ...), ceci pouvant en outre entraîner des incohérences au niveau global du système.

Pour éviter ceci, il faut munir la ressource d'un mécanisme d'allocation afin de *limiter* l'accès à la ressource. Chaque processus utilise la ressource seulement si l'allocateur lui en donne la permission.

L'allocateur "parfaitement" sûr, ce qui est le cas le plus répandu, assure au processus (ayant la permission d'utilisation) le bon service de la part de la ressource (les autres processus ne gêneront en rien). Ceci fait que dans ce cas, le mécanisme de vérification devient superflu.

Si l'allocateur est "imparfait", cette assurance n'est pas donnée et le mécanisme de vérification est nécessaire.

On aura ainsi deux mécanismes complémentaires :

- . allocateur (imparfait) : il permet ou non à un processus de commencer l'utilisation de la ressource ;
- . mécanisme de vérification : il entre en jeu pendant l'utilisation effective de la ressource et fait qu'un processus arrête l'utilisation s'il détecte une incohérence entre le service rendu et le service demandé (il y a quelqu'un d'autre qui est concurrent à l'utilisation) et ré-essaiera plus tard.

Ces considérations s'appliquent pour la ligne multipoint régie par l'auto-allocation :

- . ressource : ligne multipoint,
- . utilisation : émission d'un message,
- . service demandé : véhiculer un bit sur la ligne,
- . service rendu : information réellement véhiculée par la ligne.

Les deux mécanismes complémentaires de l'auto-allocation sont :

- . détection de "ligne libre" (allocateur "imparfait"),
- . détection de "conflit" (mécanisme de vérification).

Ces deux types de détection sont au niveau de chaque station, ce qui assure la modularité du mécanisme d'auto-allocation.

### 2.2.1.1. Détection de ligne libre

On définit deux états de la ligne :

- . ligne libre : personne n'émet sur la ligne et donc une nouvelle émission est permise (peut commencer),
- . ligne occupée : au moins une station émet sur la ligne et toute nouvelle émission est interdite.

Toute station qui veut émettre teste la ligne et commence à émettre seulement si elle a trouvé "ligne libre".

Dès qu'une station commence à émettre, la ligne devient "occupée", ce qui entraîne : si plusieurs stations émettent sur la ligne, alors elles ont dû commencer au même instant.

N.B. Tout ceci permet d'avoir le mécanisme de détection de panne suivant : si pendant l'émission une station détecte "ligne libre", alors il y a une panne dans le mécanisme de communication.

Définissons concrètement l'état "ligne libre". La ligne est libre quand personne n'émet sur la ligne et dans ce cas elle se positionne à une certaine valeur que l'on notera avec REPOS.

Exemple :

- . circuits d'émission en collecteur ouvert : REPOS = "1",
- . circuits d'émission en émetteur-suiveur : REPOS = "0".

On considèrera seulement les modes de transmission qui vérifient l'assertion suivante :

en cours d'émission d'un message, la durée pendant laquelle la ligne a la valeur REPOS d'une manière continue, est bornée supérieurement.

N.B. Cette hypothèse ne constitue pas une forte restriction, puisque pour tous les modes de transmission, à l'exception du NRZ, il y a une durée maximale entre deux changements consécutifs de polarité de la ligne et ceci pour que les stations réceptrices puissent se resynchroniser.

Notons par TOCC cette borne supérieure. On a l'énoncé :

la ligne est libre si elle a la valeur REPOS d'une manière continue pendant au moins la durée TOCC.

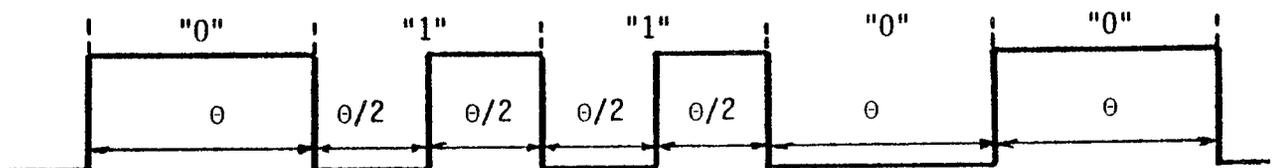
Remarque : On a dit que la ligne devient "occupée" dès qu'une station commence à émettre. Pour ceci il faut donc que toute station commence l'émission par la valeur  $\overline{\text{REPOS}} \neq \text{REPOS}$ .

On va déterminer la durée TOCC dans quelques cas.

#### a) Emission en code bi-phase

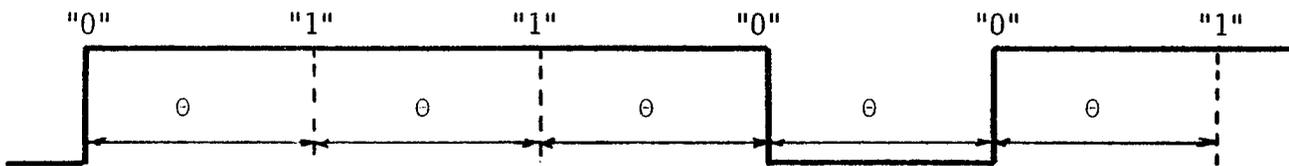
Si  $\theta$  est la durée du bit, alors pendant l'émission la durée entre deux changements consécutifs de polarité est au maximum de  $\theta$ .

Il faut donc prendre  $\text{TOCC} > \theta$ , par exemple  $\text{TOCC} = 2.\theta$ .



b) Emission en code NRZI et format SDLC des messages

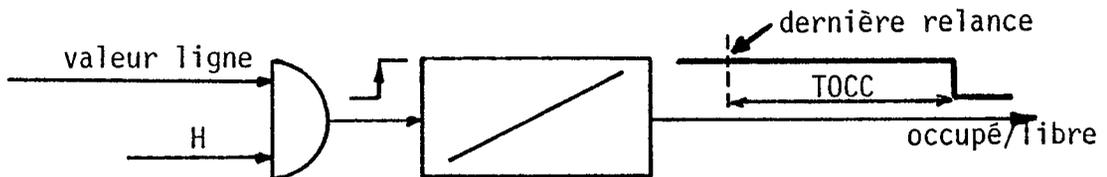
En NRZI le "0" se traduit par un changement de polarité de la ligne et le "1" par pas de changement.



Par ailleurs, dans le format SDLC, à l'émission a lieu l'insertion d'un "0" après toute succession de cinq "1", en dehors des flags qui contiennent six "1" consécutifs. Ceci fait qu'avec le format SDLC et le code NRZI de transmission, la durée entre deux changements de polarité de la ligne est au maximum  $7.\theta$  où  $\theta$  = durée du bit.

Il faut donc prendre  $TOCC > 7.\theta$ , par exemple  $TOCC = 8.\theta$ .

La détection de ligne libre se réalise très simplement à l'aide d'un monostable redéclenchable. Le monostable est relancé en permanence si la ligne n'est pas au repos. S'il n'est pas relancé, sa sortie tombe à zéro au bout de la durée  $TOCC$ , ce qui signifie "ligne libre".



N.B. H=horloge rapide (d'échantillonnage)

FIG.6. Détection de ligne libre

### 2.2.1.2. Détection de conflit

La détection de conflit est le mécanisme de vérification de la cohérence entre le service demandé (information émise) et le service rendu (information véhiculée par la ligne).

Le conflit est la situation où plusieurs stations envoient (en même temps) des informations différentes sur la ligne. Dans ce cas, il est évident qu'il existe au moins une station pour laquelle il y a incohérence entre l'information émise et l'état de la ligne : elle détecte un conflit.

Toute station qui détecte un conflit s'isole de la ligne et cesse d'émettre (elle ré-essaiera plus tard).

Il est évident qu'une station qui ne détecte pas de conflit continue à émettre.

Si TX est l'information émise et RX la valeur de la ligne, il y a conflit si  $TX \neq RX$ .

En réalité, même si la station est seule à émettre, elle détectera  $TX \neq RX$  (pendant une courte durée) chaque fois que TX change de valeur et ceci à cause du temps mis par le signal pour traverser les couches logiques, parcourir les fils, ... On appelle ce conflit "conflit physique", à distinguer du conflit logique dû à un conflit entre les stations.

Il faut filtrer les conflits parasites dûs uniquement au changement de la valeur émise.

A et B étant respectivement les points de fabrication de TX et RX, on va noter par TPOS le temps mis par le signal de A à B (figure 6). Il est clair qu'un conflit qui dure moins de TPOS est un conflit parasite.

Ceci nous amène à dire qu'en général un conflit physique est un vrai conflit (conflit logique) et on cesse d'émettre si ce conflit dure plus qu'une certaine durée notée TCNF. Pour éliminer les conflits parasites, il faut prendre  $TCNF \geq TPOS$ . Comme on le verra plus loin, la valeur de TCNF est essentielle pour le choix d'une stratégie précise d'auto-allocation.

Remarque : du point de vue de la sécurité, la détection de conflit peut être vue comme étant le mécanisme d'accusé de réception au niveau de l'échange élémentaire. L'accusé de réception est identique à l'information émise. Si l'accusé arrive (pas de conflit), on continue à émettre sinon (conflit détecté), on arrête l'émission (et on ré-essaie plus tard).

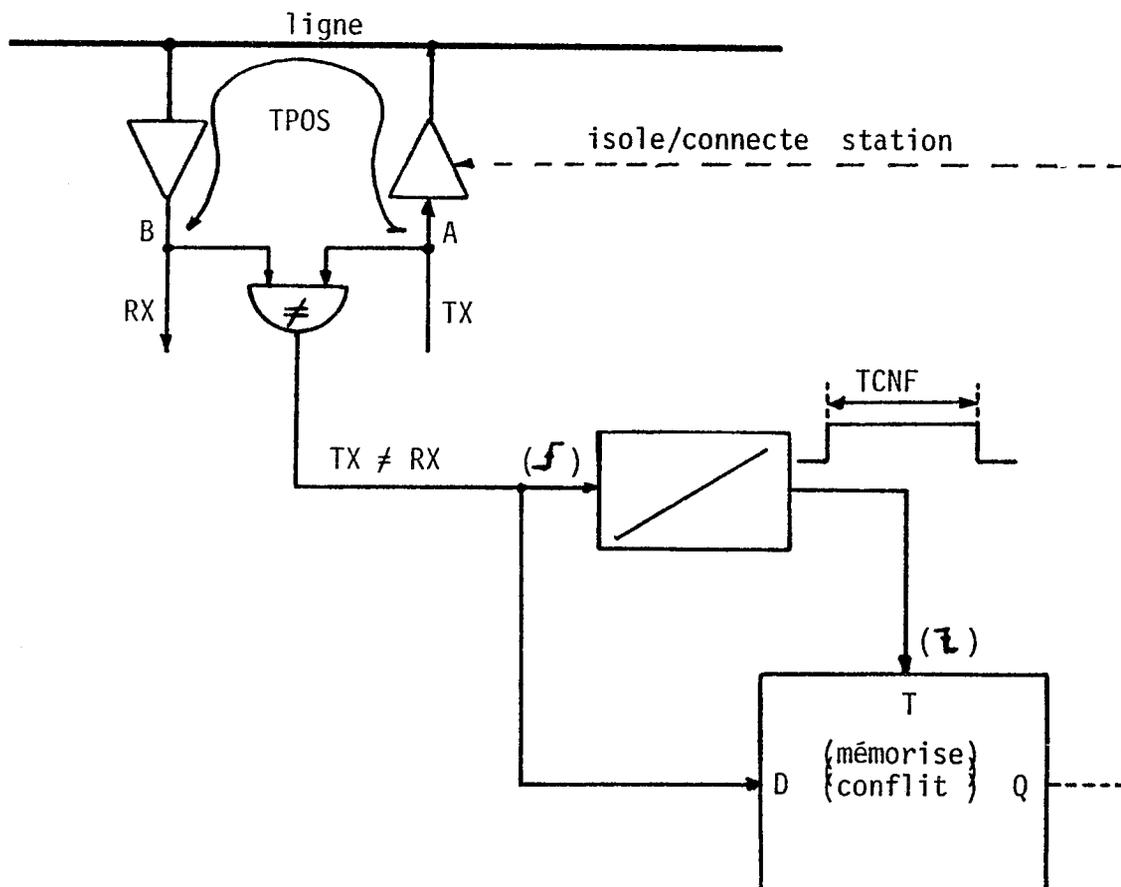


FIG.7. Détection du conflit

### 2.2.1.3. Algorithme de l'auto-allocation

Les deux mécanismes de détection présentés ci-dessus sont complémentaires :

- . la détection de ligne libre permet de réaliser la politique : premier arrivé, premier servi ;
- . la détection de conflit permet de choisir quand il y a plusieurs "premiers arrivés".

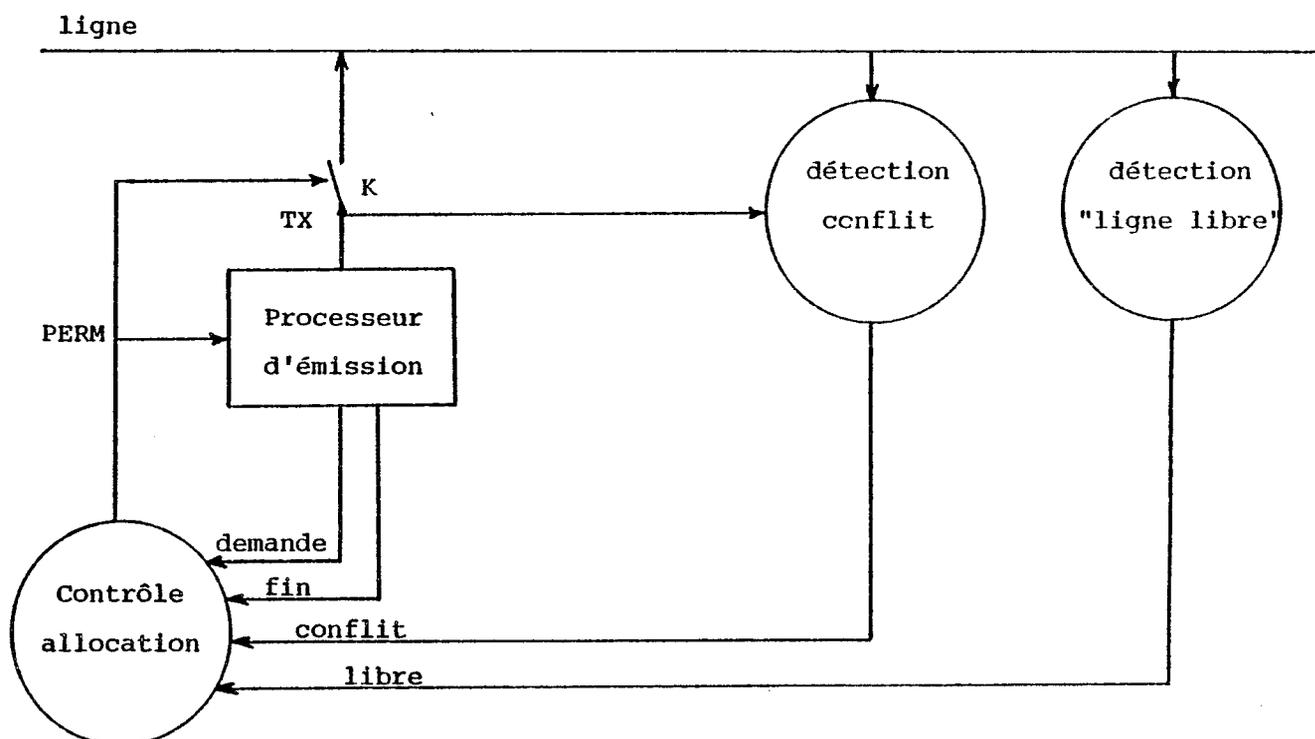


FIG.8. Schéma de principe de l'auto-allocation

N.B. Le contact fonctionnel K se réalise physiquement par l'amplificateur de ligne. Sa fermeture/ouverture se réalise par la mise en état off/on de l'amplificateur.  
PERM = 1  $\Leftrightarrow$  émission permise (amplificateur en état on).

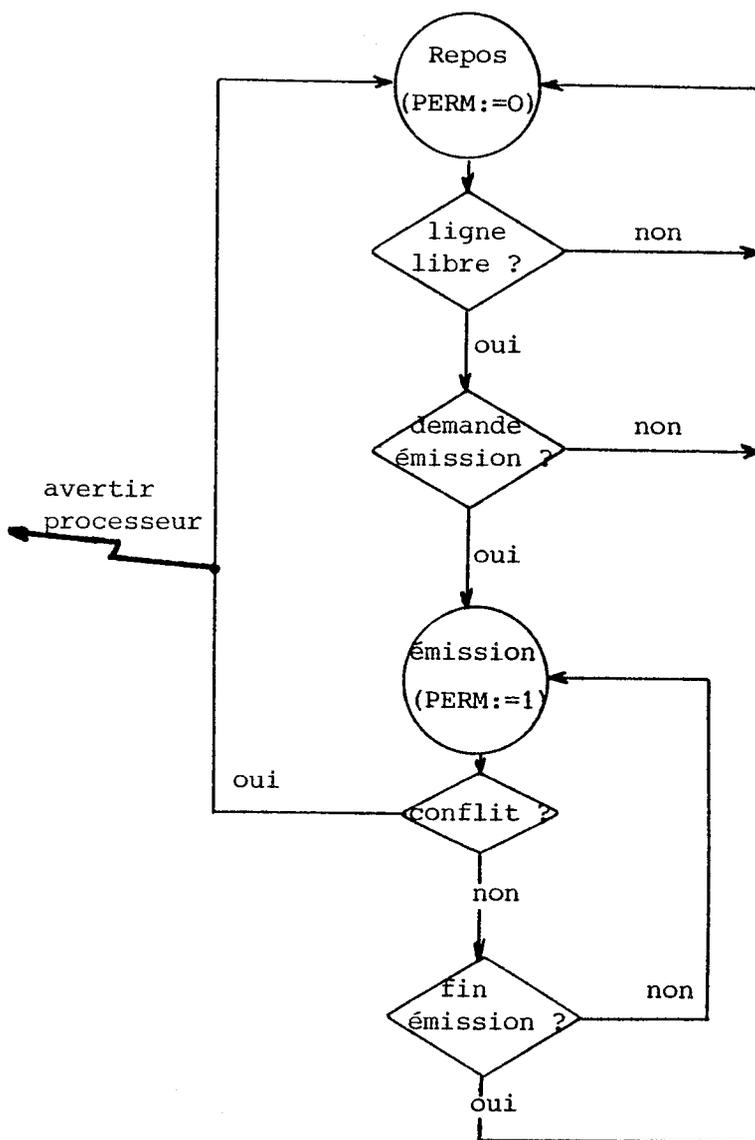


FIG.9. Algorithme de l'auto-allocation

Regardons l'algorithme complet pour l'émission d'un message.

Quand le processeur d'émission veut émettre un message, il va faire une demande au mécanisme d'auto-allocation. Dès que la permission lui est accordée (PERM := 1), il va commencer à émettre son message (bits en série). Si un conflit est détecté en cours d'émission, le mécanisme d'auto-allocation va retirer la permission (PERM := 0) en avertissant l'émission. Le processeur d'émission devra se réinitialiser (pour ré-émettre le même message) et faire une nouvelle demande au mécanisme d'auto-allocation. Si l'émission se déroule sans incidents (sans conflit), le processeur d'émission avertit l'auto-allocation à la fin de l'émission et l'auto-allocation va retirer la permission (PERM := 0) en le déconnectant ainsi de la ligne.

D'après l'algorithme ci-dessus, on voit qu'en cas de conflit, la station concernée doit ré-initialiser son coupleur d'émission. Donc le taux de conflits a une influence, dans le sens de l'augmentation, sur la charge du système. Si l'augmentation de la charge, due au taux de conflits, s'avère gênante, alors il faut diminuer le nombre de conflits. On peut réaliser cela si le mécanisme d'auto-allocation fait le test "ligne libre" avec une certaine période TTEST pour diminuer la probabilité que plusieurs stations fassent en même temps le test de "ligne libre".

Un autre intérêt de la période de test est la possibilité de réaliser une priorité entre les stations. On munira les stations plus prioritaires d'une période TTEST petite et les stations moins prioritaires d'une période TTEST grande.

### 2.2.2. Types d'auto-allocation

Les principes de l'auto-allocation étant définis, il se pose le problème de l'influence de l'auto-allocation sur la transmission des données.

Suivant le comportement physique de la ligne quand plusieurs stations émettent en même temps, on distingue deux cas :

**a) l'état de la ligne correspond toujours à une des informations émises par les stations :**

cette situation s'obtient par la transmission en bande de base.

Ce cas induit ce que l'on appellera "auto-allocation déterministe", puisque, comme on le verra plus loin, si plusieurs stations émettent en même temps (concurrentes à l'émission), il existe toujours une station qui n'est pas rejetée et écoule son message en entier.

**b) l'état de la ligne ne correspond à aucune des informations émises par les stations :**

ce cas arrive dans la transmission par modulation [MET].

L'auto-allocation induite est nommée "auto-allocation probabiliste".

Si plusieurs stations *émettent en même temps*, toute station détectera à son niveau local un conflit (puisque l'état de la ligne ne correspond à aucune des informations émises). Par ailleurs, les stations réceptrices feront une mauvaise réception (n'associeront pas une information correcte à l'état de la ligne).

Donc, en cas de conflit, il faut que *toutes les stations* émettrices cessent d'émettre (et même les forcer à cesser), puisque si au moins une station continue, on est sûr que son message sera mal reçu.

Ainsi voit-on que les performances de la communication dépendent entièrement du taux des conflits, ce qui entraîne une baisse importante quand le système est chargé.

Une quelconque station ne réussit à écouler correctement son message que si au début de l'émission elle est seule à émettre (à cet instant elle est seule à avoir vue "ligne libre") : la ligne devenant "occupée", aucune autre station ne pourra émettre pendant ce temps.

Pour réaliser ceci, les stations essaient (ou ré-essaient en cas de conflit) d'émettre (en s'assurant d'abord que la ligne est libre) de temps en temps, l'intervalle entre deux essais étant aléatoire.

On a vu qu'en cas de conflit, il faut que toutes les stations cessent d'émettre, d'où le principe :

si une station détecte (à son niveau local) un conflit, elle devra cesser d'émettre. Mais, pour que les autres stations cessent aussi, cette station continuera à émettre pendant une courte durée  $t$ , durée suffisante pour que le conflit devienne visible pour toutes les stations.

L'intérêt de l'auto-allocation probabiliste réside dans son applicabilité même quand le support physique n'est pas une ligne (câble) : il peut être constitué par des ondes dans l'espace.

N.B. Dans le reste de notre exposé, on considèrera seulement *l'auto-allocation déterministe*.

### 2.2.3. Comportement idéal de l'auto-allocation

On se place dans l'hypothèse suivante : si plusieurs stations émettent en même temps, l'état de la ligne correspond toujours à une des informations émises. Autrement dit, si  $S_1, \dots, S_m$  sont les informations émises en même temps par plusieurs stations, l'information  $L$  associée à l'état de la ligne sera :

$$L = f(S_1, \dots, S_m) \text{ et } L \in \{S_1, \dots, S_m\}$$

Exemples :

- . circuits d'émission en collecteur ouvert :  $f = \text{"et"}$  logique,
- . circuits d'émission en émetteur-suiveur :  $f = \text{"ou"}$  logique.

On négligera pour l'instant le temps de propagation sur la ligne.

#### Corollaire :

quand il y a conflit au niveau global du système, il existe au moins une station pour laquelle, à son niveau local, il n'y a pas de conflit et qui continue donc à émettre.

En effet, en cas de conflit au niveau global, l'état  $L$  de la ligne correspond à une des informations émises,  $L = S_i$  (conformément à notre hypothèse). Les stations qui ont émis  $S_i$  ne verront pas de conflit, tandis que les autres le détecteront et seront rejetées.

Conformément au mécanisme de détection de ligne libre, si plusieurs stations émettent en même temps, elles ont dû commencer au même instant. La ligne étant occupée, il en résulte qu'aucune autre station, à part celles considérées, ne peut plus émettre.

Il en résulte que si la ligne est occupée, il existe au moins une station qui réussit à écouler son message.

En effet, le cas contraire signifie qu'il y a un conflit qui élimine les dernières stations restées en course, ce qui est en contradiction avec le corollaire démontré plus haut.

Par ailleurs, le message écoulé sera un message correct : le message qui transite par la ligne correspond exactement au message émis du fait que la station en question n'a pas détecté de conflit.

### Observation :

Un cas particulier est celui des messages identiques. Si deux stations ayant le même message à émettre commencent l'émission en même temps, aucune station ne détectera de conflit. En effet, si  $S_1$  et  $S_2$  sont les informations émises par les deux stations, on a  $S_1 = S_2$  à tout instant (puisque les messages sont identiques). L'état  $L$  de la ligne doit satisfaire (conformément à l'hypothèse)  $L \in \{S_1, S_2\}$  et comme  $S_1 = S_2$ , il en résulte  $L = S_1 = S_2$  à tout instant et donc aucune des stations ne détecte de conflit. Si cette situation s'avère gênante, le protocole de communication doit interdire les messages identiques. Pour cela, on associe aux stations des adresses distinctes et tout message contient au début l'adresse de la station émettrice. Ceci ne constitue pas une forte restriction, puisque en pratique les protocoles de communication incluent l'adresse de l'expéditeur dans le message pour que le destinataire sache à qui répondre. Cette politique a en plus l'avantage de permettre la détection de certaines pannes : si après l'émission de sa propre adresse, une station détecte un conflit quand logiquement elle devrait être seule à émettre, il est clair qu'une panne est apparue dans le mécanisme de communication.

L'analyse du comportement de l'auto-allocation nous amène à conclure que l'emploi de la ressource "voie de communication" est optimisé :

- . la ligne n'est jamais occupée inutilement : quand elle est occupée, il y a toujours échange d'information,
- . la ligne n'est jamais libre inutilement : dès qu'une station a quelque chose à émettre, elle occupe la ligne (si elle n'était pas déjà occupée).

Le coût de l'auto-allocation est déterminé par la durée pendant laquelle personne n'émet sur la ligne, même si des stations ont à émettre. Cette durée est le temps TOCC nécessaire pour déterminer le passage de occupé à libre de la ligne.

Le coût relatif par message de l'auto-allocation peut être donné par le rapport :  $\frac{\text{TOCC}}{\text{durée message}}$

Notons par  $Q$  la quantité d'information échangée dans le réseau pendant une certaine durée de temps. Soit  $n$  le nombre de messages dont  $Q$  est constituée. Pour ces  $n$  messages, le coût de l'auto-allocation est  $n \cdot \text{TOCC}$ . Pour échanger la quantité d'information  $Q$ , la ligne met le temps  $Q \cdot \theta$  où  $\theta$  = durée de l'information élémentaire.

Le coût relatif à  $Q$  de l'auto-allocation est  $\frac{n \cdot \text{TOCC}}{Q \cdot \theta}$

Ce coût diminue si  $n$  diminue. On a donc intérêt à échanger une quantité  $Q$  d'information en diminuant le nombre de messages (tout en augmentant leur taille).

### Observation :

Suivant la fonction  $f$  réalisée par la ligne, on peut déterminer la station "choisie" par l'auto-allocation. Prenons par exemple le cas où  $f = \text{"ou"}$  logique. Supposons plusieurs stations qui commencent à émettre en même temps et soient  $m_1, \dots, m_n$  leurs messages. En cas de conflit (des stations envoient "0", d'autres envoient "1"), la ligne se positionne à "1", les stations émettant "0" détecteront un conflit et seront rejetées, tandis que les autres continueront (jusqu'au prochain conflit).

Ceci veut dire que la station "choisie" (qui écoulera son message en entier) sera celle dont le message (complété éventuellement à droite pour avoir la même longueur que les autres) est le plus grand en tant que nombre représenté en binaire.

D'une manière analogue, quand  $f = \text{"et"}$  logique, la station "choisie" est celle dont le message (complété éventuellement à droite) est le plus petit en tant que nombre.

Comme les valeurs de messages (en tant que nombres) sont généralement équiprobables, on pourra prendre pour le mécanisme le modèle suivant : un serveur qui "choisit" au hasard une station parmi les stations ayant quelque chose à émettre.

#### 2.2.4. Comportement réel de l'auto-allocation. Calcul du débit

En considérant le comportement idéal de la ligne, on a vu les propriétés intéressantes de l'auto-allocation.

Il s'agit maintenant de voir sous quelles conditions on obtient ces propriétés avec une ligne réelle.

Pour une ligne réelle on a principalement :

- . réflexion du signal : ce phénomène est éliminé par l'adaptation électrique en bouts de ligne (avec la résistance caractéristique) ;
- . affaiblissement du signal : ce phénomène limite la longueur de la ligne ;
- . temps de propagation du signal sur la ligne : on va considérer la durée entre la génération du signal (avant l'interface d'émission) par la station émettrice et sa réception (après l'interface de réception) par la station réceptrice.

Le temps de propagation du signal sera composé de :

- . temps de propagation linéaire ; si  $\delta$  est la propagation linéaire unitaire, alors pour une longueur  $l$  la propagation sera  $l.\delta = \tau$
- . temps de montée (descente) du signal ;
- . temps TRAV pour traverser les interfaces d'émission/réception.

Pour une réalisation donnée, le temps de propagation en fonction de la longueur sera une fonction notée  $g(x)$ .

N.B. Si l'on néglige le temps de montée (descente), on obtient :  
 $g(x) = x.\delta + \text{TRAV}.$

En tenant compte du temps de propagation, on va prendre pour la ligne le modèle logique suivant :

Soient  $E_1, \dots, E_m$  les stations émettrices qui émettent en même temps. En un point quelconque  $X$  de la ligne, l'information associée à l'état de la ligne est donnée par :

$$R_X(t) = f(S_1(t - g(d_1)), \dots, S_m(t - g(d_m)))$$

où  $S_1, \dots, S_m$  = informations émises respectivement par  $E_1, \dots, E_m$

$d_1, \dots, d_m$  = distances entre respectivement  $E_1, \dots, E_m$  et  $X$

$g$  = temps de propagation

$f$  = fonction dépendant de la réalisation physique, mais respectant

$f(S_1, \dots, S_m) \in \{S_1, \dots, S_m\}$ , autrement dit, l'état de la ligne correspondant toujours à une des informations émises.

Dans la suite, on considèrera que toutes les stations ont la même fréquence d'émission (même période) : les stations ont des horloges d'émission indépendantes, de même fréquence, mais déphasées. Ceci est évidemment nécessaire pour que les stations réceptrices fassent une bonne réception quel que soit l'émetteur.

Il faut déterminer la durée TCNF détectrice de conflit, ainsi que la durée  $\theta$  du bit.

La durée  $\theta$  du bit est fonction de la dispersion  $p$  causée par l'auto-allocation. Une bonne reconnaissance de l'information par les récepteurs impose  $\theta > 2.p$  (voir annexe 3 pour plus de détails). Le calcul détaillé de la dispersion  $p$  est donné en annexe 2.

N.B. dispersion = |durée impulsion reçue - durée théorique|.

Sans restreindre la généralité, on étudie le cas suivant :

- . deux stations A et B émettent en même temps,
- . la fonction  $f$  réalisée par la ligne est le "ou" logique :  
si  $S_A, S_B$  sont les informations émises, on a :  $f = \max (S_A, S_B)$ .

Il est clair que l'auto-allocation introduit une dispersion seulement si les deux stations émettent en même temps des informations identiques (mais déphasées). Quand les informations sont différentes, une des stations détectera un conflit, cessera d'émettre et une seule station continuera à émettre, donc on n'aura plus de dispersion causée par l'auto-allocation.

Soit  $d$  la distance entre A et B. Considérons le début d'une émission : A teste la ligne, la voit libre, commence à émettre et B fait la même chose (figure 10). Soit  $\epsilon$  le déphasage entre A et B : B a un retard  $\epsilon$  par rapport à A. La station B teste la ligne, la trouve libre et commence à émettre avant que le front émis par A arrive en B, autrement, B verrait la ligne occupée. Il en résulte que  $\epsilon \leq g(d)$ .

Conformément au chronogramme de la figure 10 on a :

- . A voit un conflit physique pendant  $C_A = g(d) + \epsilon$
- . B voit un conflit physique pendant  $C_B = g(d) - \epsilon$

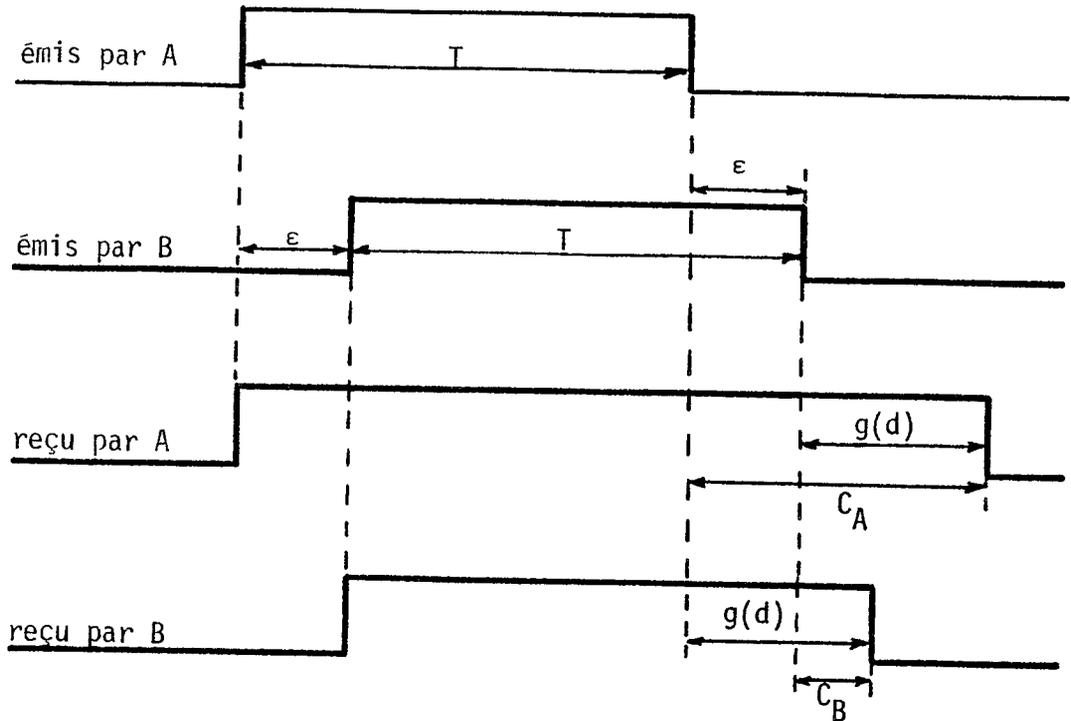


FIG.10. Signaux en début d'émission

Trois cas sont possibles :

a)  $C_A \leq \text{TCNF}$ ,  $C_B \leq \text{TCNF}$  :

A et B ne détectent pas de conflit et continuent ensemble ;

b)  $C_A > \text{TCNF}$ ,  $C_B \leq \text{TCNF}$  :

A détecte un conflit et cesse d'émettre, B ne détecte pas de conflit et continue ;

c)  $C_A > \text{TCNF}$ ,  $C_B > \text{TCNF}$  :

A et B détectent un conflit et cessent d'émettre (la ligne a été occupée inutilement).

On voit que le choix de la durée TCNF est très important.

Dans l'annexe 2 on montre qu'à tout moment de l'émission pour l'ensemble des stations la dispersion maximale est :

$$(I) \begin{cases} p = \epsilon + g(d) - g(0) & \text{si } g \text{ concave} \\ p = \epsilon + g(1) - g(1-d) & \text{si } g \text{ convexe} \end{cases}$$

Suivant le choix de la durée TCNF détectrice de conflit, on peut dégager les stratégies d'auto-allocation suivantes :

**a. L'auto-allocation choisit toujours une station**

Autrement dit, il y a toujours une station qui n'est pas rejetée. Pour cela il faut que le cas c) soit interdit, donc on doit toujours avoir  $C_B \leq \text{TCNF}$ , ou encore  $g(d) - \epsilon \leq \text{TCNF}$ . Comme  $d \leq 1$  ( $1 =$  longueur de la ligne) et  $\epsilon \geq 0$ , il en résulte que  $\text{TCNF} \geq g(1)$ . On prendra  $\text{TCNF} = g(1)$ .

Suivant la signification donnée à la première impulsion émise, on distingue deux cas :

**a.1. La première impulsion est support d'information significative pour la station réceptrice**

On a  $d \leq 1$  et  $\epsilon \leq g(d) \leq g(1)$  et en tenant compte des formules (I) ci-dessus, on obtient que la dispersion sur la première impulsion est égale à :

$$p = 2.g(1) - g(0)$$

Comme  $\theta$  doit satisfaire  $\theta > 2.p$ , on obtient

$$\theta > 4.g(1) - 2.g(0)$$

N.B. Négligeant le temps de montée (descente), on obtient :

$$\theta > 4.\tau + 2.\text{TRAV}$$

où  $\tau = 1.\delta$  ( $\delta =$  propagation linéaire)

TRAV = temps pour traverser les interfaces.

### a.2. La première impulsion n'est pas support d'information significative

Cette première impulsion est utilisée uniquement pour l'auto-allocation. Si après cette première impulsion, plusieurs stations restent en course, elles doivent satisfaire au cas a), donc  $C_A \leq \text{TCNF}$ , ou encore  $g(d) + \epsilon \leq \text{TCNF}$ .

En tenant compte de cette relation, on montre dans l'annexe 2, que la dispersion sur les bits suivant la première impulsion est égale à :

$$\begin{aligned} p &= \text{TCNF} - g(0) && \text{si } g \text{ concave} \\ p &= \text{TCNF} + g(1) - 2g(1/2) && \text{si } g \text{ convexe} \end{aligned}$$

On doit avoir  $\theta > 2p$ , donc :

$$\begin{aligned} \theta &> 2(\text{TCNF} - g(0)) && \text{si } g \text{ concave} \\ \theta &> 2(\text{TCNF} + g(1) - 2g(1/2)) && \text{si } g \text{ convexe} \end{aligned}$$

et comme on a pris  $\text{TCNF} = g(1)$  il en résulte :

$$\begin{aligned} \theta &> 2(g(1) - g(0)) && \text{si } g \text{ concave} \\ \theta &> 4(g(1) - g(1/2)) && \text{si } g \text{ convexe} \end{aligned}$$

N.B. En négligeant le temps de montée (descente), on obtient :

$$\theta > 2.\tau$$

Si l'on veut obtenir une fréquence d'émission plus élevée, on adopte la stratégie suivante :

#### a.2.1. Auto-allocation en deux étapes

A chaque station on associe une priorité distincte des autres. Dans une première étape, on envoie les bits de la priorité avec une période  $\theta$  comme pour la stratégie a.2. A la fin de cette étape, une seule station reste en course puisque les priorités sont distinctes. Dans une deuxième étape, la seule station qui est restée émet son message à la fréquence supportée par la ligne, vu que l'auto-allocation n'introduit plus de dispersion.

N.B. Si l'on veut que toutes les stations soient équivalentes, on rend les priorités dynamiques. Par exemple, si  $n$  est le nombre de stations du réseau, chaque fois que la ligne devient libre, on fait pour chaque station :

$$\text{priorité} \leftarrow \text{priorité} + 1 \text{ (modulo } n\text{)}.$$

b. L'auto-allocation peut ne choisir personne

A la suite de la première impulsion, il est possible que toutes les stations détectent un conflit et donc personne ne continue. Autrement dit, le cas c) est permis. Ceci veut dire que l'on a choisi  $TCNF < g(1)$  (autrement on a une stratégie de type a.). Si à la suite de la première impulsion il y a plusieurs stations qui continuent, alors elles satisfont au cas a), c'est-à-dire  $C_A \leq TCNF$ . Si l'on note par  $s$  la distance telle que  $g(s) = TCNF$ , on montre dans l'annexe 2 qu'après la première impulsion, la dispersion est égale à :

$$\left. \begin{array}{l} \cdot \text{ si } g \text{ est convexe :} \\ \quad p = TCNF + g(1) - g(1-s) - g(s) \quad \text{si } s \leq 1/2 \\ \quad p = TCNF + g(1) - 2g(1/2) \quad \text{si } s \geq 1/2 \\ \cdot \text{ si } g \text{ est concave :} \\ \quad p = TCNF - g(0) \end{array} \right\}$$

d'où on tire la durée du bit ( $\theta > 2p$ ).

Cette stratégie se rapproche beaucoup de l'auto-allocation probabiliste. Les performances qu'elle entraîne (utilisation efficace de la ligne) dépendent du taux de conflits, ce qui constitue un désavantage par rapport aux stratégies de type a.

Parmi les stratégies présentées ci-dessus, le choix se fait suivant le débit que l'on veut écouler.

La stratégie b. permet une fréquence d'émission plus élevée que les autres stratégies, mais le débit utile est inférieur au débit théorique à cause des conflits. Les performances (débit utile) diminuent avec l'augmentation du taux de conflits, celui-ci augmentant avec la charge du système et le nombre de stations.

Pour les stratégies de type a., le débit utile de la ligne est égal au débit théorique (fréquence d'émission). Par ordre croissant de débit permis on a : stratégie a.1, stratégie a.2, stratégie a.2.1. La stratégie a.2.1. sera préférée, surtout pour les longues distances (proches d'un kilomètre) quand le débit du réseau ne peut pas être écoulé avec la stratégie a.2.

N.B. On a vu que suivant la stratégie choisie, le message peut être précédé de signaux qui servent uniquement à l'auto-allocation. On appelle ces signaux "signaux de candidature".

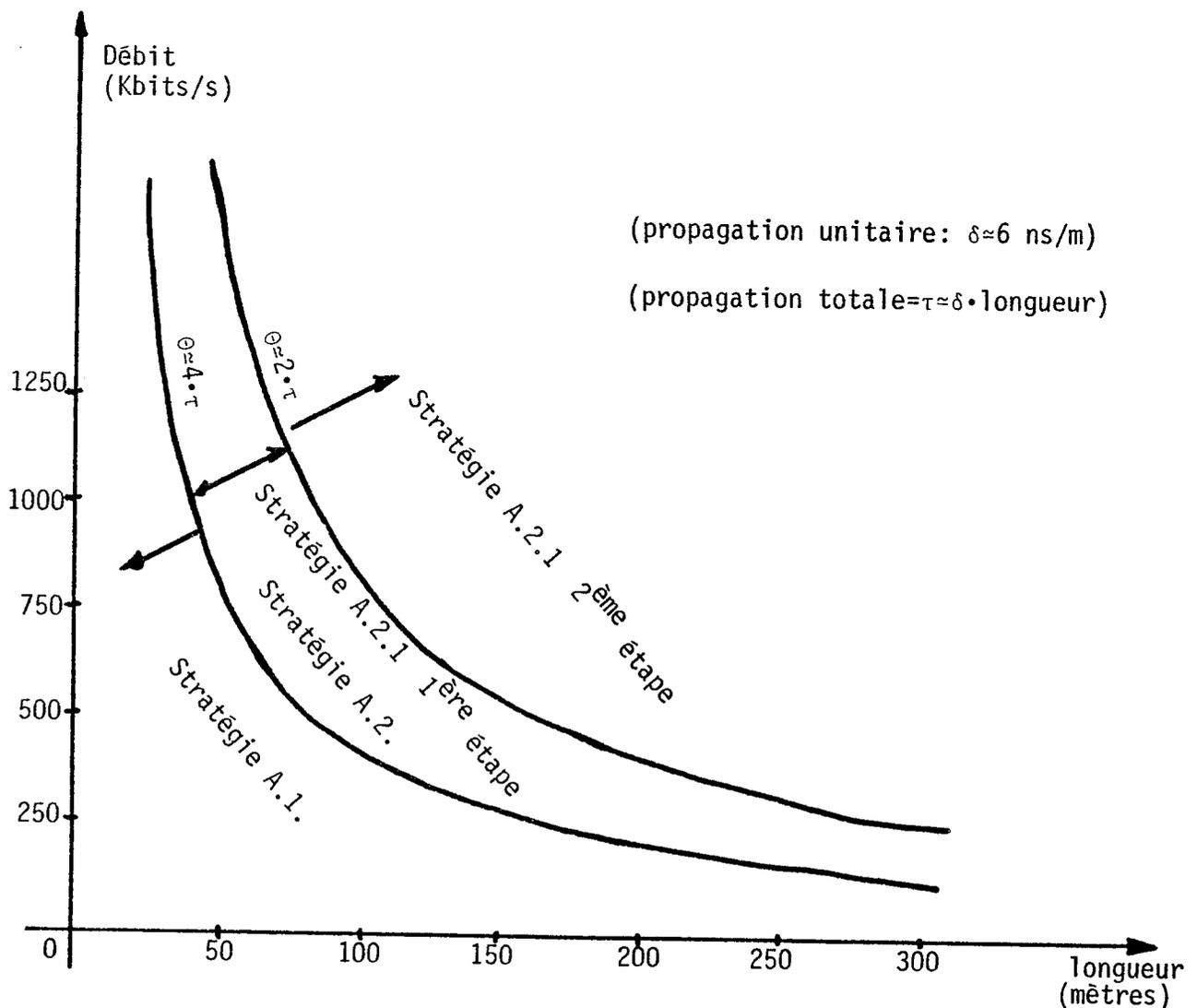


FIG.11. Estimation du débit permis par l'auto-allocation

### Observation importante

Dans l'estimation de la durée  $\theta$  du bit (et donc du débit), on n'a pas tenu compte des dispersions provoquées par la ligne et par les horloges. En tenant compte de ces dispersions, la relation qui donne la durée  $\theta$  est plus compliquée.

Avec une méthode de reconnaissance de l'information par échantillonnage, on montre dans les annexes 2 et 3 que la durée du bit, dans le cadre de la stratégie a.2 (qui nous intéresse plus particulièrement), doit satisfaire :

$$\theta > 2 \cdot \frac{\text{TCNF}' - g(0) + \mu + (1 + \alpha).h}{1 - (8N + 1)\alpha} \quad \text{si } g \text{ concave}$$

$$\theta > 2 \cdot \frac{\text{TCNF}' + g(1) - 2.g(1/2) + \mu + (1 + \alpha).h}{1 - (8N + 1)\alpha} \quad \text{si } g \text{ convexe}$$

où TCNF' = limite supérieure de la durée détectrice de conflit

g = temps de propagation

l = longueur de la ligne

h = période de l'horloge d'échantillonnage

$\alpha$  = précision de la période d'échantillonnage

N = nombre maximum de bits dont une impulsion peut être le support

(avec le format SDLC et codage NRZI, on a  $N = 7$ ).

$\mu$  = dispersion isochrone (introduite par la ligne).

### 2.2.5. Généralisation de l'auto-allocation

L'auto-allocation se généralise d'une manière naturelle au cas où le dialogue dans le réseau local se fait sur plusieurs lignes. Le réseau peut comporter plusieurs lignes pour pouvoir écouler le débit et pour des questions de tolérance aux pannes. Ces lignes sont banalisées : toute station peut émettre/recevoir sur n'importe quelle ligne. On appelle *autoroute* l'ensemble de ces lignes banalisées.

Le nombre  $k$  de lignes de l'autoroute est choisi comme suit :

- . on calcule le nombre  $k_1$  de lignes nécessaires pour écouler le débit :  
si  $D$  est le débit total d'informations échangées dans le réseau et  $\Delta$  le débit supporté par une ligne, il faut :  
$$k_1 > \frac{D}{\Delta}$$
- . on prévoit un nombre  $k_2$  de lignes supplémentaires suivant le nombre de pannes que l'on veut pouvoir tolérer.

On a  $k = k_1 + k_2$ .

Une première solution est d'avoir un mécanisme d'émission (coupleur) associé à chaque ligne, avec un mécanisme d'auto-allocation par ligne, la station répartissant les messages d'une manière aléatoire sur ses coupleurs d'émission.

Une deuxième solution est le mécanisme d'auto-allocation d'autoroute qui dérive des principes généraux de l'auto-allocation. A chaque ligne est associé un détecteur de ligne libre. Si une demande d'émission est faite, l'auto-allocation choisit une ligne libre (s'il y en a) sur laquelle aura lieu l'émission effective. Pendant l'émission, la détection de conflit se fait sur la ligne choisie. Si un conflit est détecté, on s'isole de la ligne et on avertit le processeur qui devra réinitialiser l'émission et refaire la demande.

Pour une utilisation optimale de l'utoroute, l'auto-allocation choisit au hasard une ligne parmi les lignes libres (s'il y en a).

N.B. Pour des questions de sécurité, il est possible que le processeur limite ce choix à un sous-ensemble des lignes de l'utoroute. Ceci permet de tester le fonctionnement du mécanisme de communication dans le but de localiser les pannes. Donc, à chaque demande, le processeur spécifiera l'ensemble des lignes sur lesquelles l'émission peut se faire. En fonctionnement normal, cet ensemble est constitué par l'utoroute tout entière.

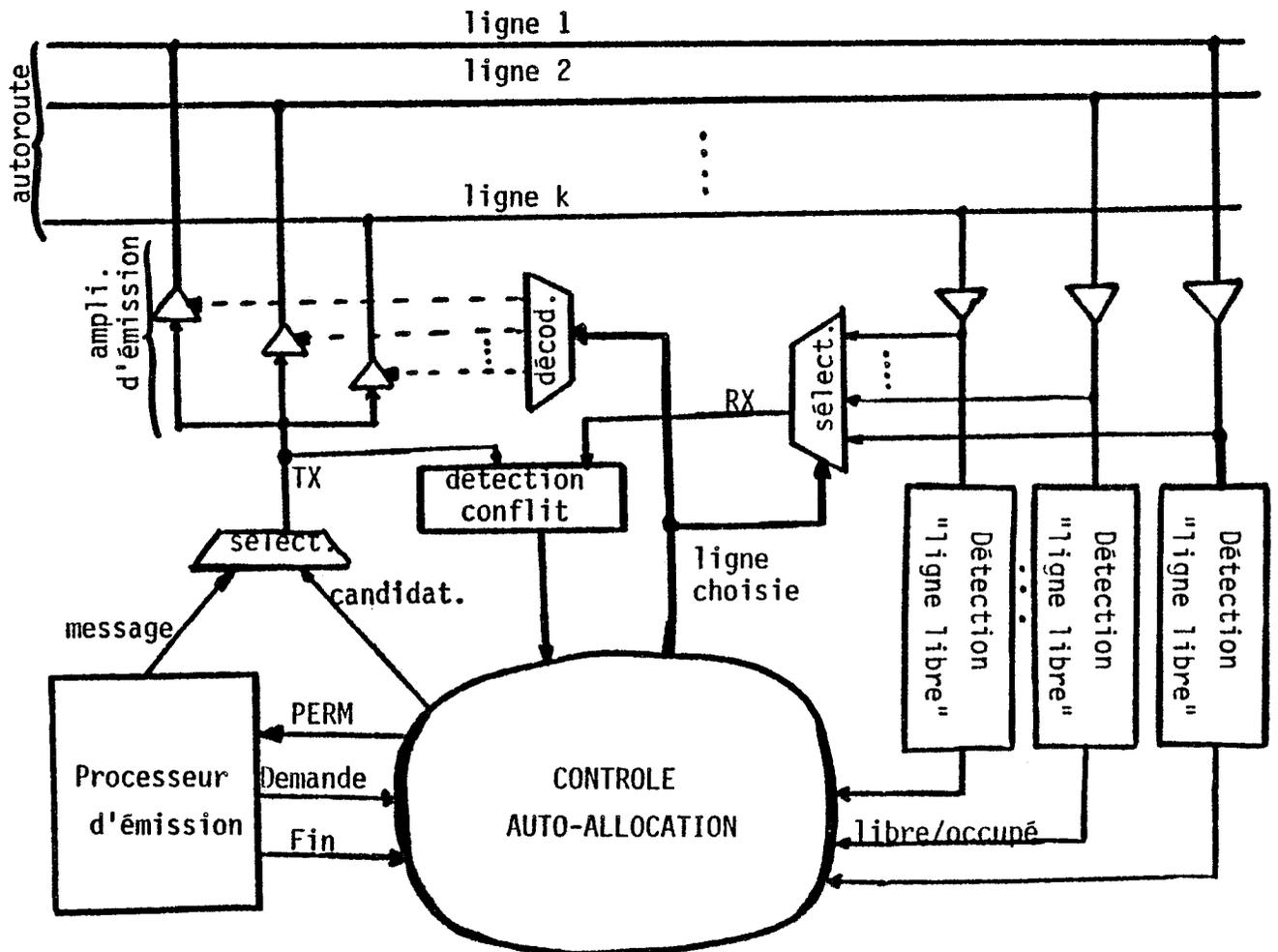


FIG.12. Schéma de l'auto-allocation d'utoroute

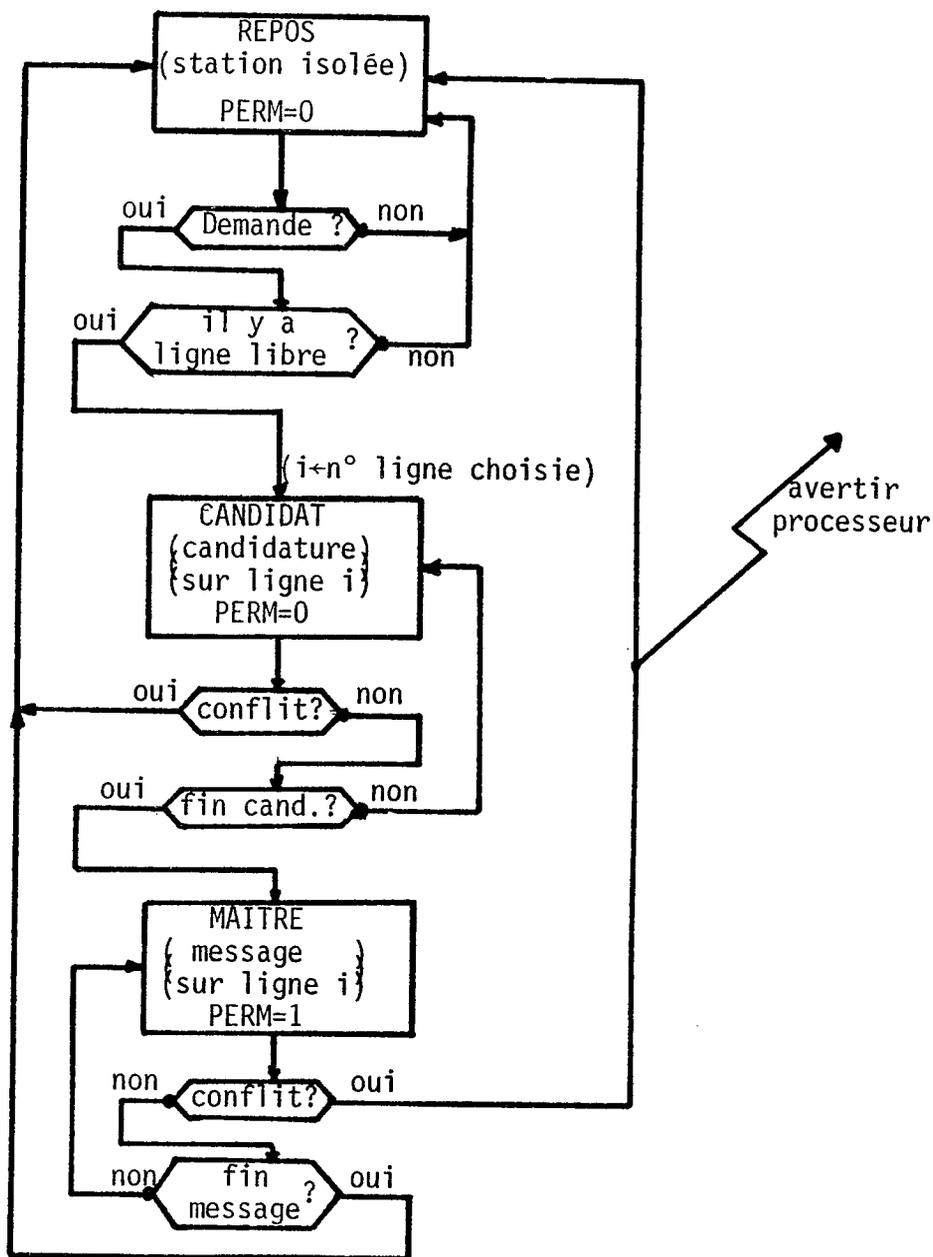


FIG.13. Algorithme de l'auto-allocation d'autoroute

Le choix entre l'auto-allocation d'autoroute et une "auto-allocation par ligne" se fera essentiellement en fonction du coût de la réalisation. L'auto-allocation d'autoroute a l'avantage que les messages arrivent aux destinataires dans l'ordre de leur génération (mise en file d'attente) par la station émettrice.

### 2.2.6. Mécanisme élémentaire d'échange

Au niveau de l'échange élémentaire, le support de l'information est un signal (onde) analogique. Suivant la forme du signal, on distingue deux types de transmission [CLA] :

- . transmission par modulation,
- . transmission en bande de base.

Dans notre cas, pour pouvoir réaliser l'auto-allocation (déterministe), on a pris la transmission en bande de base. On a vu que, pour appliquer le principe de l'auto-allocation, il faut que la ligne réalise une fonction  $f(S_1, \dots, S_m) \in \{S_1, \dots, S_m\}$  où  $S_1, \dots, S_m$  sont les informations émises en même temps par les stations concurrentes. Dans l'annexe 1 on montre des réalisations de lignes qui respectent cette propriété : fibre optique [NOG2], amplificateurs d'émission en collecteur ouvert, amplificateurs d'émission en émetteur-suiveur, transmission en différentiel.

Dans tout échange d'information on distingue deux parties [NOG1] :

- . la donnée,
- . l'horloge (repère temporel) = donne les instants où la donnée est valide.

Ces deux entités peuvent se trouver :

- . sur des supports physiques distincts : quand la distance émetteur-récepteur est très courte (dans un ordinateur par exemple),
- . sur le même support physique : pour des longues distances (liaisons série).

Dans notre cas, on "mélange" la donnée et l'horloge sur le même support. A l'émission a lieu l'opération de "codage" qui mélange la donnée et l'horloge et à la réception a lieu le "décodage" par lequel on extrait la donnée et l'horloge du signal.

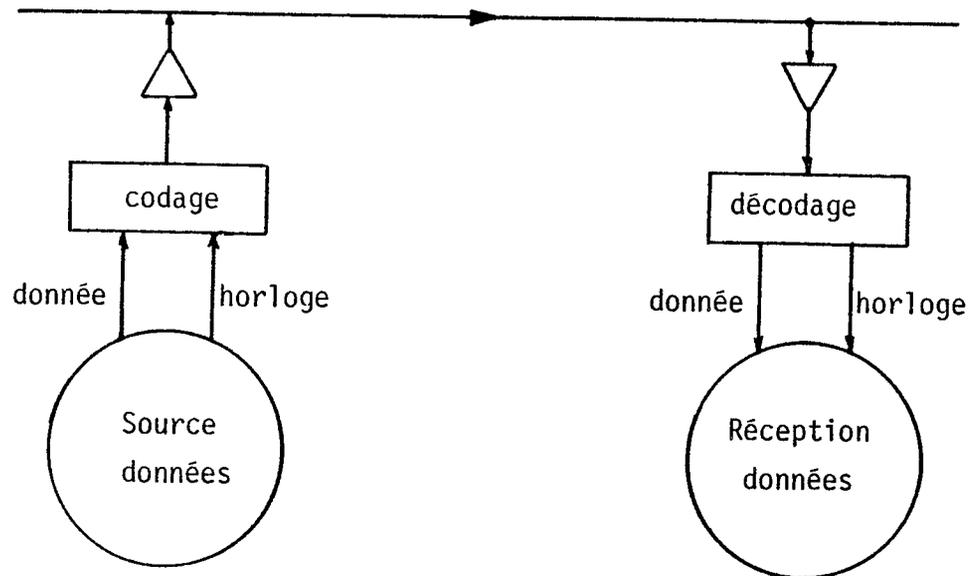


FIG.14. Principe de la transmission

Le codage/décodage repose sur le fait que les informations fournies par une impulsion (impulsion = signal entre deux changements successifs de son état) sont sa durée et son amplitude.

Suivant le codage choisi, on a un quantum d'information de durée  $u$  et la durée  $\theta$  du bit sera un multiple de  $u$  :  $\theta = q.u$ .

La série de bits se transforme sur la ligne en une série de quantum (selon l'algorithme de codage choisi). Le signal sur la ligne sera une suite d'impulsions, chaque impulsion "contenant" un ou plusieurs quantum.

N.B. L'ensemble des états du signal est supposé fini (discret). Dans la transmission série en bande de base, le signal est généralement bivalent (deux valeurs : "0" et "1"). Finalement, il importe que le récepteur fasse une bonne extraction des quantum à partir du signal.

L'émetteur et le récepteur ont la même unité de temps (plus généralement on connaît la relation qui lie leurs unités de temps respectives).

En mesurant la durée de l'impulsion, le récepteur peut déterminer le nombre de quantum qui y sont "contenus", la valeur des quantum étant déterminée à partir de la valeur (amplitude) de l'impulsion.

L'intérêt du codage est d'avoir sur la ligne un signal tel que la durée d'une impulsion ait une borne supérieure. Ceci entraîne que, pour une impulsion, la dispersion résultant du cumul des dispersions sur chaque quantum, aura une borne supérieure, ce qui permettra de déterminer les conditions (la durée  $\theta$  du bit) pour un échange correct (voir annexe 3).

N.B. Ce fait est par ailleurs important dans l'auto-allocation pour pouvoir fixer la durée TOCC détectrice de ligne libre.

Pour un codage donné on a :  $\theta = q.u$  ( $u$  = durée du quantum). Avec la méthode de reconnaissance de l'information par la mesure de la durée des impulsions (annexe 3), on montre que l'on doit avoir :

$$\theta > 2q. \frac{p + (1 + \alpha).h}{1 - (2N + 1)\alpha}$$

où  $p$  = dispersion sur la durée d'une impulsion. Cette dispersion est introduite par la précision (plutôt l'imprécision) de l'horloge d'émission, par la ligne et par l'auto-allocation

$h$  = période d'échantillonnage (unité avec laquelle on mesure la durée)

$\alpha$  = précision de la période d'échantillonnage

$N$  = nombre maximum de quantum contenus dans une impulsion.

Pour avoir  $\theta$  petit (donc grand débit), l'horloge d'échantillonnage sera donnée par un quartz (précision  $\alpha \approx 10^{-4}$ ). De plus on prend un codage pour avoir  $q$  et  $N$  petits. Dans notre cas, on a  $q = 1$  et  $N = 7$  et ceci en utilisant :

- . codage NRZI : changement de polarité de la ligne, si bit "0"; pas de changement, si bit "1"
- . format SDLC des messages : il y a au maximum six "1" consécutifs.

## 2.3. PROTOCOLE DE BASE

Le protocole de base définit le dialogue que l'on peut avoir entre les stations du réseau.

Le but visé est d'échanger des blocs de données entre les stations tout en assurant l'intégrité de l'information.

Pour que le protocole de base puisse être utilisé pour une large gamme de protocoles fonctionnels, chaque échange de bloc donnée constitue (pour le protocole de base) une unité complète, indépendante des autres échanges.

Pour avoir un dialogue cohérent et sûr, il faut que le protocole de base satisfasse les conditions suivantes :

- . spécification du début et de la fin du message : ceci afin de distinguer les messages bien formés dans l'ensemble des chaînes binaires, ce qui permet par exemple aux récepteurs d'ignorer une partie des signaux parasites ;
- . identification du destinataire du message ;
- . identification de l'expéditeur du message : ceci afin que le destinataire puisse répondre plus tard ;
- . détection des erreurs ; plus précisément, il faut :
  - un moyen (information redondante) permettant au récepteur de déterminer si le message reçu est ou non entaché d'erreurs,
  - un moyen (accusé de réception) permettant à l'expéditeur de savoir si le destinataire a bien reçu ou non le message.

### 2.3.1. Format des messages

Pour satisfaire les conditions imposées au protocole, on choisit la structure suivante pour les messages :

préambule (PRB), adresse destinataire (D), adresse expéditeur (E) informations (I), bloc contrôle (BC), postambule (PSB).

N.B. Le mode de transmission est synchrone.

Les champs PRB, D, E, BC, PSB ont des longueurs fixes, tandis que le champ I est de longueur variable. Pour la cohérence, il faut que les configurations PRB, PSB n'apparaissent pas dans le corps du message.

Au niveau du protocole de base, à chaque station est associé un ensemble d'adresses ("adresses" des processus implantés sur la station).

Un message est bien reçu par une station réceptrice si celle-ci a reconnu le préambule, si l'adresse destinataire appartient à l'ensemble d'adresses associées à la station et si à la détection du postambule le bloc contrôle calculé est correct. Autrement le message est ignoré par la station.

Remarque : on voit que le format choisi satisfait aux conditions imposées au protocole, sauf en ce qui concerne l'accusé de réception dont on discutera plus loin.

Le format proposé ci-dessus a été choisi aussi compte tenu de sa ressemblance avec le format SDLC et parce qu'il existe des circuits intégrés qui le manipulent. En fin de compte, on prend pour les messages le format SDLC [DON] légèrement modifié :

<u>flag (F)</u> ,	<u>destinataire (D)</u> ,	<u>expéditeur (E)</u> ,	<u>informations (I)</u> ,
1 octet	k bits	k bits	variable
<u>code contrôle (CRC)</u> ,	<u>flag (F)</u> .		
2 octets	1 octet		

N.B. généralement on prend  $k = 8$  (1 octet) et la longueur du champ I un multiple d'octets.

Le flag F a la configuration binaire : 0 1 1 1 1 1 1 0.

Pour empêcher l'apparition de cette configuration en dehors du début et de la fin du message, après toute succession de cinq "1" (succession qui se trouve en dehors des flags), a lieu l'insertion automatique d'un "0" qui est éliminé à la réception.

Conséquence :

avec le codage NRZI (changement de polarité de la ligne pour chaque bit "0"), la durée entre deux transitions successives pendant l'émission d'un message est au maximum de 7 périodes de bit (ce fait est utilisé par l'auto-allocation pour la détection de "ligne libre").

N.B. Le code contrôle (CRC) est calculé en effectuant la division du message (considéré en tant que polynôme) par un polynôme de degré 16.

Comme on l'a vu, à chaque station on associe un espace d'adresses reconnaissables. Une station reçoit le message si l'adresse destinataire appartient à son espace d'adresses reconnaissables.

On peut avoir deux sortes de messages :

- . message à destinataire unique : une seule station le reconnaît comme sien,
- . message à plusieurs destinataires.

Le message à plusieurs destinataires présente de l'intérêt quand quelque chose se passe dans le réseau et plusieurs stations doivent être prévenues : un seul message à plusieurs destinataires suffit. Pour avoir ce type de message, il faut que l'intersection de plusieurs espaces d'adresses reconnaissables soit non vide.

N.B. Pour les messages à plusieurs destinataires, le mécanisme de l'accusé de réception (présenté plus loin) est inopérant.

Dans la présentation du mécanisme d'auto-allocation, on a vu qu'en fonction de la stratégie adoptée, on peut faire précéder le message proprement dit de signaux dits signaux de candidature. Donc, toute station émettrice émet sur la ligne :

candidature(option), flag, destinataire, expéditeur, informations,  
code contrôle, flag.

### Observation :

Les signaux de candidature servent uniquement à l'auto-allocation et doivent être ignorés par les stations réceptrices. Pour que ces signaux soient ignorés, il faut que la chaîne binaire candidature, flag ne puisse pas former un bon message.

Soit TCAND la durée prise par les signaux de candidature. Dans le format SDLC, la longueur des messages est plus grande que 4 octets (code contrôle et les deux flags). Si la durée TCAND est plus petite que le plus petit message (4 octets), on est sûr que la chaîne candidature, flag ne forme pas un bon message. Autrement, on fait le raisonnement suivant :

avant l'émission d'un message, la ligne est libre.

Alors chaque fois qu'a lieu le passage de libre à occupé de l'état de la ligne, on inhibe la réception sur chaque station réceptrice pendant la durée TCAND.

### 2.3.2. Accusé de réception

L'accusé de réception est l'information que le destinataire d'un message envoie à l'expéditeur pour lui spécifier que le dialogue s'est bien déroulé. Le mécanisme de l'accusé de réception est un moyen d'augmenter la sécurité du système.

Il est clair qu'en cas de mauvaise émission/réception du message, aucune station réceptrice ne reconnaîtra le message comme étant pour elle et par conséquent, il n'y aura pas d'accusé de réception. Ce fait entraîne au niveau de la station émettrice la politique suivante : on définit une durée TACC comme étant le temps maximum entre la fin de l'émission et l'arrivée de l'accusé de réception.

N.B. Cette durée TACC dépend du mécanisme de communication choisi.

Si pendant la durée TACC, après la fin de l'émission, l'accusé de réception arrive, alors le message a été bien reçu. Si, au contraire, la durée TACC s'écoule sans que l'accusé n'arrive, alors on décrète qu'il y a eu panne et que le message n'a pas été reçu.

N.B. Par d'autres mécanismes (ré-émission du message, séquences de test, ..), on essaie de remédier à la panne.

Si le temps de réponse du destinataire (temps mis par celui-ci pour envoyer l'accusé) est borné, alors le mécanisme d'accusé de réception constitue un moyen "parfait" de vérification de la bonne émission/réception du message (la durée TACC est choisie plus grande que ce temps de réponse). Si le temps de réponse n'est pas borné, il est clair que ce moyen de vérification est imparfait : il existe des cas où il y a eu bonne réception, mais où le temps de réponse dépasse la durée TACC, ce qui fait que l'expéditeur prend une mauvaise décision (il décrète qu'il y a eu mauvaise réception), ce qui peut entraîner des incohérences au niveau du système. De ce fait, on s'efforce d'avoir un mécanisme d'accusé de réception où le temps de réponse du destinataire est borné.

Tout système se présentera comme une hiérarchie de niveaux, le niveau  $i$  utilisant les mécanismes du niveau  $i-1$ .

En ce qui concerne le dialogue, à chaque niveau on a un mécanisme d'accusé de réception. La notion de "bonne réception" peut avoir des acceptations différentes suivant le niveau où l'on se place dans le système.

Exemple : au niveau fonctionnel, "bien reçu" voudra dire "bien compris".

Nous nous plaçons au niveau matériel où la "bonne réception" sera la bonne réception physique du message : les bits émis transitent bien (sans panne) par la ligne physique, sont reçus par le coupleur récepteur, ce coupleur reconnaît ces bits comme formant un message et met le message à la disposition du logiciel de la station réceptrice.

Dans le contexte de la ligne multipoint auto-allouée, le principe du mécanisme d'accusé de réception est le suivant : après la fin du message émis et avant la libération de la ligne, la station qui a reçu le message va envoyer sur la ligne l'accusé de réception, noté ACK, qui sera une configuration binaire (de petite longueur). Dans le contexte du format SDLC, on va prendre pour ACK la même configuration que pour le flag :  
0 1 1 1 1 1 1 0.

N.B. Ce choix de l'accusé ACK a été fait compte tenu du fait que les contrôleurs qui manipulent le format SDLC signalent chaque fois qu'un flag est reçu.

Comme on l'a déjà vu, à la fin du message, la station émettrice "relâche" la ligne qui prendra la position "repos". La station réceptrice doit envoyer l'accusé ACK avant la libération de la ligne, donc avant que la durée (comptée à partir de la fin du message) TOCC s'écoule.

N.B. on a défini "ligne libre" = ligne "en repos" pendant la durée TOCC.

En effet, si la station réceptrice envoie ACK après la durée TOCC, il est évident que les autres stations verront la ligne libre, commenceront (éventuellement) leurs émissions et donc ACK ne pourra être envoyé correctement à cause des conflits avec les autres messages.

Notons par TREP le temps maximum (compté à partir de la fin du message émis/reçu) mis par la station réceptrice pour commencer l'envoi de ACK, envoi qui a lieu avant la fin de la durée TOCC. Il faut donc :

$$\text{TREP} < \text{TOCC}$$

Notons par TACK la durée prise pour l'envoi de ACK. La durée TACC, au bout de laquelle la station émettrice du message se rend compte de la présence/absence de l'accusé sera :

$$\text{TACC} = \text{TREP} + \text{TACK}.$$

Les paramètres du mécanisme d'accusé de réception sont :

- . ACK, l'accusé de réception ; on a pris  $\text{ACK} = 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0$  ; sa petite longueur ne pénalise pas fortement les performances du mécanisme de communication ;
- . TACK, temps nécessaire pour envoyer l'accusé de réception ACK ;
- . TREP, temps maximum (mis par la station réceptrice) entre la fin du message et le début de l'envoi de ACK ; il faut que  $\text{TREP} < \text{TOCC}$  ;
- . TACC, temps compté à partir de la fin du message, pendant lequel la station émettrice attend l'accusé de réception. On a  $\text{TACC} = \text{TREP} + \text{TACK}$ .

Le mécanisme fonctionne correctement si la station émettrice détecte correctement s'il y a eu bonne/mauvaise réception du message. On se place dans le cas où un message a un destinataire unique.

### a) Bonne réception

Considérons une station émettrice qui émet un message. Le message est bien reçu par la station réceptrice qui en est le destinataire. Conformément au mécanisme, la station réceptrice va commencer l'envoi de ACK avant la libération de ligne (avant l'écoulement de la durée TOCC). Par ailleurs, pendant l'envoi de ACK, la ligne ne pourra devenir libre

Il résulte qu'entre la fin du message considéré et la fin de l'envoi de ACK, aucune station, à part la station réceptrice considérée, ne commencera à émettre, puisque :

- . les autres stations émettrices ne pourront pas émettre du fait que la ligne n'est pas libérée,
- . la station réceptrice considérée est le seul destinataire du message et donc d'autres stations réceptrices ne pourront pas envoyer de ACK.

Donc ACK sera acheminé correctement vers la station émettrice considérée puisqu'il n'y a personne pour le perturber. La station émettrice le reçoit et ainsi il y a détection de la bonne réception.

Bien sûr il y a toujours la possibilité d'une panne sur la ligne pendant l'envoi de ACK, donc la station réceptrice ne le reçoit pas et décrète qu'il y a eu mauvaise réception du message émis (ce qui n'est pas le cas). Il y a donc là un fonctionnement incorrect du mécanisme. On doit faire les remarques suivantes :

- . on a détecté quand même une anomalie,
- . la probabilité d'une panne pendant l'envoi de ACK est très faible puisque la longueur (en tant que chaîne binaire) de ACK est petite.

b) Mauvaise\_réception

Considérons une station qui émet un message. Il y a mauvaise réception, donc aucune station réceptrice n'enverra d'accusé de réception.

Au bout du temps TOCC, la ligne va devenir "libre" et des stations émettrices pourront émettre des messages. Se pose alors la question : est-ce que le début d'un tel message ne pourrait être pris par la station qui attend ACK comme étant l'accusé de réception ?

Montrons que cela ne peut pas arriver : la durée TOCC écoulée, il y aura émission de message (par d'autres stations). A la fin de la durée TACC, la station qui attend un ACK a vu arriver des bits pendant la durée  $TACC - TOCC$  et comme  $TREP < TOCC$  et  $TACC = TREP + TACK$ , il en résulte que :

$$TACC - TOCC < TACC - TREP = TACK.$$

Donc les bits de cette durée  $TACC - TOCC$  ne peuvent pas former un ACK puisque ACK nécessite une durée TACK plus grande. La station considérée ne recevant pas ACK, va détecter ainsi le fait qu'il y a eu mauvaise réception du message.

En cas d'absence d'accusé de réception, la station émettrice utilisera la stratégie classique : on ré-émet le message un certain nombre de fois et si l'accusé de réception est toujours absent, on décrète qu'il y a panne permanente et par d'autres mécanismes on essaie de remédier à la panne. Dans ce cas on met en cause la station réceptrice plutôt que la ligne. En effet, comme la station émettrice a pu écouler son message, ceci signifie qu'elle n'a pas détecté de conflit (voir mécanisme d'auto-allocation), donc la ligne a bien véhiculé le message vers la station réceptrice.

## 2.4. PERFORMANCES DU MÉCANISME DE COMMUNICATION

Au paragraphe 2.2.4. nous avons déterminé la durée  $\theta$  du bit et donc le débit physique total permis par l'auto-allocation. Dans ce qui suit, on s'intéressera aux performances du mécanisme de communication vues du point de vue de l'utilisateur.

### 2.4.1. Efficacité du mécanisme de communication

L'efficacité du mécanisme est défini par le rapport :

$$e = \frac{\text{débit utile}}{\text{débit total}}$$

ou encore

$$e = \frac{\text{durée d'occupation de la ligne par l'information utile du message}}{\text{durée totale d'occupation de la ligne par un message}}$$

Au niveau du protocole de base, l'information utile est constituée par : adr.destinataire, adr.expéditeur, données.

Soit  $n$  le nombre moyen d'octets de cette chaîne : nombre moyen d'octets utiles. Si  $\theta$  est la durée du bit, on obtient que la durée d'occupation utile de la ligne est égale à  $8.n.\theta$ .

La figure 15 montre que la durée d'occupation totale entraînée par un message est égale à :

$$TCAND + (4 + n).8.\theta + TREP + 8.\theta + TOCC$$

L'efficacité devient :

$$e = \frac{8.n.\theta}{TCAND + TREP + TOCC + (5 + n).8.\theta}$$

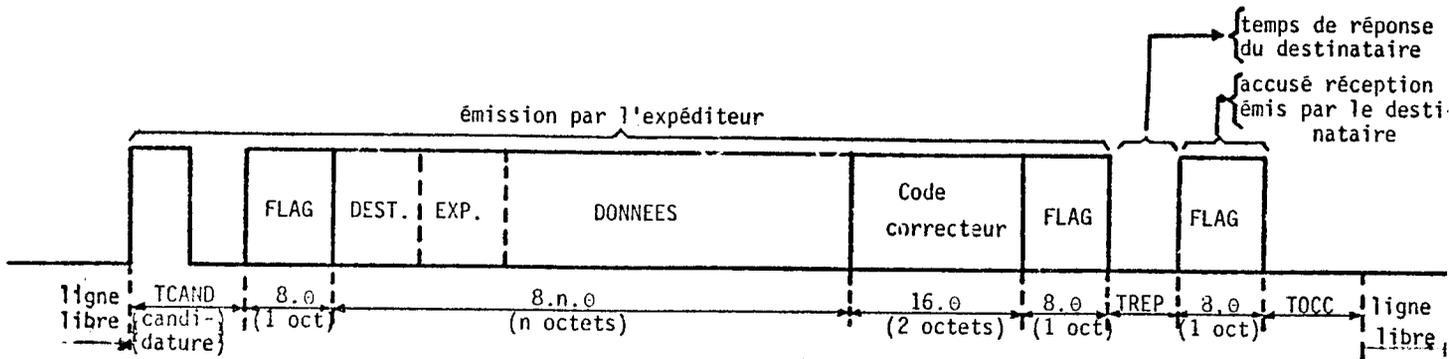


FIG.15. Occupation de la ligne à l'émission d'un message

On s'aperçoit que l'efficacité augmente avec la longueur de messages, donc les petits messages sont pénalisés.

Dans la réalisation que nous proposons plus loin (chapitre 4), on a :

$$TCAND + TREP \leq 8.0, TOCC \approx 8.0$$

L'expression de l'efficacité devient :

$$e = \frac{n}{n + 7}$$

avec  $n$  = nombre moyen d'octets (utiles) par message.

Dans l'application du mécanisme de communication à l'autocommutateur téléphonique, le nombre moyen d'octets (utiles) par message est  $n \approx 10$ . On obtient une efficacité  $e = \frac{10}{17} \approx 58,8 \%$ . Cette efficacité est relativement basse et ceci à cause d'une part de la petite longueur des messages et d'autre part de la grande sécurité (constituée par les flags, le code correcteur, l'accusé de réception) que l'on veut avoir pour les échanges.

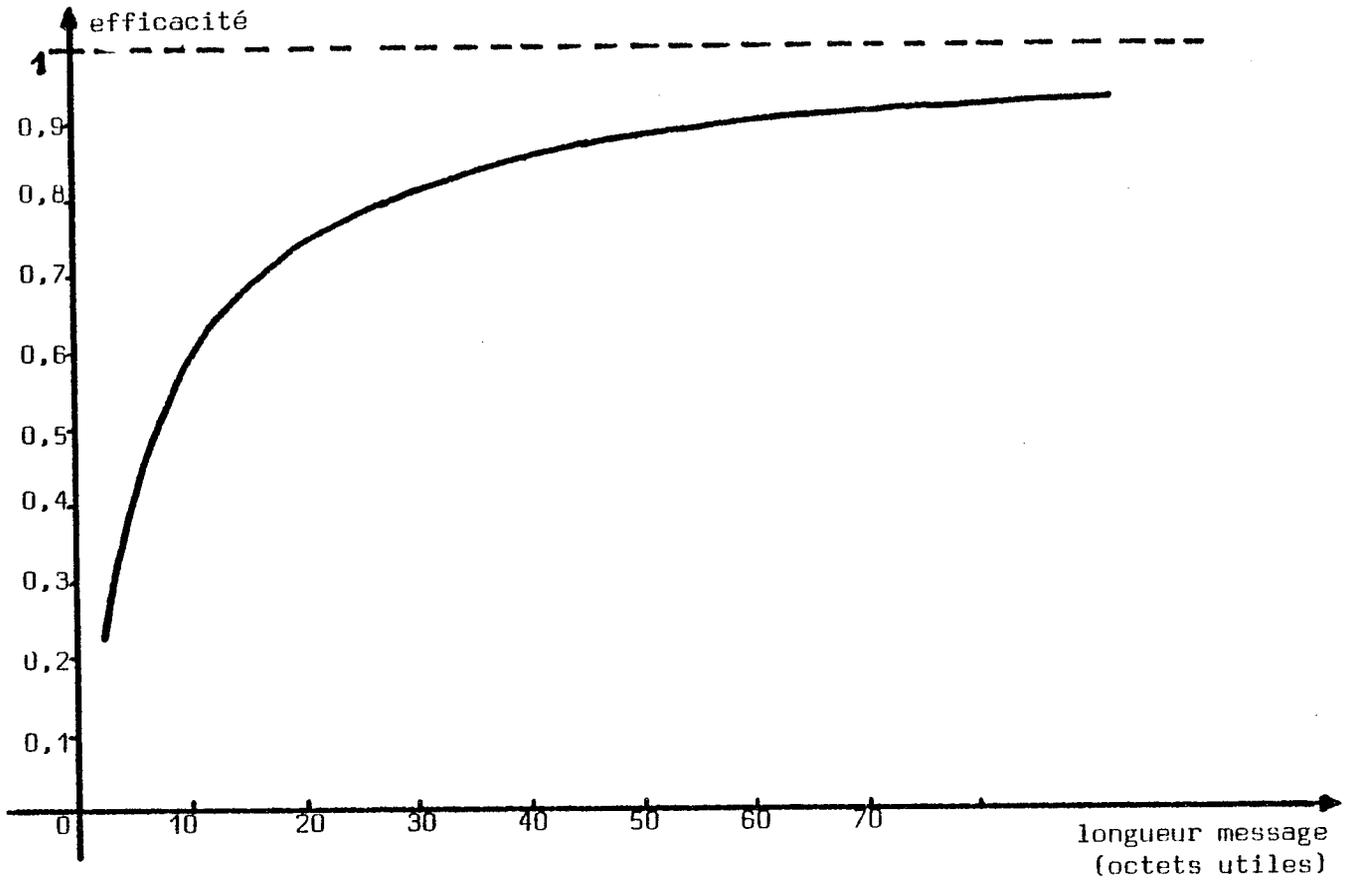


FIG.16. Efficacité du mécanisme de communication

### 2.4.2. Comportement des files d'attente

Dans le réseau se posent des problèmes concernant les performances globales du système : temps d'attente des messages, longueur des files d'attente, ..

Pour notre mécanisme de communication basé sur l'auto-allocation, on est amené d'une manière naturelle au modèle : un serveur et plusieurs files d'attente.

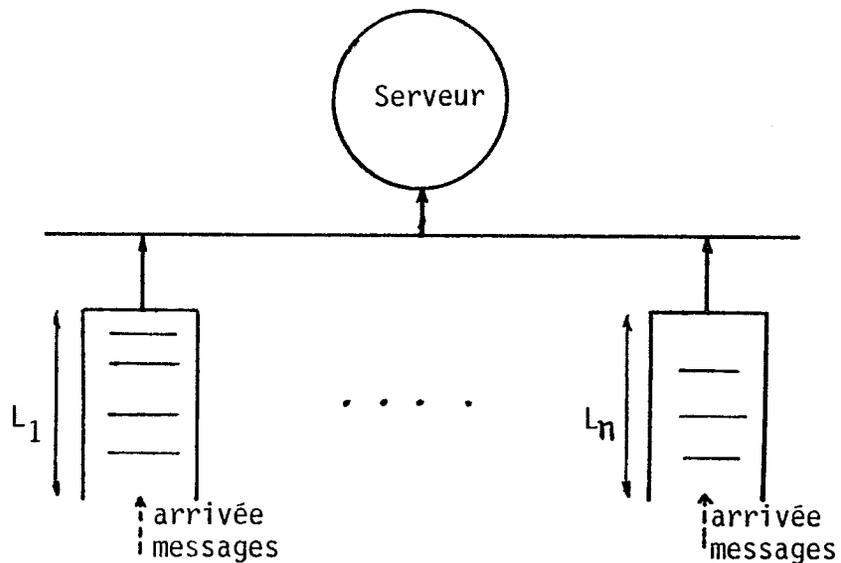


FIG.17. Modèle pour l'auto-allocation :  
Un serveur , plusieurs files d'attente

Chaque file d'attente correspond à une station du réseau : les messages générés (arrivées messages) par la station sont mis dans la file d'attente. Les files sont vidées par le serveur. Dans notre cas, le serveur est la ligne régie par l'auto-allocation.

Compte tenu du fonctionnement de l'auto-allocation, il résulte que la politique du serveur est : il prend (choisit) au hasard un message dans les files non vides.

La durée  $U$  de service du serveur sera égale au temps pendant lequel l'émission d'un message occupe la ligne ; dans notre cas  $U$  comprendra :

- . la durée TOCC nécessaire à l'auto-allocation pour détecter que la ligne est libre ;
- . la durée nécessaire pour l'émission des bits du message : si le message a  $N$  bits et  $\theta$  est la période du bit, cette durée est  $N.\theta$  ;
- . la durée TACC nécessaire à l'envoi (par la station réceptrice) de l'accusé de réception.

On voit que  $U$  dépend linéairement de la longueur du message.

Notons par  $n$  le nombre de files d'attente (le nombre de stations). Notons par  $L_i$  la longueur de la file  $i$  : c'est le nombre de messages de la file, y compris celui qui (éventuellement) est en cours d'émission. On s'intéresse principalement au temps d'attente d'un message (temps pendant lequel il reste dans la file) et à l'évolution des longueurs  $L_i$  (moyenne, écart type, ..) pour pouvoir déterminer l'espace mémoire à allouer aux files. Pour ce type de problème, on peut donner des solutions analytiques seulement dans quelques cas (essentiellement quand les arrivées sont poissonniennes), ce qui fait que la simulation reste le moyen principal pour prévoir le comportement réel du système.

Dans l'annexe 4 on étudie le cas où les arrivées sont poissonniennes et les stations sont équivalentes (même loi pour les arrivées, trafic équilibré), cas qui malheureusement est peu fréquent en pratique. A notre connaissance ce cas n'a jamais été traité. Nous n'avons pas pu lui donner de solution exacte, nous contentant de donner des majorants aux différentes quantités recherchées.

Selon les calculs développés à l'annexe 4, on trouve les résultats suivants :

[on note  $E(X^i)$  le moment d'ordre  $i$  de la variable aléatoire  $X$ ]

.  $n$  = nombre de stations

.  $c$  = taux de charge

$$c = \frac{D}{\Delta}$$

où  $\Delta$  = débit maximum supporté par la ligne (auto-allouée)

$D$  = débit (total) échangé dans le réseau (y compris les informations correspondant aux durées TOCC, TACC, ..)

.  $L_i$  = longueur (nombre messages) d'une quelconque file  $i$  ( $i = 1, 2, \dots, n$ )

On obtient :

$$E(L_i) = \frac{1}{n} \cdot \left[ c + \frac{c^2 \cdot u_2}{2(1-c)} \right]$$

$$E(L_i^2) \leq \frac{E(L^2)}{n} - \frac{c \cdot (n-1)}{n^2} \cdot E(L) + \frac{c^2 \cdot (n-1)}{2 \cdot n^2}$$

$$\text{avec } E(L) = c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)}$$

$$E(L^2) = c + \frac{3 \cdot c^2 \cdot u_2}{2 \cdot (1-c)} + \frac{c^3 \cdot u_3}{3 \cdot (1-c)} + \frac{c^4 \cdot u_2^2}{2 \cdot (1-c)^2} \quad \text{où } u_i = \frac{E(U^i)}{E^i(U)} \quad [i=2,3]$$

$U$  étant le temps d'occupation de la ligne entraînée par un message.

L'écart type sera  $\sigma_i = \sqrt{E(L_i^2) - E^2(L_i)}$ .

.  $S$  = nombre de stations qui ont quelque chose à émettre

$$E(S) < \min \left( n \cdot c, c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)} \right)$$

. PRS = temps de "présence" du message dans la file : c'est le temps écoulé entre la génération du message (sa mise en file d'attente) et son arrivée à la station réceptrice dont il est le destinataire.

$$E(\text{PRS}) \leq [1 + E(L_i)] \cdot \frac{E(S)}{c} \cdot E(U) \quad \text{ou encore}$$

$$E(\text{PRS}) \leq \left[ 1 + \frac{1}{n} \cdot \left( c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)} \right) \right] \cdot \frac{E(S)}{c} \cdot E(U)$$

(  $E(U)$  étant la durée moyenne d'un message et

$E(S)$  étant le nombre moyen de stations concurrentes

c'est à dire dont les files d'attente sont nonvides)

On va dimensionner l'espace mémoire alloué à la file d'attente d'une station tel que la probabilité de débordement soit plus petite qu'une valeur  $p$  (constante du système). La formule de Bienaymé-Tchebychef donne :

$$\text{prob} ( |L_1 - E(L_1)| > \beta ) \leq \frac{\sigma_1^2}{\beta^2}$$

où  $L_1$  = nombre de messages de la file

$$\sigma_1^2 = E(L_1^2) - E^2(L_1)$$

( $\sigma_1$  = écart type)

On prend  $\frac{\sigma_1^2}{\beta^2} \leq p$  d'où  $\beta \geq \frac{\sigma_1}{\sqrt{p}}$

La file peut contenir  $\beta + E(L_1)$  messages, donc il faut allouer à la file la taille mémoire égale à :

$$\left( \frac{\sigma_1}{\sqrt{p}} + E(L_1) \right) \cdot \text{taille message}$$

pour que la probabilité de débordement soit inférieure à  $p$ .

Exemple numérique :

$n = 50$  stations

$\Delta = 320$  Kbauds - débit supporté par la ligne

$D = 280$  Kbauds - débit échangé dans le réseau

On a  $c = \frac{D}{\Delta} = 0,875$ .

On fait l'hypothèse que la longueur des messages est presque constante, donc la durée  $U$  de service est presque constante, d'où :  $U_2 \approx 1$ ,  $U_3 \approx 1$ .

On obtient :

. nombre moyen de messages dans une file :  $E(L_1) = 0,07875$

. écart type :  $\sigma_1 = \sqrt{E(L_1^2) - E^2(L_1)} < 0,738$

En allouant à la file une taille mémoire égale ou supérieure à  $7,45 \times$  taille message, on est sûr que la probabilité de débordement est inférieure à  $0,01$ .

Toutefois, étant données les majorations faites (dans le calcul de l'écart type), il paraît vraisemblable qu'en pratique on pourra se contenter de tailles mémoire moins importantes.

. "Présence" d'un message dans le système : temps présence  $< 4,8 \times$  durée message (l'attente proprement dite est inférieure à  $3,8 \times$  durée message).

## 2.5. CONCLUSION

Le mécanisme de communication dont nous venons de présenter les principes nous semble très bien adapté aux systèmes décentralisés (répartis) réalisés sous forme de réseaux.

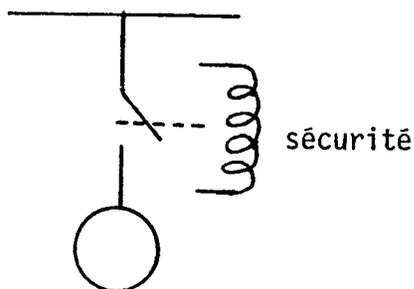
Le mécanisme respecte les propriétés d'extensibilité, flexibilité, asynchronisme, uniformité, ... discutées au paragraphe 1.1, propriétés que l'on doit avoir pour les systèmes répartis.

Ces propriétés sont assurées essentiellement par le mécanisme d'auto-allocation. La caractéristique principale de notre mécanisme est le fait de rendre complètement équivalentes les stations du point de vue du dialogue.

Un autre intérêt du mécanisme de communication est constitué par sa sûreté de fonctionnement :

. une station en panne n'empêche pas le dialogue dans le reste du réseau.

N.B. cette assertion demande à être nuancée : en effet un collage d'un circuit directement relié à la ligne rend celle-ci inutilisable. Pour éviter cela, il est préférable que les mécanismes qui détectent les pannes sur une station l'isolent *physiquement* de la ligne (relais électromagnétiques).



- . Le mécanisme de communication détecte des pannes :
  - au niveau de l'échange élémentaire (par l'intermédiaire de l'auto-allocation) :
    - .. la ligne devient libre pendant l'émission (normalement elle devrait être occupée),
    - .. détection d'un conflit après l'émission de l'adresse expéditeur (quand normalement il n'y a plus qu'une seule station qui émet)
  - au niveau du protocole de base : par l'intermédiaire du mécanisme d'accusé de réception.

Le désavantage du mécanisme de communication, et plus spécialement de l'auto-allocation, est sa limitation à des réseaux géographiquement locaux (quelques centaines de mètres).

## CHAPITRE 3

### IMPLÉMENTATION DU MÉCANISME DE COMMUNICATION

### 3.1. ARCHITECTURE DU RÉSEAU

On s'intéresse au système qui globalement se présente comme un ensemble de groupements géographiquement locaux de processeurs (figure 18). Chaque groupement constituera un réseau local.

Dans l'optique d'une architecture répartie, le réseau local est conçu pour traiter le plus possible le problème là où il se pose. Pour cela, le réseau local concernera surtout des courtes distances (centaines de mètres) comme par exemple : usine, bureaux, central téléphonique.

De la localité du traitement, il résulte que le débit d'information échangé dans le réseau local est largement supérieur au débit échangé entre le réseau local et le reste du réseau global. (Le débit est fonction décroissante de la distance).

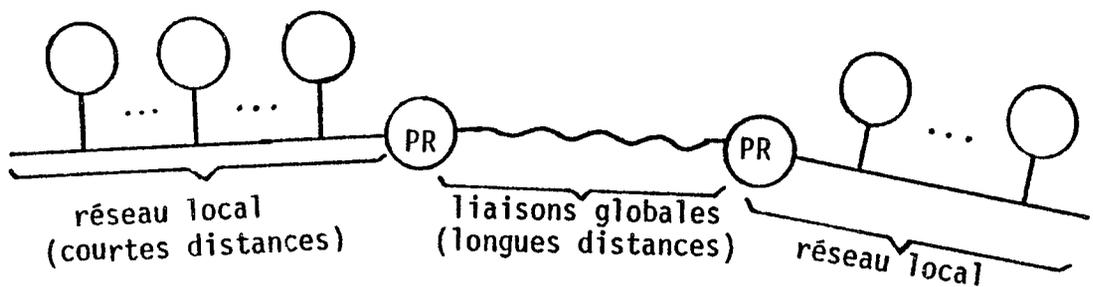


FIG.18. Architecture du réseau global

Pour communiquer avec le reste du réseau global, chaque réseau local dispose d'une station spécialisée : le processeur de routage noté PR.

Sur les stations du réseau sont implémentés les processus qui collaborent pour réaliser le système objet.

Comme on l'a déjà vu, on considère trois niveaux pour le dialogue : protocole fonctionnel, protocole de base, échange effectif.

N.B. Le niveau "protocole fonctionnel" peut être composé de plusieurs niveaux que l'on ne détaille pas dans notre exposé.

Dans notre étude, on s'intéresse à l'implémentation du mécanisme de communication constitué par les deux derniers niveaux, dont les principes ont été présentés au chapitre 2.

Une question qui n'a pas été abordée au chapitre 2 est le problème classique des noms. Chaque processus a un nom qui lui est accordé une fois pour toutes.

Au niveau fonctionnel, les processus dialoguent entre eux par noms, indépendamment de leurs emplacements géographiques. Un processus peut dialoguer seulement avec les processus dont il connaît les noms. Ce dialogue par noms a l'avantage de rendre la conception des processus indépendante de leur implémentation géographique. Pour réaliser ceci, il faut donc que dans chaque station la partie réception puisse reconnaître les noms.

Généralement, le mécanisme de communication dispose dans chaque station de deux tables :

- . table des noms des processus existant dans le réseau et qui est utilisée à l'émission des messages : le message est émis seulement si le nom du destinataire se trouve dans la table ;

. table des noms des processus existant sur la station et qui est utilisée à la réception des messages : un message est bien reçu (et donc on renvoie un accusé de réception) seulement si le nom du destinataire se trouve dans la table.

Ces tables sont dynamiques : chaque fois qu'un (ou plusieurs) processus entre ou sort d'une station, le reste du réseau est averti (par la station en question ou par quelqu'un d'autre, comme par exemple la console opérateur) de ce changement par des messages appropriés, ce qui permet la mise à jour des tables. Initialement, ces tables sont vides et elles se remplissent au fur et à mesure que les processus sont activés.

On voit donc que dans ce cas la structure hiérarchique du système comporte un niveau entre le protocole fonctionnel et le protocole de base et que l'on nommera "protocole système".

Le protocole système servira à charger/détruire le code des processus implantés sur la station, à la mise à jour des tables des noms, à la détection des pannes, ..., donc en général à tout ce qui touche à la reconfiguration du réseau.

Un cas spécial est celui du processeur de routage PR :

- . du côté du réseau local, il reçoit tous les messages dont le destinataire ne se trouve pas dans le réseau local (mais existe dans le réseau global) et les ré-émet dans le réseau global ;
- . du côté du réseau global, il reçoit tous les messages dont le destinataire se trouve dans le réseau local et les ré-émet dans le réseau local.

### 3.2. STRUCTURE D'UNE STATION

On va regarder la station surtout du point de vue du mécanisme de communication.

Fonctionnellement, la station se compose de trois parties :

- . processeur de traitement : exécute le programme d'application,
- . coupleur(s) d'émission,
- . coupleur(s) de réception.

Le flux de l'information entre ces trois parties est montré dans la figure 19.

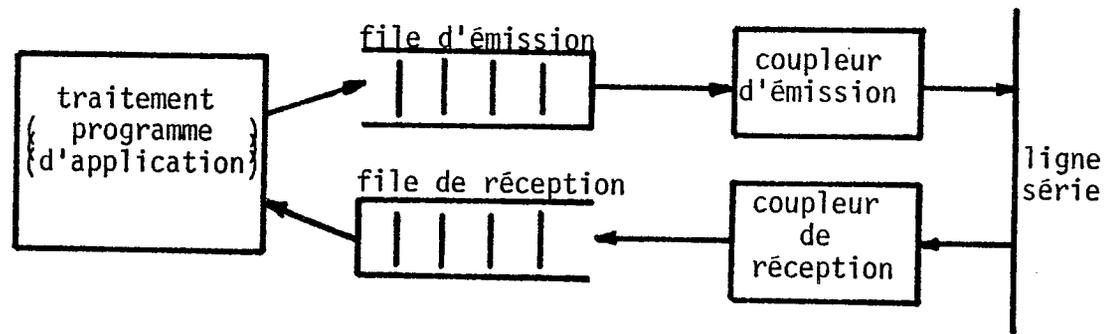


FIG.19. Flux de l'information dans une station

Par coupleurs on entend la partie matériel + logiciel qui réalise le mécanisme de communication (les niveaux inférieurs de la hiérarchie).

Le coupleur d'émission envoie le message et reçoit l'accusé de réception.

Le coupleur de réception reçoit le message (si c'est pour lui) et envoie l'accusé de réception.

Les messages sont envoyés en format SDLC.

L'accusé de réception, noté ACK, sera un flag (01111110).

Les parties distinguées plus haut doivent avoir entre elles un fonctionnement asynchrone, asynchronisme qui découle de l'asynchronisme général du système. Ceci se réalise à l'aide de files d'attente.

Le coupleur d'émission comprend :

- . mécanisme d'auto-allocation,
- . mécanisme de codage de l'information : il réalise le codage NRZ → NRZI,
- . contrôleur : émet le message en format SDLC et reçoit l'accusé de réception,
- . file d'émission (zone mémoire) : contient les messages en attente d'émission,
- . mécanisme qui réalise les échanges dans le sens file d'émission vers contrôleur,
- . logiciel : réalise la gestion de la file d'émission, l'initialisation d'émission des messages, la récupération des erreurs, ...

Le coupleur de réception comprend :

- . mécanisme de reconnaissance (décodage) de l'information : il réalise le décodage NRZI → NRZ,
- . contrôleur : reçoit le message en format SDLC et envoie l'accusé de réception,
- . file de réception (zone mémoire) : contient les messages reçus (en attente de traitement),
- . mécanisme qui réalise les échanges dans le sens contrôleur vers file de réception,
- . logiciel : réalise la gestion de la file de réception, l'initialisation de la réception des messages, la reconnaissance (éventuelle) des adresses destinataires des messages, la récupération des erreurs, ...

On préconise comme contrôleurs l'utilisation des circuits du commerce (COM 5025, ADLC, SIO, MPCC). La réalisation détaillée des coupleurs est montrée au chapitre 4.

Pour réaliser la station, on préconise l'utilisation massive des micro-processeurs qui fournissent une puissance de traitement très bon marché. Du point de vue réalisation, on a deux types d'architecture pour la station :

a) Architecture centralisée (mono-processeur)

Pour cette solution, on aura un seul processeur (microprocesseur) qui exécute le programme d'application et gère les entrées-sorties.

En cours d'émission (ou de réception) d'un message, les échanges entre les contrôleurs et la zone mémoire (files d'attente) où se trouvent les informations, se feront en parallèle (mode d'accès direct mémoire) avec le fonctionnement du processeur.

N.B. Des échanges en mode programmé (contrôlés par le micro-processeur) sont en général à écarter à cause de la forte diminution des performances que ceci entraîne.

Au début et à la fin des messages émis ou reçus et en cas de conflit détecté par l'auto-allocation, le programme d'application est interrompu au profit du programme d'entrée-sortie.

Le programme d'entrée-sortie réalise l'initialisation de l'émission (ou de réception) de message, la reconnaissance des adresses destinataires (en réception), la récupération des erreurs, la gestion des files d'attente.

Cette solution entraîne une baisse des performances de la partie application de la station. Cette baisse dépend du nombre de messages qui circulent dans le réseau et peut devenir assez importante.

## b) Architecture multiprocesseur

Cette réalisation respecte le découpage fonctionnel de la station (figure 19). La station est ainsi composée de plusieurs processeurs spécialisés :

- . processeur(s) de traitement : exécute la fonction (application) associée à la station,
- . coupleur(s) d'émission (réception) : réalise tout le mécanisme de communication de base.

Chaque coupleur comporte un microprocesseur qui contrôle par programme toute la procédure de communication (y compris les échanges élémentaires mémoire - contrôleur).

Le dialogue entre un coupleur et le processeur de traitement se fait uniquement par le mécanisme de boîte à lettre (file d'attente) : le processeur de traitement met (ou prend) un message dans la file, tandis que le coupleur prend (ou met) un message de la file.

On va se placer dans le cas général où le dialogue dans le réseau s'effectue sur une "autoroute" formée de  $k$  lignes banalisées (voir § 2.5 chap. 2). Suivant la réalisation de l'auto-allocation, on a deux solutions :

- . auto-allocation d'autoroute : c'est le mécanisme décrit au § 2.2.5 ; dans ce cas, on aura  $k+1$  coupleurs unidirectionnels :
  - un coupleur d'émission doté du mécanisme d'auto-allocation d'autoroute qui émettra au hasard sur une des lignes libres,
  - $k$  coupleurs de réception (un coupleur par ligne) ;
- . auto-allocation individuelle par ligne : dans ce cas on aura  $k$  coupleurs bidirectionnels (un par ligne). Chaque coupleur fait l'émission/réception des messages sur la ligne qui lui est associée.

Ces deux solutions seront discutées plus en détail dans le paragraphe concernant l'application à l'autocommutateur téléphonique.

En conclusion, nous préconisons une structure multiprocesseurs (coupleurs réalisés avec des microprocesseurs) à cause du rapport performances/coût très intéressant. En effet, le microprocesseur offre une grande puissance de traitement à un faible coût. Son faible coût permet de le considérer comme un quelconque boîtier, ce qui entraîne son utilisation massive, même si sa puissance de traitement n'est employée que partiellement.

La faible taille des programmes (quelques centaines d'octets) permet dès maintenant l'utilisation des microprocesseurs à programme incorporé. Comme la tendance actuelle est vers l'intégration (sur une seule "puce") d'un micro-ordinateur, on aura donc l'essentiel des fonctions coupleur réalisé avec un seul boîtier.

Enfin, la puissance des coupleurs permet de leur incorporer des fonctions annexes, surtout en ce qui concerne la sécurité (comme on le verra dans le cas de l'application au central téléphonique).

### 3.3. APPLICATION : AUTOCOMMUTATEUR TÉLÉPHONIQUE

Les principes du mécanisme de communication ont été validés et sont en cours d'implémentation dans le cadre d'une étude d'autocommutateur téléphonique menée conjointement par l'ENSIMAG et le CNET (contrat 76.378).

Le but final de l'autocommutateur est d'établir (ou rompre) des connexions entre les équipements téléphoniques (postes d'abonnés par exemple).

L'étude s'est appuyée au départ sur les spécifications de l'autocommutateur E10, le but visé étant une structure répartie permettant une puissance quatre fois plus grande (6 000 Erlangs au lieu de 1 500).

Les contraintes concernant la sûreté de fonctionnement, l'extensibilité, l'adaptabilité à la demande (puissance adaptée aux besoins), ont dirigé l'étude vers le choix d'une structure répartie (décentralisée) réalisée sous la forme d'un réseau de stations spécialisées, les liaisons du réseau étant série.

Les principales fonctions de l'autocommutateur téléphonique sont les suivantes :

- . interface avec les équipements téléphoniques (unités de raccordement),
- . commande : réalise l'algorithme téléphonique (analyse de la numérotation, actions vers les autres organes, taxation des conversations, ..),
- . traduction : réalise essentiellement la correspondance numéro d'annuaire - adresse physique de l'équipement,
- . connexion : assure la connexion entre les équipements téléphoniques,
- . gestion , surveillance, et observation du trafic.

La répartition de ces fonctions est présentée dans [ANC2] et [LEC].

Chaque fonction est implémentée sur une ou plusieurs stations, le nombre de stations réalisant la fonction dépendant des performances désirées, de la tolérance aux pannes, de la disponibilité, ..., et constitue un paramètre pour une implémentation particulière (figure 20).

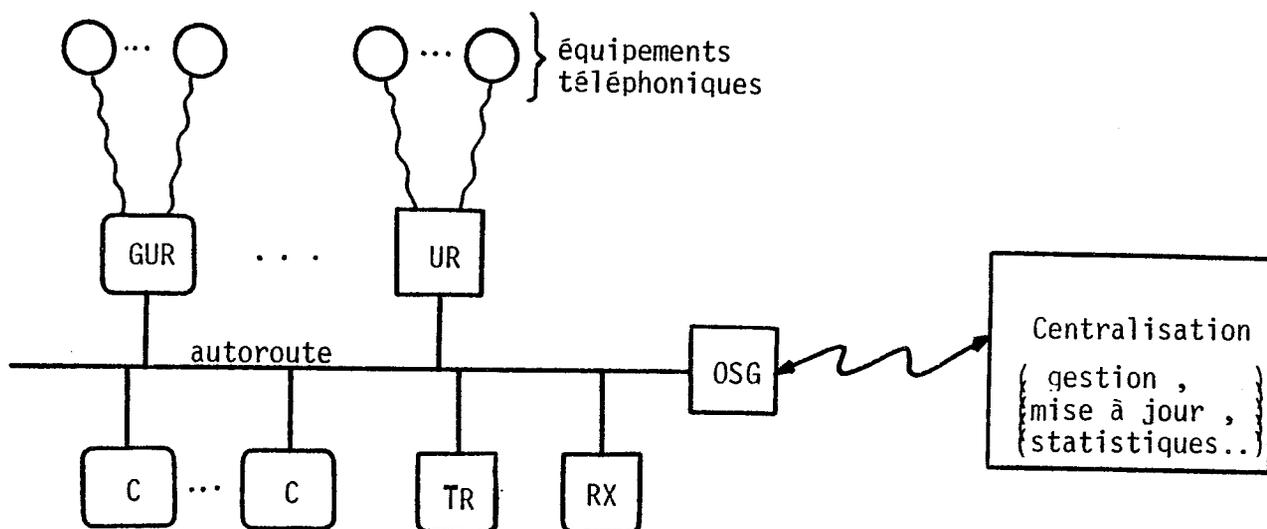


FIG.20. Architecture de l'autocommutateur téléphonique

GUR , UR - interface avec les équipements téléphoniques

C - commande

TR - traducteur

RX - réseau de connexion

OSG - organe de gestion et surveillance

L'autocommutateur est prévu pour 60000 abonnés (trafic téléphonique de 6000 Erlangs), ce qui entraîne l'utilisation d'une centaine de stations.

L'autocommutateur est disposé dans une grande salle avec environ 150 m au maximum entre deux stations du réseau et on pourra donc appliquer l'auto-allocation.

Le débit échangé dans le réseau est estimé à environ 70 K octets/s (dont la moitié est utile).

Pour écouler le débit et pour des questions de tolérance aux pannes, on prévoit une *autoroute* formée de  $k = 3$  lignes banalisées.

Les stations sont réalisées à base de microprocesseurs M 6800.

Le coupleur que l'on propose (voir chapitre 4) permet une cadence de transfert de 40 K octets/s en mode half-duplex (la moitié en full-duplex).

N.B. On veut une cadence élevée pour que le débit global du réseau puisse être écoulé même en cas de panne d'une partie de l'autoroute.

Les coupleurs d'une station peuvent être organisés de deux manières :

a) Coupleurs unidirectionnels (figure 21)

On aura un coupleur spécialisé en émission (il émet sur la première ligne libre) et  $k = 3$  coupleurs (un par ligne) spécialisés en réception. Dans ce cas, le coupleur d'émission comprend en particulier le mécanisme *d'auto-allocation d'autoroute*.

Pour augmenter la fiabilité du mécanisme de communication, on introduit de la redondance dans les coupleurs : chaque coupleur possède tous les mécanismes (auto-allocation, programmes, ..) pour *pouvoir* fonctionner dans les deux sens (émission ou réception). Le sens de fonctionnement d'un coupleur est un paramètre. A un moment donné on a un coupleur paramétré en émission et les autres en réception.

Si un coupleur de réception tombe en panne, la station peut toujours fonctionner (elle reçoit sur les autres lignes, vu qu'elles sont banalisées). Si le coupleur d'émission tombe en panne, pour que la station puisse fonctionner (et donc émettre), alors on reconfigure : un parmi les coupleurs de réception change de sens, devient coupleur d'émission et l'état final est équivalent au cas où un coupleur de réception tombe en panne.

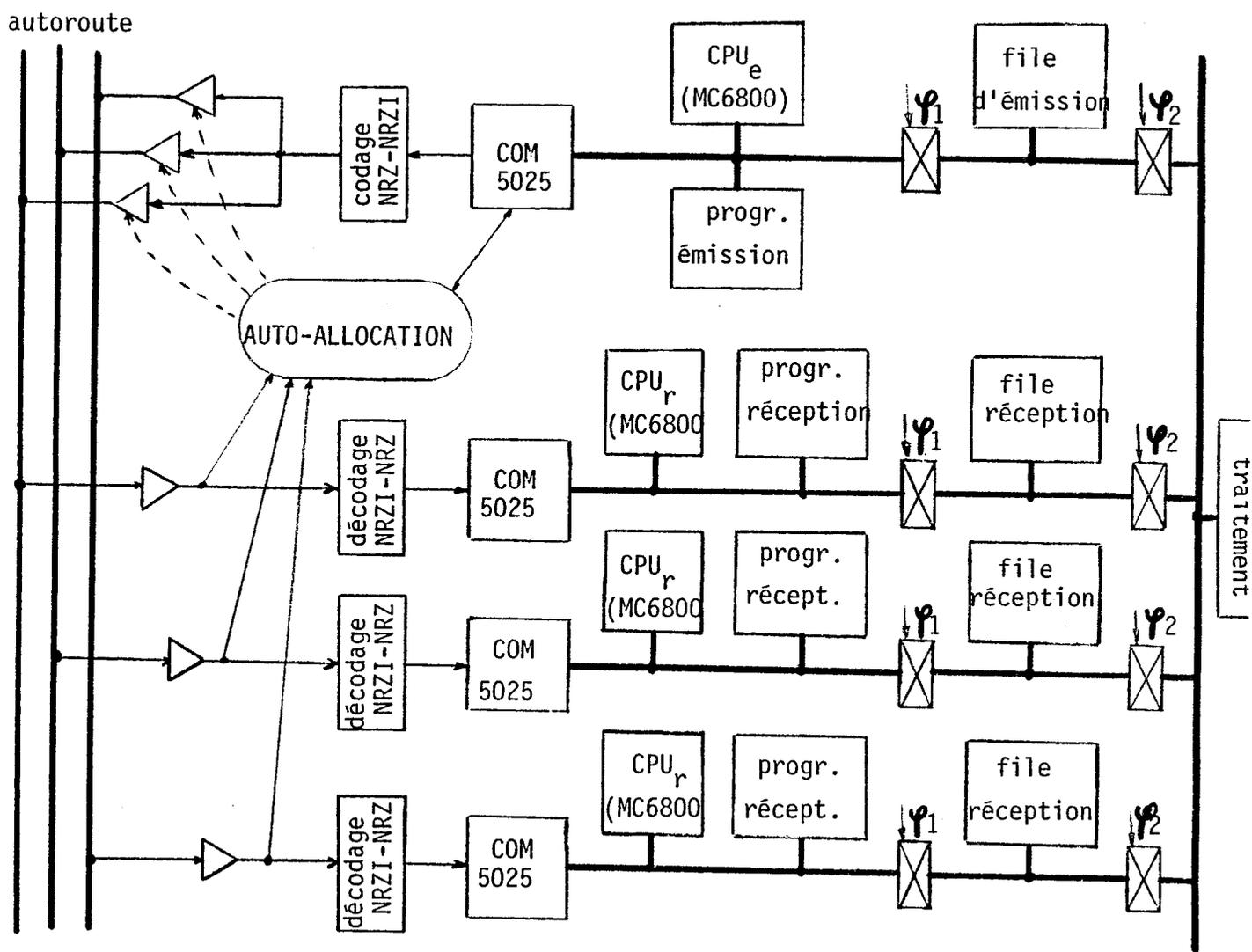


FIG.21. Station avec des coupleurs unidirectionnels

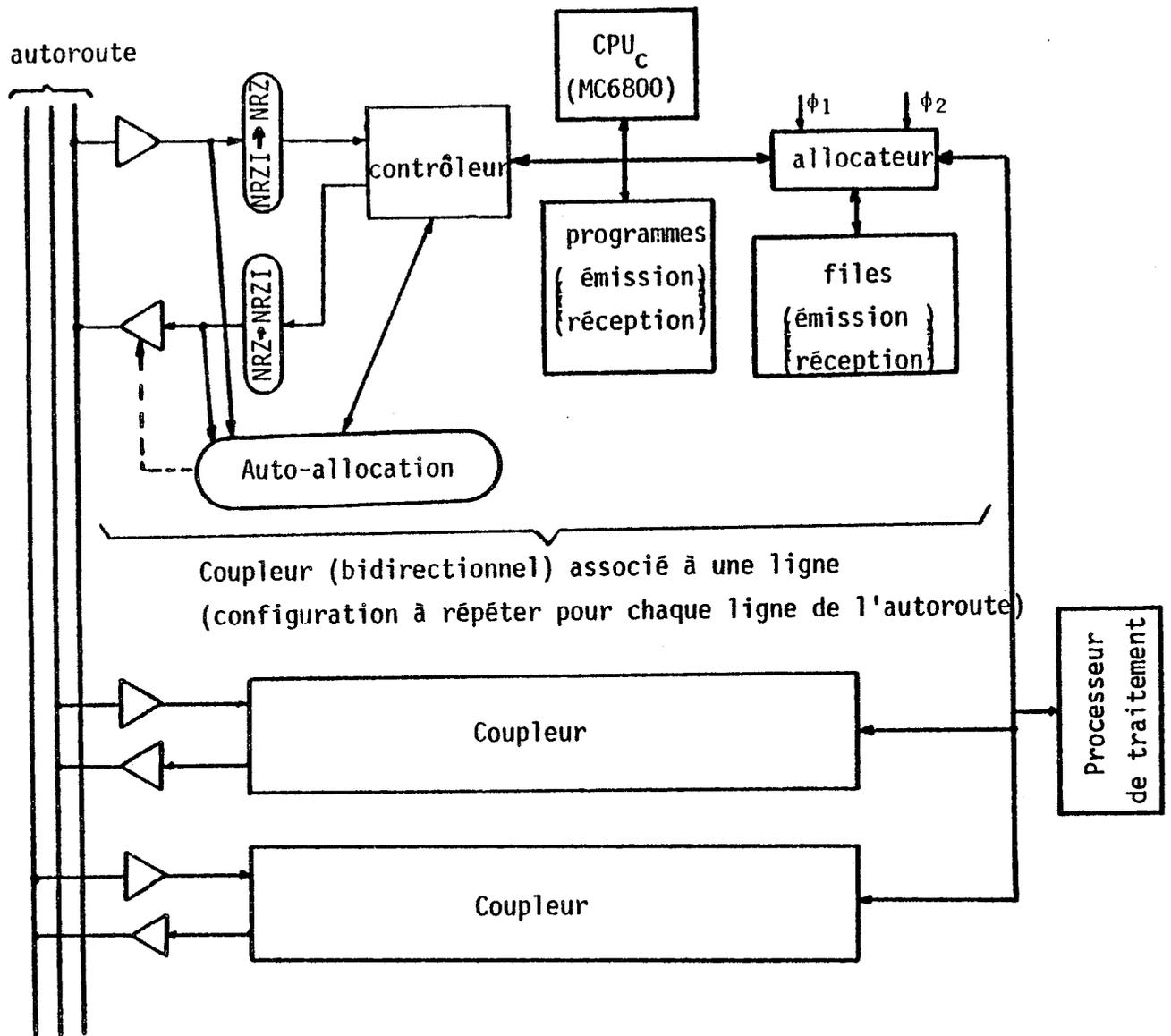


FIG.22. Station avec des coupleurs bidirectionnels

### b) Coupleurs bidirectionnels (figure 22)

Dans ce cas on aura  $k = 3$  coupleurs, chacun associé à une ligne de l'auto-route. Chaque coupleur fait l'émission/réception sur la ligne associée et il travaille en half-duplex (autrement il s'écouterait parler !). Donc, la cadence de 40 K octets/s par coupleur est maintenue.

Les messages générés (qui doivent être émis) par la station sont répartis sur les coupleurs avec une politique d'uniformisation de leur charge : par exemple, chaque nouveau message généré est attribué au coupleur dont la file d'attente est la plus petite.

Une telle organisation supporte très bien les pannes : si un coupleur tombe en panne, la ligne qui lui est associée devient inutilisable pour la station, mais elle pourra toujours émettre/recevoir sur les autres lignes.

### Comparaison des solutions a) et b)

- . Point commun : les deux solutions sont équivalentes du point de vue des performances.
  
- . Avantage de la solution b par rapport à a) : la solution b) comporte  $k = 3$  coupleurs et donc est moins chère que la solution a) qui comporte  $k + 1 = 4$  coupleurs.  
La solution b) étant moins complexe, son taux de pannes (et donc du nombre d'interventions) est plus petit que pour la solution a).

- . Avantage de la solution a) par rapport à b) : dans la solution a) il y a un seul coupleur d'émission et les messages arrivent aux destinataires dans l'ordre de leur génération par la partie traitement de la station. Dans la solution b) cette propriété n'est pas respectée : un message  $m_2$  généré après  $m_1$  peut arriver à destination avant  $m_1$  si  $m_2$  a été attribué à un coupleur dont la ligne est peu chargée. Ceci peut être incompatible avec la cohérence fonctionnelle du système.

Par ailleurs, en cas de panne d'un coupleur, on a :

- avec la solution a) : inutilisation *partielle* d'une ligne par la station : on ne peut plus recevoir sur cette ligne, mais on peut émettre sur elle ;
- avec la solution b) : inutilisation *totale* d'une ligne par la station : on ne peut ni émettre ni recevoir sur cette ligne.

Le choix entre les solutions a) et b) va se faire suivant les contraintes de fonctionnement du système :

- . pour les stations qui nécessitent l'émission des messages dans l'ordre de leur génération, on choisit la solution a) (coupleurs unidirectionnels),
- . pour les autres on prend la solution a) ou la solution b) suivant que le facteur prépondérant est la standardisation (même réalisation dans toutes les stations) ou le coût.

L'étude du système qui nous concerne a montré qu'il n'est pas nécessaire d'avoir l'émission des messages dans l'ordre de leur génération. Ceci nous a amené à choisir la solution b) : un coupleur bidirectionnel (émission/réception) par ligne et par station, comme étant moins chère et plus fiable.

Un aspect très important dans l'autocommutateur est constitué par la sûreté de fonctionnement. Au moment où ce document a été écrit, les problèmes concernant la sûreté de fonctionnement dans le réseau constitué par l'autocommutateur, n'ont pas encore reçu de solutions définitives. Néanmoins, l'étude menée conjointement par l'Equipe d'Architecture des Ordinateurs de l'ENSIMAG et le CNET, a déjà permis de dégager les principes pour aborder et résoudre le problème de la sécurité [COU].

Toujours dans l'optique de l'architecture répartie, on a choisi une politique de décentralisation des fonctions concernant la sécurité.

De par leur grande fiabilité (à cause de la redondance), leur puissance de traitement et leur habilité à communiquer avec le reste du réseau, les coupleurs seront les endroits privilégiés pour l'implémentation des fonctions concernant la sécurité.

Le mécanisme de communication constitue le noyau "sûr" du système. A cause de la redondance, on suppose que sur chaque station le mécanisme de communication fonctionne toujours (ou presque) : il existe toujours un coupleur en état de fonctionnement.

Pour la sécurité, il y a deux types de problèmes :

- . détection de la panne,
- . décision à prendre en cas de panne.

Pour la détection des pannes, la politique générale choisie est celle de l'auto-test des organes, complété par la suspicion mutuelle.

Chaque station est composée de plusieurs processeurs : les  $k = 3$  coupleurs et le(s) processeur(s) de traitement.

Chaque processeur dispose d'un mécanisme de détection de panne : la catastrophe algorithmique [COU] ; c'est un mécanisme d'auto-test de micro-ordinateur (microprocesseur + bus données/adresses + mémoire). Il est principalement composé de :

- . protection mémoire,
- . relance périodique d'un chien de garde,
- . détection des codes invalides.

Les travaux de B. Courtois [COU] montrent que la catastrophe algorithmique est un moyen simple, sûr et très efficace pour la détection des pannes.

A part la catastrophe algorithmique, chaque coupleur disposera d'un mécanisme d'auto-test spécifique au système de communication. Ce dernier concerne : la connexion physique à la ligne, l'auto-allocation, le codage/décodage NRZ ↔ NRZI de l'information, le mécanisme d'accusé de réception, le contrôleur de procédure.

Dès qu'une "anomalie" (qui peut être due à un quelconque coupleur connecté à la même ligne) de communication est détectée par le coupleur, celui-ci se déconnecte physiquement (à l'aide d'un relais) de la ligne : il se met en mode local.

N.B. Une panne détectée par la catastrophe algorithmique du coupleur, le déconnectera aussi de la ligne.

En mode local, la sortie série est rebouclée sur l'entrée série et en déroulant des séquences de test (en s'envoyant à soi-même un message), le coupleur détectera s'il y a panne chez lui. S'il n'y a pas panne, le coupleur se reconnecte à la ligne. Cet auto-test permet donc de déconnecter les coupleurs fautifs qui par leur présence pourraient empêcher les autres de fonctionner.

Au niveau matériel, les anomalies qui déclenchent l'auto-test de la communication sont :

- . l'auto-allocation détecte un conflit à l'émission d'un "1" sur la ligne ; la ligne réalisant le "ou" logique, elle devrait se positionner à "1" et donc on ne devrait pas avoir de conflit ;
- . on détecte "ligne libre" pendant l'émission, quand la ligne devrait être occupée ;
- . la ligne reste à "1" pendant plus de 8.0 (durée d'un octet), ce qui normalement ne devrait pas se produire vu notre mécanisme de communication : fonction "ou" réalisée par la ligne, format SDLC des messages, codage NRZI de l'information émise sur la ligne.

Au niveau logiciel, l'auto-test de la communication dans un coupleur est déclenché principalement par :

- . l'auto-allocation qui détecte un conflit après l'émission de l'adresse expéditeur (quand normalement il ne doit plus y avoir de concurrents) ;
- . le nombre d'accusés de réception absents dépasse un certain seuil ;
- . l'attente à l'émission d'un message dépasse un certain seuil.

Les pannes non détectées par l'auto-test (local) des organes seront à la charge de la "suspicion mutuelle" : c'est le mécanisme par lequel une station doute du bon fonctionnement d'une autre. Cette suspicion se trouve à deux niveaux :

- . au niveau de la communication (par le mécanisme d'accusé de réception) : un coupleur émetteur qui ne reçoit pas d'accusé de réception, doute du bon fonctionnement de son correspondant ;
- . au niveau fonctionnel : une station (sa partie traitement) met en doute une autre quand cette dernière ne respecte pas l'algorithme de dialogue fonctionnel (pas de réponse, réponses incohérentes, ..).

Dans le contexte d'un système décentralisé, où les stations ont des prérogatives équivalentes, la détection des pannes par la suspicion mutuelle est assez délicate, vu que celui qui suspecte peut être à son tour suspecté. La suspicion mutuelle mettra en jeu tout le réseau par les messages de suspicion/vérification échangés entre les stations.

Le principe de détection qui a été choisi repose sur la remarque réaliste suivante : si une station suspecte plusieurs partenaires, ou si elle est suspectée par plusieurs, c'est cette station qui a la plus grande probabilité d'avoir un organe fautif.

Toute panne détectée (par auto-test ou suspicion) isolera l'organe fautif du reste du système pour éviter la propagation des fautes. Le mécanisme de communication (de la station concernée par l'organe fautif) avertira (par des messages sur l'autoroute) le reste du réseau de l'occurrence de la panne. Il va déclencher sur l'organe fautif des séquences de test qui permettront de déterminer la gravité de la panne (panne permanente, transitoire, localisation de la panne) pour une éventuelle réinsertion dans le système.

En conclusion on peut dire que le mécanisme de communication (coupleurs) joue un rôle essentiel dans la sûreté de fonctionnement du réseau. Il est le noyau "sûr" du système. De par sa modularité, on peut facilement réaliser la redondance (en augmentant la "taille" de l'autoroute) pour augmenter la fiabilité. De par sa grande fiabilité, on peut toujours communiquer avec une station et savoir ce qui s'y passe : une station n'est jamais muette.

## CHAPITRE 4

### RÉALISATION DU COUPLEUR

Par coupleur nous entendons l'implémentation du mécanisme de communication représenté par les deux derniers niveaux de la hiérarchie de dialogues : protocole de base et échange élémentaire.

Pour la réalisation du coupleur, on s'est placé dans le contexte de l'auto-commutateur téléphonique présenté au chapitre 3.

Pour des questions de performance, modularité, sûreté de fonctionnement et en tenant compte du faible coût des microprocesseurs, on a envisagé le coupleur sous la forme d'un processeur d'entrées/sorties.

Du point de vue de la structure, le coupleur aura une architecture classique à base de microprocesseurs (figure 23).

Dans tout ce qui suit, on considèrera le cas du coupleur bidirectionnel : au coupleur est associée une ligne série sur laquelle il émet/reçoit en half-duplex.

Le coupleur est composé des parties suivantes :

- . mécanisme d'auto-allocation,
- . codage/décodage NRZ - NRZI de l'information au niveau de l'échange élémentaire,
- . contrôleur de format : émet/reçoit le message en format SDLC,
- . mécanisme d'accusé de réception,
- . files d'attente par l'intermédiaire desquelles a lieu le dialogue entre le coupleur et le processeur de traitement,
- . microprocesseur, avec son programme qui contrôle tout le micro-système.

On a choisi comme microprocesseur le MC 6800 de Motorola (ou SF.F 96800 de SESCOSEM [NTEC1]). Des études comparatives du MC 6800 et du 8080 d'Intel (les microprocesseurs qui couvrent la plupart du marché), n'ont pas montré de supériorité déterminante de l'un sur l'autre [PRE].

Par contre, le MC 6800 nous était déjà très familier ; il dispose d'un langage d'utilisation aisée et nos contacts industriels nous orientaient vers ce choix.

Pour obtenir de hautes performances, on va utiliser la version rapide MC 6800 B (fréquence maximale = 2 MHz). On a pris un cycle microprocesseur d'environ 0,6 microsecondes et on obtient un débit maximum permis par le coupleur de 41,1 K octets/s.

Le contrôleur (de format SDLC) choisi est le COM 5025 de SMC (ou son équivalent 2652 MPCC de RTC), tenant compte de sa disponibilité sur le marché et de son insertion facile dans le micro-système.

#### Observation :

L'allocateur mémoire qui gère l'accès à la zone commune constituée par les files d'attente, a été réalisé en tenant compte de la caractéristique du microprocesseur MC 6800 : le cycle microprocesseur est constitué de deux phases et l'accès à la mémoire est fait pendant la deuxième phase.

Pour les deux microprocesseurs (du coupleur et de la partie traitement), on utilisera des phases complémentaires.

On aura une horlogerie unique qui délivre les phases  $\phi 1$  et  $\phi 2$  d'un cycle microprocesseur. Le microprocesseur de traitement accèdera aux files pendant  $\phi 2$  et le microprocesseur coupleur pendant  $\phi 1$ .



On a fabriqué deux mots d'état :

. ETAE pour l'émission ; il est composé de :

- bit 7 = ACNF + TSA : indique si un conflit ou une erreur d'émission s'est manifesté ;
- bit 6 = TBMT . TXACT : indique si le contrôleur est prêt à émettre un nouvel octet ;
- bit 5 = RACK .  $\overline{\text{FIN}}$  : indique l'arrivée de l'accusé de réception ;
- bit 4 = MACC .  $\overline{\text{FIN}}$  : indique si la fenêtre de durée TACC s'est écoulée ;
- bit 3 =  $\overline{\text{FIN}}$  : indique la fin du message
- bit (2:0) = 000

. ETAR pour la réception ; il est composé de :

- bit 7 = RSA +  $\overline{\text{RXACT}}$  : indique la fin du message reçu ou une erreur de réception ;
- bit 6 = RDA : indique qu'un nouvel octet a été reçu ;
- bit (5:0) = 000000

Ces deux mots d'état ont été conçus dans le but d'optimiser les séquences programme qui contrôlent l'émission/réception effectives des messages.

## Réalisation du mécanisme d'auto-allocation

Le mécanisme d'auto-allocation, dont les principes ont été présentés au chapitre 2, communique avec le reste du coupleur par les variables suivantes :

### . Entrées :

DEM : demande d'émission ;  $DEM = 1 \Leftrightarrow$  demande

FIN : fin émission ;  $FIN = 0 \Leftrightarrow$  fin

TX : valeur (logique) émise sur la ligne

RX : valeur (logique) de la ligne.

### . Sorties :

ACNF : arrêt de l'émission pour cause de conflit ;  $ACNF = 1 \Leftrightarrow$  arrêt

PERM : permission d'émission du message

CNX : connexion à la ligne de l'amplificateur d'émission ;

$CNX = 1 \Leftrightarrow$  connexion.

Pour l'auto-allocation on a choisi la stratégie a.2 (cf. § 2.4.2.) : avant les bits du message, on envoie sur la ligne une impulsion (signal de candidature) dont le rôle est de réduire le déphasage entre les stations concurrentes à l'émission.

N.B. Quand  $PERM = 0$ ,  $CNX = 1$ , on envoie sur la ligne le signal de candidature.

Quand  $PERM = 1$ ,  $CNX = 1$ , on émet sur la ligne les bits du message.

On rappelle les deux paramètres temporels du mécanisme :

. TCNF : durée détectrice de conflit. On a vu qu'il faut  $TCNF \geq g(l)$  où  $l$  = longueur de la ligne et  $g(l)$  = temps de propagation du signal d'un bout à l'autre de la ligne ;

. TOCC : durée détectrice de ligne libre. Avec le format SDLC et émission en NRZI, on a vu qu'il faut  $TOCC \approx 8.\theta$  ( $\theta$  = durée du bit).

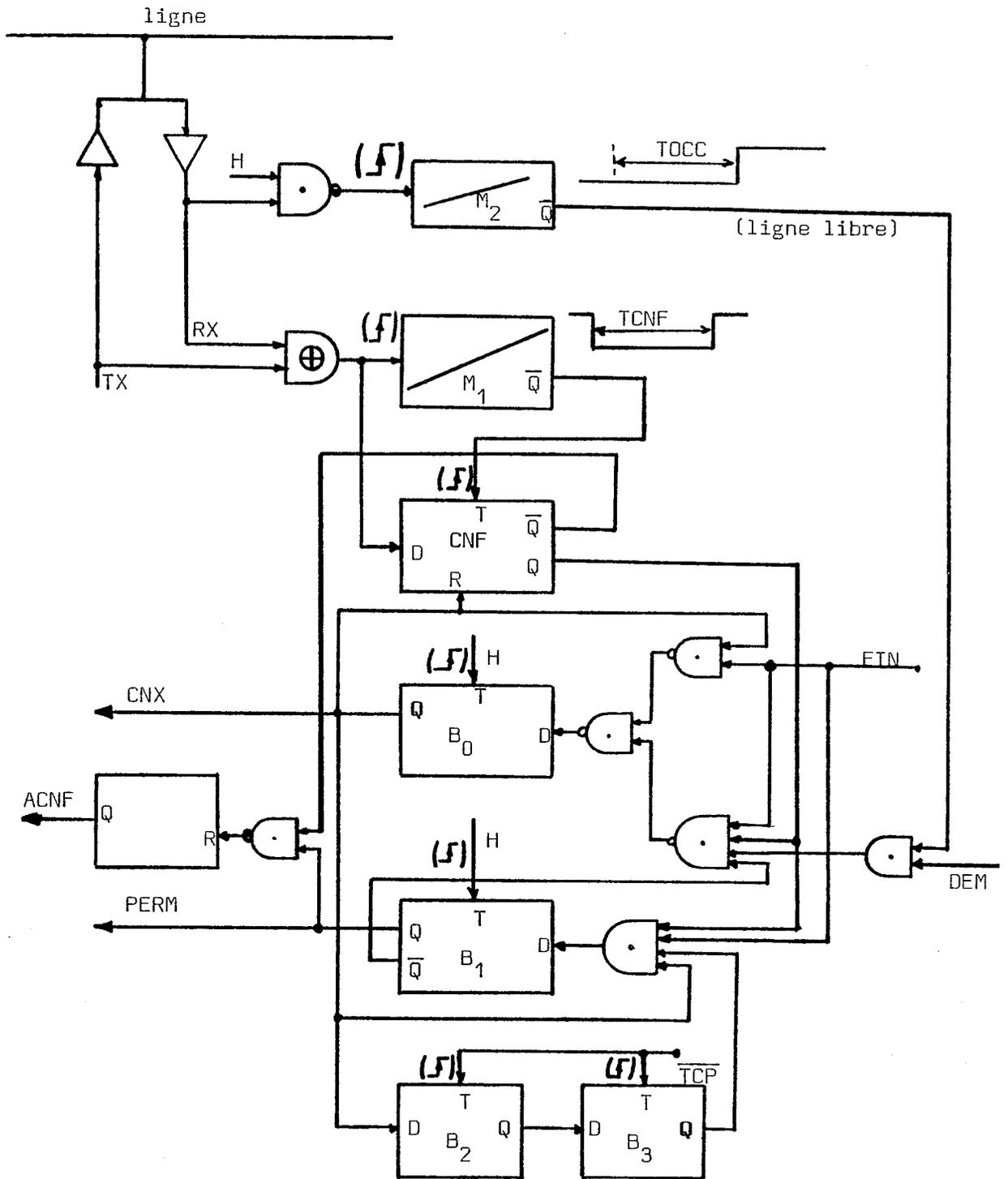


FIG.24. Schéma du mécanisme d'auto-allocation

$M_1$  - monostable détecteur de conflit

CNF- mémorisation du conflit

$M_2$  - monostable détecteur de "ligne libre"

$B_2, B_3$ -bascules qui indique la fin du signal de candidature

$\overline{TCP}$ - horloge d'émission de la station

H - horloge (rapide) de base : fait évoluer l'automate

## Contrôleur COM 5025 de SMC (ou 2652 de RTC [NTEC2])

Pour les renseignements détaillés, on se reportera à la notice du constructeur [NTEC3].

Ce circuit manipule plusieurs formats parmi lesquels le format SDLC tel que nous l'avons décrit au chapitre 2.

A part les broches permettant sa connexion au bus (données + adresses) du microprocesseur, le circuit échange avec son environnement les signaux suivants :

### . Partie réception :

RSI (entrée) : entrée donnée série

RCP (entrée) : horloge réception : prend en compte l'entrée série sur le front montant

SFR (sortie) : indique la reconnaissance d'un flag

RXENA (entrée) : validation de la réception

RXACT (sortie) : indique la reconnaissance d'un message

RDA (sortie) : indique qu'un nouveau caractère (octet) a été reçu

RSA (sortie) : indique la fin du message ou une erreur de réception

### . Partie émission :

TSO (sortie) : sortie donnée série

TCP (entrée) : horloge d'émission : la donnée sort sur le front montant (elle est donc stable sur le front descendant)

TBMP (sortie) : indique que le contrôleur est prêt à recevoir (en vue d'émission) un nouveau caractère (octet)

TSA (sortie) : indique que le registre de transmission est vide : c'est une erreur (on ne fournit pas les octets à la cadence imposée)

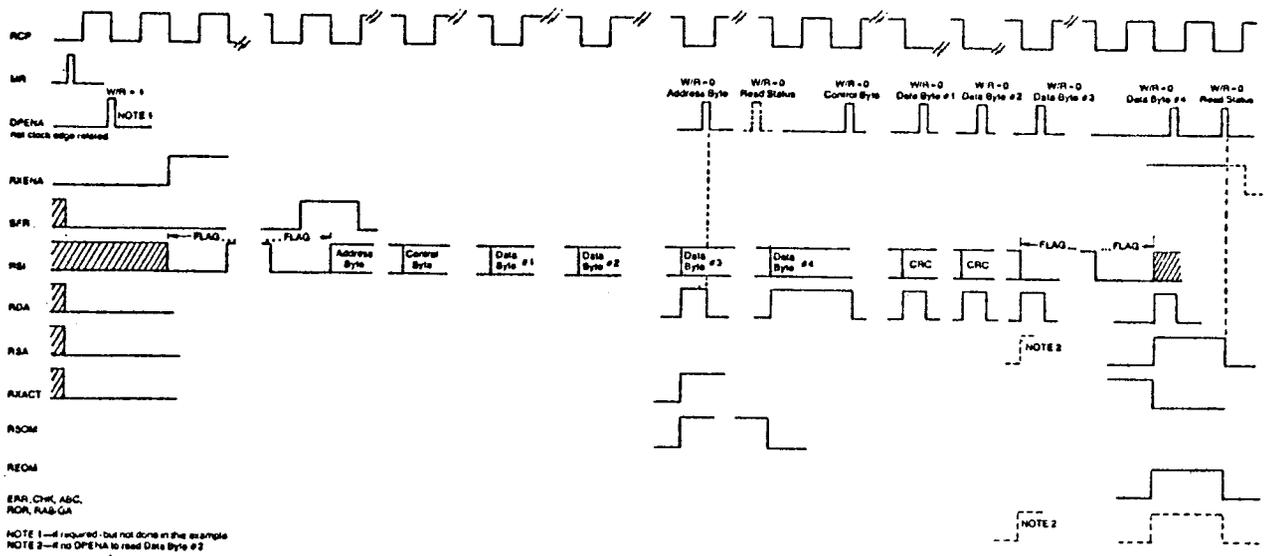
TXENA (entrée) : validation de l'émission (permission d'émission)

TXACT (sortie) : indique que la sortie série est valide.

Le contrôleur possède sept registres internes de huit bits, que l'on peut accéder par programme :

- . RD : registre réception : contient le caractère (octet) reçu,
- . TD : registre transmission : contient le caractère (octet) à émettre,
- . SSA : registre adresse : contient l'adresse de la station,
- . RS : état réception,
- . TXS : état/contrôle transmission,
- . MC : registre contrôle : permet de choisir le protocole, le code correcteur, etc ..
- . RLS : registre de sélection longueur : permet de choisir la longueur (nombre de bits) du caractère.

RECEIVER TIMING



TRANSMITTER OPERATION

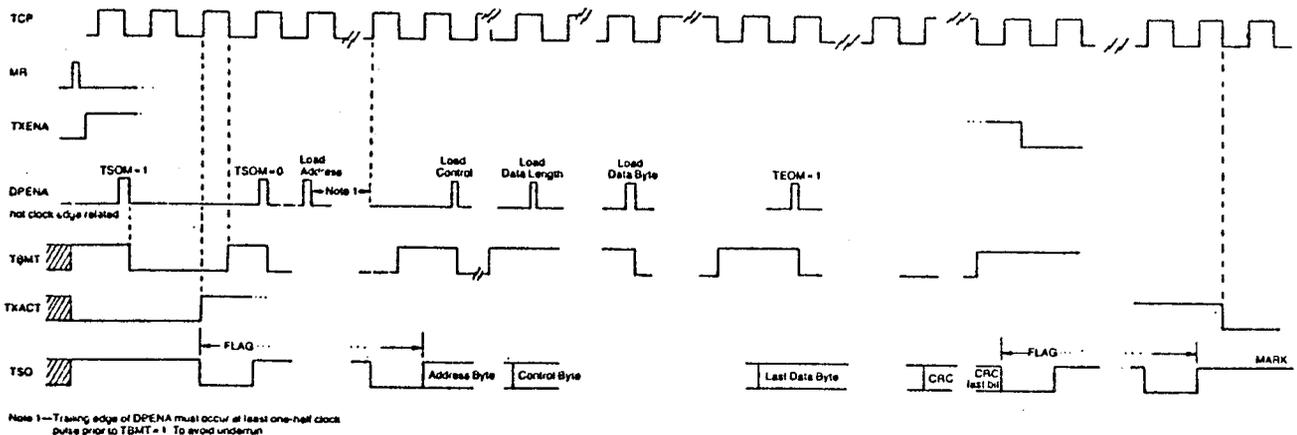


FIG.24. L'émission/réception des messages avec COM5025 (tiré de [NTEC3])



## Codage/décodage de l'information émise sur la ligne

Dans le réseau, le même support physique (la liaison) sert pour véhiculer la donnée et l'horloge.

A l'émission a lieu l'opération de codage de l'information (mélange sur le même signal de la donnée et de l'horloge) et à la réception a lieu le décodage (extraction de l'horloge et de la donnée à partir du signal), par échantillonnage et resynchronisation sur les transitions (conformément à l'annexe 3). Chaque station dispose d'une horloge d'échantillonnage rapide de même période  $h$ .

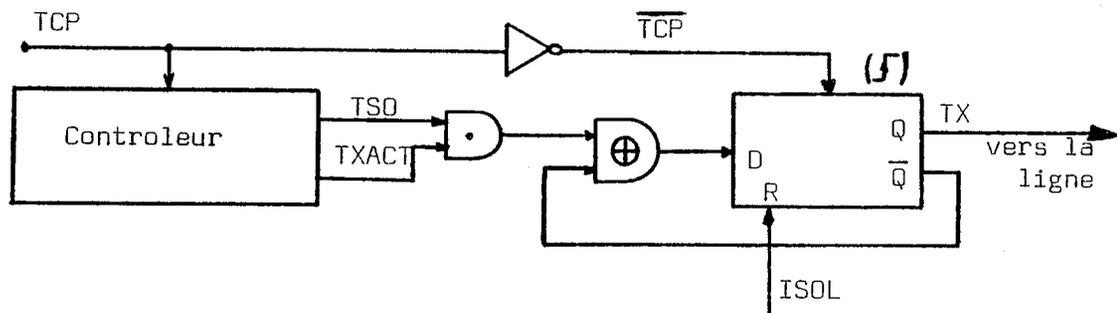
Compte tenu du format SDLC des messages, on choisit la transmission en code NRZI : changement de polarité (transition) pour "0", pas de changement pour "1".

La durée  $\theta$  est un multiple de la période  $h$  de l'horloge d'échantillonnage

$$\theta = (2m + 1).h$$

Le choix de  $\theta$  (et donc de  $m$ ) est discuté en annexe 3.

### Codage NRZ - NRZI



L'horloge TCP est l'horloge d'émission calibrée à la période  $\theta$  du bit. La donnée TSO est stable sur le front descendant de TCP (front montant de  $\overline{\text{TCP}}$ ). Quand la station est isolée de la ligne (ISOL = 0), le codage est inhibé. Tant que la donnée série TSO n'est pas valide (TXACT = 0), on considère que la donnée est "0" (donc changements de polarité) et ceci dans le but d'occuper la ligne.

### Décodage NRZI - NRZ

Pour le mécanisme de décodage, on extrait la donnée et l'horloge à partir du signal.

Si l'on émet  $0 \underbrace{1 \ 1 \ \dots \ 1}_n \ 0$  alors sur la ligne on a une impulsion de durée

$$T = (n + 1)\theta + q$$

où  $\theta$  : durée du bit

$q$  : dispersion. Sachant que la dispersion maximale est  $p$  on a  $-p \leq q \leq p$ .

N.B. La dispersion est introduite par l'imprécision des horloges d'émission, par l'auto-allocation, par la ligne.

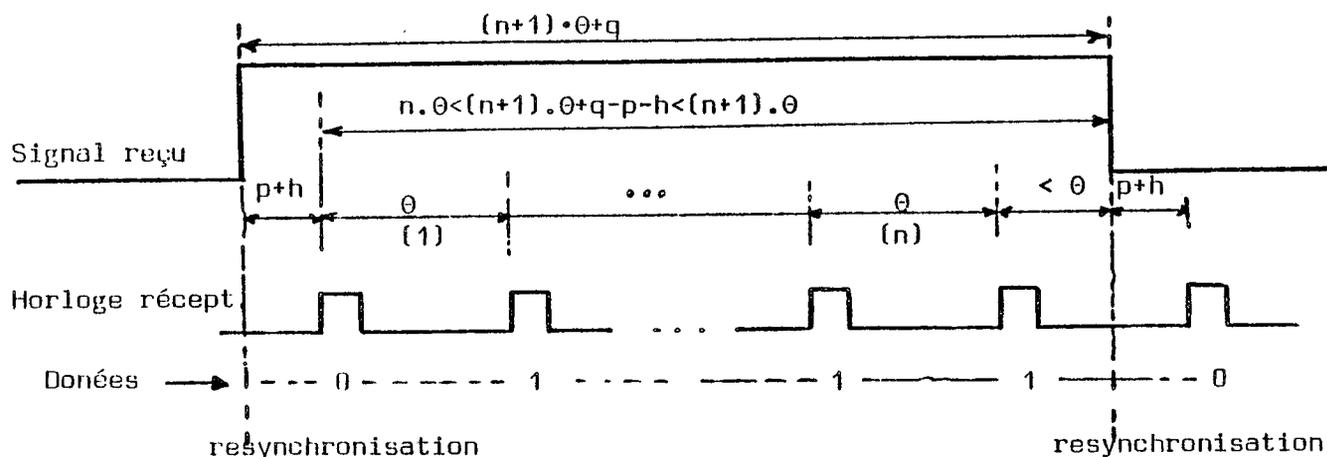
Donc si l'on reçoit une impulsion de durée  $T = (n + 1)\theta + q$  ( $n \geq 0, -p \leq q \leq p$ ) on doit reconnaître les bits suivants :

"0" : correspond à la transition de début de l'impulsion

$n \times$  "1"

"0" : correspond à la transition de fin de l'impulsion

On a le chronogramme suivant :



Pour ne pas se resynchroniser sur des parasites, il faut que ceux-ci soient filtrés. La durée  $\theta$  du bit est un multiple de  $h$  (période de l'horloge d'échantillonnage) :  $\theta = (2.m + 1).h$ . Dans l'annexe 3 on montre qu'il faut  $\theta > 2 \cdot \frac{p + (1 + \alpha).h}{1 - 15.\alpha}$  où  $p$  est la dispersion et  $\alpha$  l'imprécision de l'horloge d'échantillonnage.

On obtient qu'il faut  $(m \cdot (1 - 15.\alpha) - \frac{1}{2} - \frac{17.\alpha}{2}).h > p$

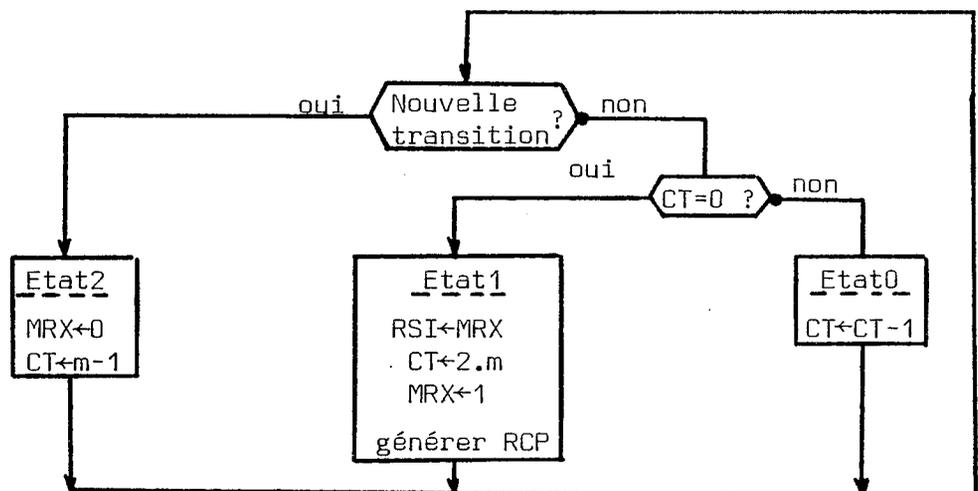
En prenant un quartz qui donne une grande précision ( $\alpha \approx 10^{-4}$ ), on voit que  $m(1 - 15.\alpha) - \frac{1}{2} - \frac{17.\alpha}{2} > m - 1$  si  $m \leq 332$  ( $\theta \leq 665.h$ ).

Sous ces conditions on va dire que le mécanisme de décodage admet une dispersion maximale  $p = (m - 1).h$ .

Notons :

- RX valeur logique du signal (filtré) de la ligne
- RCP horloge de réception extraite du signal
- RSI donnée extraite du signal
- CT contrôleur temporel
- H horloge d'échantillonnage (comptage).

L'automate de décodage NRZI  $\rightarrow$  NRZ a l'algorithme suivant :



N.B. Cet automate synchrone évolue sur l'horloge H.

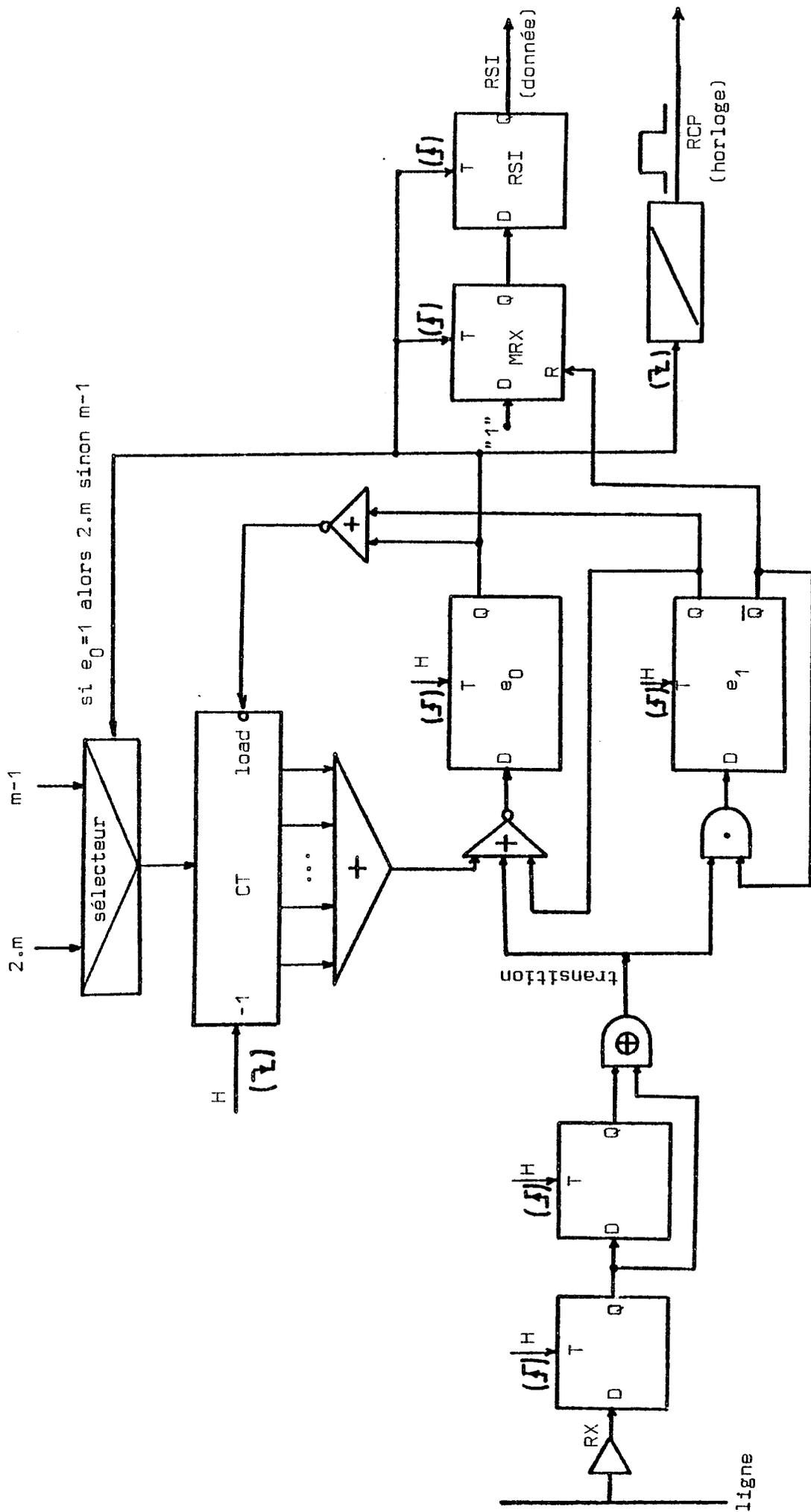


FIG.25. Schéma du décodage NRZI → NRZ

## Files d'attente

Les files d'attente constituent le seul moyen de dialoguer entre le coupleur et le processeur de traitement.

On aura deux files par coupleur :

- . FE : file d'émission (contient les messages à émettre) : elle est remplie par le processeur de traitement et vidée par le coupleur
- . FR : file de réception (contient les messages reçus) : elle est remplie par le coupleur et vidée par le processeur de traitement.

Les files sont du type : premier arrivé, premier servi (FIFO).

Les files sont implémentées dans une zone mémoire partageable par les deux processeurs.

**N.B.** On n'utilise pas des circuits intégrés qui réalisent directement la fonction FIFO, puisque ce choix nous oblige à avoir une zone tampon par où transitent les messages. Tout ceci augmente la charge du coupleur les performances restant les mêmes. D'autre part avec des files implémentées en mémoire le système est plus souple .

On écarte la solution qui consiste à chaîner des blocs tampon, vu que la gestion des pointeurs de chaînage est onéreuse avec un microprocesseur.

On utilise plutôt une structure circulaire : pour une file F on alloue un espace mémoire compact (cases mémoire contigues) de taille NF et l'incrémentement des pointeurs se fait modulo NF.

Chaque file F a un pointeur de début (DF), un pointeur de fin (FF) et une variable (TF) qui indique la taille de l'espace occupé (ou libre) de la file. Entre le pointeur de début et le pointeur de fin il y a l'espace occupé (messages en attente) et entre le pointeur de fin et le pointeur de début, il y a l'espace libre.

La zone mémoire allouée à la file F est comprise entre les adresses (constantes du système) ADF et AFF et on a :

$$NF = AFF - ADF + 1.$$

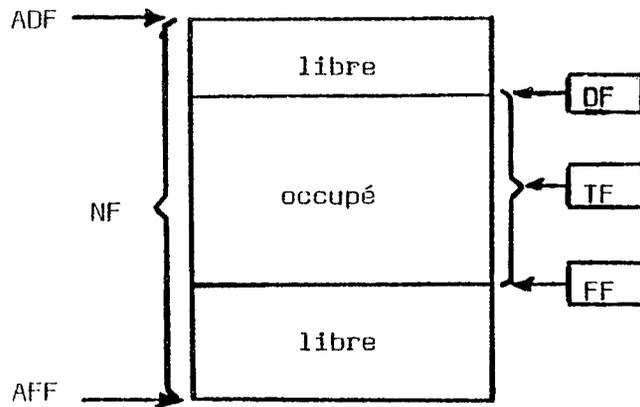
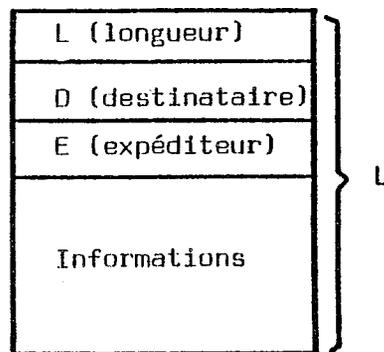
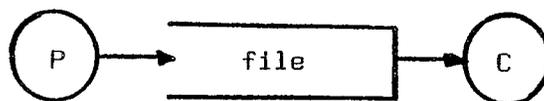


FIG.26. Structure d'une file (circulaire) F

Chaque message (bloc) de la file a la structure suivante :



La file permet un dialogue du type producteur (P) - consommateur (C) :



Soit :

F : la file

N : taille maximale de la file

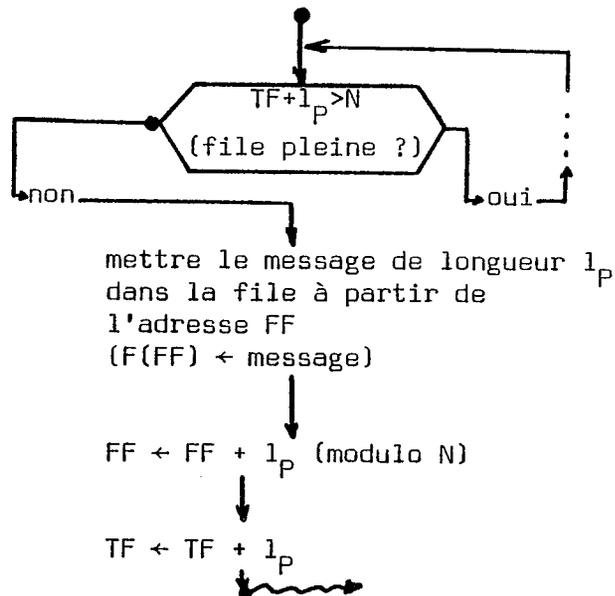
DF : pointeur de début

FF : pointeur de fin

TF : taille de la zone occupée.

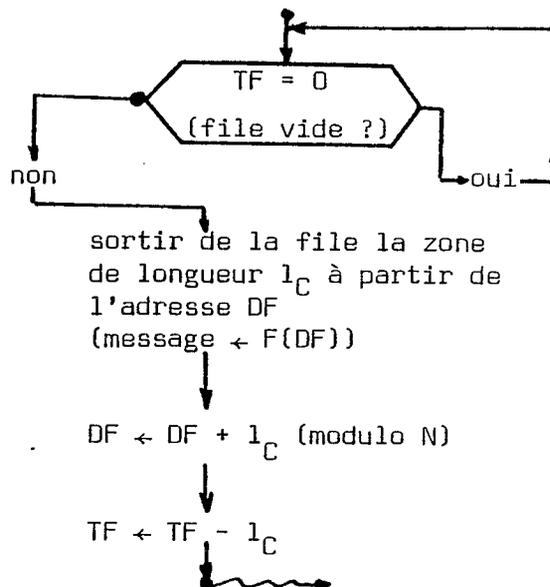
Entrée d'un message dans la file : faite par P

soit  $l_p$  la taille du message



Sortie d'un message de la file : faite par C

Soit  $l_c$  la longueur du message en tête de file (si elle n'est pas vide)



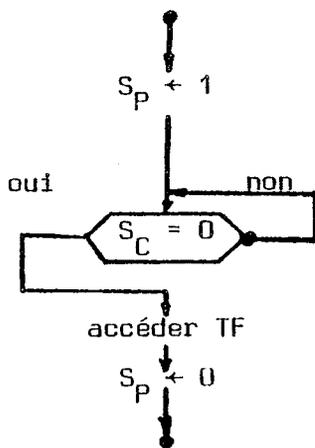
On voit que :

- . DF et la zone occupée de la file sont accédés seulement par le consommateur,
- . FF et la zone libre de la file sont accédés seulement par le producteur.

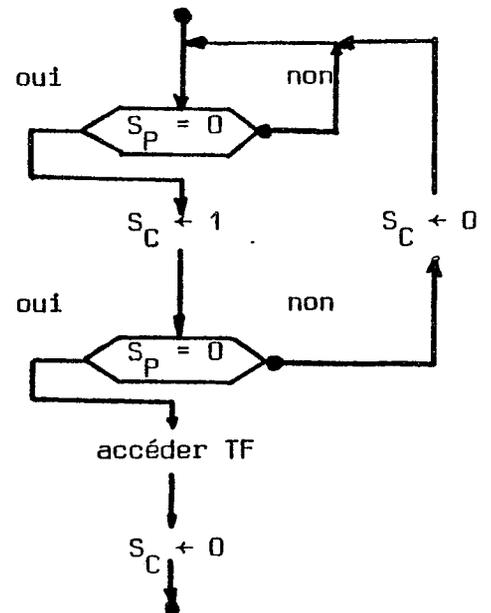
La seule variable qui est accédée par les deux processus est TF. Pour son accès, on doit donc prévoir un mécanisme de mutuelle exclusion.

Comme les microprocesseurs n'ont pas une instruction de type "test and set", ce qui permettrait d'avoir un mécanisme de mutuelle exclusion symétrique, on dissymétrise le mécanisme : on associe les indicateurs  $S_P$ ,  $S_C$  respectivement au producteur et au consommateur et on définit les primitives d'accès :

producteur (P)



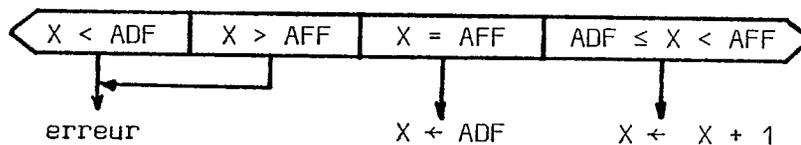
consommateur (C)



En conclusion, pour une file F (FE ou FR), on aura :

- . espace d'implantation (zone contigue de cases mémoire) : compris entre les adresses (constantes du système) ADF, AFF et de taille (constante du système)  $NF = AFF - ADF + 1$  ;
- . DF : pointeur (dynamique) de début de file ;
- . FF : pointeur (dynamique) de fin de file ;  
N.B. Il faut toujours :  $ADF \leq DF \leq AFF$   
 $ADF \leq FF \leq AFF$
- . TF : taille (dynamique) de la file ;
- .  $S_{PF}$ ,  $S_{CF}$  : indicateurs associés respectivement au producteur et au consommateur de la file, pour réaliser la mutuelle exclusion à TF.

Soit X un pointeur avec lequel on accède aux informations de la file.  
L'avancement de X se fait selon la séquence :



La file de réception FR sera caractérisée par :

- . ADFR, AFFR,  $NR = AFFR - ADFR + 1$
- . DFR, FFR
- . TFR : taille de l'espace libre
- . SR, STR : indicateurs associés respectivement au coupleur et au processeur de traitement.

La file d'émission FE sera caractérisée par :

- . AD FE, AF FE,  $NE = AF FE - AD FE + 1$
- . DFE, FFE
- . TFE : taille de l'espace occupé
- . SE, STE : indicateurs associés respectivement au coupleur et au processeur de traitement
- . ETE : variable par laquelle le coupleur indique au processeur de traitement que le message en tête de file (pointé par DFE) n'a pas pu être correctement émis (manque l'accusé de réception).

## Programmation du coupleur

Au niveau du logiciel, le coupleur comprend deux parties :

- . logiciel de sécurité,
- . logiciel de "pure" communication.

Le logiciel de sécurité comprend l'auto-test (local) de la communication, ainsi que les fonctions concernant la sécurité globale du système. Comme nous l'avons indiqué au § 3.3, le problème de la sûreté de fonctionnement de l'autocommutateur téléphonique est encore à l'étude.

Pour l'instant nous avons complètement défini le logiciel de "pure" communication. Il est composé de deux programmes (processus) :

- . programme d'émission,
- . programme de réception : c'est la routine d'interruption.

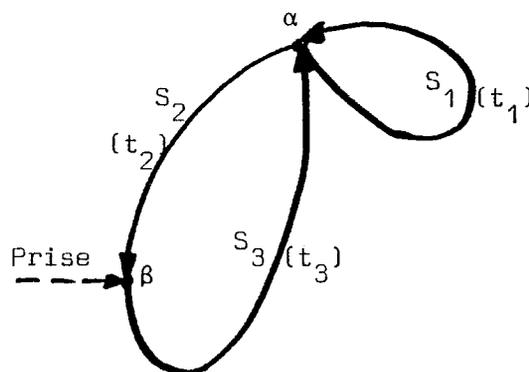
Le programme de réception est activé quand le contrôleur reçoit le début d'un message (signal  $\overline{RXACT}$ ) et plus précisément l'adresse du destinataire. Le coupleur ne travaille pas en full-duplex (ne contrôle pas *en même temps* l'émission et la réception des messages) à cause de la baisse de performances que ceci entraîne. Comme le coupleur émet/reçoit sur la même ligne, le full-duplex n'est pas nécessaire : il n'y a aucun intérêt à s'écouter parler. Pendant l'émission effective du message sur la ligne, l'interruption sera masquée.

Le programme d'émission effectue l'émission des messages qui se trouvent dans la file d'émission (file remplie par le processeur de traitement). Tout message est répété un certain nombre NMAX de fois (constante du système), tant que l'accusé de réception n'arrive pas, après quoi si l'absence d'accusé se maintient, on signale l'erreur au processeur de traitement par l'intermédiaire d'une variable ETE.

Le programme de réception reçoit les messages (corrects) dont la partie adresse destinataire est parmi les adresses associées à la station, les met dans la file de réception (vidée par le processeur de traitement) et envoie les accusés de réception.

### Evaluation des performances du coupleur

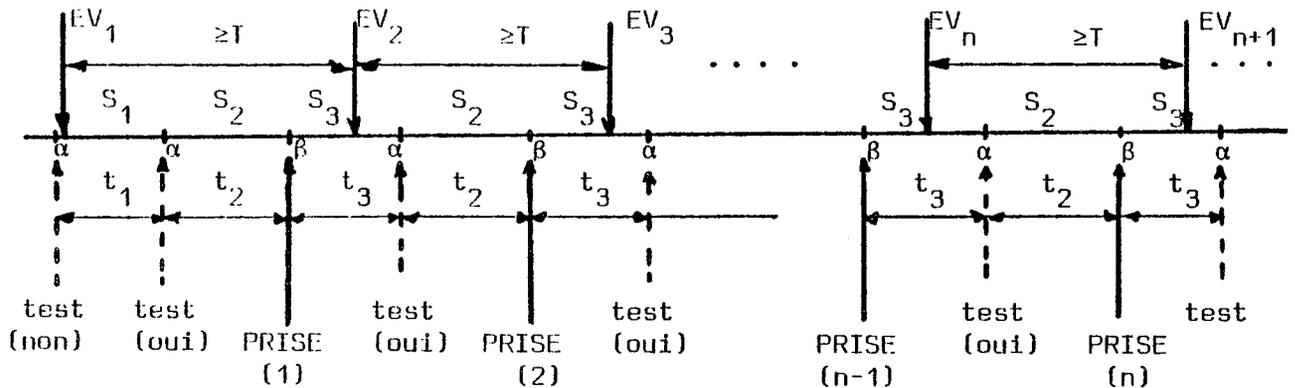
Nous allons d'abord exposer le principe adopté pour la prise en compte des événements périodiques. Considérons une succession d'événements et soit  $T$  la durée minimale entre deux événements successifs. Chaque événement doit être pris en compte. Pour le bon fonctionnement du système, il faut qu'à l'arrivée d'un événement  $E_{i+1}$ , l'événement  $E_i$  (antérieur à  $E_{i+1}$ ) soit déjà pris en compte. Pour la prise en compte des événements, on utilise la séquence cyclique représentée ci-dessous :



Au point  $\alpha$  on teste si un nouvel événement est arrivé :

- . si le test est négatif (pas d'événement), on exécute la séquence  $S_1$  (de durée  $t_1$ ) et on revient en  $\alpha$
- . si le test est positif (événement arrivé), on exécute la séquence  $S_2$  (de durée  $t_2$ ) à la fin de laquelle on fait "prise de l'événement", après quoi on exécute la séquence  $S_3$  (durée  $t_3$ ) et on revient en  $\alpha$ .

La situation la plus défavorable est la suivante :



Il faut donc  $t_1 + nt_2 + (n - 1)t_3 \leq nT$ , d'où

$$T \geq t_2 + t_3 + \frac{t_1 - t_3}{n} \quad \text{quel que soit } n = 1, 2, 3, \dots, \text{ d'où}$$

$$\begin{cases} T \geq t_2 + t_3 \\ T \geq t_2 + t_1 \end{cases}$$

On aura une programmation temps réel de l'émission/réception effective des messages suivant le principe exposé ci-dessus.

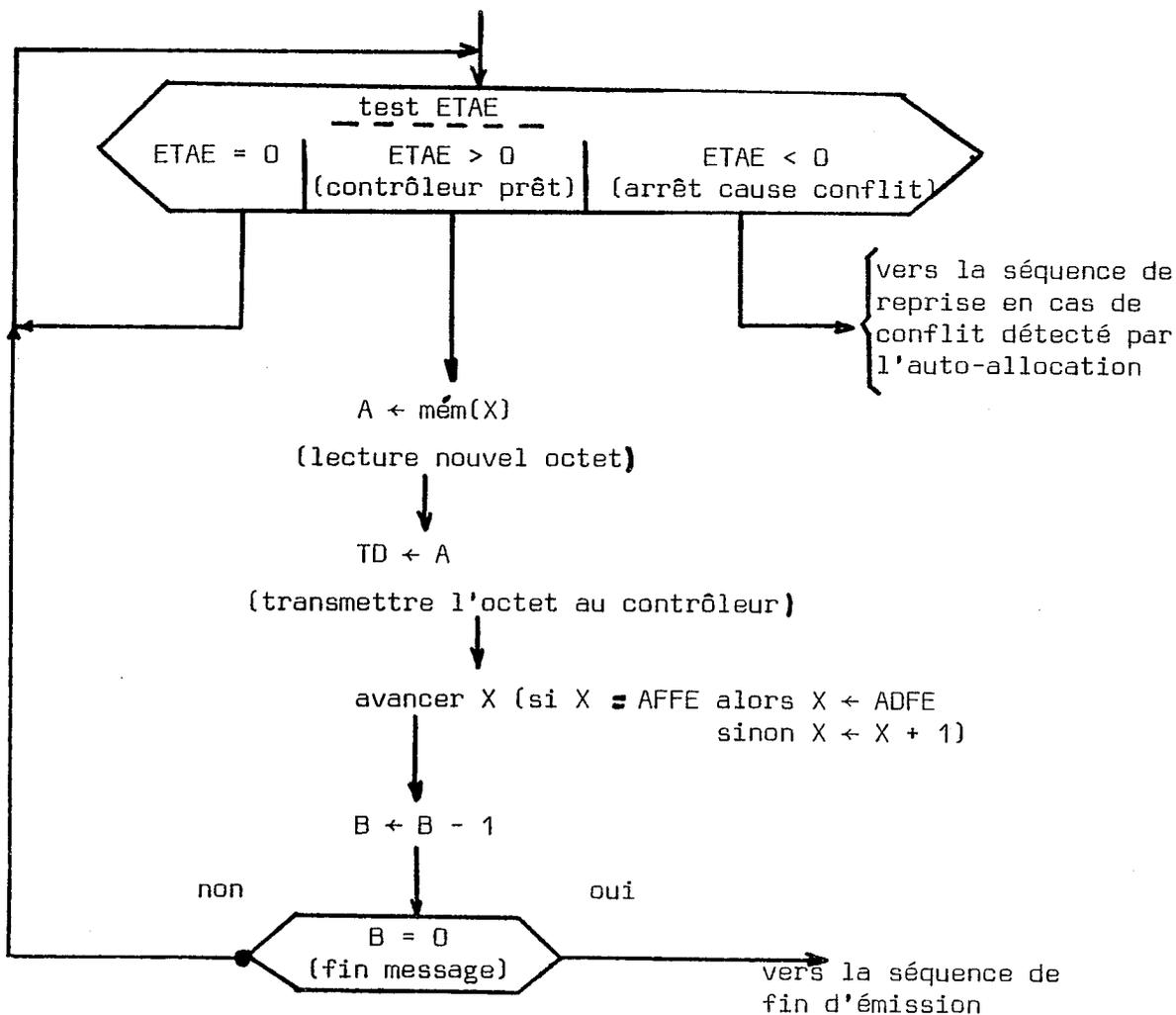
Les séquences qui contrôlent l'émission/réception effective de l'information ont été optimisées pour permettre un débit élevé d'échange.

Compte tenu des possibilités offertes par le microprocesseur 6800, on a choisi les séquences suivantes pour le contrôle de l'émission/réception effective de l'information :

. Emission\_message :

Soit X le registre index qui pointe sur le prochain octet à émettre.  
Soit B le registre qui donne le nombre d'octets qui restent à émettre.  
On rappelle que l'on a noté :  
ETAE : état global de l'émission,  
ADFE, AFFE : adresses (constantes) entre lesquelles se trouve tout l'espace de la file d'émission,  
TD : registre transmission du contrôleur COM.

On a l'algorithme :



En langage 6800 on a :

```
EDON : LDA  A, ETAE (3)
        BGT  EO      (4)
        BEQ  ITER    (4)
        JMP  ARTE    •————→ vers la séquence d'arrêt (cause conflit)
EO :    LDA  A, O(X) (5)
        STA  A, TD   (4)
        CPX  AFFE    (4)
        BNE  E1      (4)
        LDX  ADFE    (4)
        DECA B       (2)
        BNE  EDON    (4)
        BRA  FINE
E1 :    INX          (4)
        DECA B       (4)
        BNE  EDON    (4)
FINE :  { : séquence fin d'émission
```

N.B. Entre parenthèses, on a mis la durée (en cycles microproces-  
seur) de chaque instruction.

Si  $T_E$  est la durée d'un octet, on trouve qu'il faut :

$$T_E \geq 38 \text{ cycles}$$

si  $\theta$  est la durée du bit, on a  $T_E = 8\theta$  , donc il faut  $\theta \geq \frac{38}{8}$  cycles.

. Réception message :

Soit X le registre index qui pointe sur la case mémoire où il faut écrire le prochain octet reçu.

Soit B le registre qui donne le nombre maximum d'octets que l'on peut encore recevoir (ou encore la place qui est disponible).

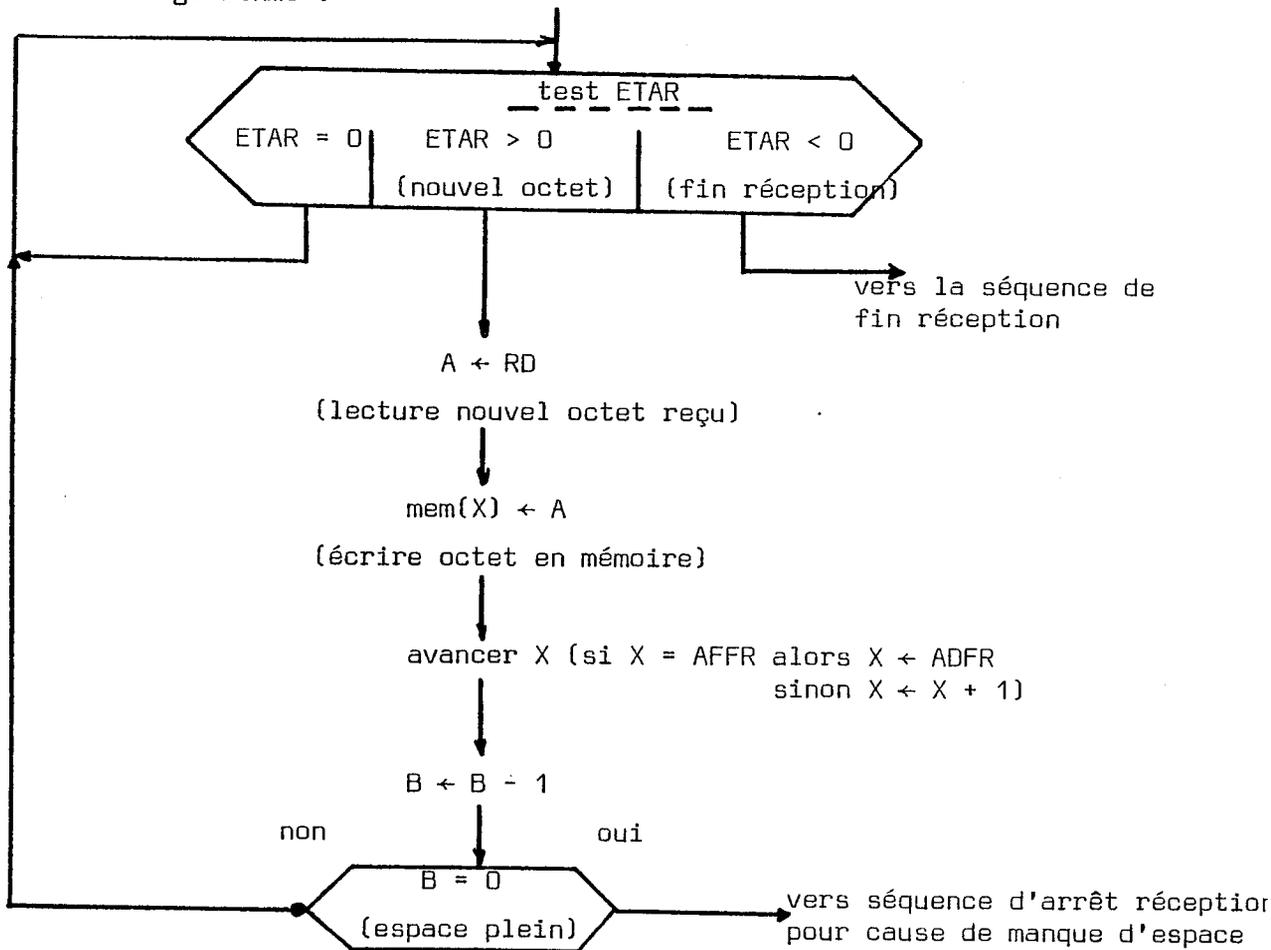
On rappelle que l'on a noté :

ETAR : état global de la réception,

ADFR, AFFR : adresses (constantes) entre lesquelles se trouve tout l'espace de la file de réception,

RD : registre réception du contrôle COM.

On a l'algorithme :



En langage 6800 on a :

```
RDON : LDA  A, ETAR (3)
        BGT  RD  (4)
        BEQ  RDON (4)
        JMP  FINR      → vers la séquence de fin réception
RO :   LDA  A, RD  (3)
        STA  A, O(X) (6)
        CPX  AFFR  (4)
        BNE  R1    (4)
        LDX  AFDR  (4)
        DECA B     (2)
        BNE  RDON  (4)
        BRA  ARTR
R1 :   INX          (4)
        DECA B     (2)
        BNE  RDON  (4)
```

ARTR : { séquence d'arrêt réception (manque d'espace)

Si  $T_R$  est la durée d'un octet reçu, on trouve qu'il faut

$$T_R \geq 38 \text{ cycles.}$$

Si  $\theta$  est la durée (nominale) du bit et compte tenu que l'on peut avoir une dispersion de  $\theta/2$  sur un octet, il en résulte que :

$$T_R \geq 8\theta - \theta/2, \text{ d'où } \theta \geq \frac{76}{15} \text{ cycles.}$$

En conclusion, la durée du bit doit satisfaire  $\theta \geq \frac{76}{15}$  cycles.

Avec un cycle microprocesseur de  $0,6 \mu\text{s}$ , on obtient  $\theta \geq 3,04 \mu\text{s}$ .

De là on obtient que les coupleurs supportent un débit de  $41,1 \text{ K octets/s}$  (en half-duplex).

## CONCLUSIONS

Le mécanisme de communication que nous avons présenté dans ce document entre dans sa phase finale de réalisation et devrait déboucher dans un proche avenir, dans le cadre de la collaboration ENSIMAG - CNET, sur une maquette de commande d'autocommutateur comportant plusieurs stations.

L'étude menée dans le contexte d'une architecture décentralisée à grande sûreté de fonctionnement, nous a permis de voir d'une part que le mécanisme de communication constitue "l'épine dorsale" du système (ce que l'on prévoyait) et d'autre part sa très grande importance dans la sûreté de fonctionnement globale.

Le mécanisme de communication que nous avons étudié semble bien adapté aux réseaux locaux complètement décentralisés, dans lesquels les stations ont les mêmes prérogatives. Sa modularité et le fait d'être complètement réparti sur les stations, permettent d'avoir des systèmes bien adaptés aux besoins et ceci suivant deux directions :

- . débit (performances) : par la "taille" de l'autoroute,
- . nombre de stations.

L'étude menée dans le contexte d'une application précise, l'autocommutateur téléphonique, nous a amené à une réalisation à base de microprocesseurs qui confère au mécanisme une grande souplesse, puissance et sûreté de fonctionnement globale.

Les critiques que l'on peut faire au mécanisme sont :

- . limitation à un périmètre géographique relativement restreint et ceci à cause de l'auto-allocation,
- . protocole d'échange, qui tout en ayant une grande sécurité, est relativement pauvre, faisant essentiellement de la commutation de messages.

Néanmoins, notre mécanisme de communication est suffisamment général pour s'adapter à une large gamme d'applications. En utilisant ce système fondamental d'échange de messages, on peut aisément réaliser un protocole de dialogue élaboré au niveau supérieur du système.

Un autre point à noter est celui du coût de la réalisation. Notre réalisation des coupleurs coûte assez cher. Dans le cadre de l'autocommutateur téléphonique, ce prix se justifie dans la mesure où les contraintes de sécurité de fonctionnement sont très sévères. Néanmoins, ce coût pourrait être abaissé par l'intégration des fonctions du coupleur sur un seul circuit : un contrôleur évolué de communication série.

A ce jour à l'ENSIMAG, deux stations "maquettes" sont sur le point d'être terminées et serviront à la validation du mécanisme de communication :

- . une station réalisée suivant la présentation donnée au chapitre 4,
- . une autre station, réalisée par P. NICOLOPOULOS, dans une technologie différente, le coupleur étant structuré autour du microprocesseur 8 X 300 de RTC.

Cette dernière solution, tout en étant très chère, permet une approche beaucoup plus fine de la communication et servira d'une part à observer plus finement le déroulement des échanges et d'autre part à dégager des spécifications pour un éventuel contrôleur évolué intégré.



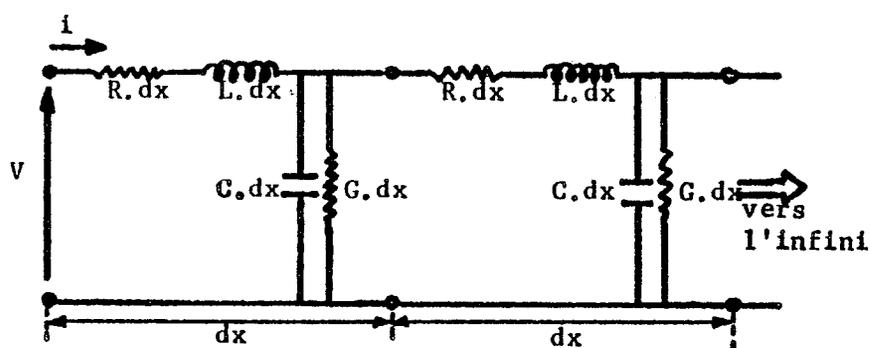
## ANNEXE 1

### CARACTÉRISTIQUES ÉLECTRIQUES DES LIGNES LONGUES

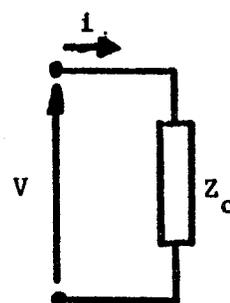
Nota : les résultats théoriques concernant les lignes sont tirés de [KEN].

Une ligne est caractérisée par les paramètres distribués suivants :

- . L : inductance unitaire
- . R : résistance unitaire
- . C : capacité unitaire
- . G : conductance unitaire



a) Paramètres distribués



b) Circuit équivalent

FIG. A1. Modèle pour une ligne longue

La ligne (sans perte) sera caractérisée par les grandeurs suivantes :

- .  $Z_C$  : impédance caractéristique ; elle est égale à :

$$Z_C = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \quad \left(\frac{\omega}{2\pi} = \text{fréquence du signal}\right)$$

Pour des hautes fréquences ( $R \ll \omega L$ ,  $G \ll \omega C$ ), on obtient :

$$Z_C = R_0 \approx \sqrt{\frac{L}{C}}$$

- . propagation uniforme du signal sur la ligne

La vitesse de propagation est :  $v = \frac{1}{\sqrt{LC}}$

La durée de propagation par unité de longueur sera :  $\delta = \sqrt{LC}$

Pour une ligne de longueur  $l$ , le temps total de propagation sera :

$$\tau = l \cdot \delta = l \cdot \sqrt{LC}$$

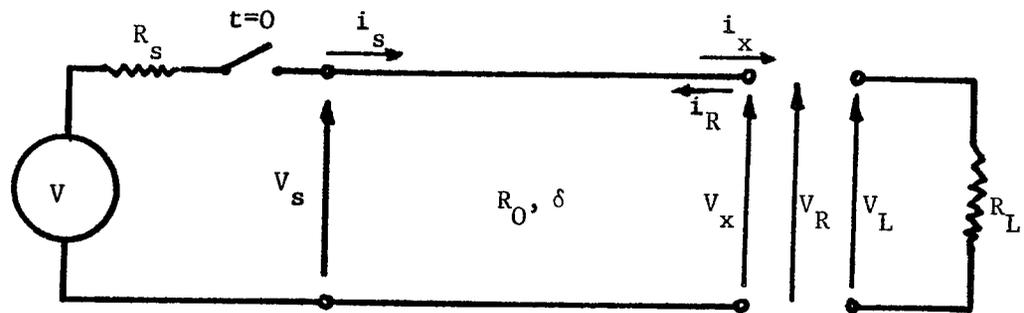


FIG.A2. Phénomène de réflexion

Considérons une ligne terminée par la charge  $R_L$  et soit  $R_S$  la résistance interne de la source de tension. En bout de ligne a lieu la réflexion de l'onde incidente (figure A.2). Le coefficient de réflexion, c'est-à-dire le rapport entre l'amplitude (tension, courant) de l'onde réfléchie et l'amplitude de l'onde incidente est :

$$\rho_L = \frac{R_L - R_0}{R_L + R_0} \quad \text{au niveau de la charge}$$

$$\rho_S = \frac{R_S - R_0}{R_S + R_0} \quad \text{au niveau de la source}$$

L'onde émise par la source arrive après le temps  $\tau = l.\delta$  à la charge où a lieu la réflexion avec le coefficient  $\rho_L$ . L'onde réfléchie arrive à la source après  $\tau$  où a lieu la réflexion avec le coefficient  $\rho_S$ . L'onde nouvellement réfléchie retourne vers la charge, etc .. (figure A.3).

Dans la transmission des données, pour éliminer le phénomène de réflexion, il faut adapter la ligne, c'est-à-dire prendre  $R_L = R_0$  (dans ce cas,  $\rho_L = 0$ ). La ligne sans pertes est un cas idéal, le signal étant véhiculé (avec la vitesse  $v = \frac{1}{\delta}$  constante) sans que sa forme soit modifiée.

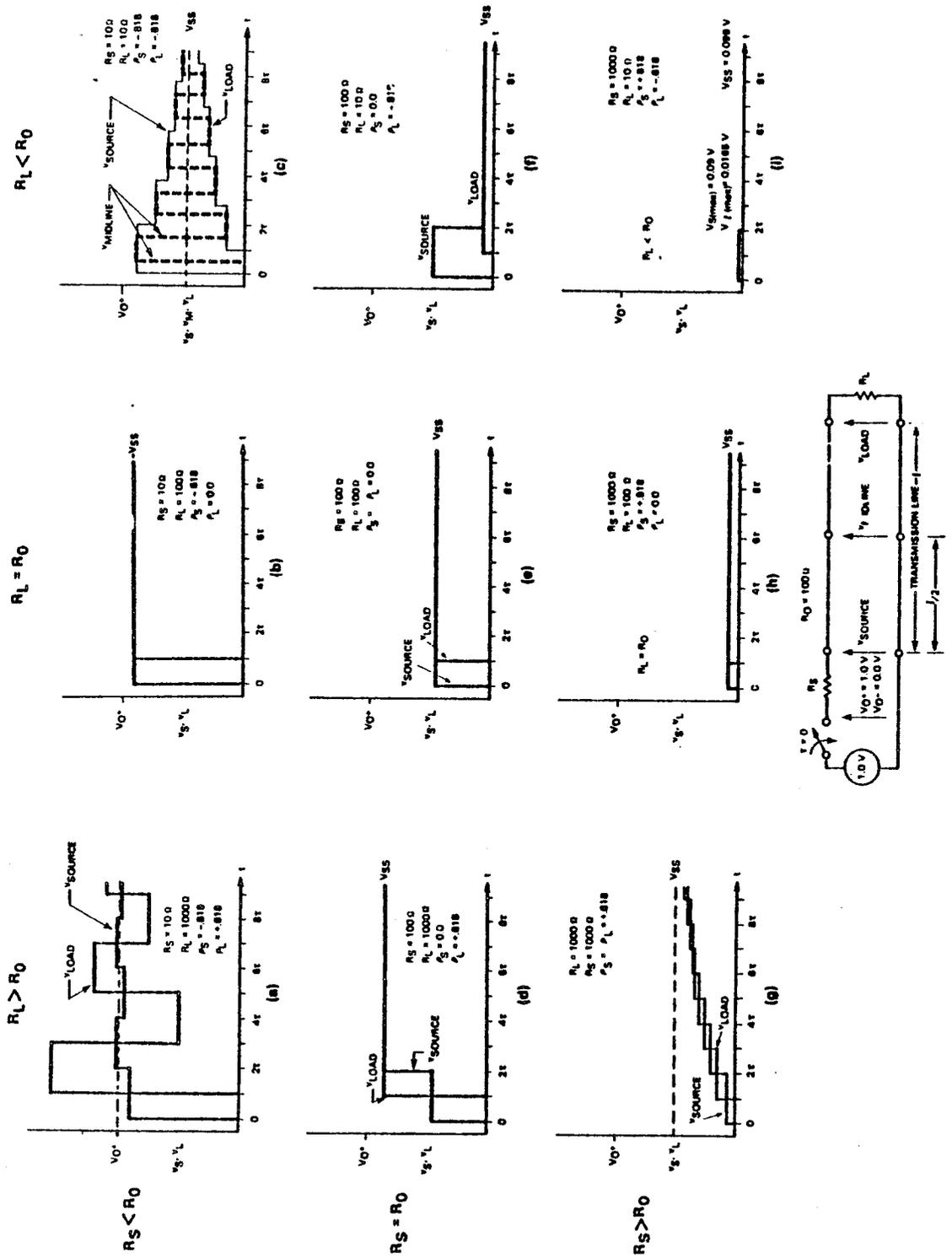


FIG.A3. Formes d'onde des tensions source , charge (tiré de [KEN])

En réalité, les lignes sont avec pertes et se comportent comme un filtre passe bas avec pertes. Les principaux phénomènes qui produisent ceci sont : l'effet de peau (distribution du courant dans le conducteur), l'effet de proximité (interactions entre les conducteurs), pertes par radiation, pertes dans le diélectrique. Ceci fait que les paramètres distribués  $L$ ,  $R$ ,  $C$ ,  $G$ , dépendent de la fréquence.

Dans la transmission du signal on doit donc tenir compte de :

- . la dispersion,
- . l'atténuation.

Ainsi, une transition appliquée à un bout de la ligne devient-elle de plus en plus "arrondie" au fur et à mesure que le signal se propage le long de la ligne. On peut dire que le temps de propagation d'une transition est égal à : propagation uniforme + temps de montée (descente). On va noter  $t(x)$  le temps total de propagation pour une distance  $x$ .

Pour les câbles où l'atténuation du signal spécifie entièrement la fonction de transfert (du filtre constitué par la ligne), ce qui est le cas généralement pour le câble coaxial, on peut appliquer la méthode de Nahman pour prévoir le temps de montée (descente).

Dans la transmission sur une ligne, a lieu le phénomène d'interférence inter-symboles : une nouvelle onde arrive au récepteur avant que l'onde précédente ait atteint sa valeur finale. La dispersion ainsi introduite est appelée dispersion isochrone : différence entre la durée théorique d'une impulsion émise et la durée réelle de l'impulsion reçue. On la mesure pratiquement à l'aide du diagramme de l'oeil : on émet une suite aléatoire de "0" et "1" et on superpose, sur l'unité de temps (durée du bit), toutes les transitions (figure A.4).

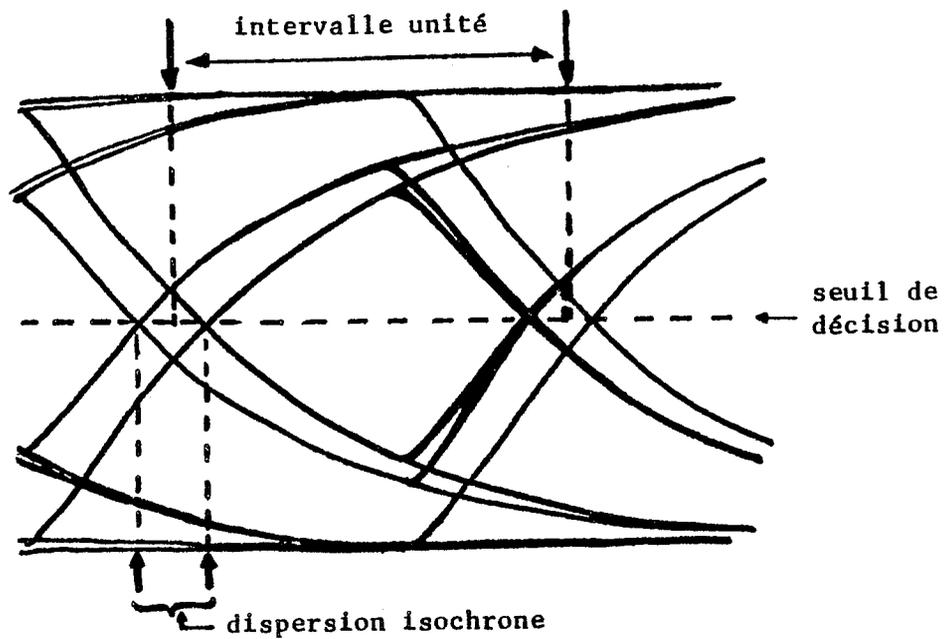


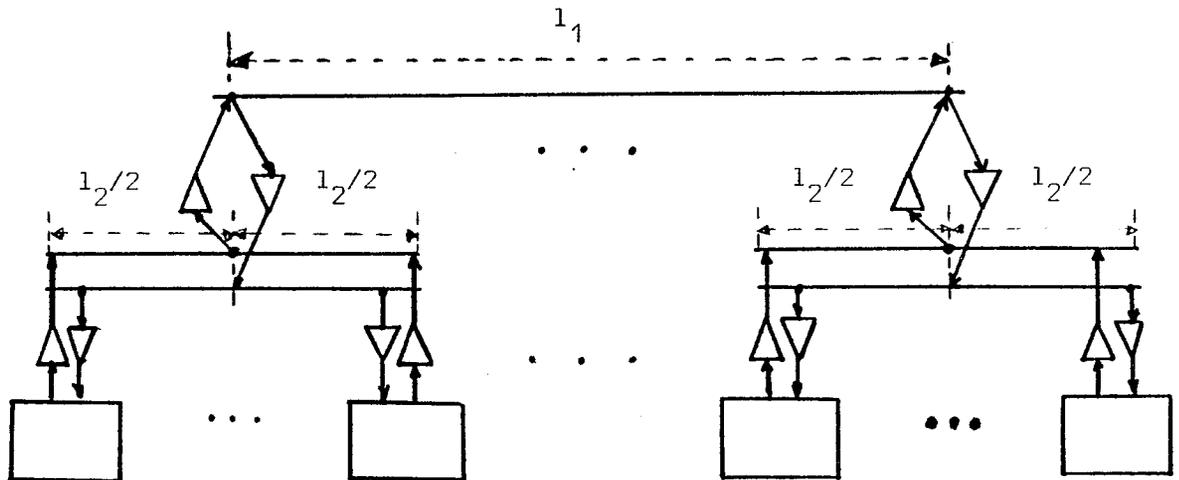
FIG.A4. Diagramme de l'oeil

Dans le cas où l'on a une liaison multipoint partagée entre plusieurs stations, les précautions suivantes sont à prendre :

- . adaptation de la ligne aux deux bouts par des résistances de terminaison égales à la résistance caractéristique de la ligne ;
- . sortance de chaque émetteur : le courant que l'émetteur peut fournir doit être plus grand que le courant consommé par les récepteurs, les émetteurs non actifs et les résistances de terminaison ;
- . la résistance équivalente des postes (combinaison en parallèle des résistances d'entrée de récepteur et des résistances de sortie des émetteurs inactifs), doit être beaucoup plus grande que la résistance caractéristique de la ligne pour des raisons d'adaptation de la ligne.

Ces considérations font que le nombre de postes émetteur/récepteur que l'on peut connecter sur une ligne est limité (de l'ordre de la quinzaine pour les circuits couramment utilisés).

Pour aller au-delà de cette limite, ce qui est le cas pour l'autocommutateur téléphonique, on pense utiliser une topologie "arborescente" comme indiqué dans la figure suivante :



Si  $n$  est le nombre de postes permis sur une ligne, avec une telle disposition on peut avoir  $n^2$  stations dans le réseau. La distance maximum parcourue par le signal est égale à :

$$l = \frac{l_2}{2} + l_1 + \frac{l_2}{2} = l_1 + l_2$$

Cette solution a le désavantage d'augmenter la quantité de câbles et de diminuer la fiabilité de la voie d'échange par l'introduction d'amplificateurs supplémentaires.

Pour pouvoir appliquer le principe d'auto-allocation envisagé, on a vu (chapitre 2) qu'il faut que la ligne réalise une fonction  $f(S_1, \dots, S_m)$  avec  $f \in \{S_1, \dots, S_m\}$  où  $S_1, \dots, S_m$  sont les informations émises par les stations concurrentes et  $f$  doit être la même en tous points de la ligne. Cette fonction  $f$  dépend des amplificateurs d'émission choisis.

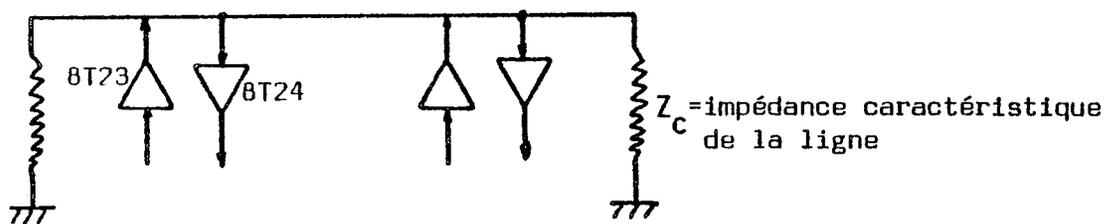
### a) Ligne monofilaire

Dans ce cas, la longueur de la ligne est limitée (environ 150 m) à cause de l'affaiblissement du signal, pour des longueurs plus grandes le bruit devenant comparable à l'amplitude du signal.

#### . Amplificateur en collecteur ouvert

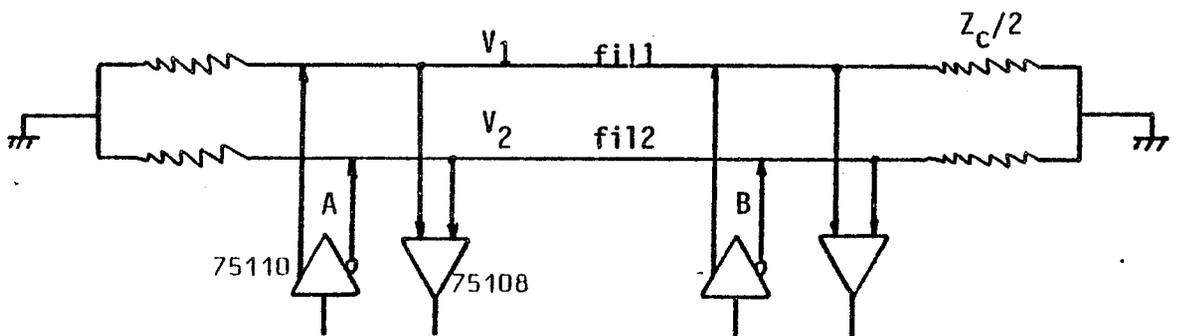
La ligne réalise le "et" logique et quand personne n'émet, celle-ci prend l'état de repos : REPOS = 1.

#### . Amplificateur en émetteur-suiveur (8T23 par exemple)



La ligne réalise le "ou" logique et son état de repos est : REPOS = 0.

### b) Ligne bifilaire (transmission différentielle)



Les circuits d'émission sont généralement des générateurs de courant (par exemple 75110). L'état de la ligne est donné par la différence de potentiel  $V_1 - V_2$  :

$$V_1 - V_2 \leq -s : "0"$$

$$V_1 - V_2 \geq s : "1"$$

Le seuil  $s$  dépend du circuit de réception ( $\approx 25$  mV pour 75107)

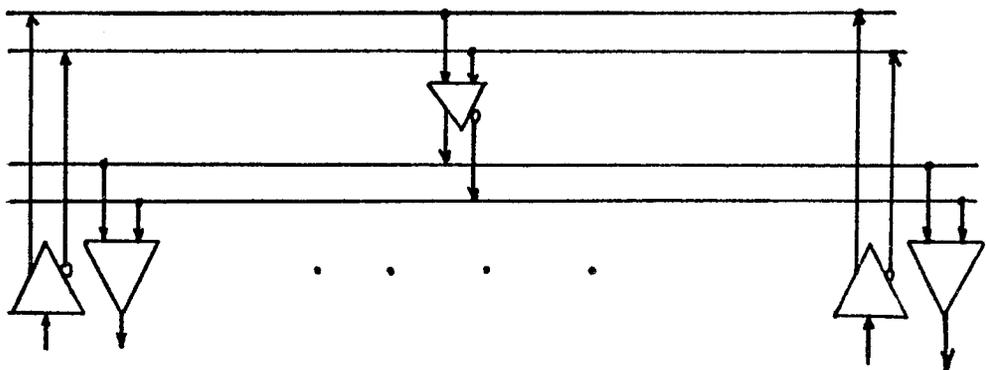
Ce mode différentiel assure une meilleure protection contre le bruit et permet des longueurs plus importantes (jusqu'à 1 Km). Mais le mode différentiel ne peut être utilisé tel quel dans le cas de l'auto-allocation. Supposons deux stations A et B qui émettent des informations différentes : A émet "1" et B émet "0".

L'amplificateur A émet sur fil 1 (ceci entraîne  $V_{1A} = V$ ) et s'isole du fil 2 (sortie haute impédance). De même, B émet sur fil 2 (ce qui entraîne  $V_{2B} = V$ ) et s'isole sur fil 1. Mais le signal sur fil 1 arrive affaibli en B ( $V_{1B} < V$ ) et le signal sur fil 2 arrive affaibli en A ( $V_{2A} < V$ ). Ceci fait que :

- en A :  $V_{1A} - V_{2A} > s$  (donc "1")
- en B :  $V_{1B} - V_{2B} < -s$  (donc "0").

Ainsi aucune station ne voit-elle de conflit.

Autrement dit, la fonction  $f$  (réalisée par la liaison) dépend de l'endroit sur la ligne et on ne peut pas appliquer l'auto-allocation. On peut s'en sortir en réalisant la configuration ci-après :

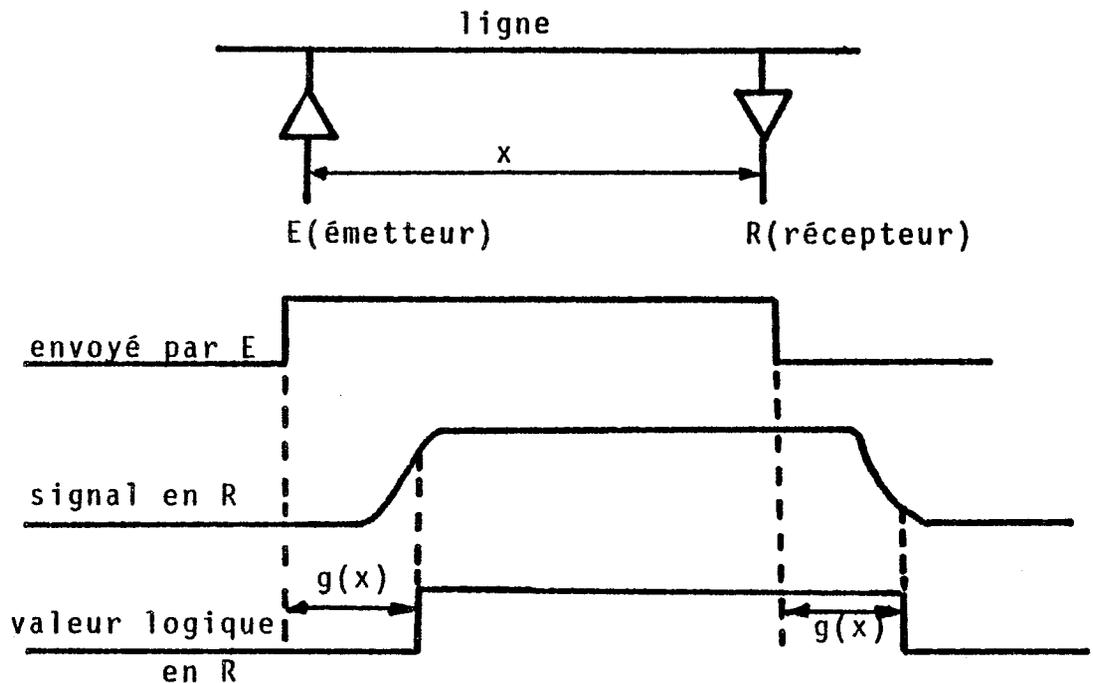


L'amplificateur qui relie les deux parties permet à toutes les stations de recevoir la même information : en tout point de réception, l'état de la ligne est donné par  $f(S_1, \dots, S_m) \in \{S_1, \dots, S_m\}$ ,  $f$  étant la même partout, où  $S_1, \dots, S_m$  sont les informations émises par les stations concurrentes.

## ANNEXE 2

CALCUL DE LA DISPERSION INTRODUITE PAR L'AUTO-ALLOCATION

On va considérer une ligne adaptée (pour éliminer les réflexions du signal) et on va noter avec  $g(x)$  le temps de propagation du signal sur une distance  $x$  (y compris le temps de montée/descente).



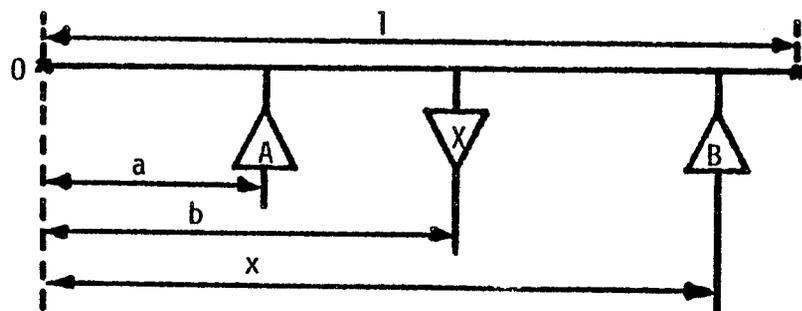
On considère que la ligne a le fonctionnement suivant :

si  $S_1(t), \dots, S_m(t)$  sont les signaux (valeurs logiques) émis (au temps  $t$ ) en un point  $X$  de la ligne, le signal reçu est donné par :

$$R_x(t) = f(S_1(t - g(d_1)), \dots, S_m(t - g(d_m))) \text{ avec } f(S_1, \dots, S_m) \in \{S_1, \dots, S_m\}$$

où  $d_i$  ( $i=1, \dots, n$ ) = distance entre  $X$  et l'émetteur du signal  $S_i$ .

Pour simplifier la présentation, et sans restreindre la généralité, on va prendre  $f = \text{"ou"}$  logique et deux émetteurs A et B. Notons par  $l$  la longueur de la ligne.



Pour le calcul de la dispersion, on va considérer seulement le temps pendant lequel A et B émettent la même information, puisque autrement un des deux sera rejeté (par détection de conflit) et après l'auto-allocation n'introduira plus de dispersion (il n'y a qu'un seul qui émet).

N.B. Dans une première étape on négligera :

- . la dispersion isochrone due à la ligne et aux seuils de décision des récepteurs,
- . la dispersion due à l'imprécision des horloges d'émission.

On va considérer (par exemple) que A a commencé à émettre d'abord et B après la durée  $\epsilon$  (ce déphasage se maintiendra jusqu'à la détection d'un conflit). Soit  $d$  la distance entre A et B. On va prendre une origine à un bout de la ligne telle que :

$$b = \text{abscisse (B)} \geq \text{abscisse (A)} = a$$

Soit  $x$  l'abscisse d'un point quelconque X sur la ligne.

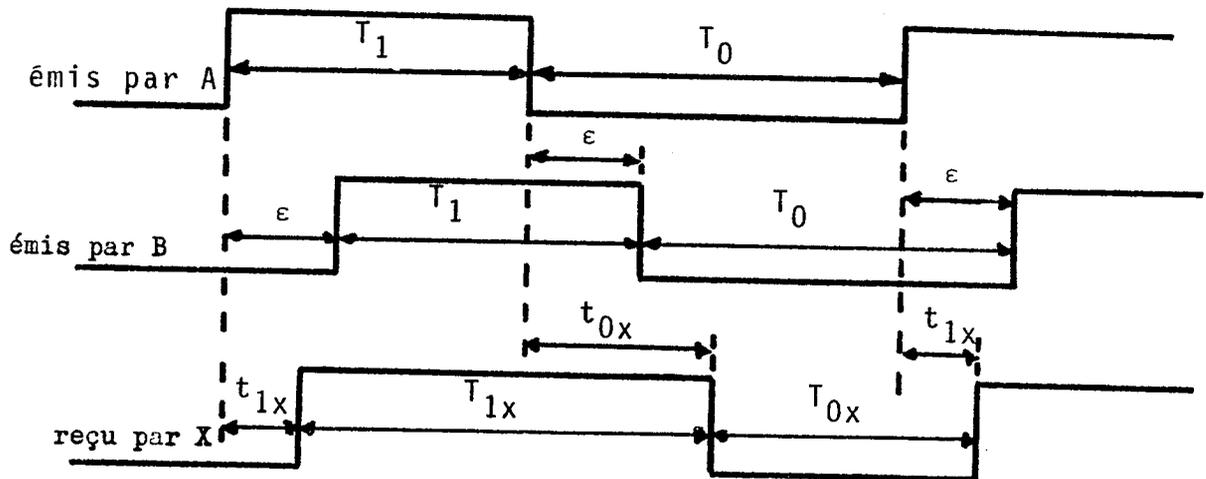


FIG.A5. Chronogramme des signaux émis/reçus

D'après ce chronogramme, on voit que :

$$T_{1X} = T_1 - t_{1X} + t_{0X}$$

$$T_{0X} = T_0 + t_{1X} - t_{0X}$$

donc la dispersion est  $p = |t_{0X} - t_{1X}|$

Comme la ligne réalise le "ou" logique, il résulte :

. le signal "monte" en X quand au moins un front montant émis arrive en X, d'où :

$$t_{1X} = \min (g(|a - x|), \epsilon + g(|b - x|))$$

. le signal "descend" en X quand tous les fronts descendants émis arrivent en X, d'où :

$$t_{0X} = \max (g(|a - x|), \epsilon + g(|b - x|))$$

La dispersion devient :

$$p = |\epsilon + g(|b - x|) - g(|a - x|)|$$

où  $a, b, x, \epsilon \in [0, 1]$  et  $b - a = d$  (distance entre A et B).

On va étudier la fonction :

$$q = \epsilon + g(|a + d - x|) - g(|a - x|)$$

où  $x \in [0, 1]$ ,  $a + d \leq 1$ , donc  $0 \leq a \leq 1 - d$

La fonction  $g$  est dérivable plusieurs fois.

Comme  $g$  est croissante (le temps de propagation augmente avec la distance) sa dérivée première  $g'$  sera positive.

On va étudier la fonction  $q$  suivant le signe de la deuxième dérivée de  $g$  (signe de  $g''$ ).

Etablissons d'abord les résultats suivants :

si  $0 \leq u \leq 1-d$  alors :

$$(i) \begin{cases} g'' \geq 0 & \text{implique} & g(u+d) - g(u) \leq g(1) - g(1-d) \\ g'' \leq 0 & \text{implique} & g(u+d) - g(u) \geq g(1) - g(1-d) \end{cases}$$

En effet :

si  $g'' \geq 0$  alors  $g'$  est croissante et on a  $g'(u+d) - g'(u) \geq 0$

Donc  $g(u+d) - g(u)$  est croissante d'où la relation cherchée.

En particulier on a :

$$g(1-a-d) - g(1-a) \geq g(1-d) - g(1)$$

$$g(d) - g(0) \leq g(1) - g(1-d)$$

si  $g'' \leq 0$  par un raisonnement analogue on a la relation

recherchée. En particulier on a :

$$g(1-a-d) - g(d) \leq g(1-d) - g(1)$$

$$g(d) - g(0) \geq g(1) - g(1-d)$$

Etude de la fonction  $q = \epsilon + g(|a+d-x|) - g(|a-x|)$

Suivant la valeur de  $x$  on distingue 3 cas :

1.  $0 \leq x \leq a$  On obtient :

$q = \epsilon + g(a+d-x) - g(a-x)$  dont la dérivée est

$$\frac{dq}{dx} = -g'(a+d-x) + g'(a-x)$$

a)  $g'' \geq 0$

$g'$  étant croissante il résulte  $\frac{dq}{dx} \leq 0$  donc  $q$  est décroissante d'où

$q(a) \leq q(x) \leq q(0)$  ou encore  $\epsilon + g(d) - g(0) \leq q(x) \leq \epsilon + g(a+d) - g(a)$

Conformément à (i) on a  $g(a+d) - g(a) \leq g(1) - g(1-d)$  donc

$$\epsilon + g(d) - g(0) \leq q(x) \leq \epsilon + g(1) - g(1-d)$$

b)  $g'' \leq 0$

$g'$  étant décroissante on déduit  $\frac{dq}{dx} \geq 0$  et d'une manière analogue

au cas  $g'' \geq 0$  on obtient

$$\epsilon + g(1) - g(1-d) \leq q(x) \leq \epsilon + g(d) - g(0)$$

2.  $a \leq x \leq a+d$  On a :

$$q = \epsilon + g(a+d-x) - g(x-a)$$

$$\frac{dq}{dx} = -g'(a+d-x) - g'(x-a) \leq 0 \quad \text{puisque } g' \geq 0$$

q étant décroissante on obtient  $q(a+d) \leq q(x) \leq q(a)$  ou encore

$$\epsilon + g(0) - g(d) \leq q(x) \leq \epsilon + g(d) - g(0)$$

3.  $a+d \leq x \leq 1$  On a :

$$q = \epsilon + g(x-a-d) - g(x-a)$$

$$\frac{dq}{dx} = g'(x-a-d) - g'(x-a)$$

a)  $g'' \geq 0$

$g'$  étant croissante ceci impliquera  $\frac{dq}{dx} \leq 0$  (donc q décroissante).

On a  $q(1) \leq q(x) \leq q(a+d)$  ou encore

$$\epsilon + g(1-a-d) - g(1-a) \leq q(x) \leq \epsilon + g(0) - g(d)$$

Tenant compte de (i) on obtient :

$$\epsilon + g(1-d) - g(1) \leq q(x) \leq \epsilon + g(0) - g(d)$$

b)  $g'' \leq 0$

$g'$  étant croissante on obtient que la fonction q est croissante.

D'une manière analogue au cas  $g'' \geq 0$  on obtient :

$$\epsilon + g(0) - g(d) \leq q(x) \leq \epsilon + g(1-d) - g(1)$$

En conclusion des cas 1,2,3 et tenant compte de (i) on obtient :

$$\epsilon + g(1-d) - g(1) \leq q \leq \epsilon + g(1) - g(1-d) \quad \text{si } g'' \geq 0$$

$$\epsilon + g(0) - g(d) \leq q \leq \epsilon + g(d) - g(0) \quad \text{si } g'' \leq 0$$

Donc la dispersion introduite par l'auto-allocation satisfait :

$$(ii) \begin{cases} p \leq \epsilon + g(1) - g(1-d) & \text{si } g'' \geq 0 \text{ (g convexe)} \\ p \leq \epsilon + g(d) - g(0) & \text{si } g'' \leq 0 \text{ (g concave)} \end{cases}$$

où .  $\epsilon$  = déphasage entre les stations concurrentes

.  $d$  = distance entre les stations concurrentes

Comme il a été montré au chapitre II, paragraphe II.2.4., après la première impulsion les stations qui restent encore en course doivent satisfaire  $\epsilon + g(d) \leq \text{TCNF}$

Si g est concave ( $g'' \leq 0$ ) on a  $p \leq \epsilon + g(d) - g(0) \leq \text{TCNF} - g(0)$

Supposons que g est convexe ( $g'' \geq 0$ ) .

Notons avec s la distance qui vérifie  $g(s) = \text{TCNF}$  .

De  $\epsilon + g(d) \leq \text{TCNF}$  et tenant compte de  $\epsilon \geq 0$ ,  $g$  fonction croissante, il résulte que  $d \leq s$ .

La dispersion devient :

$$p \leq \epsilon + g(d) + g(1) - g(1-d) - g(d) \leq \text{TCNF} + g(1) - g(1-d) - g(d)$$

avec  $d \leq s$

On montre facilement que le maximum de la fonction  $g(1) - g(1-d) - g(d)$ , avec  $g'' \geq 0$ ,  $d \leq s$  est égal à :

$$\begin{array}{ll} g(1) - g(1-s) - g(s) & \text{si } s \leq 1/2 \\ g(1) - 2.g(1/2) & \text{si } s \geq 1/2 \end{array}$$

En conclusion la dispersion maximale introduite par l'auto-allocation, après la 1<sup>ère</sup> impulsion est égale à :

$$(iii) \left\{ \begin{array}{l} \cdot \text{ si } g \text{ concave } (g'' \leq 0) \\ P = \text{TCNF} - g(0) \\ \cdot \text{ si } g \text{ convexe } (g'' \geq 0) \\ P = \text{TCNF} + g(1) - g(1-s) - g(s) \quad \text{si } s \leq 1/2 \\ P = \text{TCNF} + g(1) - 2.g(1/2) \quad \text{si } s \geq 1/2 \\ \cdot \text{ où } g(s) = \text{TCNF} \end{array} \right.$$

Si  $\text{TCNF} = g(1)$  (donc  $s=1$ ) on obtient :

$$\begin{array}{ll} p = g(1) - g(0) & \text{si } g \text{ concave} \\ p = 2.(g(1) - g(1/2)) & \text{si } g \text{ convexe} \end{array}$$

Observation :

Dans les calculs ci-dessus, on n'a pas tenu compte des dispersions introduites par la précision des horloges d'émission et par la ligne (dispersion isochrone).

Notons  $u$  = période *théorique* de l'horloge d'émission (durée du quantum d'information)

$\alpha$  = erreur (appelée improprement précision) de l'horloge

$$\text{On a } (1 - \alpha).u \leq u_{\text{réel}} \leq (1 + \alpha).u$$

Donc à l'émission d'une impulsion de durée *théorique*  $n.u$  (contenant donc  $n$  quantum), il y a une dispersion de maximum  $n.\alpha.u$ .

Pour voir l'influence de la précision des horloges, on reprend le chronogramme des signaux émis/reçus (figure A.6).

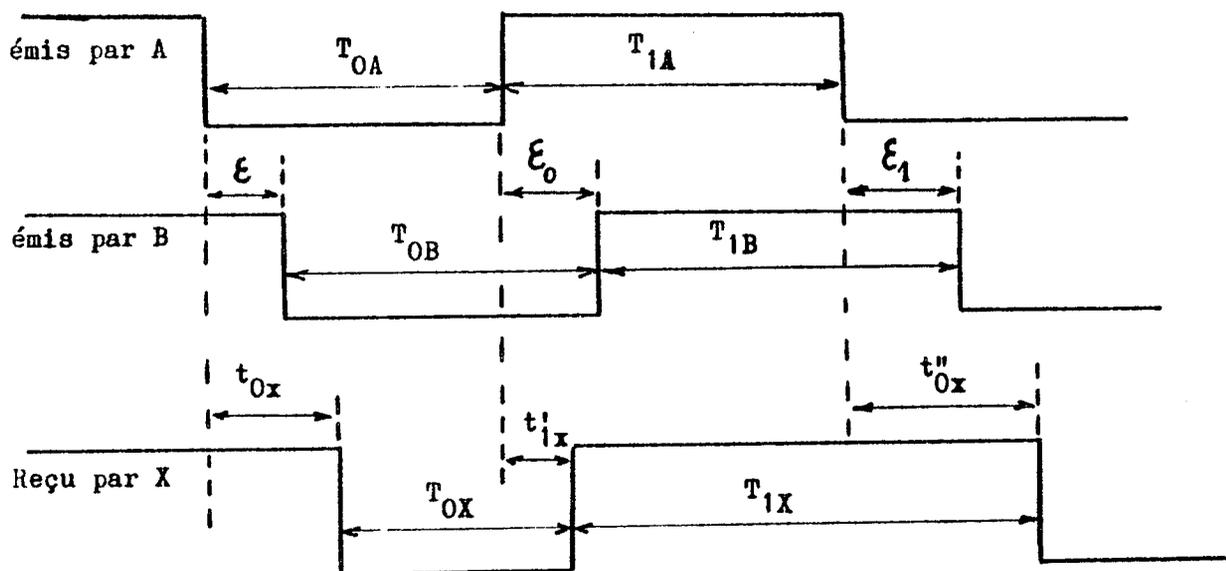


FIG.A6. Signaux émis/reçus , tenant compte des dispersions introduites par les horloges d'émission

On a :  $T_{0A} = T_0 + e_{0A}$

$T_{1A} = T_1 + e_{1A}$

$T_{0B} = T_0 + e_{0B}$

$T_{1B} = T_1 + e_{1B}$

où  $T_0, T_1$  - durées théoriques des signaux

$e_{0A}, e_{1A}, e_{0B}, e_{1B}$  - dispersions dues aux horloges

Conformément à la figure A.6 on voit que les impulsions reçues par un quelconque récepteur X ont les durées suivantes (on note a, b, x, les abscisses de A, B, X, et d = distance entre A et B) :

$$T_{0x} = T_{0A} - t_{0x} + t'_{1x} = T_0 + e_{0A} - t_{0x} + t'_{1x}$$

$$\text{avec : } t_{0x} = \max(g(|a-x|), \epsilon + g(|b-x|))$$

$$t'_{1x} = \min(g(|a-x|), \epsilon + g(|b-x|))$$

$$T_{1x} = T_{1A} + t''_{0x} - t'_{1x} = T_1 + e_{1A} + t''_{0x} - t'_{1x}$$

$$\text{avec : } t''_{0x} = \max(g(|a-x|), \epsilon_1 + g(|b-x|))$$

$$\text{D'autre part } \epsilon_0 = \epsilon + e_{0B} - e_{0A}$$

$$\epsilon_1 = \epsilon_0 + e_{1B} - e_{1A} = \epsilon + e_{0B} + e_{1B} - e_{0A} - e_{1A}$$

La dispersion maximale sera donnée par :

$$p = \max(|e_{0A} - t_{0x} + t'_{1x}|, |e_{1A} + t''_{0x} - t'_{1x}|)$$

Si N est le nombre maximum de quantaux contenus dans une impulsion alors les dispersions  $e_{0A}, e_{1A}, e_{0B}, e_{1B}$  sont comprises entre  $-\alpha.N.u$  et  $\alpha.N.u$

En étudiant tous les cas possibles pour  $t_{0x}, t'_{1x}, t''_{0x}$  on obtient

$$p = |\epsilon + g(|b-x|) - g(|a-x|)| + 3.\alpha.N.u$$

A ceci il faut ajouter la dispersion isochrone  $\mu$  introduite par la ligne . Comme la fonction  $\epsilon + g(|b-x|) - g(|a-x|)$  a déjà été étudiée ci avant et on a trouvé ses bornes [formules (ii) et (iii)] , on peut déduire la dispersion totale , après la 1<sup>ère</sup> impulsion (donc on a  $\epsilon + g(d) \leq \text{TCNF}$ ):

si g concave

$$P = \text{TCNF} - g(0) + 3.\alpha.N.u + \mu$$

si g convexe

$$P = \text{TCNF} + g(1) - g(1-s) - g(s) + 3.\alpha.N.u + \mu \quad \text{si } s \leq 1/2$$

$$P = \text{TCNF} + g(1) - 2.g(1/2) + 3.\alpha.N.u + \mu \quad \text{si } s \geq 1/2$$

$$\text{où } g(s) = \text{TCNF}$$

avec :

.TCNF = durée détectrice de conflit

.l = longueur de la ligne

.g = temps de propagation en fonction de la distance

.u = durée du quantum d'information

N.B. Avec le codage NRZ on a  $u = \theta$  = durée du bit

. $\alpha$  = précision de l'horloge d'émission

.N = nombre maximum de quantaux contenus dans une impulsion

. $\mu$  = dispersion isochrone introduite par la ligne



## ANNEXE 3

RECONNAISSANCE DE L'INFORMATION ÉLÉMENTAIRE

CALCUL DU DÉBIT

On se place dans le cas où sur la ligne la donnée et l'horloge sont mélangées. Pour réaliser ce "mélange", l'information échangée entre émetteur et récepteur est véhiculée par la ligne sous forme codée.

On choisit (suivant le codage envisagé) un quantum d'information qui est la plus petite unité d'information véhiculée par la ligne dans le système considéré. Notons par  $u$  la durée du quantum. La durée  $\theta$  du bit sera un multiple de  $u$ , soit  $\theta = q.u$ .

Exemple : en biphase le quantum est le demi-bit, d'où  $q = 2$ .

L'émetteur va envoyer sur la ligne une suite de quantum (suivant l'algorithme de codage). Le récepteur va reconnaître la suite de quantum et par décodage (fonction inverse du codage), va déterminer les bits.

De par la définition du quantum, il résulte que chaque impulsion (signal entre deux transitions) envoyée sur la ligne est support d'un nombre entier de quantum.

Avoir une réception correcte revient à l'énoncé suivant :

*"quand la station émettrice envoie une impulsion support de  $n$  quantum, il faut que la station réceptrice reconnaisse dans l'impulsion vue par elle exactement  $n$  quantum".*

La reconnaissance de l'information se fait par échantillonnage, en comptant la durée de chaque impulsion (on se "resynchronise" sur chaque transition).

Notations :

$h$  = période nominale de l'horloge d'échantillonnage : les stations du réseau ont la même unité de temps.

N.B. Les stations ont des horloges d'échantillonnage indépendantes, de même période  $h$ , mais déphasées.

$\alpha$  = précision de la période d'échantillonnage ; dans une station quelconque, la période réelle  $h'$  d'échantillonnage satisfait :

$$(1 - \alpha).h \leq h' \leq (1 + \alpha).h$$

$u$  = durée du quantum ; la durée du quantum est un multiple de  $h$  :

$$u = m.h$$

$\theta$  = durée du bit : c'est un multiple de la durée du quantum :

$$\theta = q.u$$

$p$  = dispersion du signal sur la ligne (dépend de la ligne, du mécanisme d'auto-allocation, ...).

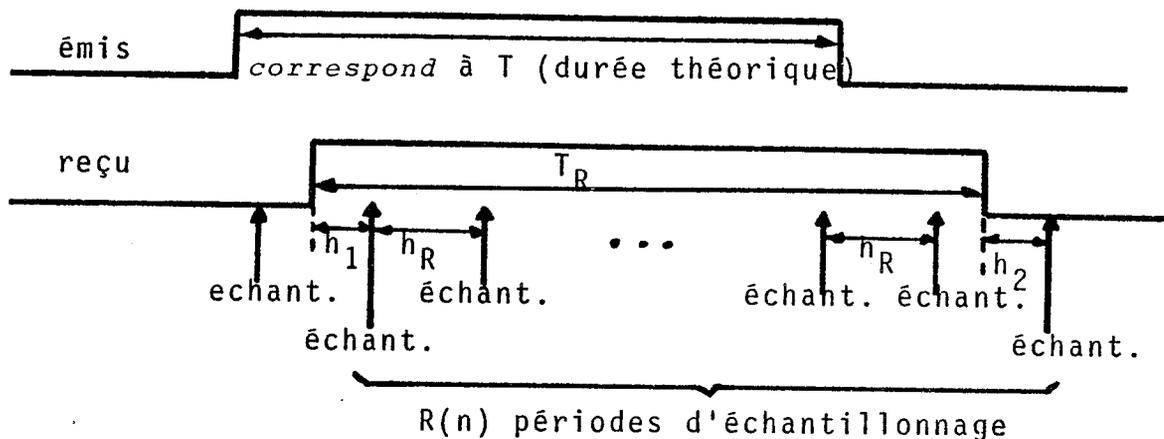
Si  $T$  est la durée *théorique* de l'impulsion émise et  $T_R$  la durée de l'impulsion reçue, on a :

$$T - p \leq T_R \leq T + p$$

Supposons l'émission d'une impulsion support de  $n$  quantum d'information, donc sa durée théorique est :

$$T = n.u$$

L'impulsion reçue (par la station réceptrice) est échantillonnée avec une période  $h_R$  où  $(1 - \alpha).h \leq h_R \leq (1 + \alpha).h$



Par échantillonnage de l'impulsion reçue, le récepteur va trouver  $R(n)$  périodes d'échantillonnage et conformément au chronogramme on a :

$$R(n).h_R = T_R - h_1 + h_2, \text{ où } 0 \leq h_1 \leq h_R, \text{ } 0 \leq h_2 \leq h_R$$

$$\text{et } T - p \leq T_R \leq T + p$$

Calculons la borne inférieure  $R_i(n)$  et la borne supérieure  $R_s(n)$  de  $R(n)$ . Comme on connaît les valeurs extrêmes de  $T_R$ ,  $h_1$  et  $h_2$ , on trouve tout de suite :

$$R_i(n) \geq \frac{n.u - p - (1 + \alpha).h}{(1 + \alpha).h}$$

$$R_s(n) \leq \frac{n.u + p + (1 + \alpha).h}{(1 - \alpha).h}$$

Il en résulte le principe suivant :

*"soit une impulsion émise et soit  $R$  le nombre de périodes d'échantillonnage que le récepteur a trouvé dans l'impulsion.*

*Le nombre  $n$  tel que  $R_i(n) \leq R \leq R_s(n)$ , sera le nombre de quantum d'information reconnus par le récepteur".*

Ce principe entraîne la condition :

$$R_s(n) < R_i(n + 1), \text{ autrement il y a ambiguïté.}$$

En remplaçant  $R_s(n)$  et  $R_i(n + 1)$  par leurs valeurs, on obtient que la durée  $u$  du quantum doit satisfaire :

$$u > 2 \frac{p + (1 + \alpha).h}{(1 - (2n + 1)\alpha)}$$

Notons par  $N$  le nombre maximum de quantum dont un créneau peut être un support ; donc  $n \leq N$ . Comme pour la durée du bit on a  $\theta = q.u$ , il résulte :

$$\theta > 2.q. \frac{p + (1 + \alpha).h}{1 - (2N + 1)\alpha}$$

Le débit sera donné par  $\Delta = \frac{1}{\theta}$

Codage biphasé :

Le quantum est le demi-bit, donc  $q = 2$ .

Pour chaque bit on a au moins un changement de polarité de la ligne, donc un créneau est support de deux demi-bits (quantum) au maximum, d'où  $N = 2$ . Il en résulte :

$$\theta > 4 \cdot \frac{p + (1 + \alpha) \cdot h}{1 - 5\alpha}$$

Codage NRZI et format SDLC du message :

Le quantum d'information est le bit, donc  $q = 1$ .

Dans le format SDLC tout message comporte au maximum six bits "1" consécutifs. Par ailleurs, en NRZI, chaque bit "0" se traduit par un changement de polarité de la ligne.

Il résulte qu'un créneau est support d'au maximum 7 bits, d'où  $N = 7$ .

Donc la durée du bit satisfait :

$$\theta > 2 \cdot \frac{p + (1 + \alpha) \cdot h}{1 - 15\alpha}$$

La valeur de la dispersion  $p$  est donnée à la fin de l'annexe 2.



## ANNEXE 4

ÉTUDE DU MODÈLE :

UN SERVEUR ET PLUSIEURS FILES D'ATTENTE

Rappel

Soit un intervalle de temps  $t$  .

Soit  $R$ =nombre d'arrivées poissonniennes pendant  $t$  .

On a  $\text{Prob}\left(\frac{R=n}{t}\right) = \frac{e^{-a \cdot t} \cdot (a \cdot t)^n}{n!}$  , le coefficient  $a$  caractérisant le processus. Pour un intervalle de durée aléatoire  $U$  , indépendante des arrivées, la loi de  $R$  (nombre d'arrivées poissonniennes de coeff.  $a$ ) est connue .

En particulier on a :

$$E(R) = a \cdot E(U) = c$$

$$E(R^2) = a \cdot E(U) + a^2 \cdot E(U^2) = c + c^2 \cdot u_2$$

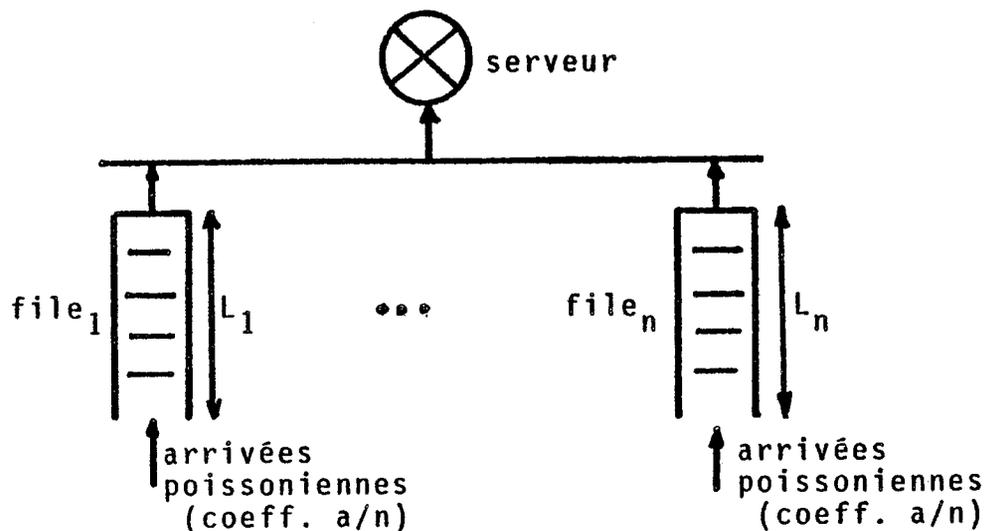
$$E(R^3) = a \cdot E(U) + 3 \cdot a^2 \cdot E(U^2) + a^3 \cdot E(U^3) = c + 3 \cdot c^2 \cdot u_2 + c^3 \cdot u_3$$

avec :  $c = a \cdot E(U)$

$$u_i = \frac{E(U^i)}{E^i(U)} \quad (i=2,3,\dots)$$

Dans la suite nous allons considérer la situation suivante:

- . arrivées poissonniennes (pour l'ensemble du système) de coeff.  $a$
- .  $n$  files d'attente (stations). Les arrivées sont distribuées d'une manière équiprobable sur les files , donc pour chaque file on a des arrivées poissonniennes de coefficient  $\frac{a}{n}$  .
- . un serveur de durée de service  $U$ . La durée de service est indépendante des arrivées et des tailles des files
- . le serveur choisit "au hasard" le message entête d'une file nonvide.



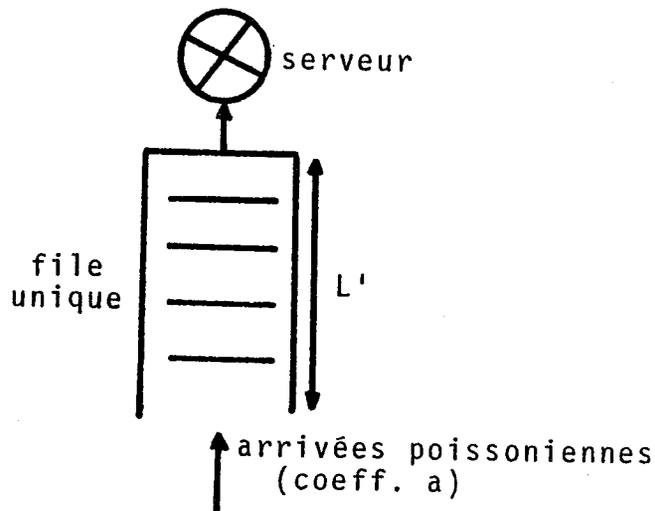
Notons avec  $L_i$  la longueur (nombre de messages) de la file  $i$ , y compris le message éventuellement en cours d'être servi. Les hypothèses de départ font que les files sont statistiquement équivalentes : les variables aléatoires  $\{L_i\}_{i=1,\dots,n}$  ont la même loi.

N.B. Les  $\{L_i\}$  ne sont pas indépendantes.

Notons  $L=L_1+\dots+L_n$  ; donc  $L$  représente le nombre de messages présents dans le système (y compris celui en cours de service).

Regardons maintenant la situation :

- . même serveur
- . mêmes arrivées (coeff.  $a$ )
- . 1 seule file d'attente



Si  $L'$  est la taille de la file (y compris le message en cours de service) il est évident que  $L' = L$  (dans les deux cas le serveur fait la même quantité de travail). Le cas de la file unique a déjà été étudié dans la littérature [LEG] et la loi de  $L$  est connue. En particulier on a :

$$\text{Prob}(L=0) = 1 - c$$

$$E(L) = c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)}$$

$$E(L^2) = c + \frac{3 \cdot c^2 \cdot u_2}{2 \cdot (1-c)} + \frac{c^3 \cdot u_3}{3 \cdot (1-c)} + \frac{c^4 \cdot u_2^2}{2 \cdot (1-c)^2}$$

où  $c = a \cdot E(U)$       taux de charge

$$u_i = \frac{E(U^i)}{E^i(U)} \quad (i = 2, 3)$$

Donons une interprétation plus précise au taux de charge  $c$  dans le cas du mécanisme de communication. Si l'unité d'information (le bit) est écoulee dans le temps  $\theta$ , le débit maximum du mécanisme sera :  $\Delta = \frac{1}{\theta}$ . Soit  $D$  le débit total que l'on veut écouler par le mécanisme de communication. Pendant une durée de temps  $T$  on veut donc écouler en moyenne la quantité  $T.D$  d'information qui prendra le temps  $T.D.\theta$ .

Cette quantité d'information est structurée en  $N$  messages (nombre moyen), d'où la durée moyenne d'un message est  $E(U) = \frac{T.D.\theta}{N}$ . Pendant  $T$  le nombre  $N$  de messages sera le nombre d'arrivées poissonniennes (de coeff.  $a$ ), donc  $N = a.T$ .

On obtient  $E(U) = \frac{D.\theta}{a}$ , d'où

$$c = a.E(U) = D.\theta = \frac{D}{\Delta} = \frac{\text{débit à écouler}}{\text{débit maximum}}$$

Pour les  $n$  files d'attente on s'intéresse aux grandeurs suivantes :

1. Moyenne des  $\{L_i\}_{i=1, \dots, n}$

Comme  $L = L_1 + \dots + L_n$  et les  $\{L_i\}$  sont statistiquement équivalentes il résulte :  $E(L) = n.E(L_1)$  d'où

$$E(L_i) = \frac{1}{n} \cdot \left( c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)} \right) \quad (i=1, 2, \dots, n)$$

2. Ecart type des  $\{L_i\}$

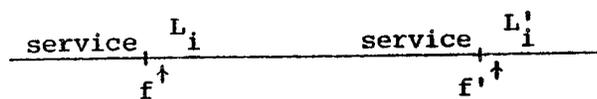
On veut donc  $\sigma_i = \sqrt{E(L_i^2) - E^2(L_i)}$  ( $i=1, 2, \dots, n$ )

On a  $L^2 = \sum_1^n L_i^2 + \sum_{i \neq j} L_i \cdot L_j$ . Comme les  $L_i$  sont équivalentes il résulte :

$$E(L^2) = n.E(L_1^2) + n \cdot (n-1) \cdot E(L_1 \cdot L_2)$$

On peut montrer que  $E(L_1 \cdot L_2) = \frac{c}{n^2} \cdot E(L) - \frac{c^2}{2 \cdot n^2}$

Observation : Pour montrer cela le principe est le suivant :  
On considère deux instants  $f, f'$  situés juste après 2 fins de service consécutives



Pour une file  $i$  on note avec  $L_i, L'_i$  la longueur de la file en  $f$  et respectivement en  $f'$ . On a :

$$L'_i = L_i - \delta_i + R_i$$

$\uparrow$  messages entrés  
 $\downarrow$  messages sortis

On a  $\delta_i = 1$  si  $L_i > 0$  et si la file  $i$  a été choisie par le serveur, sinon  $\delta_i = 0$

Tenant compte du fait que l'on est en régime permanent (donc  $L_i$  et  $L'_i$  ont la même loi) et des hypothèses sur les arrivées poissonniennes on obtient  $E(\delta_i) = E(R_i) = \frac{c}{n} = \text{Prob}(\delta_i = 1)$

$$\text{Pour les files 1 et 2 on a : } L'_1 = L_1 - \delta_1 + R_1$$

$$L'_2 = L_2 - \delta_2 + R_2$$

En multipliant ces deux relations et tenant compte du régime permanent, de l'équivalence statistique des deux files, des hypothèses sur les arrivées poissonniennes on obtient que

$$E(L_1 \cdot \delta_2) = E(L_1) \cdot \frac{c}{n} - \frac{c^2}{2 \cdot n^2} = E(L) \cdot \frac{c}{n^2} - \frac{c^2}{2 \cdot n^2}$$

On a vu que  $\delta_2 = 1$  si  $L_2 > 0$  et file 2 choisie, sinon  $\delta_2 = 0$ ; donc  $L_2 \geq \delta_2$ .

Il résulte  $L_1 \cdot L_2 \geq L_1 \cdot \delta_2$  ou encore  $E(L_1 \cdot L_2) \geq E(L_1 \cdot \delta_2)$  d'où la relation que l'on voulait montrer.

Comme on a  $E(L^2) = n \cdot E(L_1^2) + n \cdot (n-1) \cdot E(L_1 \cdot L_2)$ , et tenant compte de l'inégalité vérifiée par  $E(L_1 \cdot L_2)$  on obtient finalement :

$$E(L_1^2) \leq \frac{E(L^2)}{n} - \frac{n-1}{n^2} \cdot c \cdot E(L) + \frac{c^2 \cdot (n-1)}{2 \cdot n^2}$$

L'écart-type devient :

$$\sigma_i = \sqrt{E(L_i^2) - E^2(L_i)} < \sqrt{\frac{E(L^2)}{n} - \frac{n-1}{n^2} \cdot c \cdot E(L) - \frac{E^2(L)}{n^2} + \frac{c^2 \cdot (n-1)}{2 \cdot n^2}}$$

### 3. Nombre de stations concurrentes (nombre de files nonvides)

Soit  $S$  le nombre de stations concurrentes. On a  $\text{Prob}(S=0) = \text{Prob}(L=0) = 1-c$

Pour une file  $i$  notons :

$$S_i = \begin{cases} 1 & \text{si file } i \text{ nonvide } (L_i > 0) \\ 0 & \text{si file } i \text{ vide } (L_i = 0) \end{cases}$$

Notons  $B_0 = \text{Prob}(L_i > 0)$ ; donc  $\text{Prob}(S_i = 1) = B_0$  d'où  $E(S_i) = B_0$ .

On a  $S = S_1 + \dots + S_n$  d'où  $E(S) = E(S_1) + \dots + E(S_n) = n \cdot B_0$

Il est trivial que  $\text{Prob}(L > 0) = \text{Prob}(L_i > 0) + \text{Prob}(L_i = 0 \text{ et } L > 0)$  donc

$B_0 = \text{Prob}(L_i > 0) \leq \text{Prob}(L > 0) = c$  d'où  $E(S) \leq n \cdot c$

Comme d'autre part (nombre concurrents)  $\leq$  (nombre total de messages présents) il résulte  $E(S) \leq \min(n \cdot c, E(L))$  ou encore

$$E(S) \leq \min(n \cdot c, c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)})$$

4. Temps d'attente d'un message

Notons avec PRS le temps de présence d'un message dans le système (y compris la durée U de service). L'attente sera :

$$ATT = PRS - U \quad \text{donc} \quad E(ATT) = E(PRS) - E(U)$$

Considérons une file quelconque, par ex. la file 1. De son point de vue tout se passe comme s'il y avait un serveur de durée de service V. A l'arrivée d'un message dans la file 1 il y trouve  $L_1$  messages et le temps PRS de présence de ce message satisfait (en moyenne)

$$E(PRS) \leq (1 + E(L_1)) \cdot E(V)$$

On peut montrer que  $E(V) = \frac{E(S)}{c} \cdot E(U)$

Le principe pour montrer cela est le suivant.

Notons avec T la durée entre deux fins successives de service sur la file 1. La moyenne  $E(T)$  est égale à la durée moyenne séparant les arrivées successives sur la file 1 (pour une durée très longue et en régime stationnaire on doit avoir nombre arrivées  $\approx$  nombre sorties). Les arrivées étant poissonniennes de coefficient  $\frac{a}{n}$  on trouve donc

$$E(T) = \frac{n}{a}$$

Notons avec f, f' deux fins successives de service sur la file 1.

La durée séparant f et f' est  $T = Q + V$  où

V = durée de service

$$Q = \begin{cases} 0 & \text{si en f la file n'est pas vide } (L_1 > 0) \\ \text{durée jusqu'à l'arrivée d'un message si en f la file} \\ & \text{est vide } (L_1 = 0) \end{cases}$$

$$\text{Donc : } \text{Prob}(Q=0) = \text{Prob}(L_1 > 0) = B_0$$

$$\text{Prob}(Q < x) = \text{Prob}(L_1 = 0 \text{ et arrivée message pendant } x)$$

Tenant compte des propriétés des arrivées poissonniennes (de coeff.  $\frac{a}{n}$ ) on a :

$$\text{Prob}(Q < x) = \text{Prob}(L_1 = 0) \cdot (1 - e^{-\frac{a}{n} \cdot x}) = (1 - B_0) \cdot (1 - e^{-\frac{a}{n} \cdot x}) ; \text{ on obtient :}$$

$$E(Q) = (1 - B_0) \cdot \frac{n}{a} . \text{ Comme } E(T) = E(Q) + E(V) \quad \text{on a}$$

$$E(V) = \frac{n \cdot B_0}{a} = \frac{n \cdot B_0}{a \cdot E(U)} \cdot E(U) = \frac{E(S)}{c} \cdot E(U)$$

En conclusion

$$E(\text{PRS}) \leq (1 + E(L_1)) \cdot \frac{E(S)}{c} \cdot E(U)$$

La valeur de  $E(L_1)$  a déjà été calculée. On trouve finalement :

$$E(\text{PRS}) \leq \left(1 + \frac{1}{n} \cdot \left(c + \frac{c^2 \cdot u_2}{2 \cdot (1-c)}\right)\right) \cdot \frac{E(S)}{c} \cdot E(U)$$

[les quantités  $c, u_2, S$  ont déjà été calculées]



B I B L I O G R A P H I E

- [ANC1] F. ANCEAU  
*Contribution à l'étude des systèmes hiérarchisés de ressources dans l'architecture des machines informatiques*  
Thèse de Doctorat d'Etat, Université de Grenoble, Décembre 1974.
- [ANC2] F. ANCEAU, J.F. PONS, A. ZAMBRANO  
*Autocommutateur décentralisé*  
Rapport final contrat CNET 75 9B 533, ENSIMAG, Grenoble, Novembre 1976.
- [CLA] J. CLAVIER, M. NIQUIL, G. COFFINET & F. BEHR  
*Théorie et technique de la transmission des données*  
Masson & Cie Editeurs, 1972.
- [COU] B. COURTOIS  
*Détection des pannes et disponibilité dans les organes centraux d'un autocommutateur téléphonique à structure répartie*  
Rapport de contrat CNET 76 9B 378, ENSIMAG, Grenoble, Avril 1978.
- [DEG] B. DEGREDEL  
*Communication et protocole dans une architecture multi-micro-processeurs*  
Thèse de Docteur Ingénieur, Université de Lille, Avril 1977.
- [DON] R.A. DONNAN & J.R. KERSEY  
*Synchronous data link control : a perspective*  
IBM Systems Journal, n° 2, 1974.

- [DRO] D. DROMARD  
*Représentation d'une procédure de transmission par des automates d'états finis*  
Thèse de Docteur Ingénieur, Institut de Programmation, Paris VI, Juin 1974.
- [KEN] KENNETH M. TRUE  
*The interface handbook line drivers and receivers*  
Fairchild Semiconductors, 1975.
- [LEC] C. LECERF  
*Rapport sur le projet ENSIMAG : Architecture d'un autocommutateur à structure répartie*  
Rapport interne FTI/RCI/PLC/10, CNET Lannion, Novembre 1977.
- [LEG] P. LE GALL  
*Les systèmes avec ou sans attente et les processus stochastiques*  
Dunod, 1972.
- [MAR] M. MARINESCU & P. NICOLOPOULOS  
*Mécanisme de communication par bus série pour des réseaux informatiques locaux*  
Congrès Informatique AFCET, Novembre 1978.
- [MET] R.M. METCALFE & D.R. BOGGS  
*ETHERNET : distributed packet switching for local computer networks*  
C.ACM, vol 19, n° 7, Juillet 1976.
- [NIS1] L.B. NISNEVITCH & G.G. STETZIURA  
*Système multicanal à priorité décentralisée*  
Compte rendu de l'Académie des Sciences de l'URSS, Cybernétique Technique, 1971 (en russe).
- [NIS2] L.B. NISNEVITCH & E. STRASBOURGER  
*Decentralized priority control in data communication*  
Second Symposium on Computer Architecture, Houston, Janvier 1975.

- [NOG1] G. NOGUEZ  
*Etude d'un modèle temporel des systèmes séquentiels*  
Thèse de Doctorat ès-Sciences, Institut de Programmation, Paris VI,  
Septembre 1975.
- [NOG2] G. NOGUEZ  
*Contrôleur de transmission par mono-fibre optique (PRT)*  
Rapport interne, Institut de Programmation, Université Paris VI,  
1977.
- [POU] G. POUJOLAT  
*The CORAIL building block system*  
EUROMICRO Newsletter, Octobre 1975.
- [PRE] E. PRESSON  
*Fiches de microprocesseurs*  
Rapport de Recherche n° 52, ENSIMAG, Novembre 1976.
- [SOM] SOMMER  
*COBUS, a firmware controlled data transmission system*  
EUROMICRO Symposium, Venise, Octobre 1976.
- [REA] C.C. REAMES & M.T. LIU  
*A loop network for simultaneous transmission of variable length  
messages*  
Second Symposium on Computer Architecture, Houston, Janvier 1975.



Notices techniques

- [NTEC1] *Microprocesseur S.F.F. 96800 et circuits associés*  
SESCOSEM, Thomson-CSF, Division semi-conducteurs, 1978.
  
- [NTEC2] *Contrôleur de communication multiprotocoles, 2652 MPCC*  
RTC, 1978.
  
- [NTEC3] *Multi-protocol universal synchronous receiver/transmitter, COM 5025*  
SMC Microsystems Corporation.

AUTORISATION DE SOUTENANCE

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

VU le rapport de présentation de :

- Monsieur F. ANCEAU, Maître de Conférences à l'Institut  
National Polytechnique de GRENOBLE

Monsieur Marian MARINESCU

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

Grenoble, le 4 Septembre 1978

**Ph. TRAYNARD**  
Président  
de l'Institut National Polytechnique  
*P.O. le Vice-Président,*

