

## DEUXIEME PARTIE

### LES GRANDS CHOIX DU NANORESEAU

---

Le Nanoréseau actuel est le résultat de nombreux compromis entre des choix techniques les plus variés. Se limiter à décrire son fonctionnement point par point, couche par couche, ne permet pas d'expliquer les choix fondamentaux qui ont permis ce résultat.

L'objet de cette seconde partie est d'expliquer les quelques options de base qui ont assuré la cohérence interne du produit.

- Les contraintes de l'électronique ont poussé à choisir tel système (que permet de faire l'électronique ? ).
- L'analyse des besoins des utilisateurs a conduit à sélectionner tels services (quelles sont les demandes des pédagogues ? ).
- L'expérience des premières versions a incité à élaborer telle procédure d'échange (comment arriver au résultat cherché ? ).
- La nécessité de maîtriser tous les cas d'erreur a fait opter pour telle solution (comment éviter les dysfonctionnements ? ).

Le Nanoréseau n'est pas une accumulation de petites solutions à des petits problèmes techniques, c'est d'abord le choix de fondations solides, constituées d'un nombre limité d'options de base.

## 21ème partie : CHAPITRE 1

LE CAHIER DES CHARGES INITIAL

Dans ce premier chapitre, nous présentons le cahier des charges tel qu'il s'est esquissé au cours de nos premières expériences. Il est fonction :

- du domaine pédagogique, distinct de la bureautique
- des utilisateurs, pédagogues et non informaticiens.
- du volume de mémoires des machines cibles : les fichiers à transférer sont au maximum de quelques dizaines de kilo octets.

## I ) CONVIVIALITE

-----

Les utilisateurs sont des enseignants ou des personnes sans formation initiale. Les ordres doivent être donnés selon une syntaxe claire, parfaitement cohérente avec l'application.

## II ) RAPIDITE

-----

Ceci est vrai de toutes les utilisations des ordinateurs dans le contexte pédagogique, pour deux raisons :

- les lenteurs à un niveau quelconque provoquent un désengagement de l'activité intellectuelle (dissipation).

- Un ordinateur qui paraît lent induit l'idée que le travail demandé est difficile. Les utilisateurs penseront que si déjà lui "souffre" pour exécuter la tâche demandée, alors, eux n'ont aucune chance d'y parvenir.

### III ) MOBILITE

-----

A tout moment il peut être important de déplacer un poste de travail. Le système doit autoriser cette modification de configuration sans perturber le travail en cours dans les autres postes.

### IV ) RUSTICITE

-----

Cela se traduit surtout par le choix du matériel grand public, de préférence au matériel sophistiqué. En particulier les câbles de liaison sont de simples câbles utilisés en HI-FI.

### V ) COUT MODESTE

-----

Le surcoût de connection d'un poste de travail à un réseau existant ne doit pas dépasser 20 % de la valeur du poste de travail.

### VI ) SYSTEME ANTI - PANNE

-----

Un réseau complet ne peut être hors usage lorsqu'un élément quelconque de l'ensemble est défaillant, fut-il le serveur. Il s'agit d'un principe que nous nous étions fixé dans les premières versions du Nanoréseau, où un complément de Basic, intégré dans le boîtier réseau, procurait l'autonomie des postes vis à vis du serveur. Pour des raisons indépendantes de notre volonté, ce service n'a pas été repris dans la version IPT. Nous avons eu maintes occasions de le regretter.

### VII ) TRANSPARENCE AUX CODES

-----

Certains systèmes de communication n'acceptent d'échanger que les codes ASCII. Or toute plage mémoire (Basic, logiciels en langage machine, écrans, ...) est à priori composée d'octets quelconques. D'où l'importance de ce point.

### VIII ) LES DIMENSIONS MAXIMALES DU DISPOSITIF

---

Le réseau doit desservir au moins une dizaine de machines réparties sur un rayon d'une vingtaine de mètres : ce sont les caractéristiques d'un petit laboratoire d'EAO.

A l'exception du point 6, toutes les caractéristiques de ce premier cahier des charges ont été satisfaites au delà de nos souhaits.

## 2ième partie : CHAPITRE 2

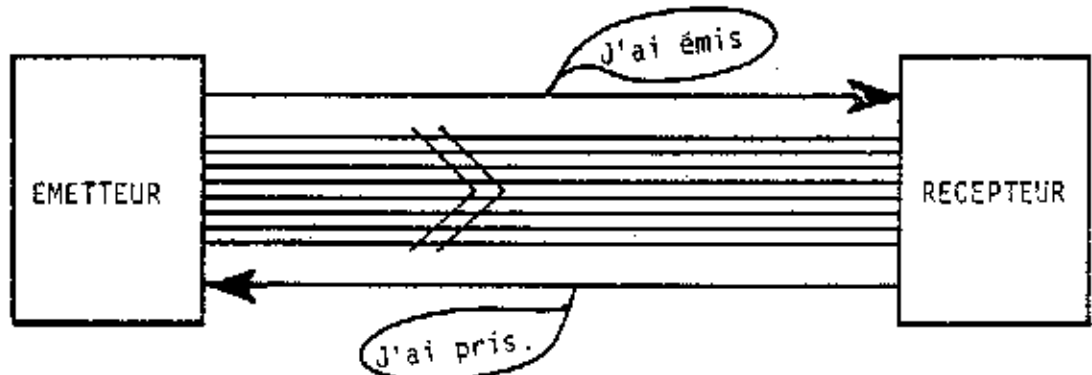
## LES CHOIX DE L'ELECTRONIQUE

Les techniques proposées par l'électronique pour résoudre les problèmes d'échanges sont nombreuses, et le choix n'est pas toujours facile, d'autant plus que toutes les conséquences d'un choix ne sont pas perceptibles au moment où se prend la décision.

Dans le cas présent, on pouvait choisir entre transmissions série ou parallèle ; à l'intérieur des transmissions série, plusieurs techniques étaient possibles ; Une fois le boîtier contrôleur HDLC décidé, comment procéder pour la prise de ligne ; quelle vitesse de transmissions, etc ... Tous les choix ultérieurs concernant la manière d'organiser les échanges sont tributaires de ces options préliminaires.

## I ) LE CHOIX ENTRE SERIE ET PARALLELE

Il s'agit d'un problème habituel par exemple dans le choix des imprimantes. Les procédures d'échange sur des ports parallèles sont classiques et permettent d'atteindre des vitesses de transfert élevées.



Le standard de communication IEEE s'impose dès qu'il faut envisager un retournement du sens des échanges. Or ce standard présente les inconvénients suivants :

- Il limite les échanges à de courtes distances (20 mètres).
- Il n'est pas transparent aux codes.
- Il suppose un contrôleur maître ; or aucun poste ne doit être indispensable dans le réseau.

Les liaisons série sont en principe plus lentes : chaque octet doit être sérialisé, soit à priori un temps 8 fois supérieur. Ensuite il faut ajouter au message des indicateurs permettant au récepteur de recomposer sans erreur des octets à partir de trains de bits 0 ou 1 (bit start, bit stop ou caractères de synchronisation).

Ces inconvénients sont contrebalancés par des arguments techniques importants :

- Le passage d'un câble de liaison à 2 conducteurs est plus aisé, et donc moins coûteux que le passage d'un câble de liaison de 16 conducteurs comme dans le bus IEEE 488.

- On peut, sur un signal unique, appliquer un système de protection élaboré, impossible à étendre à un ensemble de 16 signaux (la transmission par signaux complémentés double le nombre de conducteurs).

Ces arguments confirment l'opinion que nous nous étions faite après l'exploitation du bus IEEE sur les ordinateurs CBM : portée trop réduite, et câbles trop lourds.

## II ) TECHNIQUE DE TRANSMISSION SERIE

---

Dans un tel projet, il faut de toute évidence s'orienter vers les circuits hautement intégrés (VLSI) disponibles sur le marché. Seuls les contrôleurs HDLC ont apporté une solution satisfaisante.

- Ils permettent la transmission de codes quelconques.
- Ils offrent en surplus une vérification de haut niveau sur le contenu du message échangé, le CRC (voir p 117).

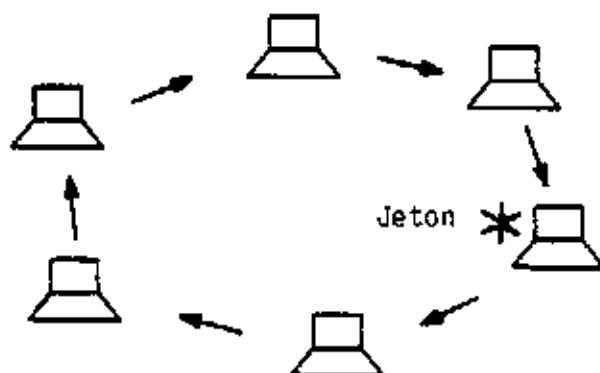
(L'histoire a montré qu'une autre solution aurait pu être envisagée : la réalisation d'un circuit spécialisé VLSI, intégrant ce contrôleur ainsi que le reste de l'électronique associée. Cette solution supposait une production minimum de 10000 pièces.)

## III ) PROCEDURE DE PRISE DE LIGNE

---

En HDLC, il n'y a pas de poste maître comme en IEEE. Lorsqu'un poste veut prendre la ligne pour émettre un message ; il doit soit en avoir l'autorisation, soit la prendre de force.

\* Soit une autorisation d'émettre (= un "jeton") circule d'un poste à l'autre. Tout poste désirant prendre la ligne attend de recevoir ce jeton qui lui en donne le droit. C'est le système le plus sûr.



\* Soit chaque poste décide d'occuper la ligne, après avoir vérifié qu'elle est libre. Cette décision peut coïncider avec une autre décision prise simultanément par un autre poste, d'où risque de collision à prendre en compte.

Il n'y a pas de meilleure technique, et les spécialistes [cf R.Parker et SF.Shapiro, [A9] ou F.Roste, [L3],p.41] estiment que l'on s'oriente vers la coexistence de ces deux systèmes :

- Le système à jeton permet de garantir un passage très régulier du droit de parler ; il est avantageux dans les traitements "temps réel", mais n'optimise pas le débit d'information. Il s'imposera, semble-t-il, pour les échanges à hautes vitesses (cf. H.C.Salwan [A1]).

- Le système "CSMA" (Carrier Sense Multiple Acces) ne garantit pas la possibilité d'émettre un message dans un délai donné, mais permet d'optimiser le débit (intérêt en bureautique).

Nous avons choisi le système CSMA pour les raisons suivantes :

- Nos ordinateurs n'ont qu'un seul processeur, et lui imposer de veiller à la circulation du jeton revient à l'handicaper inutilement. En pratique, les systèmes à jetons sont équipés d'un processeur supplémentaire spécialisé dans la gestion du réseau ("frontal").

- Le système du jeton impose une configuration figée, car chaque poste doit savoir à qui transmettre le jeton reçu. Ceci est contraire à notre cahier des charges.

Plusieurs variantes de techniques CSMA sont expliquées dans la littérature (voir [A9] ou [L3, p.41]). Sur le Nanoréseau, la ligne doit être à l'état de repos stable avant la prise de ligne. L'utilisation d'un délai aléatoire permet de réduire la probabilité de collisions.



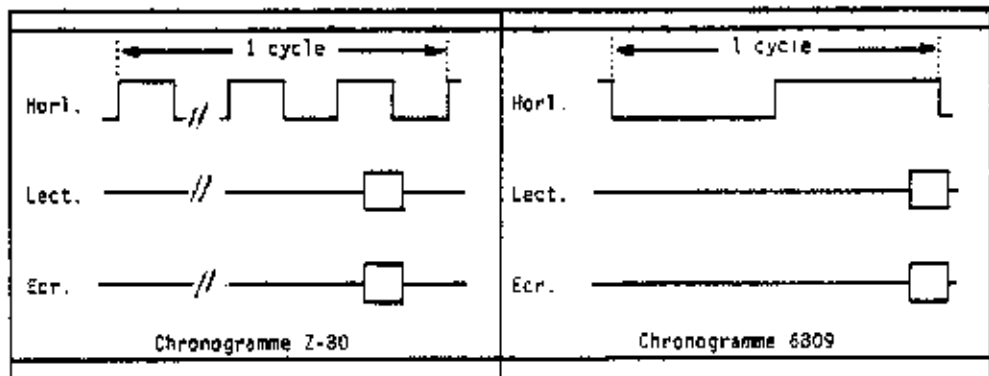
#### IV ) PROBLEME DES RECEPTIONS D'APPELS

Lorsqu'un appel est lancé sur la ligne, seul le poste désigné doit se reconnaître et amorcer la réponse. Nous nous sommes trouvé devant une alternative :

- Soit choisir le contrôleur SIO (Serial Input Output de ZILOG). C'est le contrôleur le plus ancien. Il est équipé du système de reconnaissance automatique d'adresse, c'est à dire que lors d'une réception de trame, il ne transmet une interruption au processeur que si cette trame lui est adressée. Avec un tel contrôleur, qu'importe le nombre de trames circulant sur la ligne, le poste ne sera dérangé qu'à bon escient.

- Soit choisir le contrôleur ADLC (Advanced Data Link Controller, de Motorola), plus simple, mais démuné du système de reconnaissance automatique d'adresse : toute trame doit provoquer une interruption si l'on ne veut pas rater un message. Dans ce cas, il importe de protéger le poste des interruptions inutiles.

On peut imaginer d'utiliser le SIO, plus favorable sur le plan reconnaissance automatique d'adresse, sur le T07/M05 équipé d'un processeur de la famille Motorola (6809). La gestion des signaux à l'intérieur des cycles d'horloge dans les deux familles (Zilog et Motorola) n'est pas identique :



Les informations ne sont pas validées au même instant, si bien qu'un dispositif mixte (avec des éléments des deux familles) ne peut garantir une fiabilité suffisante. Nous avons donc choisi le contrôleur le moins intéressant pour conserver la compatibilité (le T07/M05 est équipé du processeur 6809 de MOTOROLA), et avons développé un dispositif permettant de minimiser les inconvénients de cette non reconnaissance d'adresse.

#### V ) LE CHOIX DES VITESSES

-----

Le contrôleur utilisé supporte des horloges de transmission de 1 MHz, soit un octet toutes les 8  $\mu$ s. Le logiciel n'est capable de desservir le contrôleur qu'à la vitesse moitié. L'expérience nous a appris qu'il faut chercher les vitesses maxima, l'horloge à 500 KHz est facile à fabriquer à partir de l'horloge du processeur : la vitesse de 500 kilo bits par seconde a été retenue, soit le transfert d'un écran en moins d'un tiers de seconde (16000 octets échangés).

#### CONCLUSION

Les choix essentiels sur le plan électronique sont donc définis. Il s'agit de transmissions série, effectuées par un contrôleur HDLC. L'horloge est à 500 Kbits/s. Les prises de ligne se font par saisie brutale, après surveillance de l'état de la ligne pendant un délai aléatoire. Chaque requête déclenche une interruption sur chaque poste.

Les petits problèmes techniques annexes (choix des amplificateurs de ligne, manière de déconnecter les contrôleurs lors d'échanges entre postes extérieurs, détection d'horloge, ...) sont des détails sans répercussion sur le reste du dispositif.

## 2ième partie : CHAPITRE 3

LE CHOIX DES FONCTIONS A IMPLANTER

Le but premier de notre travail était de permettre le chargement de logiciels. L'infrastructure à élaborer pour réaliser cet objectif permettait d'offrir à bon compte d'autres services, d'autres "fonctions".

Le terme fonction n'est peut-être pas évident. Une personne utilisant des logiciels d'EAO pour apprendre les mathématiques n'a pas accès aux fonctions du réseau, sauf lors des consultations du répertoire des logiciels disponibles, et le chargement de tel logiciel sélectionné. Celui qui crée les logiciels a des besoins plus vastes : il doit sauvegarder, rappeler, utiliser des fichiers, appeler des images, etc. Chacun de ces besoins correspond en principe à une fonction.

Au départ de nos travaux sur le Nanoréseau, nous n'avions que fort peu d'idées sur les fonctions à implanter. Peu à peu nous sommes parvenu à cerner plusieurs ensembles de besoins distincts :

- les services normaux d'un système d'exploitation de disques,
- les échanges entre postes,
- les services d'impression.

Dans ce chapitre, nous passons en revue toutes les fonctions qui, peu à peu, se sont avérées nécessaires et suffisantes pour satisfaire nos besoins en pédagogie.

## I ) LES SERVICES D'UN SYSTEME D'EXPLOITATION DE DISQUES

### A ) Le chargement de logiciels

C'était notre premier objectif. Le chargement de logiciel par lecteur de cassettes est trop lent et trop sujet aux incidents ; le lecteur de disquettes est trop coûteux et trop délicat dans le contexte pédagogique. Les quelques milliers d'octets du programme devaient pouvoir être chargés en quelques secondes.

Un logiciel Basic Microsoft est constitué de lignes identifiées et rangées par un numéro. Chaque ligne est enregistrée en mémoire de la façon suivante :

AA AA	NN NN	Texte	00
adresse ligne suivante	numéro ligne (Hexa)	(les mots Basic sont codés)	Terminateur de ligne

Exemple : Voici un logiciel Basic :

```
10 A = RND * 10 + 5
20 FOR I = 0 TO A : BEEP : NEXT
```

Voici comment il s'implante en mémoire du MO5 :  
(avec PNTL = pointeur ligne, pointe la ligne Basic suivante ;  
code 20 = blanc, inutile ; FL = fin de ligne Basic)

4B07	4B 1B 00 0A 41 20 D4 20 FF 9F 20 C9 20 31 30 20
	PNTL lgn10 A = RND * 1 0
4B17	C7 20 35 00 4B 33 00 14 81 20 49 20 D4 20 30 20
	+ 5 FL PNTL lgn20 FOR I = 0
4B27	BB 20 41 20 3A 20 A3 20 3A 20 82 00 00 00 30 00
	TO A : BEEP : NEXT fin fichier

Sur certains ordinateurs (CBM par exemple), l'adresse de début de logiciel est fixe. Dans ce cas, après un échange, il suffit de mettre à jour les pointeurs de fin de Basic et de début de variables.

Sur le T07/MO5, l'adresse de début de logiciel est variable (présence ou non de lecteur de disque, de réseau, nombre de fichiers ouverts ...). En plus de la mise à jour des pointeurs de fin de Basic et de début de variables, il faut rectifier chaque pointeur de ligne (adresse de la ligne suivante) après chaque chargement.

### B ) Le chargement de plage mémoire

Certains logiciels utilisent des sous-programmes en langage machine (fichiers binaires), qu'il faut pouvoir charger en mémoire à partir du serveur. Cette possibilité induit le souhait de pouvoir implanter ces fichiers binaires à tout endroit de la mémoire, d'où un paramètre optionnel précisant la nouvelle adresse d'implantation.

Le problème s'est posé de savoir s'il convenait d'interdire certains chargements de plages mémoires qui, par exemple, écraseraient par inadvertance la "pile" du processeur. Nous avons choisi d'ignorer toute censure dans la mesure où le matériel ne pouvait pas souffrir d'écritures intempestives ; tout est permis, aux risques et périls de l'utilisateur. L'expérience a montré par la suite que chaque censure interdisait des ouvertures non imaginées au départ.

### C ) La sauvegarde

La sauvegarde de logiciels ou de plages mémoire est indispensable dès que le serveur n'est plus de même type que le poste de travail. Elle ne se justifiait pas sur notre réseau prototype CBM : le serveur et les postes de travail étant identiques, une simple saisie de logiciel par le poste équipé du lecteur de disquettes permettait cette sauvegarde.

Sur Nanoréseau, le serveur ne peut pas être utilisé pour les développements : Il n'a pas les mêmes codes Basic, les mêmes services disponibles, les mêmes dimensions d'écran, les mêmes couleurs ... Le développement ne peut se faire que sur les postes de travail. La fonction de sauvegarde est indispensable.

La sauvegarde pose problème lorsqu'un fichier de même nom que celui proposé existe déjà au serveur :

- soit la sauvegarde est refusée : c'est l'option que nous avons retenue dans les versions anciennes du Nanoréseau, où un ordre de sauvegarde spécial autorisait l'écrasement.

- soit l'écrasement est automatique : c'est l'option du Basic Microsoft, qui a été repris dans la version définitive, par souci de cohérence avec le système d'exploitation de disques.

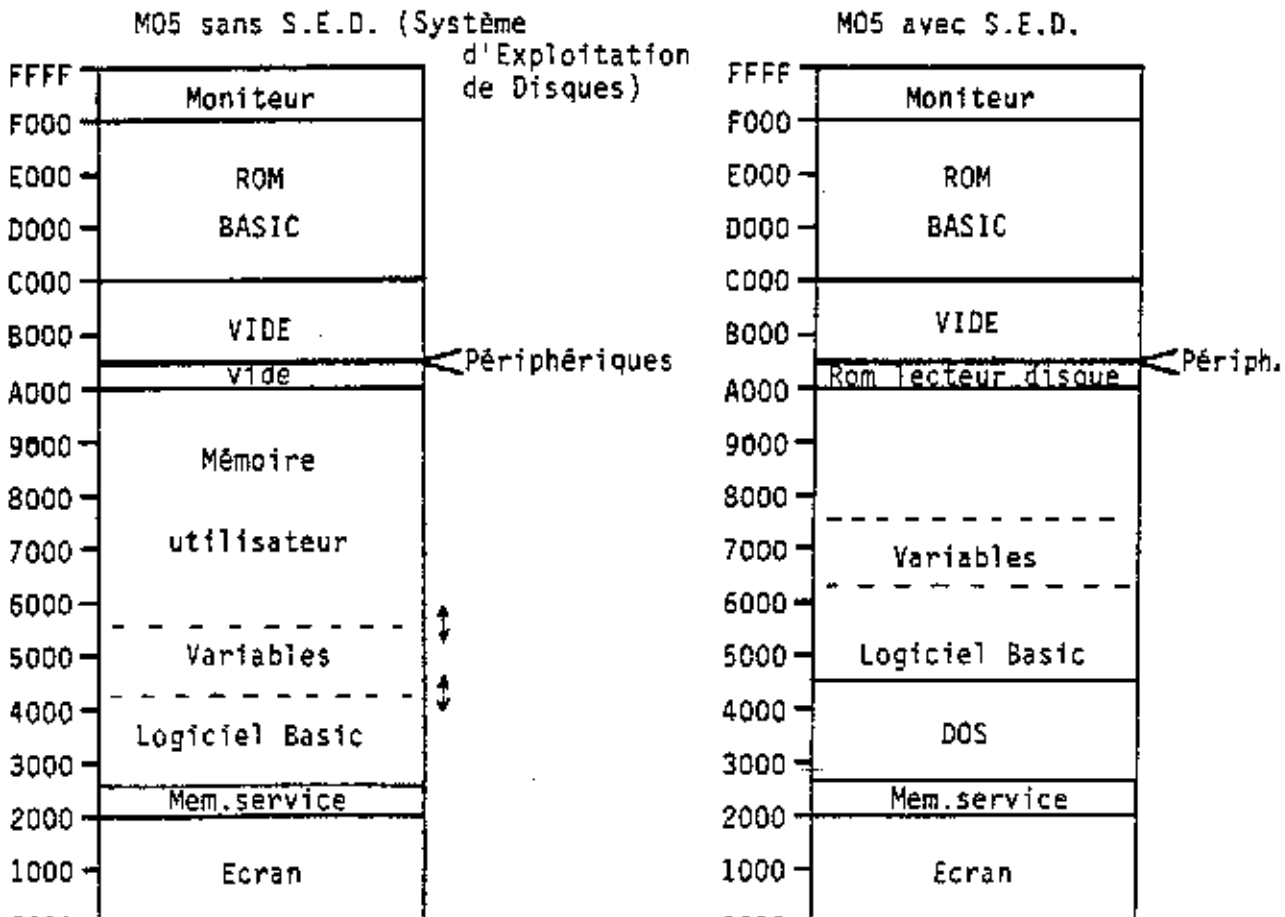
### D ) Le Répertoire

La fonction "consultation de répertoire" semblait peu utile au début de nos travaux, mais dès les premiers essais sur le terrain, elle s'est avérée nécessaire : il est désagréable, en effet, de travailler sur un poste sans savoir quels logiciels sont disponibles au serveur.

L'implantation de cette fonction impose l'usage d'un tampon. En effet, lors de la réception de la réponse du serveur à une demande de répertoire, le réseau apporte dans le poste un nouveau code ASCII toutes les 17  $\mu$ s, alors que pour afficher ce code ASCII sur l'écran, il faut plus d'une milli seconde (pour afficher un caractère sur l'écran, il faut 8 écritures dans la page caractère et 8 écritures dans la page couleur !). D'où le passage imposé par un stockage provisoire.

### E ) La gestion de fichiers

Le domaine de la pédagogie des mathématiques nous a peu porté vers les gestions de fichiers. Par contre nous avons réussi à détourner sur le Nanoréseau ancien l'ensemble du "DOS" fourni par THOMSON avec son lecteur de disquettes.



Il s'agit d'un complément de Basic occupant 8K octets de mémoire et offrant [D6] :

- le Système d'exploitation de Disques (SED) avec fichiers à accès séquentiel et fichiers à accès direct.
- des services outils graphiques (DRAW, CIRCLE, PAINT...).

Le Nanoréseau était encore une réalisation modeste, et l'intégration de ce DOS ouvrait de belles perspectives. En particulier, nous avons enrichi les fonctions du serveur en y intégrant l'émulation du lecteur de disque T07/M05 (logiciels mis au point par M.X.Coine). Tout échange avec un enregistrement du disque était dévié vers le serveur.

Cette solution présentait cependant 3 difficultés :

1 - Un problème de propriété de logiciels : Une Société A peut elle diffuser des logiciels appartenant à une société B, légèrement modifiés par un laboratoire d'université ?

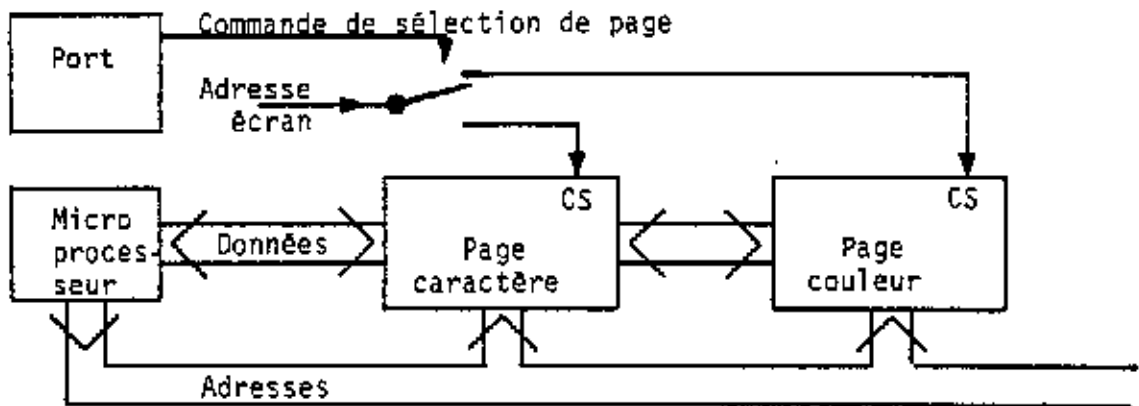
2 - Le lecteur de disques était émulé sur serveur. Un poste de travail jouissait de tous les avantages d'un lecteur de disques, mais côté serveur, il n'apparaissait qu'un fichier global-disque par poste.

3 - Un poste de travail n'accède qu'à un disque fictif de 80 K octets, à l'image des disques du T07 de l'époque. (En fait 2 disques privés et un disque commun en écriture seule).

Pour le Plan Informatique Pour Tous, THOMSON a adapté son "DOS" de manière à rendre disponible sur le réseau tous les services proposés sur les T07/M05 équipés de lecteurs de disquettes (gestion de fichiers et ordres graphiques). En principe, tout logiciel fonctionnant avec lecteur de disque fonctionne sur Nanoréseau, et inversement. Seules les fonctions que nous qualifions de "propriétaire de disque" ont été éliminées (initialisation, recopie de disque, etc..).

## F ) La notion de fichier écran

L'usage pédagogique des premiers Nanoréseaux a fait émerger un besoin non ressenti initialement. Les utilisateurs réclamaient de pouvoir sauver et rappeler des écrans. Un écran est composé de deux pages mémoire de 8000 octets, logées aux mêmes adresses dans le champ de mémoire du processeur, et sélectionnées par un système de commutation de banques.



Le choix de types de fichiers disponibles n'était pas satisfaisant. Il existait en effet des fichiers :

XXXX.BAS	logiciels basic
XXXX.BIN	pages mémoires
XXXX.DAT	fichiers de données

mais rien qui satisfasse les images écran formées de deux fichiers jumeaux et de l'information "couleur du pourtour".

Nous avons signalé ce besoin au constructeur lorsqu'il nous a relayé pour les développements de sa partie dans la version définitive. La sauvegarde et le rappel d'écran ont été intégrés au Basic, et traités avec une technique de compactage d'image pour éviter d'encombrer les disques avec 16000 octets par fichier. Ce compactage, ainsi que le décompactage, est exécuté dans le T07/M05. On trouve donc maintenant des fichiers :

XXXX.MAP	fichiers écran
----------	----------------

Malheureusement cette technique restitue ces images un peu trop lentement, par bandes verticales.



## II ) LES ECHANGES ENTRE POSTES

-----

Historiquement, ces échanges ont été d'abord utilisés pour mettre au point les logiciels de transfert. Pour simplifier les tests, nous avons utilisé les transferts de plage mémoire écran. Les pédagogues ont tout de suite réparé les exploitations possibles de tels échanges.

Plusieurs échanges ont été progressivement mis au point dans les versions anciennes, et sont repris dans la version définitive :

→ saisie par un poste de la plage mémoire d'un autre poste, et recopie, soit aux mêmes adresses, soit à partir d'une adresse précise.

→ saisie par un poste de l'écran d'un autre poste.

Ceci était facile sur GEM (1000 octets à transférer) ; ce qui devient complexe sur T07/M05 (deux fois 8000 octets à transférer + informations diverses).

→ saisie du logiciel d'un autre poste (avec ou sans lancement)

→ concaténation (= mise bout à bout) de logiciels pour permettre la réalisation collective de logiciels sans passer par le serveur.

La difficulté des fonctions d'échanges entre postes réside dans le choix du garde fou : les quatre services cités conviennent à l'usage pédagogique car les postes sollicités ne sont pas perturbés par la demande. La règle du jeu est là, très claire :

Chaque poste peut lire à loisir dans les autres postes  
mais ne peut jamais y écrire.

Au niveau du vocabulaire, nous utilisons le terme "pillage" (non destructif). Dans la version définitive, THOMSON a utilisé le terme "CLONE". Comme dans une photocopie, l'original n'est jamais perturbé.

Une protection supplémentaire permet par ailleurs d'interdire ces pillages, par exemple dans les cas de contrôle de connaissances.

En conséquence chaque poste devient "Mémoire de Masse" en lecture seulement, accessible aux autres postes. Un logiciel peut être appelé s'il se trouve dans les disques du serveur, ou dans un autre poste, ce qui s'avère très confortable à l'usage : l'utilisateur appelle tel logiciel non pas parce qu'il en connaît le nom, mais parce qu'il le voit s'exécuter sur un poste voisin.

### III ) L'IMPRESSION

-----

La notion d'impression recouvre trois besoins distincts qui nécessitent trois traitements très différents :

- ➔ Pour le développeur, il importe d'imprimer des listages de logiciels.

(la suite d'octets enregistrés en mémoire du T07/M05 est traduite en un texte ASCII, qui est envoyé sur l'imprimante au lieu d'être envoyé sur l'écran.)

- ➔ Pour le pédagogue, il importe surtout d'imprimer les copies d'écran.

(Les pixels de l'écran doivent être regroupés verticalement 8 par 8 pour fournir les octets permettant de commander l'imprimante à aiguille.)

- ➔ Pour l'utilisateur polyvalent, il importe de sortir sur imprimante des textes composés par logiciels, plutôt que les afficher sur l'écran.

(Il s'agit de codes ASCII comme dans le premier cas. Il faut bloquer la sortie sur imprimante tant que le texte à afficher n'est pas déclaré complet par un ordre de fermeture de fichier : il ne faudrait pas que plusieurs demandeurs voient leurs textes mélangés !).

La disparité des imprimantes est telle que dans la version ancienne du Nanoréseau, les listages sont envoyés sur l'imprimante du serveur, tandis que les copies d'écran sont envoyées sur l'imprimante d'un poste de travail (qui effectue d'abord une saisie d'écran).

Dans la version actuelle, les 3 services attendus sont pris en charge par le serveur, sur son imprimante. Le standard IBM aura au moins rendu ce service éminent de résoudre le casse-tête des incompatibilités d'imprimantes.

#### IV ) L'INTEGRATION AU BASIC

-----

A l'intérieur d'une application, il importe que la façon de donner les ordres réseau soit cohérente avec la façon de donner les autres ordres. En Basic, cette intégration a été menée très loin par nous-même sur les versions anciennes du Nanoréseau, puis a été achevée par le constructeur dans la version définitive.

La notion d'"Intégration au Basic" suppose que les ordres doivent pouvoir être donnés :

- ① - en mode "commande", directement au clavier.
- ② - en mode "programme" (inclus dans les logiciels, par exemple pour demander le chargement de fichiers binaires).
- ③ - avec des paramètres. Exemple :  
A\$ = "FIC" : LOAD A\$

Dans la Version 1, cette intégration était sommaire, nous avons exploité la technique connue de tous les développeurs des CBM et TRS-80 (déviation de la routine "Acquisition du caractère suivant", implantée en mémoire vive.

Dans la Version 2, l'utilisateur employait toujours les mêmes codes (!E5 pour demande de saisie d'écran du poste 5) mais les commandes étaient intégrées au Basic (nouveaux codes) : nous avons retrouvé le dispositif mis au point par Microsoft pour autoriser l'extension de la table des codes (cf. l'implémentation des ordres du DOS).

Nous avons alors intégré les commandes suivantes :

!\$M	sauvegarde avec écrasement de plage mémoire
!\$	sauvegarde avec écrasement de logiciel
!C	concaténation (mise de logiciels bout à bout)
ID	répertoire
!E	saisie d'écran
!LM	chargement de plage mémoire
!L	chargement de logiciel
!P	impression du listage de logiciel
!R	charger puis exécuter un logiciel
!SM	sauvegarde de plage mémoire sans écrasement possible
!S	sauvegarde de logiciel sans écrasement possible
!V	vérifier (la conformité entre disque et logiciel)
!VM	vérifier (la conformité entre disque et mémoire)
!Z	ordre ouvert à tout développement ultérieurs.

On retrouve pratiquement l'équivalent de toutes ces fonctions dans la version définitive (voir 3ième partie), en plus de la gestion des fichiers de données.

## V ) LES FONCTIONS FUTURES

-----

L'usage quotidien du Nanoréseau dans des circonstances les plus variées fait pressentir l'émergence de nouveaux besoins qui ne pourront aboutir que sur les machines futures, plus "étouffées" en électronique que les TO7/MO5 actuels. Nous n'en citerons que 2 exemples.

### A ) Messagerie

-----

Chaque poste va pouvoir envoyer des messages aux autres postes. Techniquement, ceci est facile, mais la libre disposition d'une telle fonction dans le monde pédagogique risque de créer beaucoup de perturbations chez les élèves ! C'est d'ailleurs contraire aux normes que nous nous sommes fixées, de ne jamais écrire dans un autre poste.

Un boîtier de synthèse de parole très satisfaisant est sorti sur le marché, pour TO7/MO5, compatible avec le Nanoréseau. Un tel dispositif pourrait ouvrir un début de solution, dans la mesure où il n'y a pas écriture sur l'écran.

### B ) Les lutins

-----

De nombreux ordinateurs "grand public" offrent de base une animation d'écran par "lutins". Indépendamment de l'aspect "gadget", ces fonctions sont précieuses pour l'animation des écrans dans le domaine pédagogique, les objets glissant sur un paysage sans le détruire.

La solution technique au problème des lutins est peu satisfaisante sur TO7/MO5. Leur mise au point nécessite un gros investissement en logiciel, ainsi que des tampons importants pour sauvegarder l'image recouverte. Il faut synchroniser les substitutions d'images sur le rafraîchissement d'écran pour éviter des effets visuels désagréables. Le résultat est pauvre comparé aux lutins gérés par circuits électroniques.

La limite d'exploitation des lutins ainsi réalisés est fixée par une contrainte déconcertante : le fond ne doit jamais changer lorsqu'un lutin se déplace (l'ancien fond serait régénéré). Pour la même raison, deux lutins simultanément mobiles ne doivent pas se croiser.

Les enseignants demanderont d'appeler un lutin comme on appelle un fichier au serveur, ou comme on appelle l'écran d'un poste voisin. On se fabriquera ainsi des banques de lutins, qui seront amenés et déplacés sur écran avec le crayon optique par exemple.

#### En conclusion :

La panoplie de fonctions proposées semble actuellement satisfaire les utilisateurs les plus exigeants. Elle est trop riche dans un premier temps. Lorsqu'on demande à quoi sert le Nanoréseau, nous répondons : "c'est un système qui permet de partager un lecteur de disques ainsi qu'une imprimante". Ce n'est qu'après avoir bien assimilé ce premier niveau de service qu'il est possible de comprendre l'intérêt des services supplémentaires et leur exploitation pédagogique.

## 2ième partie : CHAPITRE 4

## LES PROCEDURES DE DIALOGUE

---

Nous avons utilisé le sigle HDLC pour préciser le type de contrôleur de communication. Ce sigle signifie Contrôle de Liaison de données de haut niveau ("High Level Data Link Control"). Il définit un protocole; c'est à dire un ensemble de conventions adoptées par la communauté internationale des spécialistes des télécommunications [L7]. Les industriels ont ensuite fabriqué des circuits intégrés permettant une implantation plus facile de ces règles. Ces circuits intégrés se bornent à échanger les octets, sans analyser les conventions liées aux "mots de contrôle" en particulier. Nous avons choisi le contrôleur HDLC, nous n'avons pas choisi le protocole.

Dans ce chapitre, nous expliquons les raisons du non respect du protocole HDLC : la cohérence interne de ce protocole ne serait pas respectée dans notre cas. Il a donc fallu créer une procédure distincte cohérente avec les données de notre situation.

## I ) LE COURRIER ET LE TELEPHONE

---

Deux modes de communication sont universellement connus : le courrier et la téléphone.

Dans le courrier, un document a une existence propre : le contenu, le contexte du message, les noms de destinataire et d'expéditeur, tout est regroupé dans une "lettre". Le chemin parcouru par la lettre pour atteindre son destinataire n'a aucune importance. L'échange est réalisé "en temps différé". Si l'on veut être sûr que la lettre est correctement parvenue à destination, il suffit de demander un accusé de réception, et d'attendre un temps "raisonnable".

Les messages doivent tenir dans les enveloppes. On peut imaginer que pour les grosses informations, on envoie des suites de lettres numérotées. Ainsi le destinataire, avant même de décacheter les enveloppes, sait s'il en manque une.

Le téléphone est tout différent : il faut que le correspondant soit présent et disponible. Lorsque la communication est établie, toute l'information s'échange "en temps réel". Quand la communication s'achève, le demandeur a toutes les garanties que le message a été reçu.

Il s'avère que le protocole HDLC permet de traiter le courrier, et que le Nanoréseau est du type téléphone. Lors de notre projet de réseau avec boîtier "communicateur" extérieur, il était prévu d'implanter HDLC. C'était normal dans la mesure où le communicateur jouait le rôle de "boîte à lettre". Pratiquement tous les réseaux professionnels travaillent ainsi.

## II ) LES PROBLEMES D'IMPLANTATION DE HDLC

Il est hors de question d'implanter tout HDLC dans un réseau local de la taille envisagée. Il n'y a pas de noeud de communication, tous les postes sont sur un bus unique. Il est impensable que l'ordre d'arrivée des messages soit différent de l'ordre d'émission. Tout une partie du protocole est abandonnée d'office.

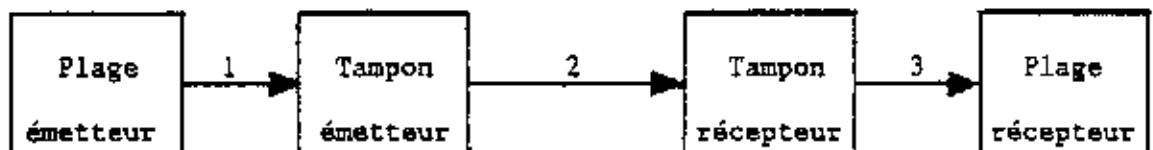
La logique interne de HDLC veut que les messages soient autonomes (les "paquets" de TRANSPAC). Ce système implique les points suivants :

1) Chaque poste de travail doit disposer d'un tampon en émission, et d'un autre en réception, car un paquet n'est interprété qu'à posteriori. Ces tampons peuvent être internes ou externes à l'ordinateur.

2) Les échanges s'effectuent entre plages mémoires limitées par la taille des tampons. Il faut donc envisager de tronçonner les envois.

3) Dans chaque poste, il faut réserver la place nécessaire aux logiciels d'échange ; plus ces échanges sont complexes, plus il faut de place.

4) Tout échange implique trois copies successives, ce qui augmente inutilement les délais :



L'implantation de HDLC est disproportionnée à votre situation. La technique suivante contourne ces inconvénients.



### III ) LA SOLUTION "TELEPHONE" : le NANORESEAU

-----

Il s'agit d'affectuer des échanges directs de mémoire à mémoire en "temps réel" et non plus en "différé". Les tampons n'existent plus. Le système de sécurité à mettre en oeuvre est spécifique. On joue sur le facteur temps, un peu accessoire dans les échanges de type "courrier".

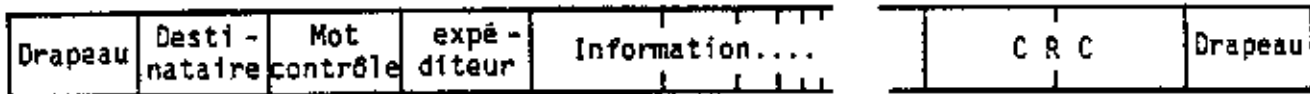
Nous travaillons en trains de messages et non en message autonome. En effet, un descripteur doit être transmis pour désigner les adresses où seront réceptionnés les octets du message. Ce descripteur doit avoir été reçu et validé préalablement, sinon il faudrait attendre la fin du message (validation par la "CRC") pour savoir où le réceptionner, ce qui serait absurde !

Il est intéressant de constater que le réseau ACORN, équipant les centres scolaires en Angleterre, a été confronté un an avant nous à ce même problème d'absence de tampon, puisqu'ils travaillaient également sur des machines à faible mémoire, et que nous avons adopté une technique semblable. Par contre, ce réseau travaille selon le système "courrier. (voir C.N.R.Dellar [A8]).

Sur Nanoréseau, nous nous sommes donc imposé que l'adresse d'implantation, et même le nombre d'octets soient connus avant la réception de toute trame (= message élémentaire au sens de HDLC). Nous nous sommes inspiré de la notion de trame de service et de trame d'information dans HDLC. Les trames de service sont limitées à deux octets : le numéro de destinataire, et le mot de contrôle. En pratique, suite à une analyse de M.C.Vieville, ayant décelé une faille possible dans les échanges ([T1]), un troisième octet a été systématiquement ajouté, donnant le numéro de l'expéditeur.

Drapeau	Desti - nataire	Mot contrôle	expé - diteur	C R C	Drapeau
---------	--------------------	-----------------	------------------	-------	---------

Les trames d'information n'ont pas de limite de longueur.



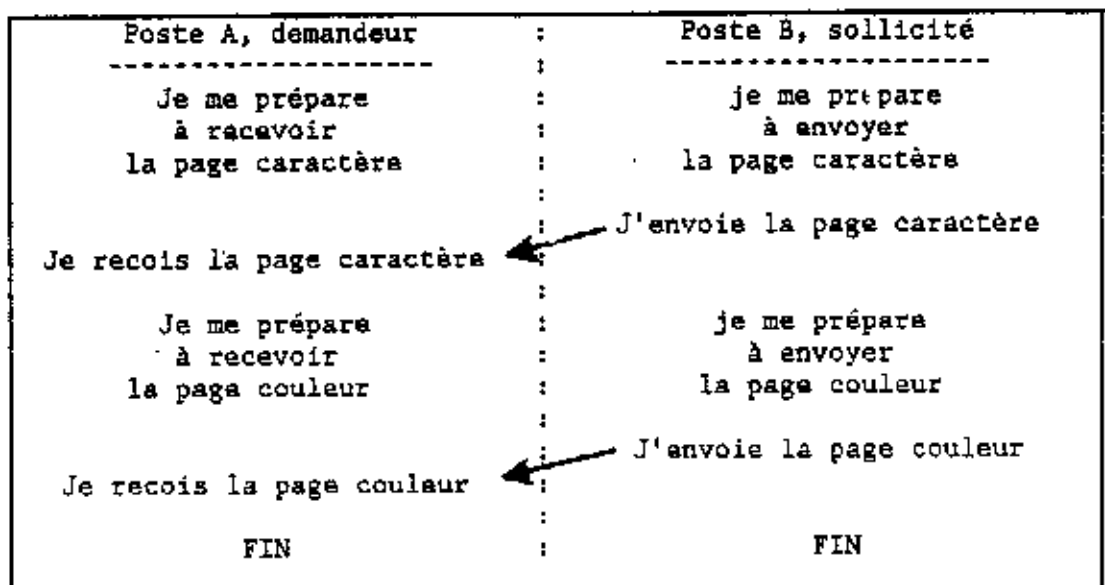
Ces deux types de trames diffèrent par une indication (bit 7) dans le mot de contrôle.

Il suffit donc d'enchaîner ces deux trames, en "trains de trames". Dans cette image, la "locomotive" est une trame de service, courte, servant de descripteur de la trame suivante, et la "wagon" contient l'information annoncée.

#### IV ) LES PREMIERS DIALOGUES ENTRE POSTES

Dans la version ancienne du Nanoréseau, nous avons utilisé le principe précédent uniquement pour amorcer l'échange entre deux postes. La transmission de la consigne étant faite, il fallait exécuter l'ordre. Chaque poste était amené à jouer sa partition, synchronisée avec la partition de l'autre.

Par exemple, suite à une demande de saisie d'écran formulée par la poste A au poste B, le processus suivant se déroulait :



Cette façon de faire était efficace, mais artisanale :

- il fallait développer deux logiciels synchrones de transfert d'écran, l'un en tant qu'émetteur, et l'autre en tant que récepteur. Soit deux logiciels de mise au point difficile.

- Le traitement des erreurs est difficile : en cas d'incident, comment un poste peut savoir où en est l'autre ? La seule parade est de reprendre à l'origine le traitement complet.

#### V ) LA TELECOMMANDE DE L'ESCLAVE PAR LE MAITRE

-----

La lourdeur de ce premier type d'échange a conduit à s'interroger sur cette technique. Plutôt que d'organiser deux logiciels distincts et synchrones, il était plus simple d'implanter un logiciel unique, implanté dans un poste MAITRE, télécommandant l'autre poste ESCLAVE. Le processus d'échange devient alors asynchrone, le poste maître travaillant à son rythme, et le poste esclave obéissant instantanément aux ordres reçus.

Cette technique a été d'autant plus facile à mettre au point qu'il s'est avéré que tous les échanges se décomposaient en suite de transferts de plages mémoire, et de consignes.

Il suffit donc de généraliser la notion de "trains de trames" présentée plus haut. Ce nombre de "trains de trames" est très limité :

- |   |
|---|
| <p>1) → ce qui suit est une consigne (donc à ranger à l'endroit réservé pour les consignes et à exécuter).</p> <p>2) → ce qui suit est une plage mémoire, dont l'adresse d'implantation et la longueur ont été définies dans la dernière consigne.</p> <p>3) → c'est au correspondant à envoyer cette même plage mémoire.</p> |
|---|

Par la suite, nous utiliserons l'expression **PLAGE TRANSFERABLE** pour désigner la plage dont les coordonnées sont précisées dans la consigne.

Il reste à ajouter deux "trains" particuliers : la prise de contact initial, et la déconnection.

Qui est maître et qui est esclave dans un échange ? Peu importe : que ce soit le poste demandeur ou le poste sollicité, la facilité de programmation reste semblable. Par référence au serveur, le poste sollicité prend le statut de poste maître car le serveur est souvent sollicité et ne peut pas être en situation d'esclave. Nous avons bien sûr introduit la possibilité de renverser le lien.

#### En conclusion

Nous nous sommes éloigné du protocole HDLC pour conserver l'avantage de la simplicité, et ne pas encombrer les mémoires avec des tampons. Ce choix nous a conduit à travailler en "trains de trames" et nous a heureusement permis d'effectuer des transferts directs de mémoire à mémoire. La technique des "trains de trames" ayant été maîtrisée, nous avons généralisé les échanges de type maître-esclave en confiant toute la responsabilité de l'échange à un seul poste.

## Zième partie : CHAPITRE 5

### LE SYSTEME DE SECURITE

---

Dans un logiciel autonome, la quantité d'états possibles est importante, mais maîtrisable. Après un temps plus ou moins long de débogage, on peut prétendre que tout fonctionne, avec une probabilité supérieure à 99,999 % par exemple.

La difficulté dans le domaine des réseaux, c'est que le nombre d'états possibles n'est pas cernable dans la mesure où l'environnement est susceptible d'une grande mouvance : il est possible d'envisager et de traiter presque tous les incidents imaginables : il reste les cas "impossibles", sans parade tant qu'il n'ont pas été identifiés.

#### I ) LES PRINCIPES DE SECURITE CHOISIS

---

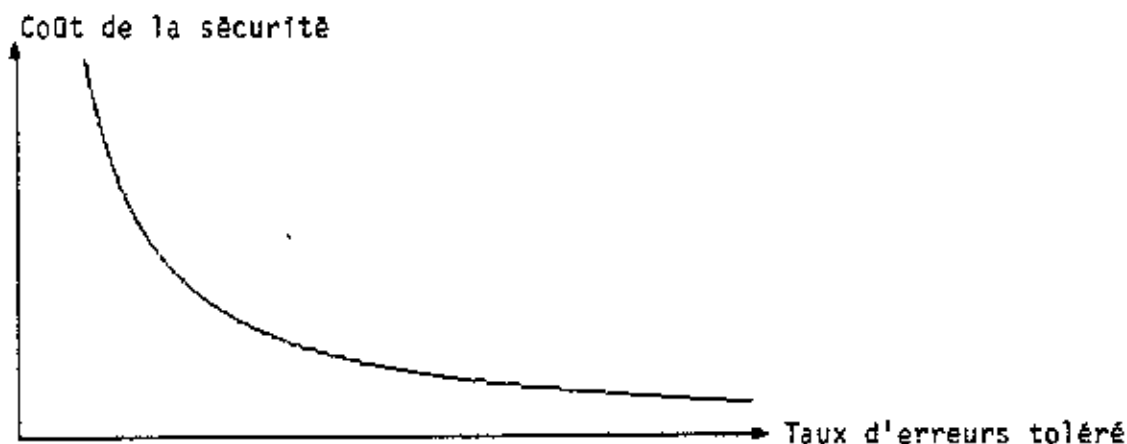
L'objectif d'un système de sécurité est de faire tendre vers zéro le pourcentage d'incidents non détectés. Dans le cas du Nanoréseau, ces incidents peuvent prendre deux formes :

- soit un poste reste bloqué en attente : il a envoyé une requête à un poste ; la requête a été acceptée et peut-être satisfaite ; mais l'ordre de déconnection n'a pas été reçu. C'est le point faible de la procédure de dialogue choisie mais ce cas n'arrive jamais ... en laboratoire. Il se reconnaît au fait que le poste bloqué génère un son de disca rayé.

- soit un ordre erroné n'a pas été détecté, et s'exécute. La parade est impossible. Les conséquences sont imprévisibles. L'identification de l'incident est hautement improbable : le MOS deviendra peut-être "fou" sans raison apparente.

Dans le domaine des réseaux, on ne peut pas parler de sécurité absolue. On peut seulement évaluer que sur 1 million d'incidents, moins de 1000 ou de 10 échapperont au contrôle.

Le coût de la sécurité est intuitivement une fonction hyperbolique du taux d'erreurs toléré : pour réduire d'un facteur  $N$  le taux d'erreur, il faut multiplier par  $N$  le coût des logiciels de sécurité. D'où un compromis à trouver : l'incident catastrophique dans les liaisons avec une navette spatiale devient anodin dans un laboratoire d'ÉAO.



Les logiciels des couches de base du Nanoréseau occupent moins de 1500 octets logés dans la mémoire morte du boîtier. L'ensemble du système de sécurité ne peut donc excéder quelques centaines d'octets.

En pratique, nous limitons les tests aux indicateurs indépendants. Il serait possible par exemple d'intégrer à chaque trame le "cheksum" (addition sur 1 ou 2 octets) des octets échangés : il s'agit de tests venant en doublure du test du "CRC" effectué automatiquement par le circuit intégré "contrôleur". Dans notre cas, cette redondance ne se justifie pas.

Nous avons choisi de nous en tenir aux règles suivantes :

- Une trame émise est à priori correcte.
- Une trame reçue est à priori altérée.
- Une trame n'est déclarée valide qu'après les tests suivants :

- 1 Le destinataire est-il correct ?
- 2 En cours d'échange, l'expéditeur est-il le même que celui de l'appel initial ?
- 3 Le numéro de séquence est-il le numéro attendu
- 4 Le mot de contrôle est-il cohérent ?
- 5 Le nombre d'octets attendus est-il vérifié?
- 6 Le "CRC" est-il correct ?

Une fois passés ces tests, la validité de l'information sera considérée totale (sachant qu'il y a une probabilité non nulle que ces tests aient été insuffisants).

## II ) LE DISPOSITIF DE SECURITE

-----

Nous allons expliquer brièvement le dispositif complet que nous avons implanté sur le Nanoréseau en recherchant la sécurité maximale pour un coût réduit. Toutes ces informations seront développées dans la troisième partie.

1 - La transmission des états 0 et 1 sur simple fils est sensible aux parasites. Nous utilisons la technique des transmissions par états complémentés sur paire torsadée, qui diminue sensiblement l'effet des parasites électriques.

2 - La sécurité maximale est assurée par le "CRC". Il s'agit d'un test interne au contrôleur de communication utilisé. Le taux de sécurité varie de 100 % (altération de quelques bits de la trame) à 99,9985 dans le cas de perturbations majeures (c'est à dire que sur 65000 trames erronées entrantes, une seule statistiquement ne sera pas détectée dans le pire des cas).

3 - Le nombre d'octets reçus est contrôlé. En cas de non correspondance avec le nombre d'octets attendus, la trame est refusée.

4 - Les postes qui ne se reconnaissent pas comme destinataires d'un échange se déconnectent du réseau tant que l'échange n'est pas achevé. Les incidents sont donc exclusivement du ressort des deux postes en communication.

5 - Le fait de travailler en "trains de trames" oblige les échanges à obéir à des enchaînements de séquences : telle trame amorce doit être suivie de tel type de trame. Toute anomalie déclenche la procédure d'erreur.

Chaque train de trames s'achève par une trame reçue par l'émetteur du train : ce dernier peut ainsi évaluer la validité du train complet.

6 - Un nouveau numéro est affecté à chaque "train de trames" et est vérifié systématiquement.

7 - Les développeurs sont invités à inclure des niveaux de sécurité supplémentaires, par exemple en Basic, le drapeau d'erreur est levé systématiquement avant chaque échange, et doit être baissé au cours de l'échange.



L'ensemble de ce dispositif offre une sécurité remarquable (voir 4ième partie, chapitre 1). Les travaux d'amélioration des sécurités s'apparentent aux travaux de débogage dans la mesure où les failles subsistantes correspondent à des phénomènes imprévus parce que imprévisibles sur papier. Ces travaux sont ingrats : chaque fois que nous avons pu détecter et corriger une faille, une nouvelle se profilait, 10 ou 100 fois moins fréquente, et donc 10 ou 100 fois plus difficile à détecter. Voir en 4ième partie les informations permettant d'évaluer le degré de sécurité atteint.

#### En conclusion

Dans ce chapitre, nous avons expliqué notre position par rapport aux problèmes d'erreur sur réseau : c'est un problème qui se pose sous la forme : quel prix accepte-on de payer pour quel niveau de sécurité. Nous avons expliqué les choix que nous avons effectués, puis nous avons résumé l'ensemble des dispositifs de contrôle installés dans le Nanoréseau.