Contribution à la réalisation d'un service de transfert de fichiers et de transfert et de manipulation de travaux sur le matériel MITRA suivant le standard NIFTP B (80)

Maurice Testa

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE

GRENOBLE

MEMOIRE

Présenté en vue d'obtenir

Le DIPLOME D'INGENIEUR C.N.A.M.

En

INFORMATIQUE

Par

Maurice TESTA

"Contribution a la réalisation d'un service de transfert de fichiers Et de transfert et de manipulation de travaux Sur matériel MITRA suivant le standard NIFTP B(80)."

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE

GRENOBLE

MEMOIRE

Présenté en vue d'obtenir

Le DIPLOME D'INGENIEUR C.N.A.M.

En

INFORMATIQUE

Par

Maurice TESTA

"Contribution a la réalisation d'un service de transfert de fichiers Et de transfert et de manipulation de travaux Sur matériel MITRA suivant le standard NIFTP B(80)."

"Contribution a la réalisation d'un service de transfert de fichiers et de transfert et de manipulation de travaux en milieu hétérogène sur matériel MITRA 625 suivant le standard NIFTP (Network Independant File Transfer Protocol) utilisant le protocole de transport ECMA 72-classe 0.

Le travail présenté dans ce mémoire a été effectué dans le cadre de la formation permanente, dans la société BULL-SEMS (Société Européenne de Miniinformatique et de Systèmes).

Je tiens à remercier chaleureusement :

Monsieur J.Y RANCHIN Professeur au Conservatoire National des Arts et Métiers qui m'a fait l'honneur d'accepter mon dossier de thèse,

Le Président du Jury ,Monsieur le Professeur L. BOLLIET, Directeur du Département Informatique à l'Institut Universitaire de Technologie de Grenoble pour l'aide qu'il m'a apportée pendant le projet.

Monsieur C. SAURY Chef du Service Grandes Affaires à la SEMS qui m'a proposé le sujet et m'a permis de le réaliser dans les meilleures conditions,

Monsieur M. HABERT Chef du Service Communication et Protocoles qui ma donné toutes les facilités pour sa réalisation,

Mes camarades de l'équipe de développement Madame F. TARIEL et Monsieur J.M. POULET Chef de projet dont les conseils m'ont été très précieux pour la rédaction de ce mémoire.

Enfin je tiens à remercier particulièrement Monsieur J. DAROLD, Chef du Groupe Documentation ainsi que Mademoiselle J. SCHNEIDER qui se sont occupés avec gentillesse et compétence de la saisie de ce document.

- TABLE DES MATIERES -

1 INTRODUCTION

Le but du projet présenté dans ce mémoire a été de réaliser un système de transfert de fichiers multilatéral entre machines hétérogènes baptisé FTS-NIFTP, à base du système de communication SEMS SCS2 sur matériel MITRA 625 suivant le standard NIFTP (Network Independant File Transfer Protocol) , utilisant le protocole de transport ECMA/72 Classe 0, pour le compte de la CCE (Commission des Communautés Européennes).

A terme l'objectif de la CCE est de pouvoir effectuer des transferts de fichiers entre les ordinateurs des différents constructeurs ayant retenu ce standard (SIEMENS, ICL, CII-HB, IBM, SEMS) interconnectés soit par son propre réseau à commutation de paquets (C.C.E. In-house Packet Switching Network) ou par d'autre réseaux de même type (Transpac, Euronet, Luxpac, DCS, entre autres).

Le projet FTS-NIFTP s'est scindé en deux étapes qui ont constitué 2 LOTS distincts consistant en la réalisation:

- D'un service de transfert de fichiers séquentiels pour le LOT 1.
- D'un service de transfert et de manipulation de travaux pour le LOT 2.

La réalisation de ces deux étapes a été avant tout un travail d'équipes (composées de trois personnes pour la première et de deux pour la seconde),un certain nombre de tâches étant effectuées en commun (définition de l'architecture du produit,découpage en modules,définition des interfaces entre les modules),les autres (étude et réalisation des différents modules, intégration,tests) étant réparties entre les différents membres du projet.

En ce qui concerne la première étape, mon travail a porté essentiellement sur l'étude et la réalisation du module principal du système, chargé d'assurer notamment la fonction session (ouverture, fermeture des chemins virtuels TRANSPORT, gestion des échanges de données avec le TRANSPORT) et la fonction transfert de fichiers conformément au protocole NIFTP B(80).

Au cours de la deuxième étape, j'ai été responsable de l'analyse des nouvelles fonctionnalités du produit et de leur implémentation.

Ce mémoire a pour but de présenter ce travail en le situant dans son cadre global.

Pour ce faire, nous allons tout d'abord nous attacher à exposer le protocole NIFTP dans son ensemble et le situer par rapport aux principaux standards de transfert de fichiers existants .

Dans un deuxième temps, nous décrirons les deux étapes de son implémentation sur machine SEMS en insistant sur la réalisation du module gestion du protocole qui a constitué la majeure partie de ma contribution au projet.

2 LE PROTOCOLE NIFTP

2.1 NIFTP ET L'ISO

2.1.1 Problèmes liés a l'interconnexion de systèmes hétérogènes.

Les exigences de l'informatique actuelle sont multiples et se caractérisent par trois grands principes :

- Ouverture .
- Hétérogénéité.
- Répartition.

En effet avec les progrès de la technologie et l'abaissement des côuts du matériel informatique , un mode d'organisation nouveau est apparu, L'architecture distribuée dans laquelle les ressources et les fonctions de traitement de l'information sont réparties dans les différents constituants d'un réseau.

De même,la nécessité d'interconnecter des systèmes de toute nature est devenue impérative, ceci en préservant un caractère d'ouverture permettant de se raccorder a des réseaux publics et de s'adapter facilement a de nouvelles normes.

De ces différentes exigences est né le modèle pour l'interconnexion des systèmes ouverts (O.S.I.) qui fournit une architecture et un ensemble de standards permettant à des systèmes de coopérer entre eux, sans préjuger des moyens de communication utilisés ni de leurs architectures internes.

Dans un tel contexte, établir des chemins de communication entre les divers systèmes ne représente qu'une partie des contraintes. Les multiples ordinateurs utilisés ayant des logiciels et du matériel différents, il est nécessaire de transformer les informations à échanger suivant des conventions les rendant indépendantes du type d'équipements utilisés, afin de pouvoir véritablement décentraliser le traitement et le stockage de ces informations.

Le protocole NIFTP va apporter une solution au problème du transfert de fichiers entre machines hétérogènes, tout en respectant les principes du modèle OSI. Nous allons montrer comment.

2.1.2 Objectif du protocole NIFTP

L'objectif du protocole NIFTP (Network Independant File Transfer Protocol) est de définir un standard de transfert de fichier dans un contexte de machines hétérogènes, indépendant du type du réseau de communication utilisé.

La raison principale qui a amené à définir ce protocole (dans les années 77-78) a été l'absence d'un standard international dans ce domaine. Ainsi il a été développé en vue d'assurer l'intérim en attendant que les travaux de l'ISO aboutissent pour satisfaire la demande d'un grand nombre d'utilisateurs de réseaux publics ou privés.

Dans le paragraphe suivant, nous allons nous attacher à montrer comment l'architecture de ce protocole va suivre celle du modèle ISO.

2.1.3 Architecture du protocole NIFTP - NIFTP de l'ISO

Le modèle ISO pour l'interconnexion des systèmes ouverts comporte sept couches .

Les entités d'une même couche communiquent entre elles a l'aide d'un protocole normalisé tandis que les entités de deux couches adjacentes communiquent a travers une interface standardisée.

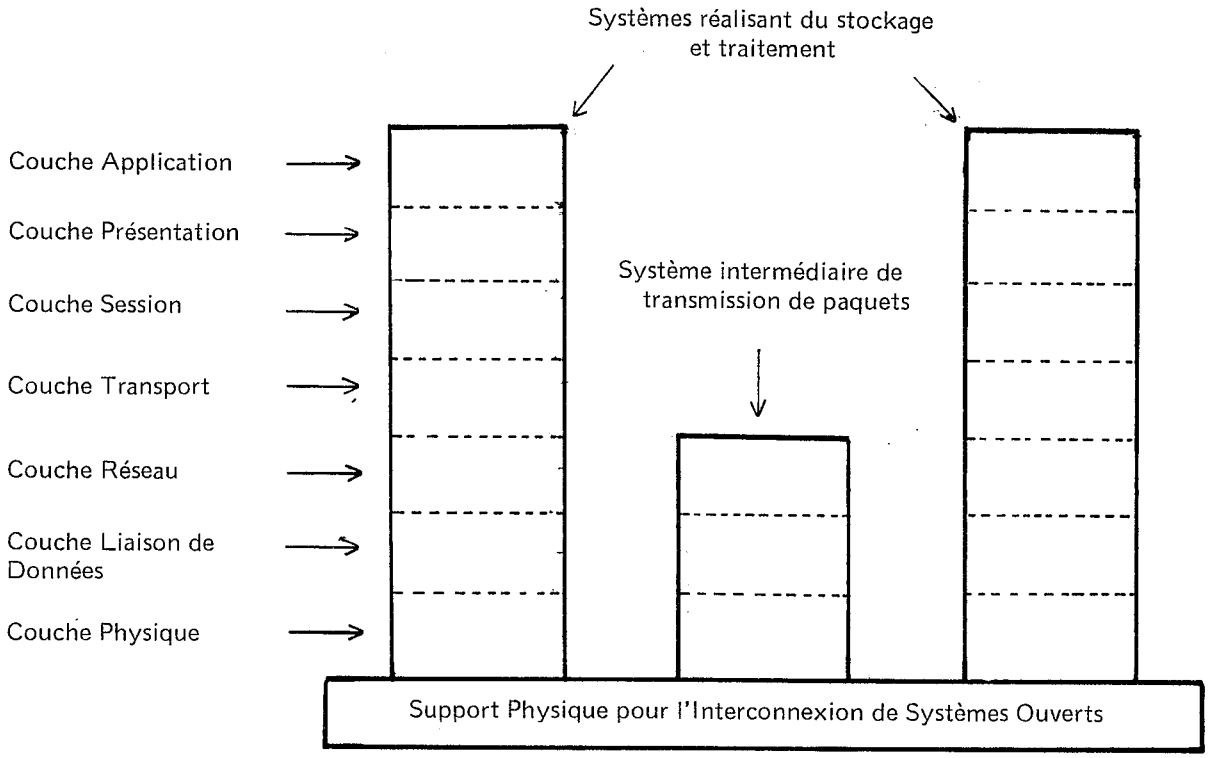


Fig 1 Modèle de Référence pour l'Interconnexion de Systèmes Ouverts

1 - Couche physique : Elle fournit l'interface électrique avec le support de transmission situé entre les équipements et permet l'usage de différents supports physiques d'interconnexion avec différentes procédures de contrôle V21, V35, X21 etc...) conformes aux Avis du CCITT.

2 - Couche liaison de données : Supporte les procédures de contrôle de liaison de données normalisées, X25, VIP etc... Elle se trouve immédiatement au-dessus de la couche physique.

3 - Couche réseau : Va assurer le routage des informations du système origine au système destination à travers un ou plusieurs noeuds intermédiaires. Pour cela, elle supporte les protocoles de connexion réseau (X21, X25).

4 - Couche transport : Elle va assurer le contrôle de bout en bout des données véhiculées entre le système origine et destinataire. Cette couche libère les couches supérieures de tout souci de transport entre elles.

5 - Couche session : Va assurer la gestion des connexions logiques entre les activités réparties ainsi que le contrôle des échanges de données (ainsi que leur synchronisation).

6 - Couche présentation : Va assurer la manipulation des données et assurer leur représentation (codage, transcodage, formattage) au profit du programme d'application.

7 - Couche Application : C'est la couche la plus élevée du modèle et contient les processus d'application exécutant le traitement des informations ainsi que les fonctions de gestion système et application.

Pour des raisons historiques (au moment de la définition du protocole NIFTP seules les couches 1, 2, 3, 4 du modèle OSI étaient définies), l'implantation de NIFTP va se faire au-dessus de la couche transport et assurer les fonctions des couches applications, présentation et session.

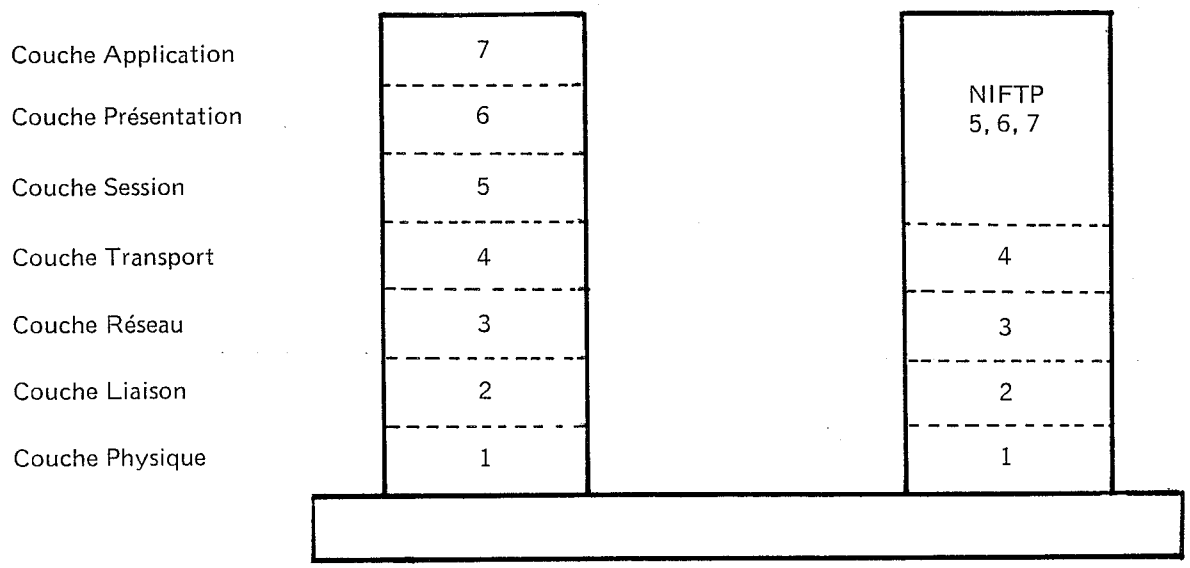


Fig 2 - NIFTP et l'ISO

Afin d'être le plus indépendant possible du réseau utilisé, le protocole s'appuiera sur un transport de fonctionnalités minimum (pas de multiplexage,de contrôle de flux ni de récupération des erreurs) .

Ainsi l'architecture NIFTP respecte globalement les principes du modèle OSI (architecture en couche, interfaces) sans pour autant suivre le découpage imposé par ce modèle pour les couches supérieures (présentation et session). Elle sera de ce fait assez lourde à implémenter . Nous verrons plus loin les avantages et les inconvénients d'une telle architecture.

2.2 PRINCIPES GENERAUX DE NIFTP

2.2.1 Protocole de contrôle et de transfert : les processus P et Q

Un transfert de fichier est initialisé à l'aide d'un stimulus externe qui peut provenir :

- d'un utilitaire exécuté par un utilisateur unique.
- d'un sous-système gérant plusieurs utilisateurs.
- d'un operating-system.

Ce stimulus identifie le processus initiateur ou processus "P", le processus répondeur étant désigné par "Q".

Ainsi deux protocoles sont nécessaires pour exécuter un transfert :

- un protocole de contrôle entre l'utilisateur et le processus P permettant le contrôle du transfert à distance.
- un protocole de transfert entre les processus P et Q.

Seul le protocole NIFTP qui est un protocole de transfert va faire l'objet de cet exposé.

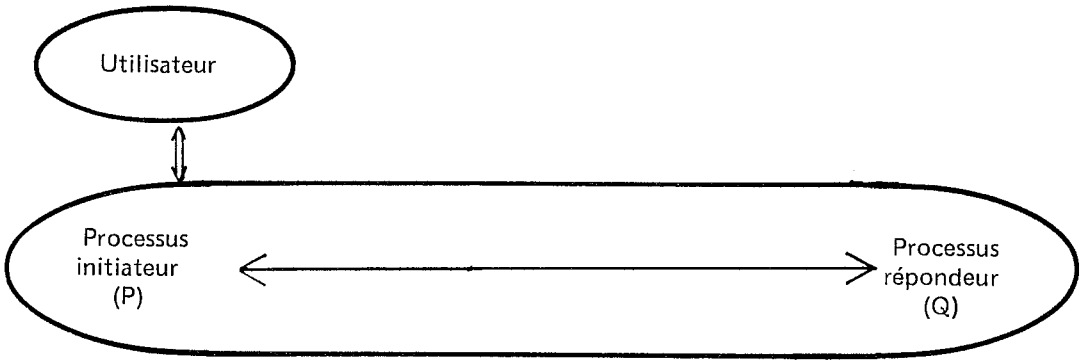


Fig 3 - Le protocole de contrôle

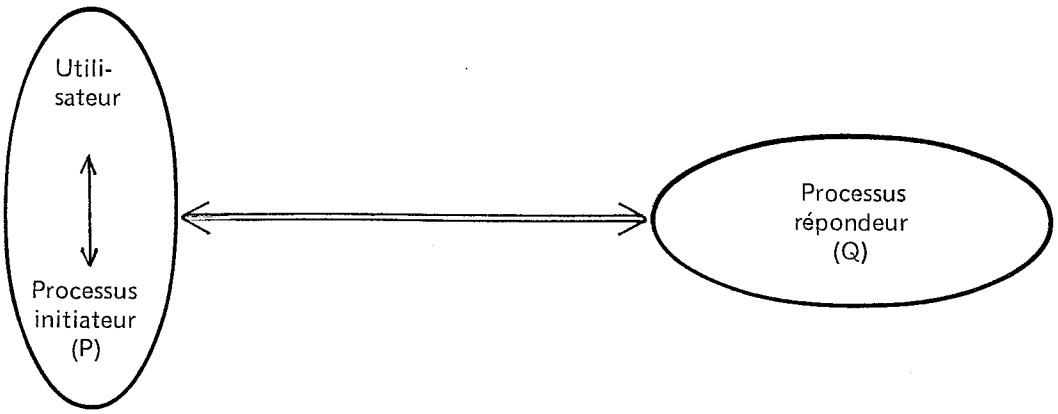


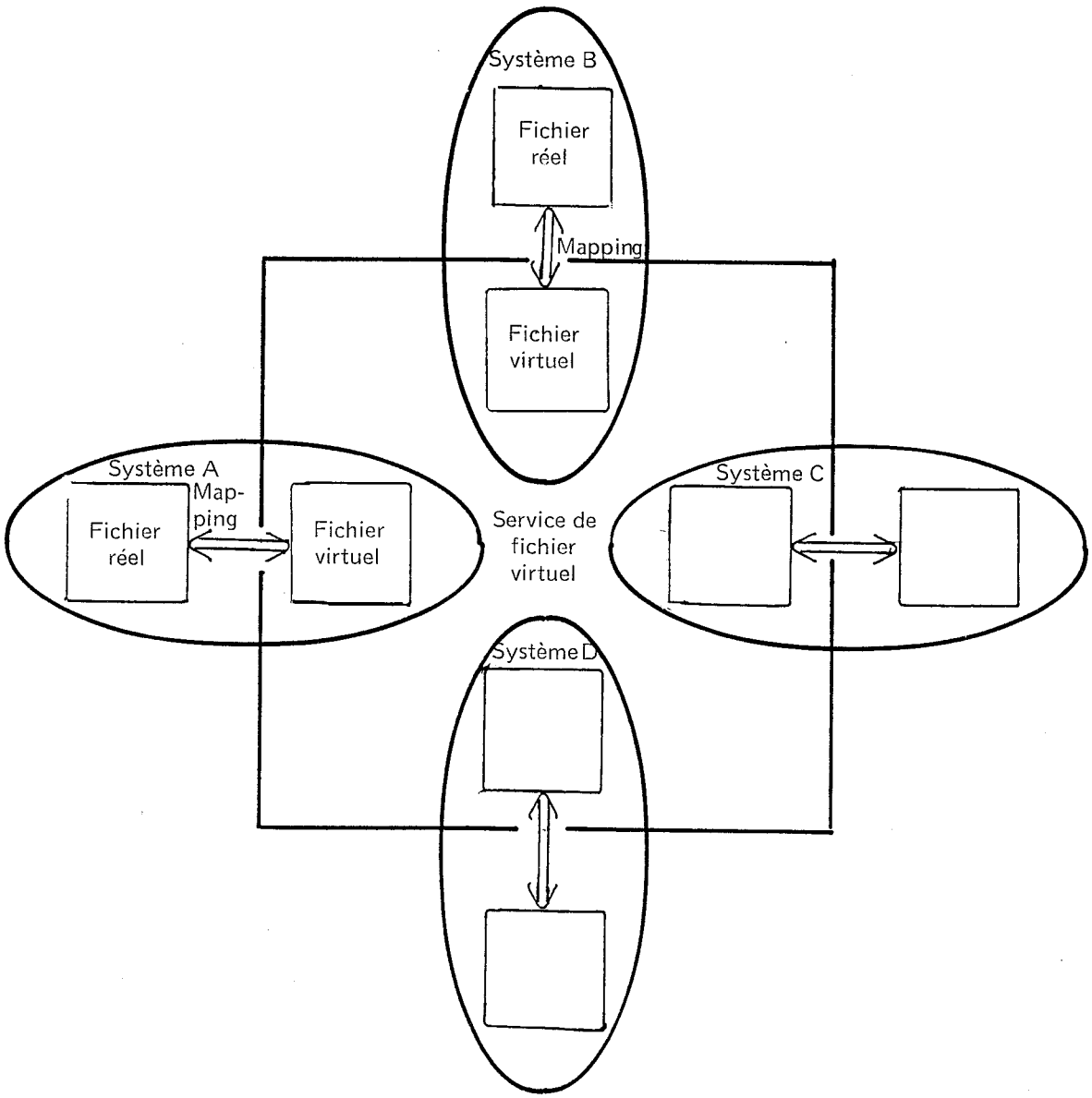
Fig 4 - Le protocole de transfert.

2.2.2 NIFTP et l'hétérogénéité - le Mapping

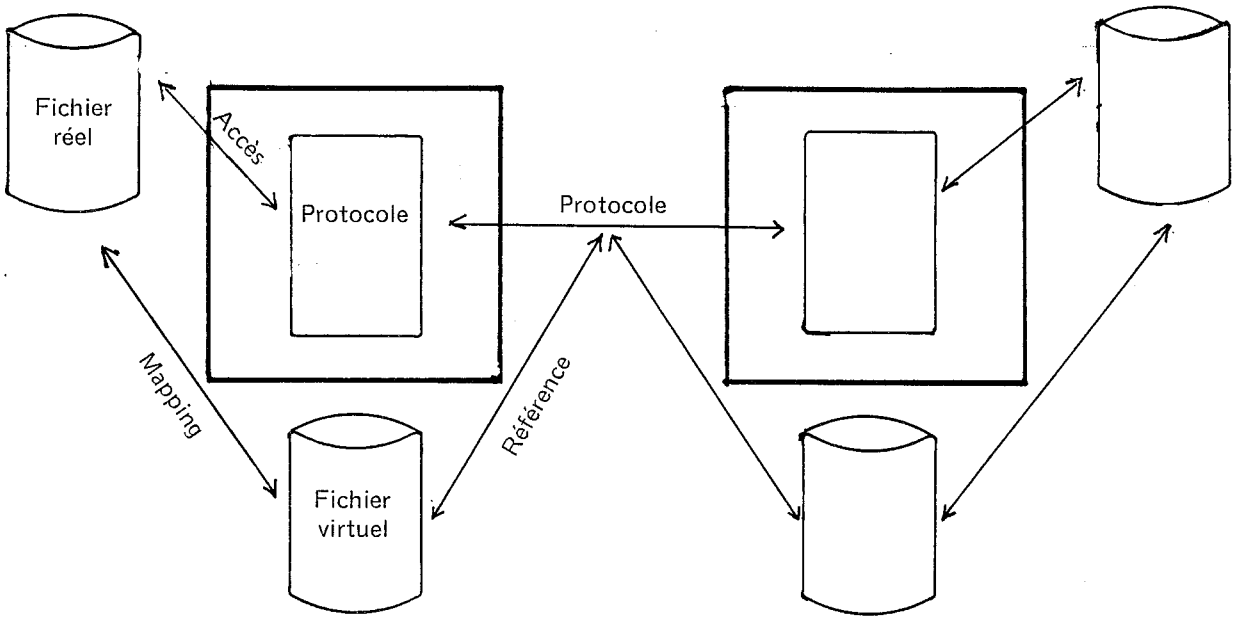
Afin de pouvoir transférer un fichier dans un contexte hétérogène, il est nécessaire de pouvoir définir ce fichier "réel" en termes d'une représentation standardisée unique baptisée "fichier virtuel" ou "virtual filestore" compréhensible par l'émetteur et le récepteur.

Cette transformation ou "mapping" du fichier réel vers le fichier virtuel ou inversement va être effectuée localement par les deux entités communicantes et va donc être fonction du système utilisé tandis que les échanges d'informations seront toujours effectués en termes d'attributs standards indépendants des systèmes utilisés.

Fig5Q04900



- Fig 5 Le Mapping.



- Fig 5 bis Le mapping .

2.2.3 Définition d'un transfert de fichier NIFTP

Un transfert de fichier NIFTP va se définir comme une transaction entre un processus initiateur "P" et un processus répondeur "Q", le fichier transmis faisant l'objet d'un "mapping" chez le processus P (transformation du fichier réel en fichier virtuel) et d'un "mapping" inverse chez Q, tous deux étant réalisés en termes d'attributs standards.

2.3 LES ATTRIBUTS NIFTP

Le protocole NIFTP contient une gamme très large d'attributs qui vont permettre d'effectuer le "mapping" des fichiers séquentiels transférés. Ces attributs sont divisés en trois classes.

2.3.1 Les attributs "Storage" ou "Q"

Ces attributs concernent le stockage du fichier sur le site distant et ont une durée de vie égale à celle du fichier et ont pour fonctions :

- d'identifier le fichier distant :
  - "nom du fichier".
  - "numéro de compte".
  - "mot de passe".
  - "nom de l'utilisateur".
- de demander des ressources chez le distant :
  - "taille du fichier".
  - "taille maximum d'un enregistrement".
- de choisir le type de périphérique utilisé pour l'édition du fichier sur le site distant pour les fichiers spoolés.
  - "type de périphérique" (numéro d'imprimante...).
- de préciser le codage du fichier distant :
  - "code de stockage" pour les fichiers "texte".

2.3.2 Les attributs de "transfert" ou T

Concernent le transfert et ont une durée de vie égale à celle du transfert. On trouve :

a) Les attributs caractérisant le transfert des données:

- "Code du transfert" (EBCDIC, IA5 ou code privé) pour les fichiers "textes", le type du fichier étant défini par l'attribut "type des données"
- "taille du mot binaire" : pour les fichiers binaires cette taille peut être différente de huit, l'attribut "format binaire" indiquant la façon dont les mots binaires sont "mappés" en octets pour le transfert.
- Le "mode d'accès" : permet d'indiquer le sens du transfert et les actions à effectuer (lecture, écriture, lecture et destruction, mise à jour, création et écriture, lecture et destruction au fur et à mesure de la lecture). D'autre part, des modes d'accès spéciaux sont réservés aux traitement de travaux à distance (give job imput : prendre le job en entrée chez le distant et le mettre en attente d'exécution sur le site local ,take job input:opération inverse avec en plus récupération de la liste d'exécution,take job output:imprimer une liste chez le distant,give job output:récupérer une liste chez le distant et l'imprimer).
- Les "facilités" : précisent si les données transmises doivent être compressées, si les requêtes de suspension, de reprise à chaud ou à froid sont permises dans le transfert.

B) attributs précisant des actions de formattage sur les données:

- "text formatting" : définit les actions de formattage à effectuer sur les données reçues. Ces actions peuvent être déclenchées par :

  - des caractères de contrôle ANSI en début d'enregistrement.

  - des caractères inclus dans le texte.

  - la fin d'un enregistrement peut déclencher un saut de ligne ou de page.

- Ces différentes actions peuvent être combinées et afin de définir de façon précise les actions des caractères de formattage la notion de "notional character matrix" est introduite. C'est une matrice dont la taille est déterminée par les attributs "largeur de page" et "longueur de page" décrivant la position des caractères dans une page logique. L'action de formattage d'un caractère est déterminée par son effet sur la position courante dans la matrice.

- 'Tabulations Horizontales' : la gestion des tabulations est simplifiée par l'utilisation de la "national character matrix". L'attribut tabulations horizontales est constituée par une chaîne de caractères, chaque caractère non blanc indiquant la position d'une fin de tabulation (voir annexe 7-4).

- "Délimiter préservation" : cet attribut précise si certaines caractéristiques du fichier virtuel chez "Q" doivent être préservées lors du mapping vers le fichier réel. (Tabulations horizontales, frontières d'enregistrements à préserver dans le fichier réel).

C) attributs relatifs à la taille du fichier transféré:

- "Taille maximum des enregistrements transférés"

- "Limite de transmission" : volume maximum de données échangées pendant la phase "donnée".

D) Les attributs liés aux possibilités de reprise :

- la "fenêtre maximum d'acquittement des marques" : donne l'intervalle maximum d'acquittement des marques lorsque cette option est utilisée .

- La "marque de reprise initiale" : cette marque définit l'endroit dans le fichier où le transfert doit reprendre lors d'une reprise a froid .

- L'"identificateur de transfert" :identifie le transfert en vue d'une reprise ultérieure.

E) attributs relatifs a la gestion du protocole :

- le "délai minimum" : définit l'intervalle de temps utilisé dans l'automate pour se protéger d'un "dead-lock". Le transfert est abandonné après expiration de ce délai.

2.3.3 Attributs "Information" ou "I"

Ces attributs sont utilisés pour véhiculer des informations :

- pour l'utilisateur :

  attribut "message d'information".

- sur l'état du transfert. :

  attribut "état du transfert".

- sur la version du protocole utilisé :

  attribut "identification du protocole".

- pour l'opérateur :

  attribut "action message".

Lorsqu'une implémentation a besoin de facilités qui ne sont pas définies dans le protocole, elle peut le spécifier par un attribut appelé "spécial options" de format et de type à définir par les implémentateurs.

2.4 DESCRIPTION D'UNE PROCEDURE DE TRANSFERT

Un transfert de fichier se divise en plusieurs phases associées chacune à un jeu de commande spécifique.

2.4.1 Phase initialisation

Cette phase est définie en terme de processus "P" et "Q" et va avoir pour but de négocier des attributs proposés par ces deux processus afin de définir l'environnement du transfert ainsi que le fichier à transférer.

Pour ce faire les commandes suivantes sont utilisées :

- SFT : Start File Transfer (P)
- RPOS/RNEG : Réponse positive/négative à la commande SFT (Q)
- GO : Entrer dans la phase data (P).

1 . Le processus "P" initiateur émet la commande SFT en proposant un jeu d'attributs vers le processus "Q" munis chacun d'un opérateur de négociation (=, <=, >= ou "ANY value") permettant de lui imposer des contraintes sur son choix (lorsque ce choix est possible c'est-à-dire que l'opérateur n'est pas '=').

2 . Le processus "Q" peut accepter le transfert si ses contraintes locales sont compatibles avec les contraintes distantes reçues dans la commande SFT et s'il peut fournir une valeur pour tous les attributs reçus dans la commande associés à un opérateur différent de '='. Dans ce cas il émet la commande RPOS qui peut comporter des attributs avec des valeurs différentes que celles fournies dans la commande SFT, accompagnés d'autres attributs s'il existe des contraintes locales à "Q" qui n'existent pas chez "P". Tous ces attributs sont émis avec l'opérateur '=' car la commande RPOS doit imposer une valeur pour chaque attribut émis. Si le processus "Q" refuse des attributs fournis par P il émet la commande RNEG accompagnée des attributs ayant causé le refus.

3 . Sur réception de RPOS, "P" vérifie que les attributs fournis sont toujours compatibles avec ses contraintes locales, si c'est le cas il émet la commande GO et passe dans la phase de transfert de données. Sinon il passe dans la phase terminaison en émettant la commande STOP accompagnée de la cause du refus.

Remarques:

- Pour les attributs qui ne sont pas fournis explicitement par l'initiateur du transfert, il est pris une valeur par défaut propre à l'implémentation.
- Lorsque "Q" reçoit un attribut inconnu pour lui, il ne rejette pas le transfert mais le précise dans sa commande RPOS. "P" sera libre de continuer ou non ce transfert. Ce mécanisme permet de faire communiquer des implémentations possédant des niveaux de sophistication différents et de pouvoir faire évoluer une implémentation indépendamment des autres.

2.4.2 Phase transfert de donnée

Cette phase est définie en termes d'"Emetteur" et de "Récepteur" et va concerner le transfert du fichier proprement dit.

2.4.2.1 Les commandes et les données

Les commandes utilisées dans cette phase sont les suivantes :

- pour l'émetteur :
  - SS : start of data: Début de l'émission des données.
  - CS : code select : sélectionne le code de transfert.
  - MS : mark point :positionner une marque.
  - ES : End of data:fin de l'émission des données.
- pour le récepteur :
  - MR : Acknowledge mark:Acquittement de marques.
  - RR : Restart Request:demande de reprise.
  - QR : Quit:Acquittement de commande de contrôle.
  - ER : Acknowledge of data:Acquittement de données.

A ces commandes s'ajoutent les données constituées par le contenu du fichier à émettre, structurées en "records" ou enregistrements de longueur quelconque découpés en "subrecords" de 63 octets maximum plus un en-tête de 1 octet pour les besoins de la transmission (voir Annexe 7-3).

La phase "données" débute lorsque après avoir déterminé dans la phase initialisation quel est le processus "émetteur" des processus "P" ou "Q", l'Emetteur émet la commande SS et se termine lorsque le récepteur émet la commande ER confirmant la réception de "l'end of data".

Entre ces deux commandes, les données du fichier sont transférées de l'Emetteur vers le Récepteur, et des commandes de contrôle sont échangées a fin d'assurer :

- le contrôle des erreurs.
- un contrôle de flux.
- un changement de code de transfert.
- la reprise ultérieure du transfert.

2.4.2.2 Le contrôle des erreurs

Un mécanisme de marques est utilisé pour controler les erreurs de transfert.

L'Emetteur pose des marques à des endroits du fichier choisis par lui, et le récepteur les acquitte une fois que les données reçues jusqu'à cette marque sont sauvegardées de façon sûre.

L'Emetteur cesse d'émettre des marques lorsque sa fenêtre d'émission des marques est pleine et attend alors des acquittements du récepteur.

Le récepteur peut acquitter chaque marque reçue ou regrouper ces acquittements jusqu'à concurrence de la "fenêtre maximum d'acquittement des marques" attribut négocié dans la phase initialisation).

2.4.2.3 Le contrôle de flux

Si la facilité "Suspension" est supportée, le récepteur peut demander la suspension d'un transfert pour une durée limitée à quelques minutes (de l'ordre de l'attribut "délai minimum" négocié dans la phase initialisation).

2.4.2.4 Reprise sur erreurs

Il existe deux types de reprises :

Reprise à chaud :

Lors d'une erreur en réception, le récepteur peut demander à l'émetteur de reprendre son émission à partir d'une marque donnée, non acquittée.

Reprise à froid - Les dossiers de transfert

Si un transfert possédant l'option reprise au niveau de l'attribut "facilités" est stoppé (sur rupture du chemin virtuel transport, sur erreur récupérable ou sur arrêt utilisateur) il pourra être repris par la suite sur demande de l'utilisateur.

Ceci nécessite de la part des deux parties de conserver toutes les informations nécessaires à la reprise de ce transfert (marque initiale de reprise, valeur des attributs négociés, marques acquittées etc...) dans un dossier de transfert.

Ce dossier est créé par "P" sur émission de la commande SFT et détruit sur refus de transfert (RNEG) ou sur réception de STOPACK avec une cause de terminaison indiquant que le transfert est terminé sans possibilité de reprise.

Chez le processus "Q", le dossier est créé sur émission de RPOS et détruit sur réception de STOP avec une cause de fin sans possibilité de reprise.

Ces dossiers sont mis à jour à chaque fois qu'il y a un changement dans l'état du transfert.

La facilité de reprise va permettre d'effectuer un même transfert en plusieurs vacations.

2.4.3 Phase terminaison

Cette phase est comme la phase initialisation décrite en terme de processus "P" et "Q" et va permettre à un transfert de se terminer.

Les commandes utilisées sont les suivantes :

- STOP : arrêt du transfert par P.

- STOPACK : confirmation d'arrêt par Q.

L'entrée dans cette phase peut être faite via la phase initialisation (sur refus de transfert par exemple) ou via la phase "donnée" (sur fin de transfert correct ou sur erreur).

Pour terminer la transaction, le processus "P" émet la commande STOP et le processus "Q" répond par STOPACK.

L'état final du transfert est véhiculé dans ces différentes commandes et précise la cause de l'arrêt ainsi que les possibilités de reprise ultérieures.

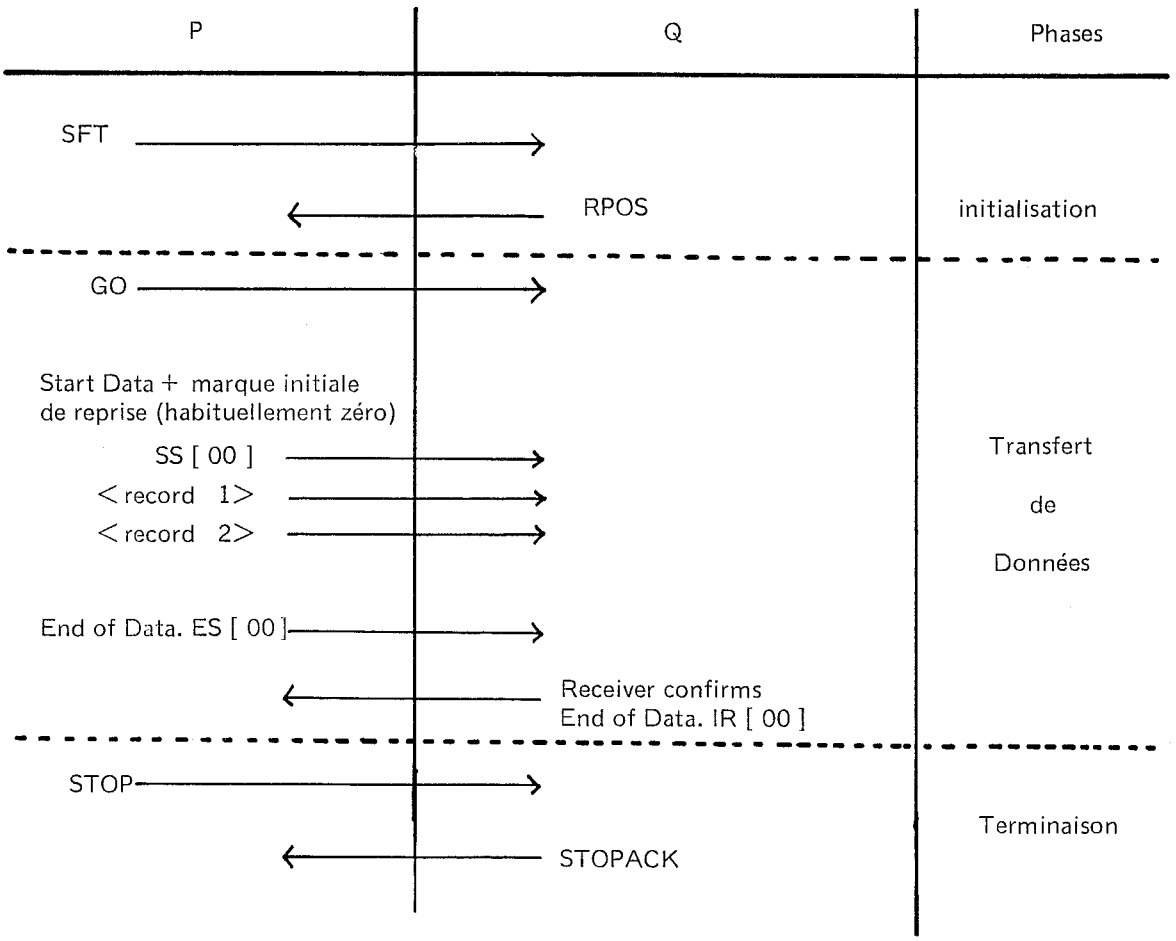


Fig 6 Schéma d'une transaction complète NIFTP.

2.5 ETUDE COMPAREE DE NIFTP VIS A VIS DES PROTOCOLES ECMA ET ISO

En l'absence d'un standard international pour les transferts de fichiers en milieu hétérogène, une multitude de protocoles ont été développés durant les années 70. Citons entre autres ARPANET (USA en 73),NIFTP (GB en 77),AUTODIN (USA en 79),VFP (ECMA en 82).

Le retard dans la définition d'un tel standard vient des difficultés rencontrées pour les instances internationales telle que l'ISO pour normaliser les dernières couches du modèle OSI : session et présentation.

Néanmoins actuellement les dernières difficultés sont ou sont en passe d'être vaincues et l'ECMA a défini un standard de transfert de fichier baptisé ECMA-85 (septembre 82) s'appuyant sur le modèle OSI, l'ISO s'apprétant à faire de même (ISO FTS).

Nous choisirons ces deux derniers protocoles pour effectuer une étude comparée vis a vis du protocole NIFTP.

L'étude détaillée de ces protocoles dépasserait le cadre de ce mémoire et nous nous bornerons à évoquer les principaux points qui les différencient en mettant l'accent sur les avantages et les inconvénients des solutions retenues, sachant que tous les trois contiennent les mêmes notions de fichier virtuel, de mapping, de phase de transfert et de négociation des attributs.

A) L'architecture :

Les architectures retenues par le protocole ECMA-85 et ISO-FTS sont différentes de celle de NIFTP en ce sens qu'elle suivent scrupuleusement le modèle OSI et respectent le découpage en couches (application, présentation, session)

- avantage : le protocole NIFTP en théorie doit apporter un gain de temps au niveau des performances, car il n'existe pas de découpage entre la couche présentation et session d'où moins d'"overhead" système.

- inconvénient : les modifications dans l'implémentation NIFTP seront moins aisées que dans les autres, le degré de modularité de l'architecture étant moindre.

B) Structure du protocole :

Le découpage d'un transfert en phases varie suivant les protocoles utilisés. Nous ne traiterons ici que des phases concernant le lancement du transfert.

Le protocole ECMA fournit le découpage le plus fin. On distingue cinq phases :

A - établissement de la session.
B - sélection du protocole et des options.
C - sélection du fichier.
D - ouverture du fichier.
E - transfert des données.

Le protocole ISO FTS ne contient que trois phases :

F - initialisation (équivalente aux phases A et B.)
G - sélection du fichier et ouverture (équivalente aux phases C et D).
H - transfert des données.

Enfin le protocole NIFTP ne comporte que deux phases:

I - initialisation (équivalente aux phases A, B, C, D).
J - transfert des données.

Au vu de ce découpage, il apparait que le protocole NIFTP, dans le cas où l'on a plusieurs transferts à effectuer vers le même site , oblige pour chaque transfert à reprendre les opérations depuis la phase d'initialisation, alors qu'avec les protocoles ISO et ECMA, on pourra ne les reprendre qu'a partir de la phase de sélection du fichier ce qui permettra d'envoyer des séries de fichiers de même type sur la même session.

C) Choix des options :

Les protocoles ECMA et NIFTP contiennent tous les deux l'attribut "Protocol ident" qui permet de préciser la version du protocole utilisée. Le protocole ECMA permet en plus de choisir une classe dans ce protocole correspondant à un jeux d'options déterminées.

D) Organisation des fichiers transférés :

Les trois protocoles supportent l'organisation séquentielle. Les protocoles ECMA et ISO supportent en plus l'organisation relative, random et séquentiel-indexé et permettent de spécifier l'emplacement et la longueur de la clé dans les enregistrements transférés dans ce dernier cas.

E) Phase transfert de donnée :

Les protocoles ECMA et ISO utilisant les concepts de couche session et présentation seront plus "coûteux" en "header" dans les données transmises que le protocole NIFTP. En effet les "subrecords" NIFTP de 64 octets maximum ne nécessitent qu'un header contrairement à ceux des protocoles ECMA et ISO qui en exigent trois (un header pour la couche présentation, un pour la couche session, un pour le protocole).

F) Conclusion :

Cette étude succinte des trois principaux protocoles de transfert de fichiers en milieu hétérogène nous montre que sur le plan purement des performance le protocole NIFTP devrait être légèrement supérieur aux autres protocoles, tandis que sur celui de la facilité d'implémentation et des évolutions, ce sont incontestablement ces derniers qui prennent l'avantage, les trois protocoles étant pratiquement équivalents sur le plan des services offerts à l'utilisateur.

3 PRESENTATION DU CONTRAT D'ACHAT DE L'AFFAIRE NIFTP

L'implémentation du protocole NIFTP sur matériel MITRA a fait l'objet d'un contrat d'achat entre la société BULL-SEMS et la C.C.E, dont l'annexe technique donne une description fonctionnelle de la fourniture demandée par le client,définissant les contraintes a respecter lors de l'implémentation.

Ces contraintes sont de quatre types:

1 - Contraintes protocoles :

L'architecture retenue devra s'appuyer sur le protocole X25/3 pour la partie réseau et ECMA/72 Classe 0 pour la partie transport ainsi que sur le protocole NIFTP pour la partie transfert.

2 - Contraintes de réalisation :

La découpe du projet en deux étapes ou LOTS ainsi que les objectifs fixés pour chacune d'elles ont déja été décrits dans l'introduction. Les chapitres suivants traiteront de leurs réalisations.

3 - Contraintes concernant les fonctionnalités offertes aux utilisateurs du produit :

Le produit NIFTP devra offrir trois types d'interfaces utilisateur:

- Une interface opérateur de site.
- Une interface opérateur.
- Une interface utilisateurs intéractifs.

Les commandes suivantes devront être offertes aux utilisateurs par le processeur NIFTP:

- Lancement d'un transfert.
- Abandon d'un transfert avec possibilité de reprise.
- Reprise d'un transfert interrompu.
- Visualisation de l'état des transferts.
- Purge des transferts en attente.

FIGNIFTP

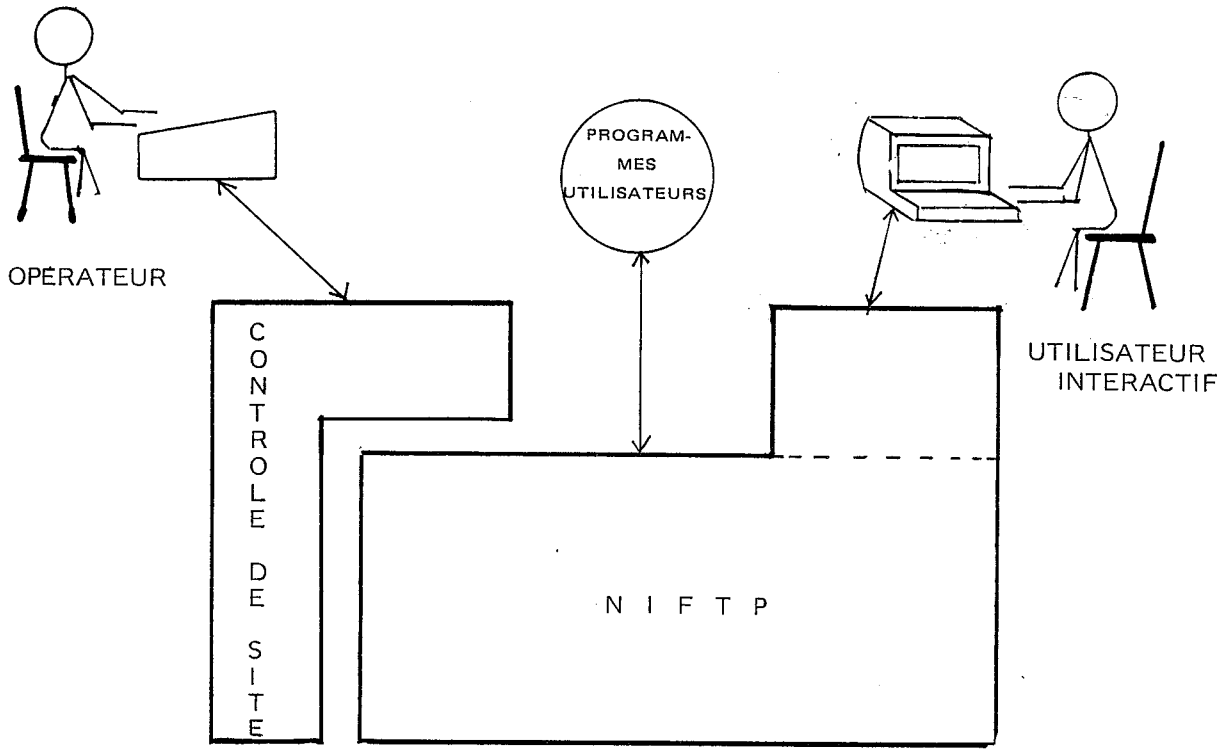


Fig 6-1 Les interfaces utilisateurs NIFTP.

4 - Contraintes de performances :

La seule contrainte de performances imposée est le nombre maximum de transferts simultanés à supporter par l'implémentation fixé à dix.

L'ensemble des contraintes exposées alliées à celles propres à la structure d'acceuil matérielle et logicielle vont être à la base de la définition de l'architecture du produit que nous allons décrire dans les chapitres suivants.

4 REALISATION DU PROJET FTS-NIFTP LOT1

4.1 L'ENVIRONNEMENT MITRA.

Dans les paragraphes précédents nous avons exposé le protocole NIFTP dans son ensemble.

Nous allons maintenant nous attacher à décrire son implémentation sur matériel MITRA 625 dans le cadre du projet LOT 1.

Auparavant il est utile de définir l'environnement matériel et logiciel qui a constitué la structure d'accueil.

L 'environnement matériel va être constitué par le mini-ordinateur MITRA 625 dont on trouvera en annexe 7-8 la description.

L'environnement logiciel va être celui du système MMT2 associé à son système de gestion de fichier FMS2 et du système de communication SEMS SCS2 que nous allons présenter.

4.1.1 Le système MMT2

Le système d'exploitation MMT2 est un système "multifonctions" et "multitâches" et permet de regrouper en fonctions indépendantes des groupes de tâches fonctionnant dans un environnement donné dans un but précis (application temps réel, batch, télespool). L'intérêt de cette structure est d'offrir une grande sécurité d'utilisation chaque groupe ayant les ressources nécessaires à son fonctionnement, les tâches d'un groupe ou fonction ne pouvant avoir d'intéractions incontrôlées avec les tâches d'un autre groupe, le partage des ressources physiques étant réglé par le système.

Ce système met à la disposition des fonctions, un certain nombre de services qui ont trait à la gestion dynamique de la mémoire, des sémaphores, des évènements, des entrées-sorties, des groupes et des tâches, des communications inter ou intra-groupe.

Nous allons examiner plus particulièrement les services FMS2 et SCS-2 sur lesquels va être bâti le service de transfert de fichier ainsi que EXOP2 qui est le système intéractif de MMT2.

4.1.2 Le système interactif EXOP

MMT2 fournit à l'utilisateur un système intéractif baptisé EXOP qui est un groupe particulier qui lui permet d'accéder à tous les services du moniteur MMT2, via une console de visualisation à l'aide d'un jeu de commandes interactives du type.

- création d'un groupe utilisateur.

- création, lancement et mise au point de tâches dans ce groupe.

- lancement et utilisation des différents utilitaires MMT2 (processeurs de gestion de fichier, debugger système,etc...).

Plusieurs utilisateurs peuvent créer simultanément leur groupe intéractif, ils bénéficient dans ce cas des propriétés des groupes (protection inter-groupe, partage des ressources).

4.1.3 Le système FMS2

Le système de gestion de fichier FMS2 fait partie du groupe système et va assurer le contrôle total des opérations effectuées sur fichier (protection inter-groupe, vérifications des droits d'accès, demandes de montages de volumes).

Les organisations de fichiers supportées par FMS2 sont les fichiers séquentiels, partitionnés, direct et séquentiels indexés. Les modes d'accès possibles sont variés (séquentiel, direct, partitionné, aléatoire et mixte). FMS2 permet de gérer des fichiers de type binaire ou alphanumérique.

4.1.4 Le système de communication SCS2

Le système SCS-2 va constituer la structure d'accueil de l'implémentation NIFTP sur matériel MITRA. C'est pourquoi il est utile de détailler l'architecture de ce logiciel ainsi que ses mécanismes internes avant de passer a la description du produit FTS-NIFTP.

4.1.4.1 L'Architecture SCS2 - SCS2 et l'ISO

L'objectif de SCS est d'offrir aux utilisateurs de machine SEMS la possibilité de réaliser des systèmes informatiques distribués, faisant coopérer les différents ordinateurs du constructeur (MITRA et SOLAR) en fournissant un service de communication indépendant de la localisation des entités communicantes et des moyens d'interconnexion utilisés (lignes HDLC, réseau local, CV Transpac).

L'architecture retenue pour SCS est conforme au modèle pour l'interconnexion des systèmes ouverts que nous avons détaillé dans le paragraphe 2-1-3.

Cette conformité avec le modèle OSI se retrouve en particulier dans la structuration en couches entraînant des mécanismes d'interface similaires et dans l'utilisation de protocoles normalisés dans les différentes couches.

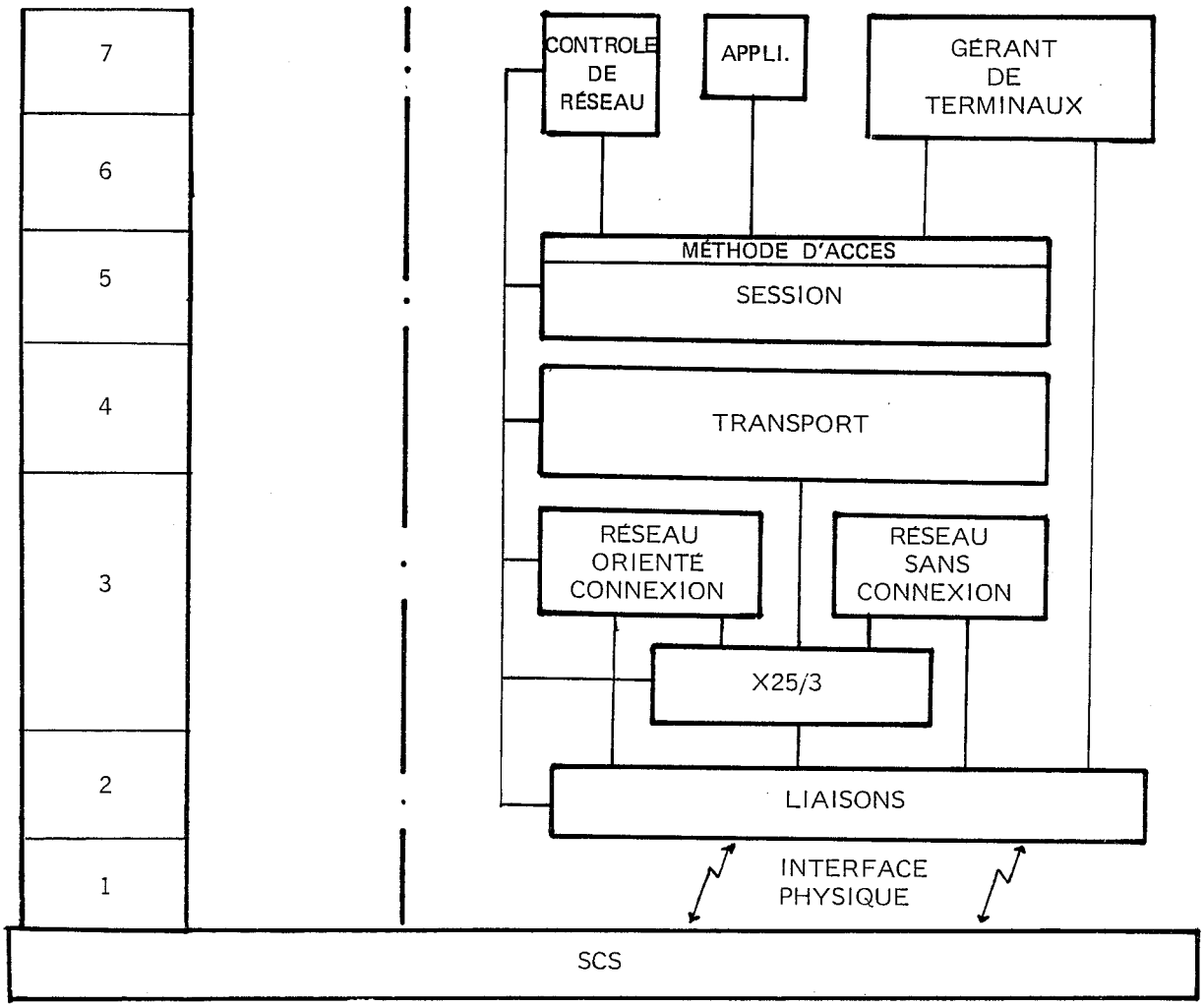


Fig 7 - SCS-2 et l'ISO

Les propriétés du système de communication SCS sont celles d'une telle architecture. Ce sont :

- la modularité (découpages des fonctions en modules interfaçant à l'aide d'un mécanisme d'interface unifié et symétrisé),

- l'ouverture vers l'extérieur:support des réseaux publics de type X25 (TRANSPAC , EURONET ou autres),

- l'Evolutivité : possibilité de modifier ou supprimer des modules d'une couche sans modifier ceux des autres couches,

- l'indépendance des entités communicantes et des moyens de communication (grâce à une méthode d'accès banalisée au service de communication) vis a vis des entités de la couche application.

4.1.4.2 Structure interne de SCS

Le système de communication SEMS SCS2 constitue un service optionnel du moniteur MMT2. Au sens MMT2, SCS-2 est un groupe et en temps que tel, il possède les propriétés d'un groupe, mais il a la particularité d'être un groupe multiplexé accessible à partir de tout autre groupe de la même façon qu'un service standard (FMS2 par exemple) par le mécanisme des appels superviseurs (CSV).

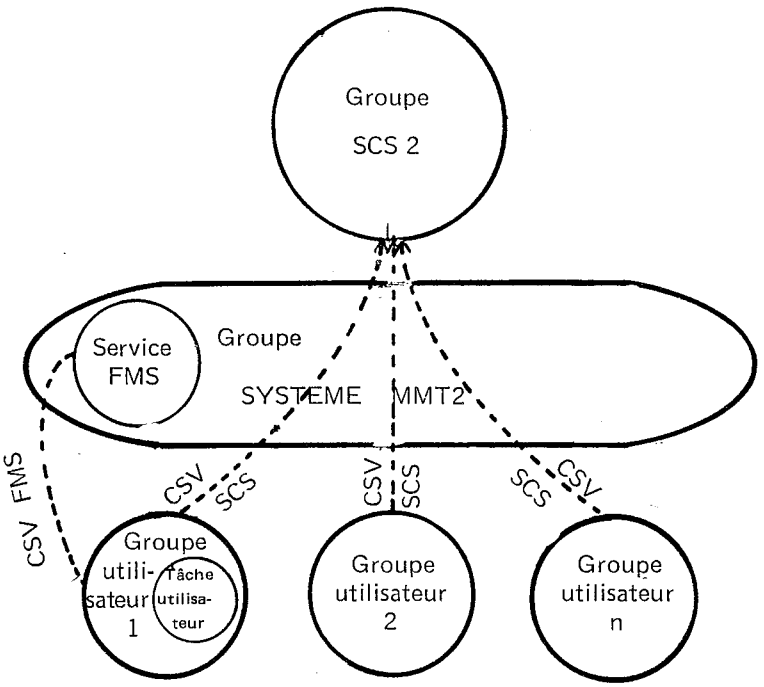


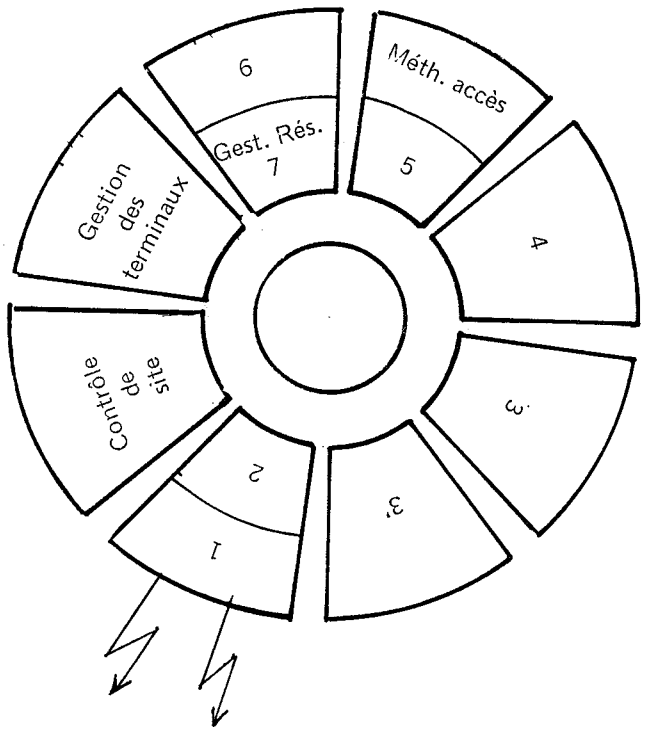
Fig 8 les groupes MMT2,SCS-2,et utilisateurs .

SCS-2 possède un noyau central contenant des fonctions communes au groupe et des modules qui communiquent entre eux à l'aide d'un interface unifié symétrisé, chaque couche au sens ISO étant constituée par un ou plusieurs modules.

SCS2-2 en plus du noyau et des modules constituant les différentes couches comprend :

- un module contrôleur de site permettant la maîtrise des paramètres d'un site, la gestion des statistiques et le contrôle des lignes (ouvertures, fermetures, visualisation des états).
- un module de contrôle réseau permettant d'étendre ces fonctions au réseau entier ainsi que d'effectuer les fonctions précédentes à distance.
- un module gestion des terminaux qui permet d'effectuer des échanges entre des processus application et des terminaux via le réseau téléphonique (terminaux vidéotex) ou via le réseau PAD Transpac.

FIG9Q04900



1 Couche physique
2 Couche liaison
3 Couche réseau (orienté connexion)
3' Couche réseau (sans connexion)
4 Couche transport
5 Couche session
6 Couche présentation

Fig 9 - Structure interne de SCS-2.

4.1.4.3 Gestion mémoire dans SCS-2

La gestion de l'espace mémoire utilisé pour les échanges de données entre les modules de SCS va être assurée par un service du noyau : le gestionnaire mémoire.

Cet espace va être initialement découpé en granules de taille fixe et chaque module pourra s'allouer une chaîne mémoire de longueur donnée constituée par un nombre de granules équivalent à la longueur demandée.

Le gestionnaire mémoire va fournir en retour un descripteur de la chaîne qui sera utilisé pour les opérations ultérieures sur cette chaîne, Ces opérations pouvant être la concaténation à une autre chaîne, la libération de cette chaîne, l'éclatement de cette chaîne en plusieurs chaînes, le transfert de données dans une chaîne, la récupération des données d'une chaîne.

4.1.4.4 La communication intermodule dans SCS-2

La communication inter-modules va être assurée par un mécanisme de files d'attentes, chacun possédant sa file d'attente propre et un évènement associé.

Un module désirant envoyer une information vers un autre, va rajouter un élément dans la file d'attente de ce module et le réveiller en positionnant l'évènement correspondant. Lorsqu'un module a fini son traitement il se met en attente sur l'évènement lié à sa file d'attente.

Un élément de file d'attente est constitué d'un pavé mémoire de longueur fixe contenant toutes les informations nécessaires à son traitement (type, indication si des données sont associées , référence, identification du module source et destination, descripteur de chaîne de données si nécessaire).

Ces éléments ou ICB (interrupt control bloc) sont regroupés en 4 types qui sont :

- élément demandant un service à un module (requête).
- élément rendant compte du déroulement d'une requête (réponse).
- élément signalant l'arrivée d'un évènement venant de la couche inférieure (indication).
- élément attestant de la prise en compte du signal venant de la couche inférieure (confirmation).

FIG10Q04900

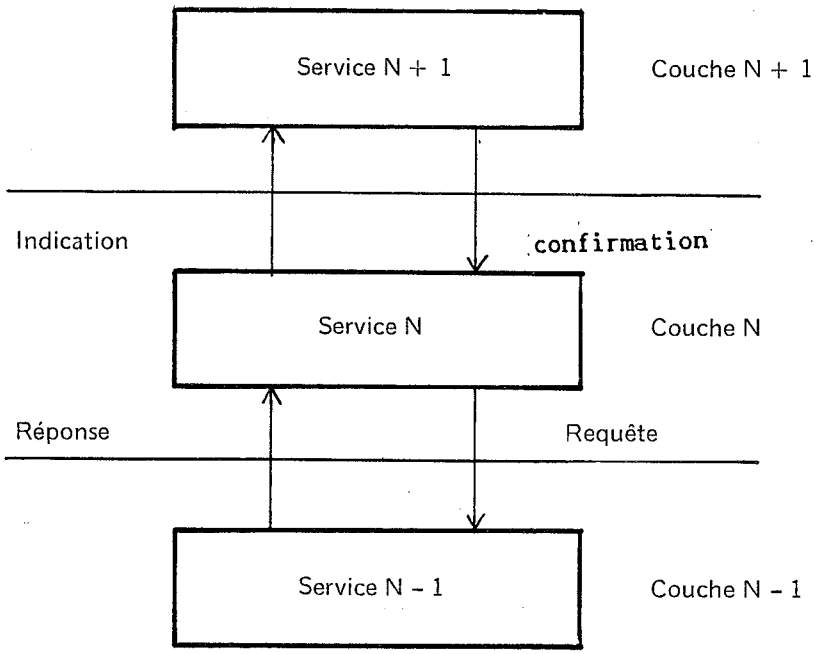


Fig 10 - Les intéractions entre les modules de SCS-2

Un service du noyau, le dispatcher, va centraliser les fonctions de communication et fournir à chaque module le moyen.

- de s'allouer un ICB libre,

- de se mettre en attente de l'arrivée d'un ICB venant d'un module,

- de poster un ICB vers un autre module,

- de libérer un ICB et de le remettre dans la file des libres.

4.2 INTEGRATION DU PROTOCOLE NIFTP DANS SCS-2

4.2.1 LES CONTRAINTES DE L'IMPLEMENTATION

L'implémentation du protocole sur MITRA devra respecter les contraintes suivantes:

1 - Les contraintes liées à l'utilisation de la structure d'accueil de SCS-2.

2 - Les contraintes liées au cahier des charges (voir ch. 3).

3 - Les contraintes de performances compte-tenu du cahier des charges.

L'architecture du produit FTS-NIFTP va résulter de ces différentes contraintes.

4.2.2 L'ARCHITECTURE GLOBALE

L'intégration du protocole NIFTP dans l'architecture SCS-2 va être rendue possible grâce aux propriétés déjà énumérées de ce système qui permettent de supprimer et de rajouter des modules dans une couche sans modifier les autres couches tout en conservant le noyau central et ses services.

Conformément à l'architecture déjà décrite dans le paragraphe 2-1-3, le protocole NIFTP va s'implanter dans les couches session, présentation et application.

Pour ce faire SCS-2 va être amputé de modules devenus inutiles tels que les modules gestion des terminaux, contrôle réseau, présentation, méthode d'accès et session.

De plus compte-tenu du cahier des charges précisant que le FTS-NIFTP doit fonctionner avec un transport ECMA/72 de classe 0, le module transport SEMS de SCS-2 va être remplacé par un module gérant un protocole de ce type (voir Annexe 7-7).

Enfin le système FTS-NIFTP étant forcément utilisé avec un réseau à commutation de paquet X25 (TRANSPAC, DCS ou EURONET), le module réseau devient inutile car ses principales fonctions (routage, connexion réseau, multiplexage des chemins virtuels sur les liaisons) deviennent redondantes avec les services assurés par ce type de réseau. Seul subsiste au niveau de cette couche le module d'adaptation des fragments transports en paquets réseaux, X25-3.

Compte-tenu des remarques précédentes l' architecture du produit FTS-NIFTP va être la suivante :

fig12Q04900

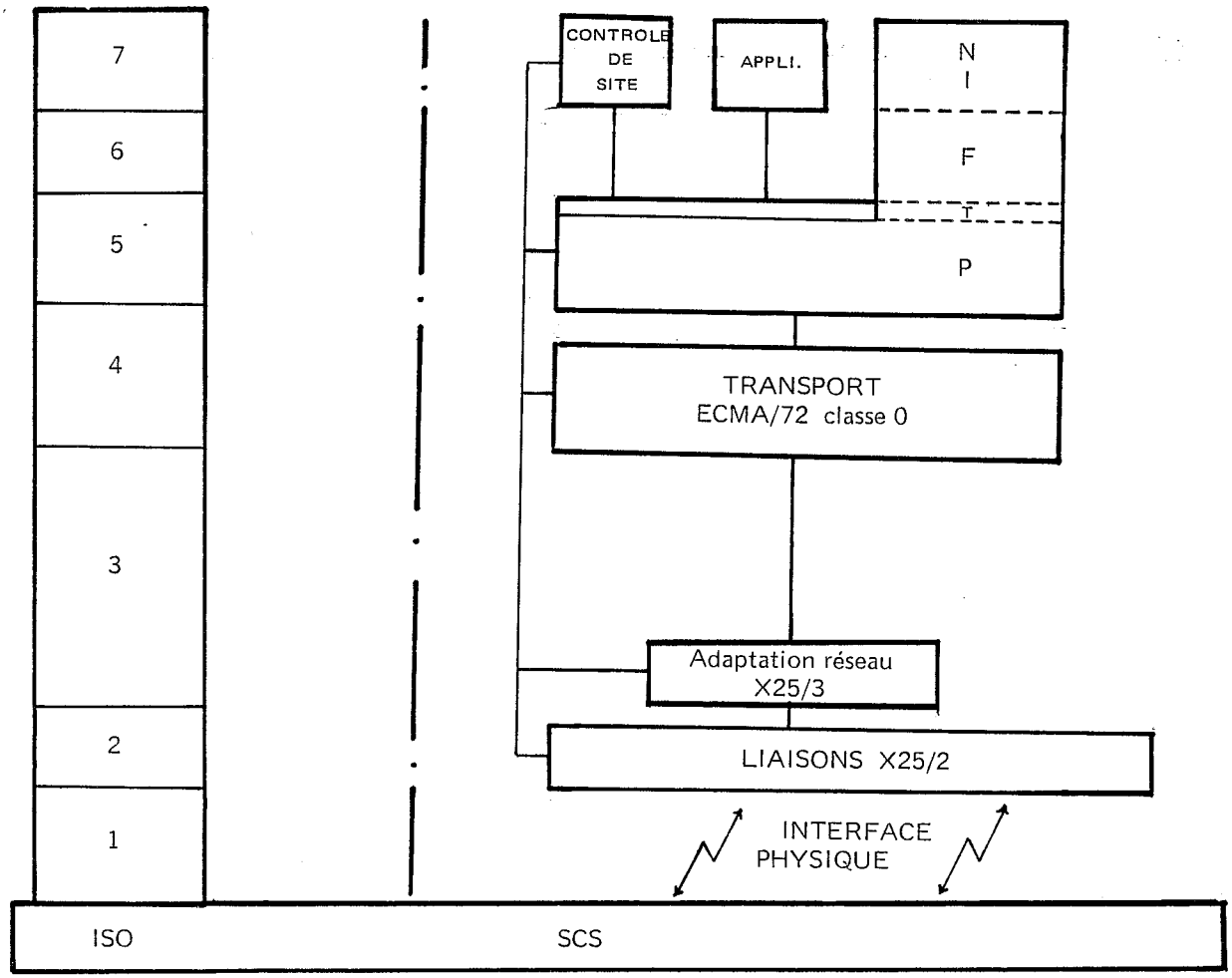


Fig 11 - Architecture NIFTP dans SCS-2.

Au sein de cette architecture, les modules NIFTP vont utiliser tous les services du noyau de SCS et en particulier vont communiquer avec les autres modules en utilisant le mécanisme standard d'ICB fourni par ce noyau en respectant les interfaces existantes.

La conséquence directe d'une telle architecture est que le système FTS-NIFTP devient un produit spécifique de transfert de fichier et de ce fait ne pourra plus assurer les mêmes services que le système SCS-2 standard l'unité d'échange entre les entités communicantes devenant le fichier et non plus le message.

4.2.3 L'ARCHITECTURE INTERNE DE LA COUCHE NIFTP.

Nous allons nous attacher maintenant à montrer comment l'architecture interne retenue pour le produit NIFTP (couches 5, 6, 7) a été élaborée en fonction des trois types de contraintes exposées ci-dessus.

Pour ce faire nous allons examiner tour à tour les solutions retenues pour répondre à chacune d'elles. Ainsi nous aboutirons à l'architecture finale du produit.

4.2.3.1 Contraintes liées à la structure d'accueil SCS-2

L'utilisation de la structure d'accueil de SCS2 va obliger les implémenteurs à respecter les interfaces existants(communication inter-modules, zone de données communes), mais en contre-partie va permettre:

- l'utilisation des services fournis par le noyau :
  - la gestion mémoire.
  - la communication inter-module (ICB).
  - la gestion des délais permettant d'assurer des délais pour le compte d'un utilisateur.
  - l'édition des messages opérateur .
  - l'acquisition des commandes opérateurs.
- l'utilisation des modules existants dans les couches 4, 3, 2:
  - le module d'initialisation du système.
  - le contrôleur de site.
  - le transport ECMA 72 (voir les caractéristiques de ce protocole en annexe 7-7).
  - le module d'adaptation aux réseaux de type X25/3.
  - le module de gestion des liaisons X25/2.
  - le contrôleur microprogrammé X25/2.

Ce faisant le nombre de services ou modules devant subir des modifications afin de répondre aux objectifs du produit, va être réduit au minimum.
Ce sont:

- le module d'édition des messages opérateurs.
  - rajout de messages spécifiques à NIFTP.
- Le module d'initialisation.
  - Prise en compte des nouveaux modules NIFTP.
- Le module d'acquisition des commandes opérateur :
  - rajout de commandes spécifiques à NIFTP.

4.2.3.2 Contraintes liées au cahier des charges

L'examen des différentes contraintes imposées par le cahier des charges (voir ch. 3) amène à définir quatre fonctions distinctes que nous regrouperons en autant de modules :

- un module méthode d'accès au service NIFTP en remplacement de la méthode d'accès SCS-2.

- un module d'administration des transferts.

- un ou plusieurs modules de gestion des transferts conforme au protocole NIFTP et assurant de plus les fonctions session vis a vis de la couche TRANSPORT ECMA 72.

- un module utilitaire permettant à des utilisateurs du système intéractif EXOP2 ou à un job du BATCH, d'accéder simplement au service de transfert de fichier via la méthode d'accès.

4.2.3.3 Contraintes de performances et d'encombrement mémoire.

La gestion simultanée de dix transferts peur être réalisée soit:

- Par une tâche unique gérant chaque transfert de facon linéaire : Solution qui entraine une mauvaise utilisation de l'unité centrale ,le temps utilisé pour effectuer les entrées sorties n'étant pas récupéré pour les autres transferts.

- Par une tâche associée a chaque transfert : Ce qui compte-tenu de l'encombrement mémoire demandé par cette solution apparait prohibitif.

- Par une tâche unique gérant les transferts en parallèle : Dans ce cas afin de ne pas favoriser un transfert par rapport à un autre, il est nécessaire de ne pas effectuer d'opérations bloquantes "lonques" dans cette tâche (création, ouverture, fermeture d'un fichier, lecture ou écriture d'un enregistrement, montage logique d'une unité).

C'est cette dernière solution qui a été retenue :

La fonction "gestion du protocole NIFTP" va être assurée par la tâche PROTOC qui va se décharger des opérations bloquantes liées au traitement des fichiers sur trois autres tâches "esclaves".

A. La tache initialisation GESINI qui va suivant le mode d'accès utilisé pour le transfert assurer une ou plusieurs des fonctions suivantes avant le début du transfert.

   - attente de montage d'un volume.
   - vérification de l'existence d'un fichier.
   - création d'un fichier.
   - destruction d'un fichier.

B. La tâche lecture GESLEC qui va assurer :

   - l'ouverture du fichier en lecture.
   - la lecture des enregistrements de fichier.
   - la fermeture du fichier.
   - des fonctions auxiliaires de traitement des articles lus que nous détaillerons par la suite.

C. La tâche écriture GESECR assurant:

   - l'ouverture du fichier en écriture.
   - l'écriture des enregistrements dans le fichier.
   - la fermeture du fichier.
   - comme pour GESLEC des fonctions supplémentaires de traitement des articles avant écriture.

Ainsi le simple examen des différentes contraintes auxquelles est soumis le produit nous permet de dégager l'architecture interne du produit FTS-NIFTP.

4.2.3.4 L'architecture interne

Le schéma suivant décrit l'architecture interne du produit FTS-NIFTP :

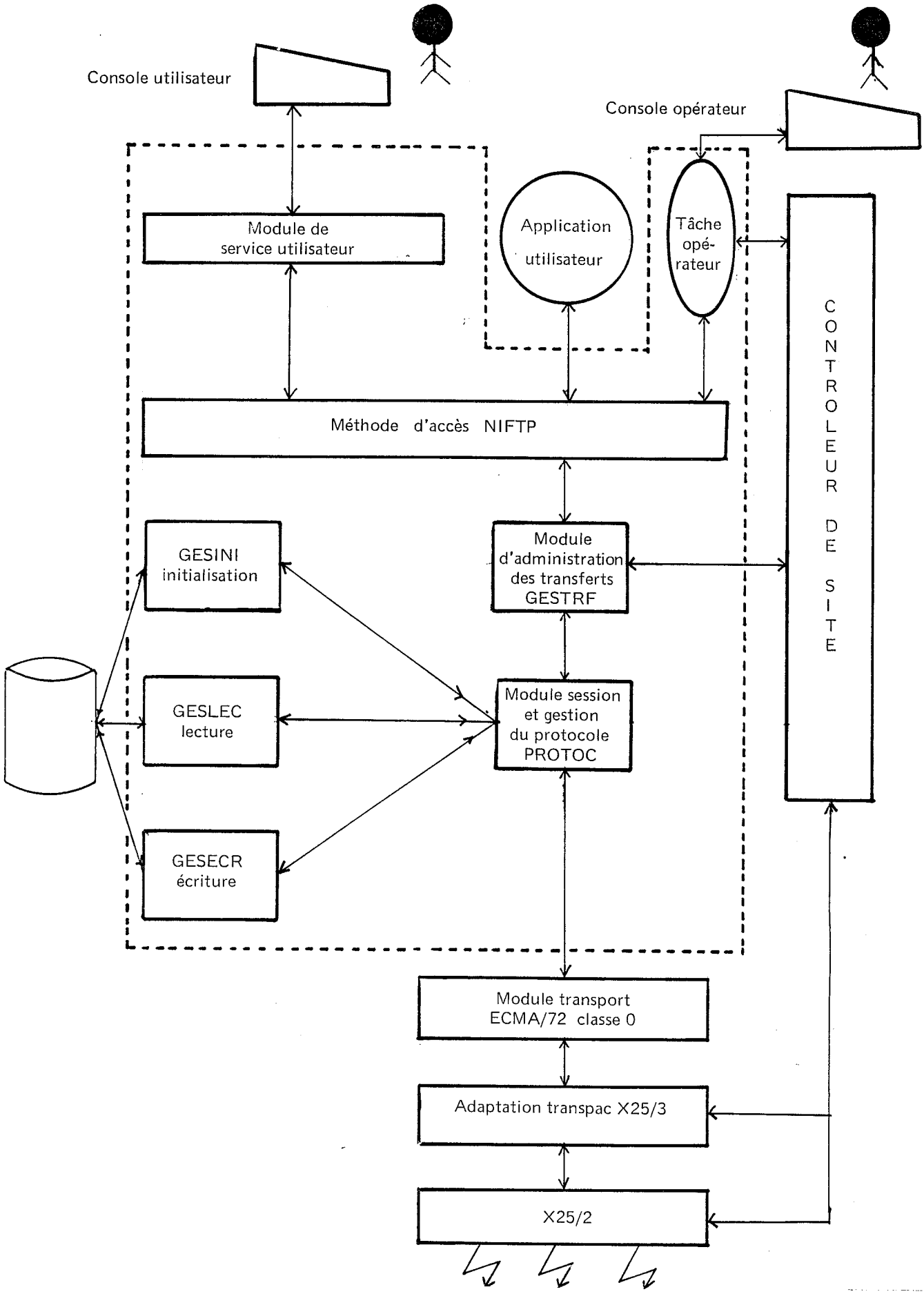


Fig 12 - Architecture interne de NIFTP

4.2.4 LES INTERFACES INTERNES NIFTP.

Dans cette architecture nous distinguerons deux interfaces privilégiées :

- l'interface méthode d'accès <-> GESTRF qui est le point d'accès au service transfert de fichier pour les utilisateurs intéractifs, les applications ou l'opérateur de site via la méthode d'accès.

- l'interface PROTOC <-> TRANSPORT qui est le point d'accès de la couche NIFTP au réseau.

- les autres interfaces ne vont concerner que le fonctionnement interne de la couche NIFTP.

Nous allons détailler ces interfaces.

4.2.4.1 Description des interfaces internes.

A l'exclusion de l'interface PROTOC-GESLEC tous les modules internes vont utiliser soit le service de communication intermodules à base d'ICB fourni par le noyau (voir paragraphe 4-1-6-4) soit un mécanisme de même type à base d'ICB réduits (ICBR) utilisé dans certains cas pour des raisons d'optimisation.

- Interface PROTOC-GESLEC

Cette interface va tenir compte du nombres maximum possible de transferts actifs simultanément (10) et des différentes fonctions que peut demander PROTOC à GESLEC. Chaque fonction va être associée à un mot dont chaque bit indiquera par son rang le numéro du transfert concerné (1 à 10) et par sa valeur si la fonction est demandée ou non. La table d'interface ACLECT va être constituée de quatre mots appelés:

- START ouverture du fichier et lancement de la lecture.

- LECT lecture des articles du fichier.

- ABORT abandon du transfert en cours.

- RSTART reprise de la lecture après un repositionnement dans le fichier de lecture à une marque donnée.

Lorsque PROTOC désire effectuer une de ces quatre opérations, il positionne le bit correspondant dans le mot associé à l'action demandée.

Les intéractions de PROTOC avec ses tâches esclaves GESLEC et GESECR ainsi qu'avec le TRANSPORT, vont faire l'objet de mécanismes qui , tout en garantissant un asynchronisme complet dans le fonctionnement de ces tâches, vont éviter d'engorger les files d'attentes associées à ces modules.

- interface GESLEC - PROTOC

Un mécanisme de contrôle de flux à l'interface GESLEC-PROTOC va être assuré à l'aide d'un compteur 'lecture' par transfert. Ce compteur est incrémenté par GESLEC lors de l'émission d'un ICB 'article' vers PROTOC pour un transfert donné. Lorsque ce compteur atteint une valeur "seuil haut" GESLEC met à zéro le bit LECT associé au transfert et cesse de lire des articles pour ce transfert.

Lorsque PROTOC déchaine un ICB article pour le compte d'un transfert, il décrémente le compteur correspondant. Si celui-ci passe par une valeur "seuil-bas" il remet à un le bit LECT associé au transfert afin de relancer GESLEC.

- Interface PROTOC-GESECR

Le mécanisme de contrôle de flux de PROTOC (Producteur) vis-à-vis de GESECR (consommateur) va comme dans le cas précédent être réalisé à l'aide d'un compteur d'écriture associé à chaque transfert. De plus, GESECR étant le consommateur final des articles émis par GESLEC, il est nécessaire d'asservir le fonctionnement de GESLEC à celui de GESECR.

Ceci va être réalisé de la façon suivante (pour chaque transfert) :

Coté récepteur

Lorsque PROTOC chaîne un ICB "article" vers GESECR il incrémente le compteur 'ecriture' de ce transfert. Lorsque ce compteur atteint une valeur "seuil haut", il va cesser d'acquitter les données reçues de la couche TRANSPORT pour ce transfert.

Coté émetteur

PROTOC va continuer d'émettre vers le TRANSPORT jusqu'à concurrence de sa fenêtre d'émission. Une fois celle-ci fermée il ne va plus consommer les articles fournis par GESLEC et grâce au mécanisme d'interface GESLEC-PROTOC, GESLEC va cesser de produire des ICB articles pour PROTOC lorsque le compteur 'lecture' vis-à-vis de PROTOC va atteindre la valeur seuil haut.

L'asservissement du producteur GESLEC vis-à-vis du consommateur GESECR est donc bien réalisé, on peut dire qu'il existe un contrôle de congestion de bout en bout entre ces deux modules.

La fin de la congestion au niveau GESECR va être concrétisée par le passage du compteur PROTOC-GESECR à la valeur "seuil bas". Lorsque GESECR consomme un ICB "article", il décrémente ce compteur. Si celui-ci passe à la valeur seuil bas, il prévient PROTOC par un ICB 'fin de congestion'. Celui-ci acquitte alors les données reçues du transport en attente d'acquittement, ce qui relance la production d'article sur le site émetteur.

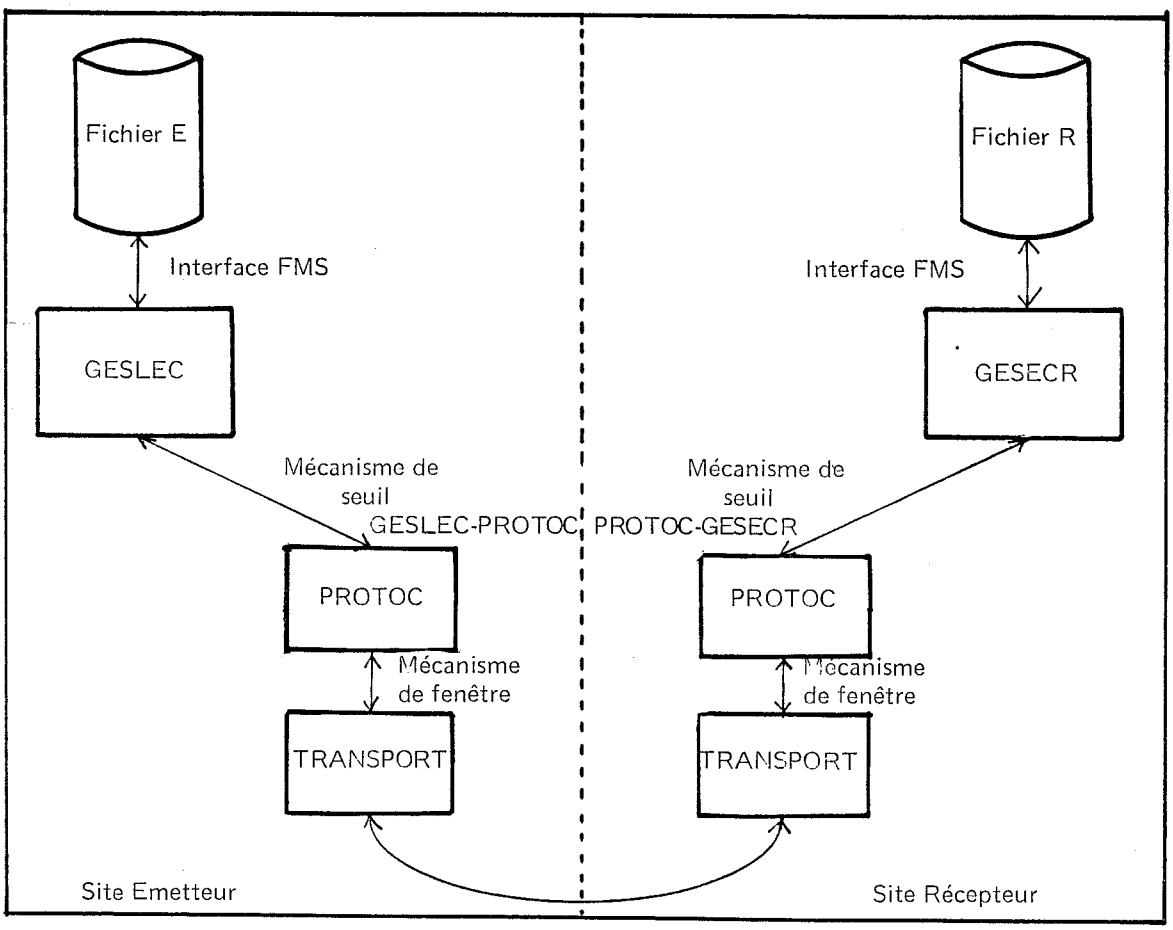


Fig 13 - Le contrôle de flux dans NIFTP.

- Interface PROTOC-TRANSPORT

Cette interface utilise le mécanisme des ICB standard associé à un mécanisme de fenêtre.

A l'ouverture d'un CV TRANSPORT, un compteur d'émission ou "fenêtre" est offerte à PROTOC par le TRANSPORT.

Lorsque PROTOC émet un ICB donné vers le transport, il décrémente cette fenêtre et s'arrête d'émettre lorsque celle-ci se ferme (devient nulle).

Lorsque le transport acquitte ces données, PROTOC incrémente cette fenêtre du nombre de données acquittées et peut reprendre l'émission des données vers le transport.

Remarque : Gestion de la fermeture de la fenêtre transport .

Sur fermeture de la fenêtre transport, PROTOC peut continuer à recevoir des ICB articles de GESLEC (jusqu'à concurrence du "seuil haut", voir interface GESLEC, PROTOC). Ces ICB ne pouvant être traités immédiatement vont être stockés dans une file d'attente propre au transfert FALEC (i) (i = numéro de transfert).
Sur réouverture de la fenêtre transport, PROTOC ira lire les ICB dans cette file en priorité.

4.2.4.2 Schéma des différentes interfaces

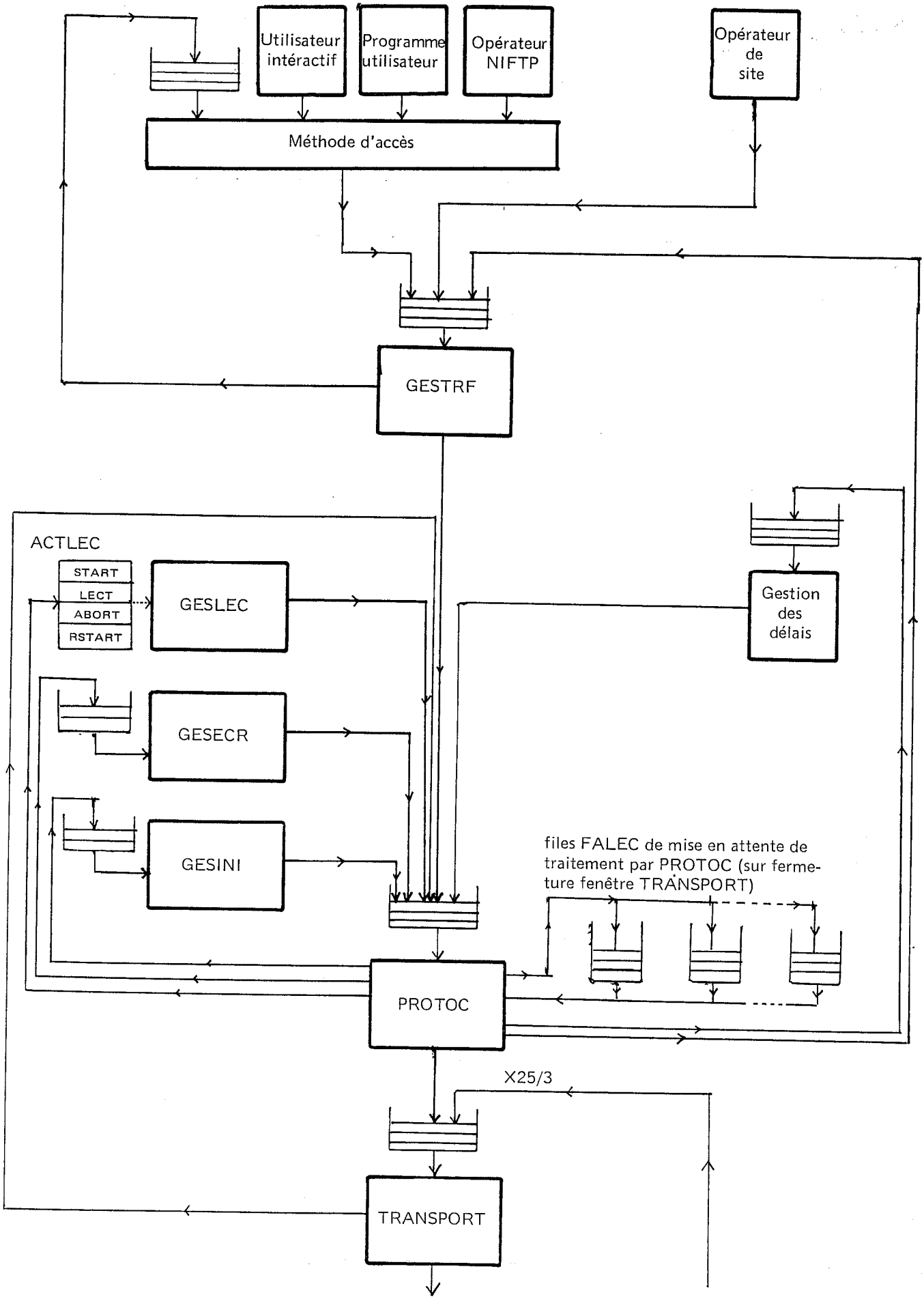


Fig 14 - Les interfaces dans l'Architecture NIFTP.

4.3 DESCRIPTION DES FONCTIONNALITES DES DIFFERENTS MODULES DE LA COUCHE NIFTP

Après avoir exposé l'architecture retenue pour le produit FTS-NIFTP dans son ensemble et les mécanismes de communication intermodules (voir Fig 14) nous allons détailler les fonctions associées à chacun des modules en insistant sur le module de gestion du protocole PROTOC.

4.3.1 LA METHODE D'ACCES NIFTP-LES INTERFACES EXTERNES.

Les interfaces offertes par la méthode d'accès vont permettre aux utilisateurs d'exercer des fonctions de contrôle et d'administration sur les transferts en fournissant des points d'accès au protocole de contrôle du transfert de fichier (voir paragraphe III-2).

Comme nous l'avons déjà évoqué il pourra exister trois types d'utilisateurs du FTS-NIFTP:

- l'opérateur de site qui est un utilisateur privilégié.
- les applications de la couche 7.
- les utilisateurs du système intéractif MMT2.

Chacun de ces utilisateurs va accéder au service FTS-NIFTP par des requêtes via une interface différente qui va avoir pour but de :

- transformer les requêtes utilisateur en ICB destinés au module de gestion des transferts GESTRF.
- fournir à l'utilisateur un moyen de se synchroniser sur la fin d'exécution d'une requête et lui transmettre le signal ou compte-rendu d'exécution remonté sous forme d'ICB par GESGRF.

Examinons ces trois types d'interfaces :

4.3.1.1 1- L'interface opérateur de site.

L'opérateur de site va disposer de deux types de commandes :

a) des commandes à destination du contrôleur de site : qui vont assurer le contrôle de l'activité réseau du site. Ce sont :

- des commandes de visualisation :
  - état d'un liaison (DLN).
  - liste des liaisons (LLN).
- des commandes de modification de l'activité réseau :
  - arrêt d'une liaison (HLN).
  - test d'une liaison (TLN).
  - mise en route de traces internes (MLN).
- des commandes de lancement et d'arrêt du système FTS-NIFTP.
  - lancement (CALL).
  - arrêt du système (HSI).

  Ces commandes sont les seules qui vont interfacer directement avec les modules spécifiques au FTS-NIFTP (sans passer par la méthode d'accès NIFTP).

B) des commandes spécifiques au transfert de fichier à destination du module d'administration des transferts GESTRF. Ce sont des commandes privilégiées en ce sens qu'elles peuvent être destinées à tous les transferts connus sur le site. L'opérateur de site va disposer des commandes suivantes :

- liste des transferts en cours (LFT).
- visualisation de l'état des transferts en cours (DFT).
- suppression d'un transfert en cours (KFT).
- purge de tous les transferts en attente de prise en compte (PFT).
- arrêt d'un transfert avec ou sans possibilité de reprise ultérieure (AFT).
- suspension d'un transfert en cours (HFT)
- continuation d'un transfert suspendu (CFT).
- reprise d'un transfert après arrêt de celui-ci (RFT).

Ces différentes commandes vont être transmises à GESTRF via la méthode d'accès NIFTP.

4.3.1.2 2- Les interfaces des programmes d'application et des utilisateurs intéractifs.

Les programmes d'application et les utilisateurs du système intéractif vont avoir les mêmes possibilités d'intéractions avec le service NIFTP à travers la méthode d'accès à l'aide d'un ensemble de requêtes.

-A l'interface programme application - NIFTP va consister en un jeu de macro-instructions correspondant aux différentes requêtes utilisateurs possibles que nous décrivons ci-après.

-B l'interface utilisateur du système intéractif - NIFTP va être constituée par des commandes adressées à un utilisateur appelé processeur NIFTP qui est en fait un programme d'application particulier utilisant le même jeu de macros-instructions qu'un programme standard. Le système NIFTP va gérer au maximum 32 sessions simultanées associées chacune à un utilisateur différent. Un utilisateur désirant utiliser le service NIFTP devra se connecter préalablement au service en donnant un numéro d'utilisateur (compris entre 1 et 32).

Contrairement aux commandes opérateurs de site, les commandes utilisateur ne pourront s'adresser qu'aux transferts lancés sous le numéro d'utilisateur donné au moment de la connexion.

Les différentes requêtes utilisateurs possibles sont :

- la connexion au service NIFTP (CNX).
- le lancement d'un transfert (SFT).
- l'arrêt d'un transfert avec ou sans possibilité de reprise (AFT).
- la suspension d'un transfert (HFT).
- la continuation d'un transfert suspendu (CFT).
- la reprise d'un transfert (RFT).
- la liste des transferts (LFT).
- la visualisation de l'état des transferts (DFT).
- la déconnexion du service NIFTP (DCNX).
- la synchronisation sur la fin d'exécution d'une requête (WFT).

Toutes les requêtes précédentes vont être prises en compte par la tâche d'administration des transferts GESTRF que nous allons détailler maintenant.

4.3.2 LE MODULE D'ADMINISTRATION DES TRANSFERTS GESTRF.

Ce module va avoir un rôle d'interface entre les utilisateurs locaux et la tâche protocole PROTOC. Pour reprendre la terminologie NIFTP elle va se partager avec celle-ci la gestion du "protocole de contrôle" des transferts, celle du "protocole de transfert" étant entièrement assurée par PROTOC.

GESTRF répond à deux types de stimuli :

- les requêtes utilisateurs provenant des différents interfaces utilisateurs.

- les indications et réponses en provenance de PROTOC.

Pour GESTRF un transfert va se dérouler en plusieurs étapes qui vont correspondre à autant d'états :

1 état "inexistant".

1 état "négociation" : transfert en négociation initiale par PROTOC.

2 état "transfert en cours" : début du transfert signalé par PROTOC.

3 état "demande de suspension" : suspension demandée à PROTOC.

4 état "suspendu" : PROTOC a signalé la suspension.

5 état "attente d'abandon": arrêt du transfert demandé à PROTOC.

6 état " attente d'abandon avec reprise" : abandon du transfert avec possibilité de reprise demandé à PROTOC.

7 état "attente de reprise": attente d'une requête utilisateur si P ou d'une indication PROTOC si Q signalant la reprise.

8 état "attente de prise en compte" : mise en attente si plus de 10 transferts actifs.

GESTRF va gérer jusqu'à 256 transferts dont 10 actifs simultanément pour le compte de 32 utilisateurs maximum.

Afin de pouvoir assurer l'administration des transferts avec le maximum de sécurité GESTRF va utiliser des fichiers de gestion qui vont contenir à chaque instant toutes les informations nécessaires à une reprise ultérieure des transferts (état du transfert, numéro d'utilisateur etc...).

GESTRF va effectuer le traitement immédiat des requêtes utilisateurs purement administratives telles que:

- la liste des transferts connus (LFT).

- la liste des états de transfert (DFT).

- la modification de la priorité de prise en compte d'un transfert en attente (MFT).

Par contre pour celles d'entre elles qui peuvent entraîner la modification de l'état d'un transfert, GESTRF va, après vérification de leur cohérence faire appel à PROTOC pour leur exécution.

Ce sont les requêtes :

- SFT : start file transfert.
- CFT : continuer un transfert.
- MFT : suspendre un transfert.
- AFT : abandonner un transfert (avec ou sans possibilité de reprise).
- RFT : reprise d'un transfert.
- PFT : purge d'un transfert en attente.
- HSI : arrêt du système NIFTP.

En fin d'exécution de tout ou partie de ces requêtes, PROTOC fournit une réponse, ce qui permet à GESTRF de suivre le déroulement du transfert. Les principales réponses venant de PROTOC sont :

- acquittement de début de transfert (ou de reprise).
- refus de transfert.
- refus de connexion transport pour le transfert concerné.
- suspension de transfert.
- continuation de transfert.
- abandon de transfert (réponse à AFT).
- acquittement de purge.
- acquittement d'arrêt du système (NIFTP).

Les évènements entrants vont être transmis par PROTOC à GESTRF sous forme d'indications. On trouve :

a) les indications ne demandant pas d'acquittement de la part de GESTRF:

- suspension de transfert venant du distant.
- continuation de transfert (après une suspension) venant aussi du distant.
- abandon de transfert.
- terminaison de transfert.

B) Les indications demandant un acquittement de la part de GESTRF :

- indication de début de transfert. GESTRF peut suite a cette indication accepter ou refuser le transfert en fonction de ses ressources.

Nous donnons ci-après un schéma fonctionnel montrant les intéractions entre la méthode d'accès GESTRF et PROTOC dans un transfert. Ce schéma ne tient pas compte du traitement interne effectué par chaque module.

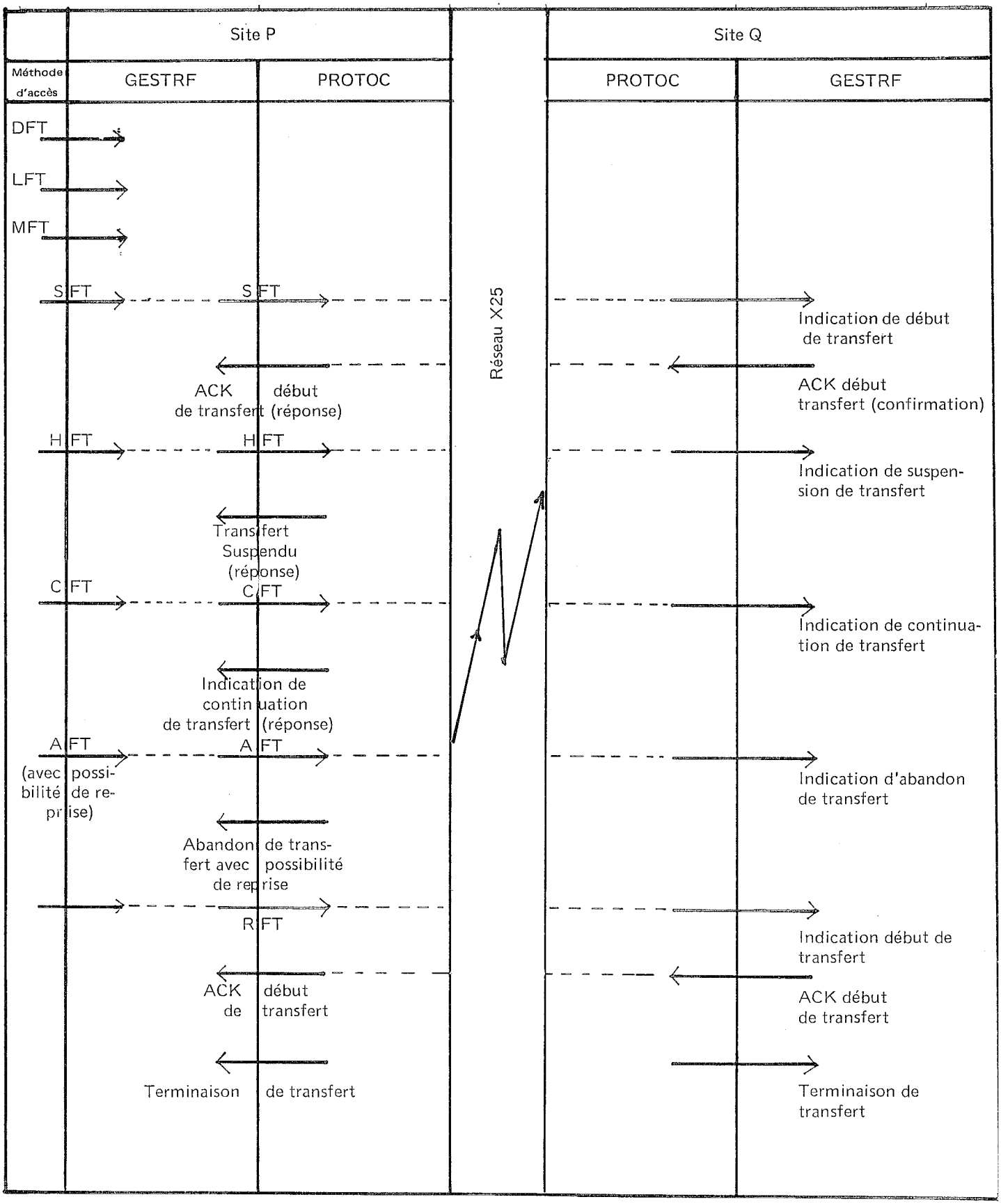


Fig 15 - Les intéractions entre GESTRF et PROTOC .

Afin de fournir un historique complet des opérations effectuées sur un site, GESTRF gère deux types de journaux :

- un journal opérateur concernant les transferts de type Q et les messages à destination de l'opérateur de site et contenant les informations suivantes :

  - heure de début et de fin des transferts.

  - diagnostics concernant l'arrêt des transferts (cause de l'arrêt, message d'information fournis par le distant, éventuellement paramètres ayant entrainé l'abandon du transfert etc...).

  - identification des transferts.

  - Messages d'informations destinés a l'opérateur.

- un journal par utilisateur contenant les mêmes informations mais uniquement relatives aux transferts lancés par cet utilisateur (de type P).

Un exemple de ces journaux est donné en annexe 7-6.

4.3.3 LE MODULE D'INITIALISATION DES TRANSFERTS GESINI.

Cette tâche va être chargée de prendre à son compte les opérations d'initialisation des transferts (vérification d'existence ou de non existence du fichier, création ou destruction du fichier suivant le mode d'accès utilisé) qui sont des opérations "longues", afin de ne pas pénaliser les autres transferts en cours.

GESINI rend compte à PROTOC du déroulement de ces opérations et lui signale si les ressources disques nécessaires au transfert sont suffisantes. En fonction de ce compte-rendu, PROTOC pourra accepter ou refuser le transfert en question.

4.3.4 LE MODULE DE GESTION DES LECTURES FICHIERS GESLEC.

Cette tâche va avoir trois fonctions principales :

A) fournir à PROTOC les articles du fichier côté émetteur du transfert avec ou sans positionnement initial dans le fichier.

B) assurer les mapping MITRA-NIFTP de ces articles après lecture en fonction des attributs négociés ("text formatting", "code de transfert", "type des données", "facilités") avant de les transmettre à PROTOC.

C) si l'option "marque" est supportée au niveau des facilités, calculer un intervalle de marques en fonction de la longueur des enregistrements du fichier traité et poser ces marques qui seront intercallées entre les ICB "articles" fournis à PROTOC sous forme d'ICB "marque". Ces marques seront matérialisées par des adresses disques correspondant à des points de reprises dans le fichier, GESLEC devant être en mesure de reprendre la lecture à partir de n'importe quelle marque déjà posée comprise dans la fenêtre d'acquittement (négociée dans la phase d'initialisation). Pour cela GESLEC gère une table des marques qui est sauvegardée sur disque à chaque changement d'état donnant pour chaque numéro de marque l'adresse disque associée. Le nombre d'éléments de cette table est donné par l'attribut "fenêtre maximum d'acquittement des marques".

4.3.5 LE MODULE DE GESTION DES ECRITURES FICHIERS GESECR.

Ce module va avoir des fonctions symétriques à GESLEC et va assurer:

- le "mapping" NIFTP-MITRA des articles reçus de PROTOC en fonction des attributs négociés ("code de stockage", "facilités", "code de transfert", "type des données", "text formatting").

- l'écriture des articles dans le fichier, côté récepteur avec ou sans positionnement initial dans le fichier.

- l'acquittement des marques reçues de PROTOC vers ce même module sous forme d'ICB "acquittement de marques". GESECR n'acquitte une marque qu'une fois que l'article antérieur à cette marque a été convenablement écrit. De même que GESLEC, GESECR va associer à ces marques des adresses disques et gère une table des marques de nombre d'éléments égal à "la fenêtre maximum d'acquittement des marques".

Dans la description précédente des tâches GESLEC et GESECR, nous avons parlé d'opérations de "mapping". Nous allons décrire ci-après comment ce mécanisme a été implémenté dans les systèmes FTS-NIFTP.

4.3.6 LA GESTION DU MAPPING NIFTP-MITRA.

Comme nous l'avons déjà décrit dans le paragraphe 2-2-3 le "Mapping" va avoir pour but de transformer le fichier à transférer, qui peut avoir des représentations différentes suivant les machines utilisées, en fichier virtuel compréhensible par les entités communicantes.

Le Mapping va se dérouler en deux étapes :

a) première étape : (phase de la négogiation initiale) .

Elle consiste à décrire les caractéristiques du fichier à transférer (nom, taille, longueur des enregistrements, nom et type du support) et les transformations à effectuer sur les données lors de la transmission (compactage, formattage, transcodage) en terme d'attributs standards NIFTP et à négocier ces attributs avec le distant afin d'avoir un ensemble cohérent d'attributs côté P et Q.

B) deuxième étape : (phase data) .

Elle consiste à effectuer sur les données les transformations imposées par les attributs négociés dans l'étape précédente, ceci entre la lecture des enregistrements par GESLEC et leur transmission vers le distant ou entre la réception des données chez le distant et l'écriture de ces données sur disque par GESECR.

Avant de décrire l'opération de Mapping sur MITRA, il est utile de rappeler les principales caractéristiques d'un fichier MITRA.

Un fichier MITRA se définit par:

- son nom (PFN).
- le nom de l'unité de disque le supportant (PVN).
- le type de l'unité de disque (DVT).
- son type (partitionné séquentiel, direct).
- le type des données (binaire ou texte).
- la forme des enregistrements (fixe ou variable).
- la longueur maximum d'un enregistrement.
- le facteur de blocage des enregistrements.

Afin de réaliser la première étape du "Mapping" ,à ces caractéristiques il est nécessaire de faire correspondre des attributs NIFTP.

Néanmoins il existe des caractéristiques qui n'auront pas à être traduites en termes d'attributs, car elles constituent des valeurs par défaut de l'implémentation. C'est le cas du type et du format du fichier. En effet seuls les fichiers de type séquentiel de format variable étant supportés il est inutile de transmettre ces informations au distant.

De même le facteur de blocage chez le récepteur ne va pas être transmis mais calculé à partir des deux attributs "taille maximum des enregistrements transférés" et "taille maximum des enregistrements chez Q".

L'attribut "nom de fichier" va servir à décrire les trois caractéristiques PFN, PVN et DVT du fichier.

Les attributs "taille de fichier", "type des données" et "taille maximum d'un enregistrement de fichier distant" vont servir à décrire les caractéristiques correspondantes du fichier.

Le problème de la description d'un fichier MITRA en termes NIFTP est donc résolu. En ce qui concerne les transformations à effectuer sur les données transmises, elles seront précisées par les attributs :

- "text formatting".
- "délimiter présentation".
- "tabulations horizontales".
- "code de transfert".
- "code de stockage".
- "facilités" (pour la compression des données).

La négociation de ces attributs va être assurée par PROTOC conformément au mécanisme préconisé dans le paragraphe 2-4-1. Dans le chapitre consacré à cette tâche nous décrirons en détail cette opération.

La deuxième étape du mapping sera réalisée par les tâches GESLEC côté émission et GESECR côté réception en fonction des attributs négociés. Au cours de cette étape les transformations suivantes pourront être effectuées sur les données transférées :

- compactage.
- transcodage.
- formattage.

4.3.7 LE MODULE GESTION DU PROTOCOLE PROTOC.

4.3.7.1 Description fonctionnelle de PROTOC

PROTOC est le module principal du système NIFTP et va avoir pour fonction d'assurer la gestion des transferts de fichiers dans leur intégralité entre des processus "P" et "Q" conformément au protocole NIFTP.

En particulier PROTOC va avoir à sa charge:

- La gestion des chemins virtuels transport (établissement, échanges de données, libération).
- Le contrôle des transferts en collaboration avec GESTRF, basé sur un protocole de contrôle propre à l'implémentation MITRA.
- Le transfert des fichiers proprement dit conformément au protocole de transfert déjà exposé.

Nous allons détailler ci-après les problèmes posés par l'implémentation de ces deux protocoles.

A) Le protocole de contrôle des transferts : A l'aide de ce protocole qui ne fait pas l'objet d'une normalisation, PROTOC en collaboration avec GESTRF va exercer un contrôle sur les transferts, exécutant les requêtes reçues de ce module et prévenant celui-ci des événements pouvant modifier l'état d'un transfert. PROTOC réagit donc à deux types de stimuli :

1 - Les commandes ou requêtes provenant de GESTRF On se reportera au chapitre 4-5-2 pour avoir la liste des requêtes émises par GESTRF et des réponses élaborées par PROTOC. Le traitement de ces requêtes pourra être effectué directement (exemple commande PURGE des transferts en attente) pour les commandes ne faisant pas intervenir le protocole de transfert, ou sous-traité à la partie protocole de transfert dans le cas contraire. Dans les deux cas,une réponse sera renvoyée une fois le traitement de la requête terminé.

2 - Les évènements internes provenant de la couche TRANSPORT signalés par le protocole de transfert et entrainant une modification de l'état d'un transfert. Dans ce cas la partie protocole de contrôle va traiter l'indication (mise à jour du contexte par exemple) et la transmettre à GESTRF (voir ch. 4-5-2).

B) Le protocole de transfert de fichier : va être strictement conforme au protocole NIFTP-B(80). L'implémentation de ce protocole va se traduire par le support de tous les automates définis dans celui-ci (voir annexe 7-5) :

- Automates de la phase initialisation du transfert.
- Automates de la phase transfert de données.
- Automates de la phase terminaison du transfert.

En outre tous les mécanismes décrits dans le protocole vont être implémentés dans PROTOC. Ce sont :

- le mécanisme de gestion des "résumption" ou reprise à froid entrainant la gestion de dossiers de transferts renfermant pour chaque transfert toutes les informations nécessaires à sa reprise ultérieure.
- le mécanisme de gestion des marques assuré en collaboration avec les tâches GESLEC et GESECR.
- le mécanisme de la négociation initiale des attributs de transfert en phase initialisation.
- le mécanisme d'allocation des identificateurs de transfert assurant l'unicité de l'identification d'un transfert dans un contexte réseau hétérogène.

4.3.7.2 L'Architecture

La tâche PROTOC va comporter deux parties :

1 - Le noyau qui va recevoir les ICB venant des modules TRANSPORT (gestion des délais), GESTRF, GESINI, GESLEC ou GESECR (voir Fig 14) et les transformer, suivant leur type :

- soit en évènements d'activation des automates NIFTP : Dans ce cas il y a appel de la partie automate qui effectuera le traitement adéquat.

- soit en évènements de gestion interne au noyau : Ces évènements seront dans ces cas traités directement par le noyau. En particulier ce sera le cas des fonctions de contrôle du transfert précédemment décrites et des fonctions sessions (gestion des CV transport).

2 - La partie automate NIFTP . Cette partie va, en fonction de l'évènement positionné par le noyau, de l'automate et de l'état courant, exécuter l'action décrite dans les tableaux d'états du protocole NIFTP, le numéro d'état et d'automate pouvant être modifiés à la fin du traitement.

4.3.7.3 Les interfaces .

Les interfaces de PROTOC vis à vis des autres modules ainsi que les différents mécanismes de contrôle de flux aux interfaces ont déjà été décrits dans le paragraphe 4-4-2 (voir figure 14).

Nous rappellerons simplement que PROTOC interface avec les modules :

- GESECR par des éléments de la file d'attente FAECR et des ICB.

- GESLEC par la table ACTLEC et des ICB.

- GESINI par des éléments de la file FAINI, et des ICB.

- GESTRF par des ICB.

- TRANSPORT par des ICB.

Après cette description générale de l'Architecture interne de PROTOC nous allons examiner plus précisément quelques détails de réalisation ayant trait à l'implémentation des mécanismes propres au protocole NIFTP et en particulier concernant la gestion :

- Des automates NIFTP.

- De l'interface avec le transport.

- Des identificateurs de transfert (gestion des reprises à froid).

- Des contextes et des dossiers de transfert (reprises à froid).

- De la négociation initiale des attributs.

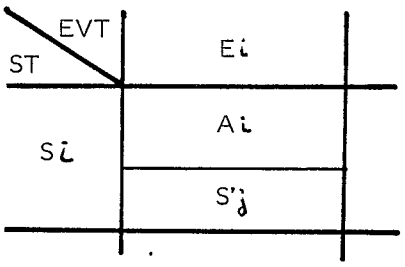
- des marques.

4.3.7.4 Implémentation des automates NIFTP.

1 - Les tables utilisées : Par souci de conformité avec le protocole NIFTP, PROTOC va utiliser les tableaux d'états tels qu'ils sont décrits dans le protocole NIFTP (voir annexe 7-5). Dans ces différents tableaux on a fait la distinction entre les différentes phases de transfert (initialisation, transmission de données et terminaison) et à l'intérieur de ces phases entre processus P ou Q (phase initialisation terminaison) ou sens émetteur ou récepteur (phase transmission de données). Ainsi on trouve les automates :

- INITQ pour la phase initialisation du processus P.
- INITP pour la phase initialisation du processus Q.
- TERMP pour la phase terminaison du processus P.
- TERMQ pour la phase terminaison du processus Q.
- SEND pour la phase transmission de donnée chez l'émetteur.
- RCVE pour la phase transmission de donnée chez le récepteur.

Chacun des éléments constituant ces tables peuvent se décrire ainsi :



Ei=Evènement recu.

Si=Etat courant automate en cours.

Ai=Action a effectuer.

Sj=Etat suivant.

Fig 16 - Détail d'un élément d'une table d'états.

Le traitement d'un évènement pourra se schématiser de la façon suivante :

1 retrouver le numéro d'automate courant en fonction de la phase et du type de processus ou du sens du transfert.

2 exécuter l'action indiquée en fonction de l'état courant et du numéro d'évènement reçu. Cette action pouvant éventuellement modifier la phase, le type de processus ou le sens du transfert.

3 armer un délai si nécessaire pour éviter un dead-lock sur attente d'un évènement et déterminer le nouvel état courant du transfert.

Afin de pouvoir effectuer ce traitement, PROTOC va associer à chaque transfert en cours un contexte automate qui va être constitué par les entités suivantes :

- Type de processus (P ou Q).
- Sens du transfert (émetteur ou récepteur).
- Nom de la phase du transfert (initialisation, donnée, terminaison).
- Numéro d'état courant dans l'automate en cours.

A l'aide de ce contexte, PROTOC sera en mesure, à partir d'un évènement positionné par le noyau, de déterminer et d'exécuter l'action correspondante de l'automate en cours.

L'information "délai à armer" lors du passage dans certains états (attente de RPOS après SFT par exemple), sera rajoutée dans les tableaux d'états au moment de leur implémentation dans PROTOC et associée à l'information Sj (état suivant).

Cette implémentation fait apparaître un problème concernant la gestion des numéros d'états et d'évènements.

En effet ces tableaux sont constitués par des événements et des états qui peuvent se retrouver dans les différents automates mais à des emplacements différents ce qui peut donner lieu à des ambiguités (Par exemple, l'évènement timed-out à le numéro 8 dans l'automate INITQ et le numéro 5 dans l'automate TERMP).

Par souci de clarté au niveau de la réalisation il est nécessaire de se définir une identification globale unique des évènements et des états, valable pour tous les automates, tout en conservant une implémentation des tableaux d'états conforme à l'annexe 7-5.

Pour ce faire les évènements et les états vont avoir une identification globale à tous les automates et une identification locale à chacun d'eux, le passage de l'une à l'autre étant assuré par deux tables associées à chaque tableau d'états :

- La table TBEVT donnant les numéros d'évènements "locaux" en fonction des numéros d'évènements "globaux".
- La table TBSTAT donnant les numéros d'états "locaux" en fonction des numéros d'états "globaux".

Le positionnement d'un évènement ou d'un état se fera toujours en utilisant l'identification "globale". La recherche de l'action à effectuer pour un automate donné se fera après avoir effectué une transformation du numéro d'évènement et d'état afin d'obtenir les numéros d'état et d'évènement "locaux" à cet automate.

2 Activation de l'automate . Le noyau va positionner le (les) évènement(s) dans une table de bit TABEV, les rangs des bits dans cette table correspondant aux numéros d'évènements globaux, et appeler la partie automate qui va effectuer les opérations suivantes :

- 1 recherche du premier bit à un dans TABEV (numéro d'évènement global EVTG).

- 2 recherche du tableau d'états (AUTO) concerné et des tables d'états TBSTAT et d'évènement TBEVT associées, en fonction du contexte d'automate CTXAUT (numéro de phase, type de processus, sens du transfert) contenu dans le contexte du transfert concerné.

- 3 calcul du numéro d'évènement local EVTL à partir de EVTG et de la table TBEVT.

- 4 calcul du numéro d'état local STATL à partir du numéro d'état courant global STATG dans le contexte et de la table TBSTAT.

- 5 calcul du numéro d'action à effectuer à l'aide du tableau d'état AUTO fonction de EVTL et STATL et exécution de l'action Ai.

- 6 mise à jour de l'état courant Sj dans le contexte et armement éventuel d'un délai Dj.

Le schéma suivant donne la configuration des tables utilisées et les différentes opérations effectuées.

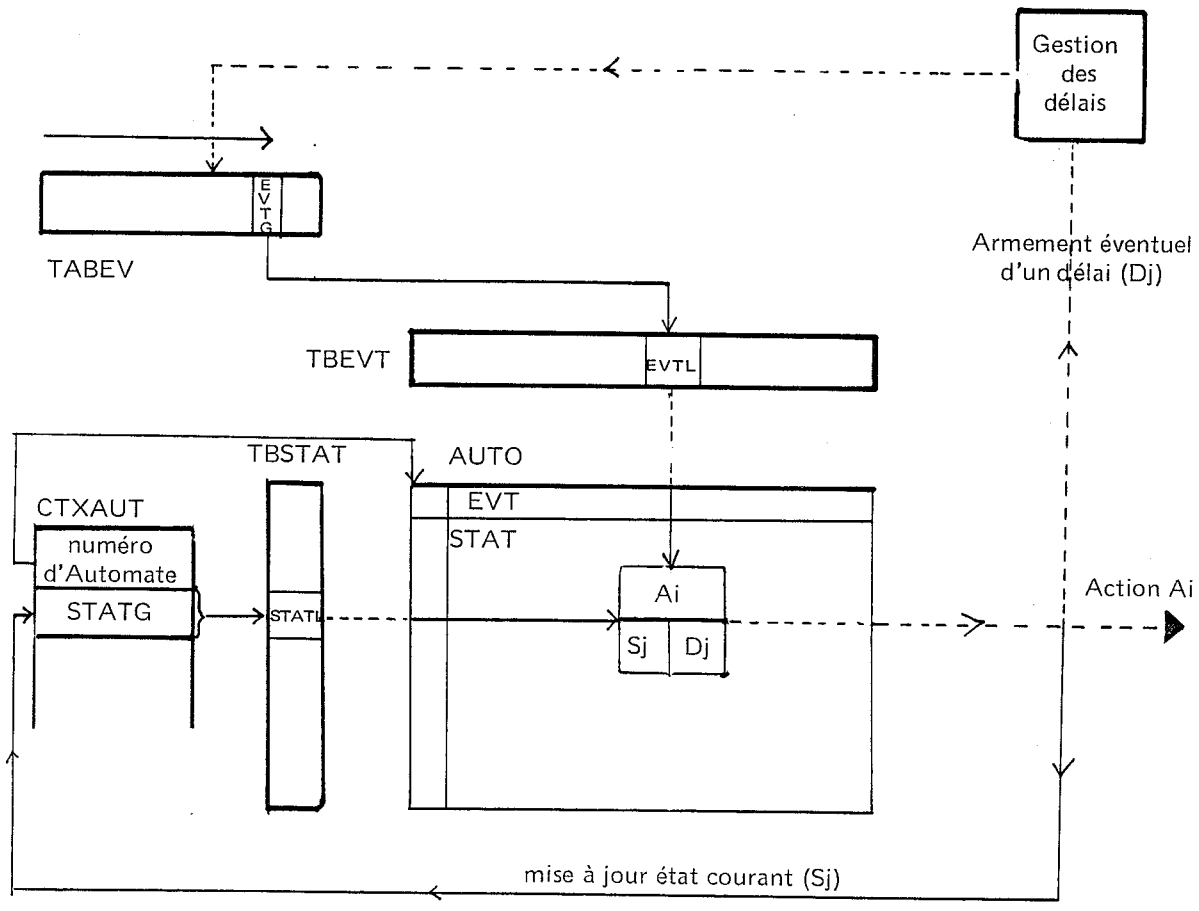


Fig 17 Gestion des automates dans PROTOC.

4.3.7.5 Gestion de l'interface PROTOC - TRANSPORT

1 Gestion des connexions transport : Le transport ECMA72 classe 0 va offrir un service de connexion permettant d'établir des chemins virtuels(CV) entre deux entités communicantes.

Ce protocole n'assurant pas le multiplexage des chemins virtuels sur une connexion réseau, il y aura une correspondance un pour un entre un CV transport et un CV réseau.

PROTOC va associer à chaque transfert un CV TRANSPORT qui sera libéré à la terminaison de celui-ci.

L'établissement de la connexion (en fonction des paramètres fourni dans la commande SFT reçue), la gestion des échanges de données (voir 2) ainsi que la déconnexion seront à la charge de PROTOC.

Au niveau de PROTOC, un CV TRANSPORT est caractérisé par un numéro de contexte fourni par celui-ci lors de la connexion. L'établissement d'une connexion réseau va se dérouler suivant le schéma suivant :

FIG18Q04900

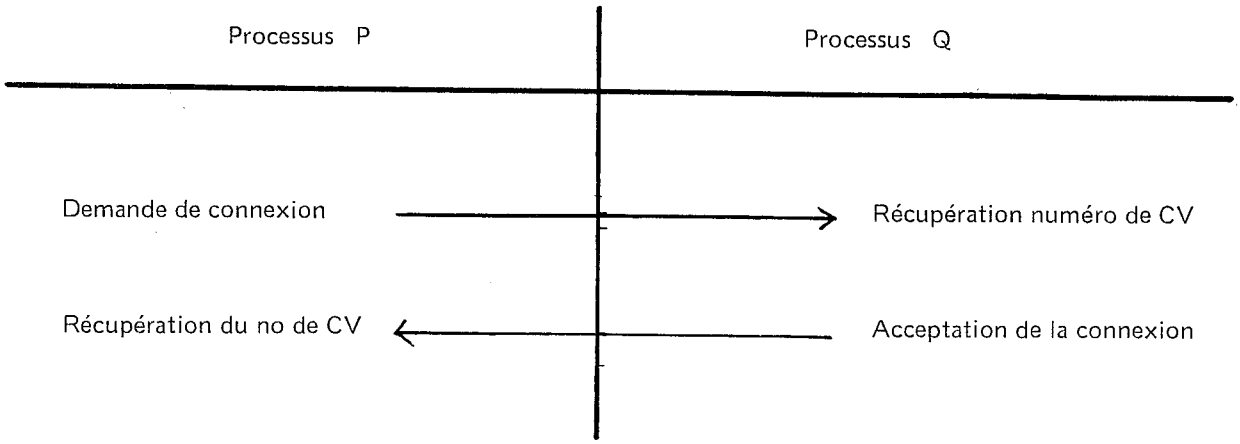


Fig 18 - Etablissement d'une connexion réseau.

La rupture d'un CV s'effectuera par une demande de déconnexion.

2 Gestion des données à l'interface PROTOC - TRANSPORT: Les deux modules PROTOC et TRANSPORT gèrent des unités de données différentes :

- pour PROTOC l'unité de donnée est l'enregistrement ou "record" NIFTP de longueur indéterminée constitué de "subrecords" de longueur maximum 64 octets, le dernier comportant l'indication "fin d'enregistrement" (EOR) dans son en-tête (voir annexe 7-2).

- pour le TRANSPORT, l'unité est la TSDU (Transport Service Data Unit) qui est, comme le "record NIFTP", une unité logique de transmission de longueur théoriquement illimitée, divisée en une ou plusieurs TPDU (transport protocol data unit), de longueur maximum négociée à l'établissement du chemin virtuel (128, 256, 1024 ou 2048 octets), la dernière TPDU comportant l'indication "fin de TSDU" (EOT) dans son en-tête.

Le transfert de données de PROTOC vers le transport et inversement va donc nécessiter une adaptation :

En émission :

PROTOC va fragmenter des données à émettre en TPDU, une unité logique de transmission constituant une TSDU, la dernière comportant l'indication EOT.

Remarque :

Il est a noter que la seule relation liant les TSDU aux enregistrements NIFTP est la suivante: Une TSDU peut contenir un ou plusieurs "subrecords", un ou plusieurs record mais pas de subrecord incomplet.

Ces TPDU vont être émises vers le TRANSPORT (à l'aide du mécanisme d'ICB) jusqu'à concurrence de la fenêtre PROTOC-TRANSPORT. Sur fermeture de celle-ci, les ICB à destination du TRANSPORT seront chainés dans une file d'attente associée au transfert (FALEC) en attendant sa réouverture (voir paragraphe 4-4-2-1).

En réception :

PROTOC va réassembler les TPDU reçues jusqu'a obtention d'une TSDU complète. L'analyse des données va commencer à partir de cet instant et va consister à reconstituer à partir des "subrecords" contenus dans la TSDU :

- soit un "enregistrement" données NIFTP,

- soit une commande d'initialisation ou de terminaison (SFT, RPOS, RNEG,...),

- soit une commande de contrôle de transfert NIFTP (MS, SS, etc...).

Une fois cette reconstitution effectuée, un évènement protocole sera positionné et la partie automate appelée.

Cette reconstitution peut s'effectuer en plusieurs étapes, un "record" NIFTP pouvant être découpé en plusieurs TSDU (voir remarque précédente en émission).

Dans ce cas, PROTOC effectuera un stockage intermédiaire des données reçues en attendant d'avoir reconstitué un "record" complet, avant de positionner l'évènement correspondant. Le seul cas ou l'évènement est positionné systématiquement est celui des commandes de contrôle qui ne sont constituées que d'un seul "subrecord".

FIG19Q04900

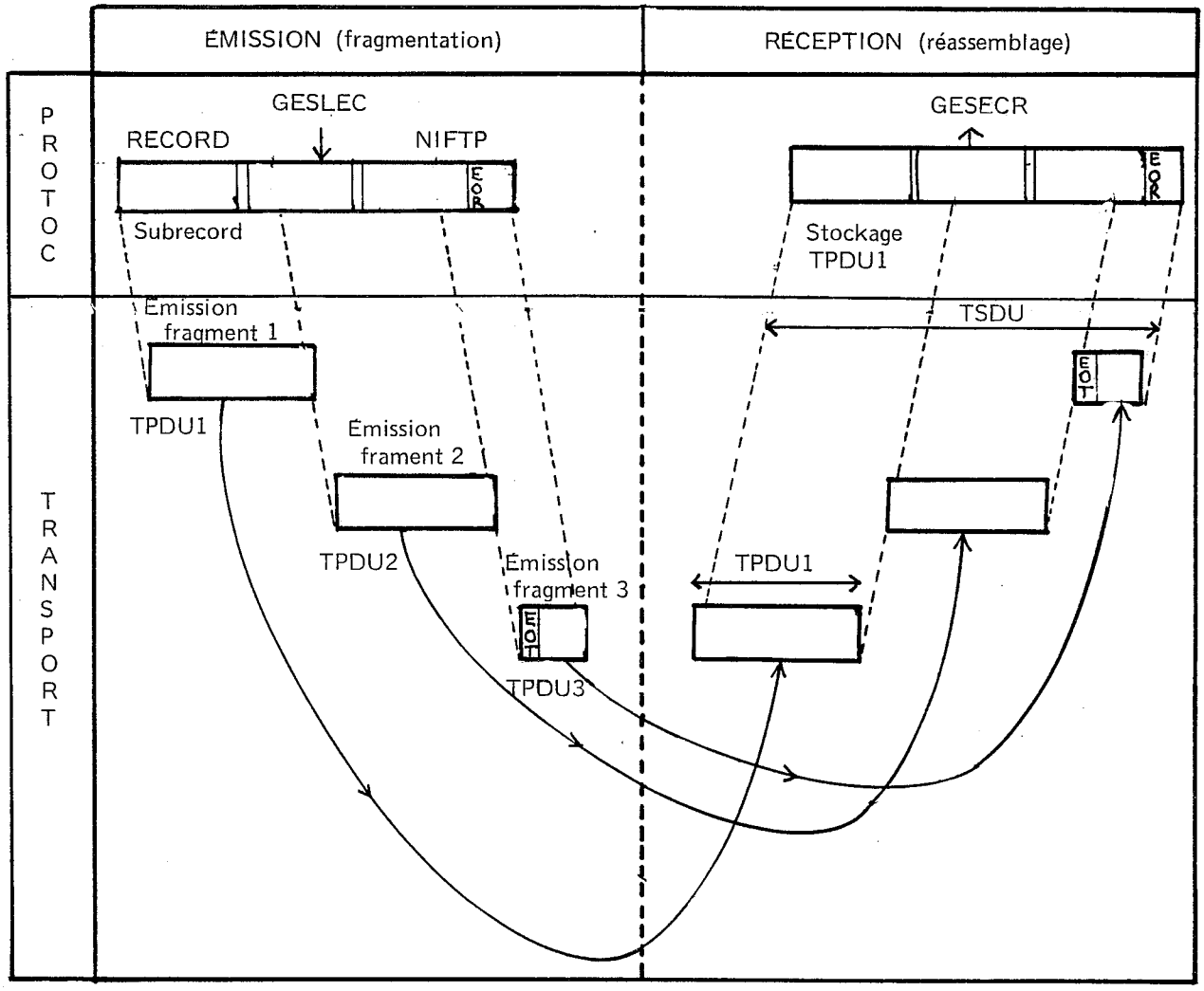


Fig 19 - Fragmentation - réassemblage dans PROTOC

4.3.7.6 Gestion des reprises à froid: les identificateurs de transfert.

L'identificateur de transfert sur un site va avoir pour rôle d'assurer l'unicité de l'identification d'un transfert donné au niveau de ce site dans un contexte réseau hétérogène en vue d'une reprise à froid du transfert.

Pour ce faire, il va être composé de deux parties :

- une partie identifiant le transfert sur le site initiateur (P) et déterminée par ce site en fonction de critères qui lui sont propres.

- une partie identifiant le site appelant côté répondeur (Q) et déterminée par celui-ci à l'aide des indications fournies à l'ouverture du CV transport (voir 4-5-7-5).

Le mécanisme de l'allocation de l'identificateur de transfert ou TFID est donc différent suivant que le site est initiateur ou répondeur :

- côté initiateur P :

GESTRF va déterminer la première partie de l'identificateur du transfert, lors de sa prise en compte par un numéro d'ordre N compris entre 1 et 255.

La deuxième partie , identification du site appelant, est sans objet pour le site initiateur et sera fixée à zéro.

L'espace des identificateurs de transfert de type P va être de la forme :

Ep = (N, 0) pour 1 =< N =< 255

La première partie du TFID est transmise par l'intermédiaire de l'attribut "transfer-ident" ou TRID dans la commande SFT afin de fournir au site Q, l'identification locale du transfert chez P.

- côté répondeur Q :

La première partie du TFID va être formée par la valeur de l'attribut "transfer-ident" reçu dans la commande SFT venant du site initiateur et constituée par :

- la première partie de l'identificateur de transfert côté P si le site initiateur est un MITRA.

- une valeur déterminée "autrement" si le distant n'est pas un site MITRA (mais identifiant toujours le transfert sur ce site de facon unique).

La deuxième partie va être constituée par le numéro s du site appelant, obtenu à l'aide des indications fournies par le TRANSPORT à l'ouverture du CV. Ce numéro de site est bien sûr unique au niveau du réseau et fixé à la génération.

L'espace des identificateurs de transfert de type Q va être de la forme:

Eq = (I, s) avec 0 =< I =< 65535 et 1 =< s =< 255

On peut montrer facilement que les espaces Ep et Eq précédemment définis sont disjoints ce qui va garantir l'unicité des identificateurs des transferts de type P ou Q sur chaque site du réseau.

Sur le schéma suivant, nous pouvons vérifier que les TFID alloués suivant le mécanisme précédemment décrit sont bien exclusifs.

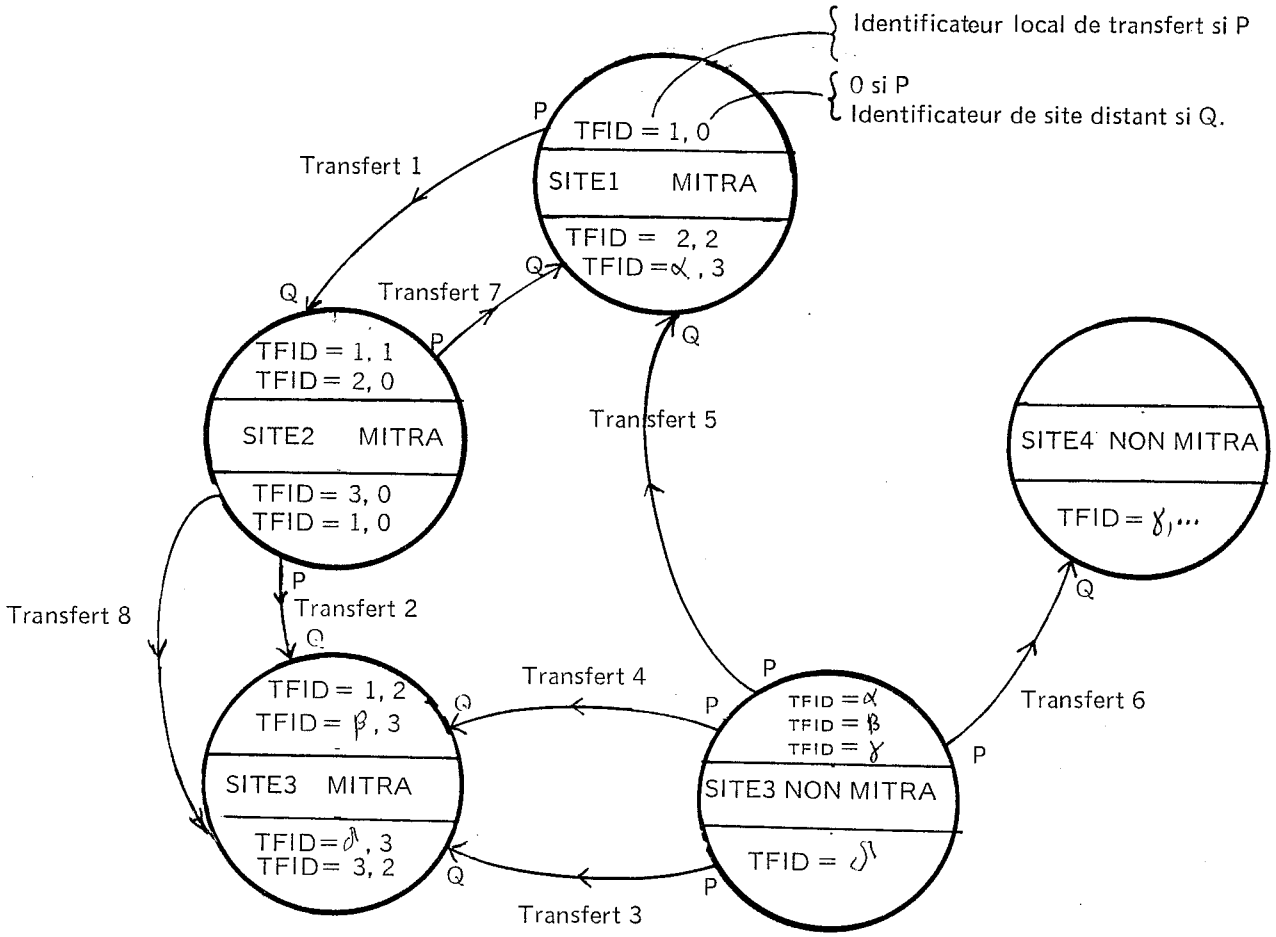


Fig 20 - Exemple d'allocations d'identificateurs de transferts.

4.3.7.7 Gestion des transferts : Contextes et dossiers de transfert.

1 Les contextes de transfert

PROTOC va associer à chaque transfert un contexte de transfert qui va être alloué dynamiquement au lancement d'un transfert et qui va contenir toutes les informations nécessaires à sa bonne exécution et en particulier :

- l'adresse du CV transport une fois celui-ci établi.
- les informations concernant la gestion de la fenêtre vers le transport .
- les informations concernant l'état des seuils avec GESLEC ou GESECR.
- le contexte automate de transfert précédemment décrit.
- les identificateurs au sens gestionnaire mémoire des différentes chaines de données en attente de traitement par PROTOC (données reçues du transport, commande Init-term en construction, "messages d'information" en attente pour GESTRF).
- des indications de gestion internes aux tâches GESLEC, GESECR et PROTOC qui se partagent le contexte.

Les informations stockées dans le contexte de transfert sont "volatiles". En effet en cas d'interruption du transfert, le contenu du contexte est perdu car celui-ci n'est pas sauvegardé sur disque. Or le protocole NIFTP impose dans certains cas de pouvoir reprendre des transferts après une interruption. Dans ce cas il est nécessaire de prévoir un deuxième type de contexte, le dossier de transfert qui va être constitué par une table pointée par un item du contexte et qui lui va être sauvegardé sur disque à chacune de ses modifications.

2 - Les dossiers de transfert

Conformément au protocole NIFTP, PROTOC va associer à chaque transfert un dossier de transfert qui va permettre la reprise d'un transfert après une interruption volontaire (commande d'abandon avec possibilité de reprise) ou non (rupture du CV transport par exemple, panne machine etc...).

Il va de soi que la possibilité de reprise suppose que cette facilité soit demandée au lancement du transfert dans l'attribut correspondant.

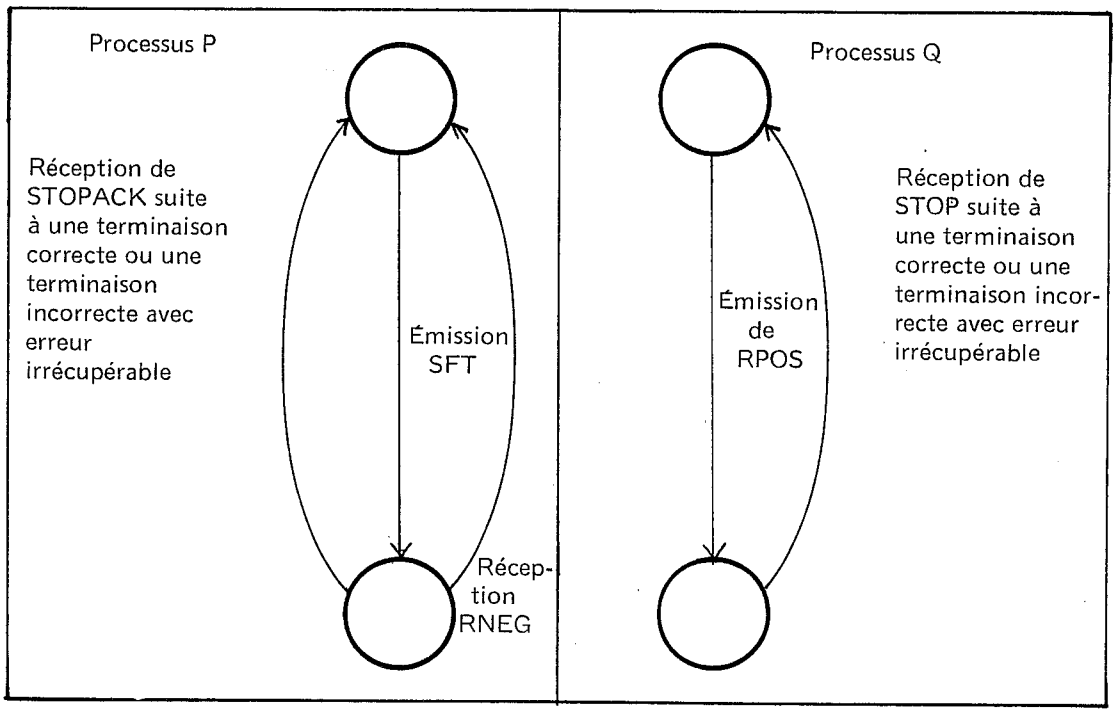
Un dossier va contenir les informations suivantes :

- les informations concernant la gestion des marques (dernière marque émise ou reçue, dernière marque acquittée en émission ou en réception, etc...).
- l'état courant de tous les attributs NIFTP de type T ou Q (qualifier, valeur) afin de pouvoir en cas de reprise être capable de reprendre le transfert dans les mêmes conditions.
- l'identificateur global du transfert TFID.
- des indicateurs internes à PROTOC concernant l'état courant du transfert.

Un certain nombre d'informations du dossier seront utilisées quelles que soit les possibilités de reprises du transfert.

Si la possibilité de reprise est supportée les dossiers seront créés sur disque et détruits conformément au protocole NIFTP suivant les schémas suivants.

FIG21Q04900



Etat 0 dossier inexistant
Etat 1 dossier existant

Fig 21 - gestion des dossiers de transfert par PROTOC

Ces dossiers seront sauvegardés sur disque dans un fichier à accès direct FDOSS.

La gestion des dossiers va être effectuée à l'aide d'une table (TABDOS implantée dans la zone donnée de PROTOC et indexée par les numéros de transfert utilisateurs (FTUS) alloués par GESTRF et contenant dans chaque poste l'identificateur de transfert TFID si le transfert existe ou NIL sinon.

Cette table est sauvegardée à chacune de ses mises à jour et constitue le premier enregistrement du fichier FDOSS.

Les dossiers de transfert sont sauvegardés dans ce fichier à une adresse qui est fonction du numéro de FTUS associé a chacun d'eux.

Les opérations possibles sur le fichier FDOSS sont de cinq types :

1 - Initialisation : Deux types d'opération suivant que le fichier existe ou non.

- si le fichier n'existe pas création du fichier FDOSS et initialisation de TABDOS.

- si le fichier existe, lecture de TABDOS.

2 - Recherche dans TABDOS d'un poste de TFID donné. Rechercher dans TABDOS un poste de TFID donné et rendre le FTUS correspondant si trouvé.

3 - Création d'un poste dans TABDOS de numéro de FTUS et de TFID donné.

- vérifier que ce poste est libre dans TABDOS.

- initialiser le poste avec le TFID.

- sauvegarder TABDOS dans FDOSS.

4 - Remplacement ou création d'un dossier de transfert dans FDOSS.

- calculer l'adresse disque en fonction du FTUS.

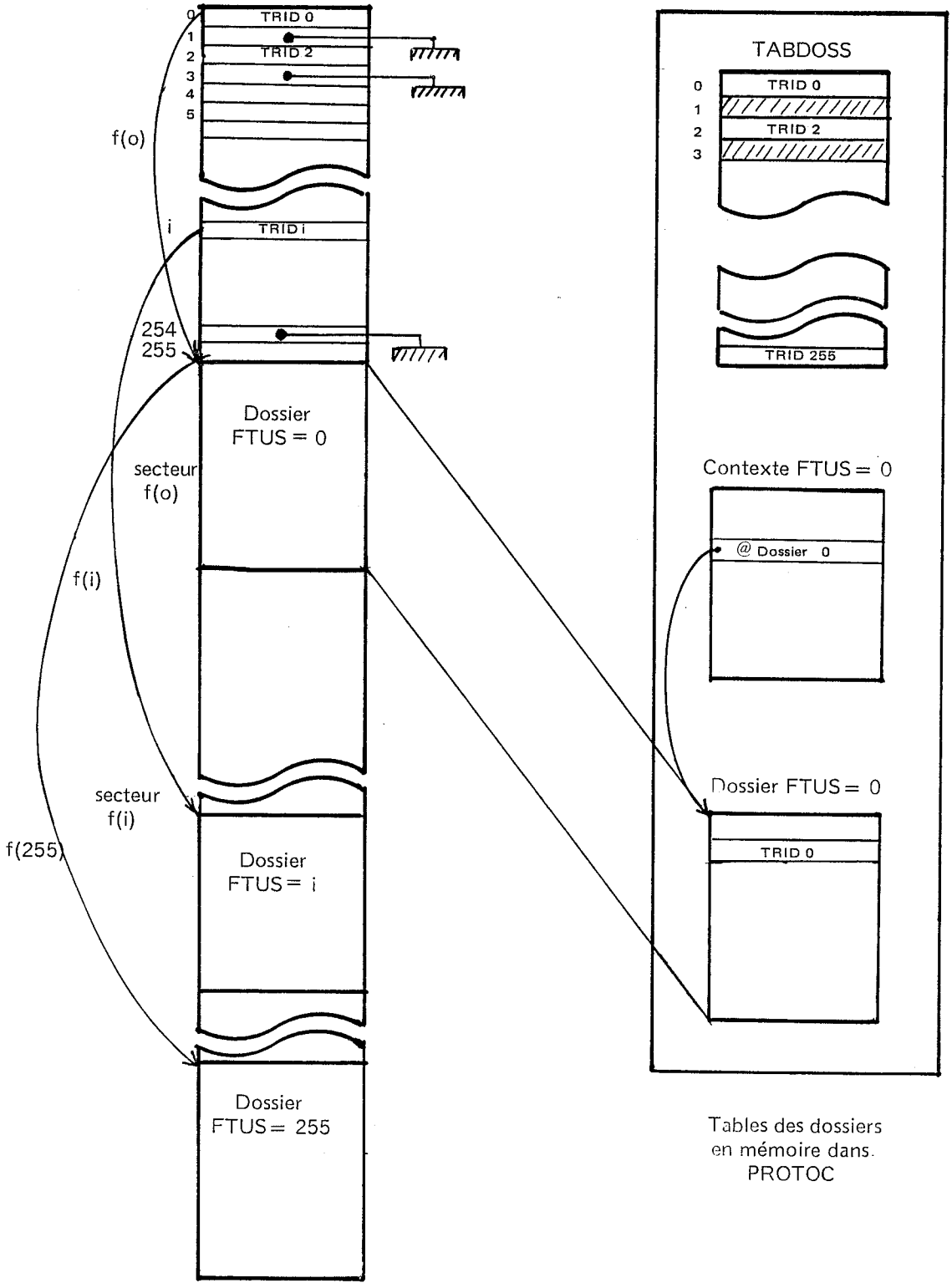
- écrire le dossier de transfert dans FDOSS.

5 - Supprimer un dossier de transfert dans FDOSS.

- supprimer le poste dans TABDOS (mise à NIL.)

- écrire TABDOS dans FDOSS.

Le schéma ci-après montre l'organisation du fichier dossier de transfert sur disque et de la table TABDOS.



Le fichier FDOSS
(sauvegarde des contextes)

Tables des dossiers
en mémoire dans
PROTOC

Fig 22 - Schéma de l'organisation des dossiers de transferts.

4.3.7.8 La Négociation initiale des attributs

1 - Les objectifs

Le but de la négociation initiale des attributs d'un transfert (Voir 2-4-1) est de déterminer la valeur des attributs NIFTP sur les sites P et Q à partir :

- des contraintes locales (P) et distantes (Q).

- des désirs de l'utilisateur initiateur du transfert.

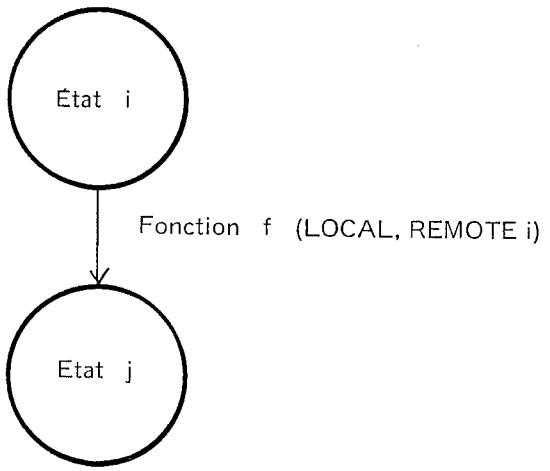
Au cours de la négociation, ces attributs vont être véhiculés dans les différentes commandes d'initialisation ou de terminaison sous forme de paramètres (voir Annexe 7-3) munis d'un opérateur de relation (qui pourra être LE, GE, EQU, ANY ou NE) afin de permettre à chacun des sites de préciser à l'autre ses contraintes minimales ou maximales.

Afin de mener à bien la négociation, chaque site va, pour un transfert donné, conserver en permanence dans deux tables appelées LOCAL et REMOTE, l'image des attributs locaux et distants négociés , valeurs associées à leurs contraintes (bornes inférieures ou supérieures),l'objectif de la négociation initiale étant d'obtenir sur chaque site des tables LOCAL et REMOTE en parfait accord sur le plan de la valeur des attributs et des contraintes.

Pour ce faire ces tables vont être modifiées au cours des différentes étapes de la négociation, matérialisées par l'échange des commandes d'initialisation, en appliquant des "fonctions de négociation" sur les éléments les constituant.

De fait, la négociation initiale va se schématiser par un graphe, chaque état du graphe correspondant à un état stable des tables LOCAL et REMOTE d'un site obtenu à partir de l'état précédent en appliquant une fonction de négociation sur une de ces tables (ou les deux).

Fig23Q04900



Etat i = état des tables LOCAL,REMOTE A l'instant i.
Etat j = état des tables LOCAL,REMOTE a l'instant j.

Fig 23 - Fonctions et états de négociation

Dans le paragraphe suivant nous allons décrire le graphe de la négociation initiale implanté dans PROTOC mais auparavant il est nécessaire de donner une description des fonctions de négociation ainsi que des tables utilisées.

2 - Implantation des tables de négociation dans PROTOC.

PROTOC va utiliser trois types de tables de négociation implantées de façons différentes.

- Une table d'initialisation LOCINI spécifique à l'implémentation servant à initialiser les tables LOCAL et REMOTE et contenant pour chaque attribut:
  - sa valeur par défaut (si elle existe),
  - l'ensemble de ses valeurs possibles,
  - ses contraintes minimales et maximales sur l'implémentation.
- La table LOCAL commune à tous les transferts qui va servir de table de travail intermédiaire réinitialisée à chaque étape de la négociation à partir soit de LOCINI soit de la table REMOTE associée au transfert.
- Les tables REMOTE spécifiques à chaque transfert faisant partie des dossiers de transferts et sauvegardées sur disque avec le dossier de transfert à chaque étape de la négociation.

Les tables LOCAL et REMOTE vont être constituées par des éléments correspondant chacun à un attribut défini dans le protocole et structuré comme un paramètre d'une commande Init/term (Voir annexe 7-3) et renfermant le numéro d'attribut, le "qualifier" ainsi que sa valeur. Le nombre des éléments de ces tables sera égal au nombre d'attributs NIFTP (voir annexe 7-1).

FGI24Q04900

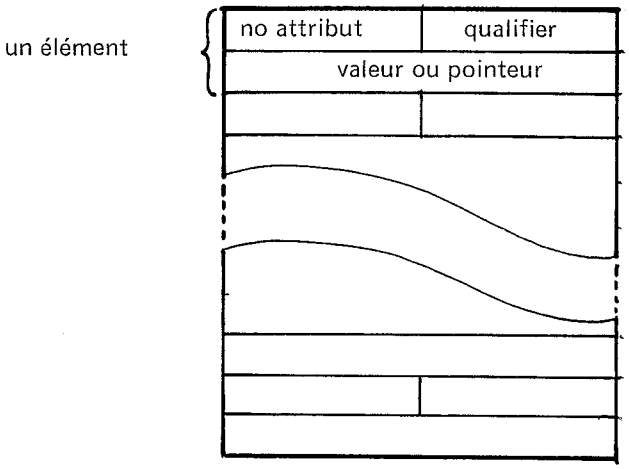


Fig 24 - Détail d'une table de négociation LOCAL et REMOTE

La zone valeur de chaque élément contiendra soit la valeur de l'attribut (cas des attributs de format "bitfield" ou "integer") soit un pointeur sur cette valeur à la suite de la table (attributs "string") ceci afin d'utiliser des tables de négociation de longueur fixe.

Ce type de structure va permettre d'effectuer facilement des opération de comparaisons entre les éléments des tables LOCAL et REMOTE pris deux à deux, ces tables étant strictement parallèles et chaque attribut de même nature y occupant le même emplacement.

Le schéma de l'implémentation de ces tables dans PROTOC est le suivant.

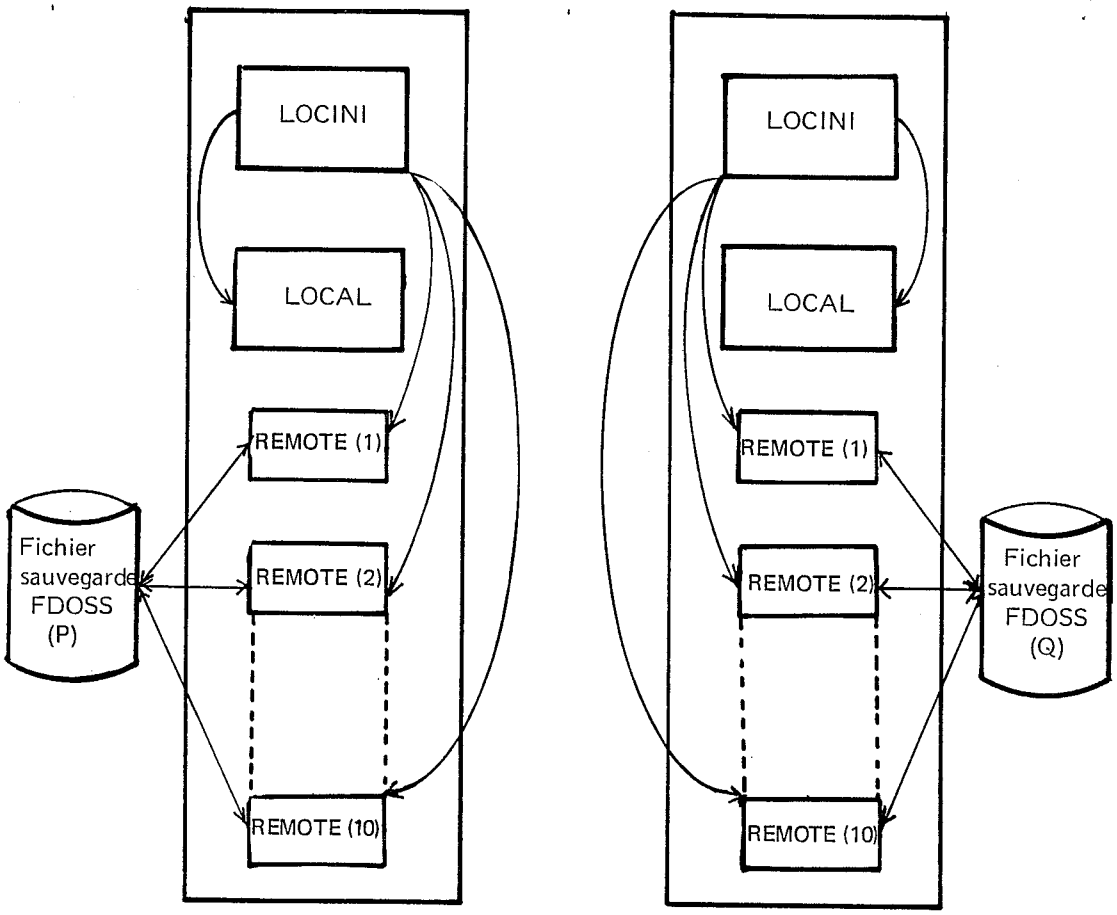


Fig 24- Les tables de la négociation initiale dans PROTOC .

3 - Les fonctions utilisées pour la négociation initiale

Afin de mener à bien la négociation il est nécessaire de se définir des fonctions de négociation élémentaires qui seront appliquées sur les paramètres reçus et sur les différentes tables précédemment décrites. La négociation initiale se résumant par la suite en un graphe d'appel à ces fonctions.

Les principales fonctions utilisées pour la négociation des attributs sont les suivantes :

A - INITAB : initialisation des tables de négociation. Au début de la négociation les tables LOCAL, REMOTE sont initialisées avec les valeurs par défaut des attributs sur chacune des implémentations, la table LOCAL contenant en plus les contraintes locales du site et la table REMOTE reflétant l'absence de contraintes distantes connues.

B - MODLOC,MODREM : Modification d'un élément d'une table. L'élément correspondant aux paramétre reçus dans une commande va être modifié dans la table en question (LOCAL ou REMOTE).

C - VERPAR : Vérification syntaxique d'un paramètre. Vérification syntaxique d'un paramètre d'une commande reçue dans la phase d'initialisation en fonction des règles données dans le protocole NIFTP (règles de syntaxe, domaine de validité...).

D - CHECK : Test de compatibilité entre deux paramètres de même nature. Vérifier que deux attributs satisfont aux contraintes locales, sont compatibles entre eux et dans ce cas déterminer un paramètre résultat affecté de l'opérateur EQU.

E - MAJLOC : Effectuer la fonction CHECK entre tous les paramètres des tables LOCAL et REMOTE d'un site et la mise à jour de la table LOCAL avec les paramètres résultats.

F - MAJREM : Comparer deux à deux les paramètres des tables LOCAL et REMOTE d'un site et modifier le REMOTE avec les paramètres du LOCAL lorsque ceux-ci sont "différents" (contrainte ou valeur) de ceux du REMOTE. Dans ce cas l'attribut modifié est rajouté s'il y a lieu dans la commande en préparation (SFT, RPOS).

G - INTERD : vérifier l'interdépendance des paramètres dans le REMOTE en fonction de règles indiquées dans le protocole NIFTP (certains attributs pouvant avoir une influence sur la valeur d'autres attributs).

4 Le graphe de la négociation initiale

Le graphe de la négociation initiale va s'inscrire dans le cadre des échanges de commandes de la phase d'initialisation qui, dans le cas ou elle aboutit, est le suivant :

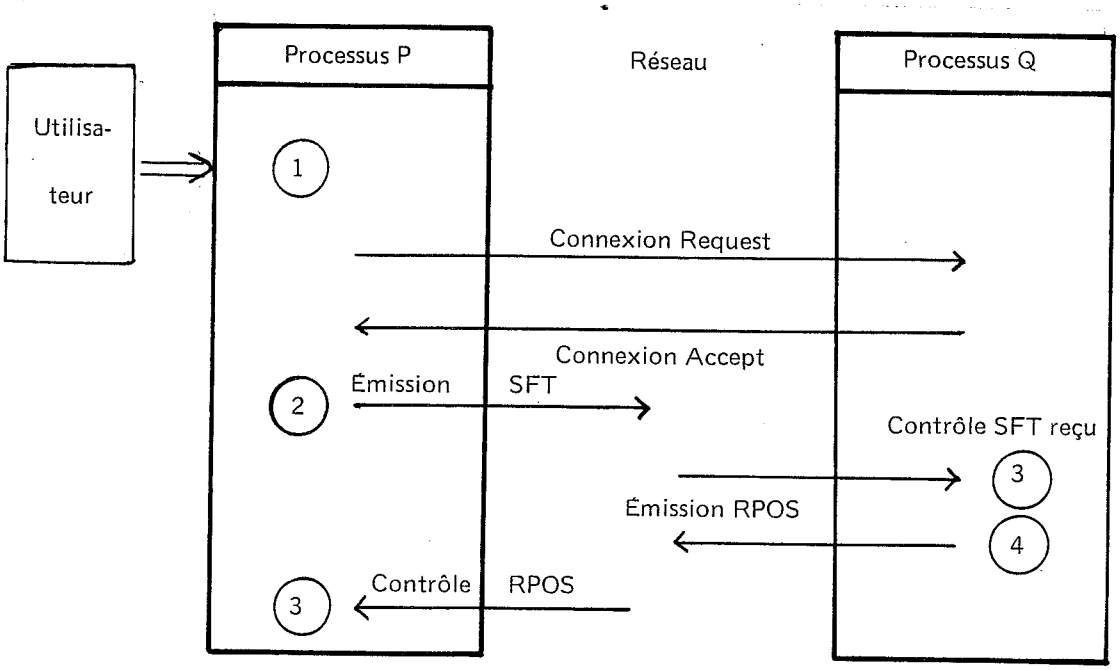


Fig 25 - Les étapes de la phase initialisation

Les différentes étapes de la négociation qui apparaissent dans ce schéma sont :

étape 1 : réception de la commande de lancement de transfert par P et contrôle des paramètres.

Etape 2 : élaboration de la commande SFT par P et émission.

Etape 3 : contrôle SFT reçu par Q.

Etape 4 : élaboration commande RPOS par Q et émission.

Etape 5 : contrôle RPOS reçu par P.

Compte-tenu des remarques précédentes, le graphe de négociation peut être représenté de la façon suivante :

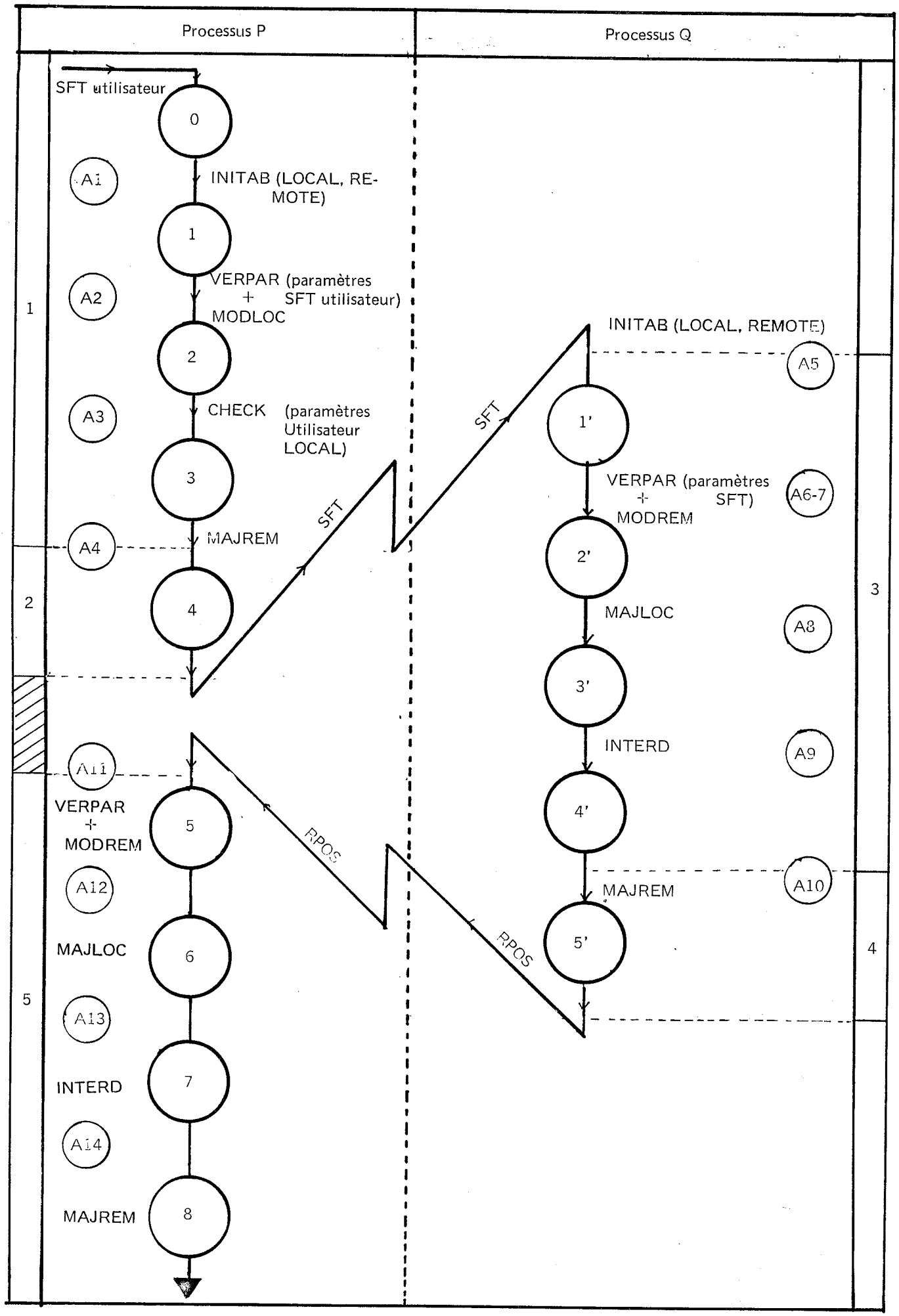


Fig 26 - Le graphe de la négociation initiale.

- Dans ce graphe nous retrouvons les étapes données dans la figure 25

Etape 1 : Réception SFT utilisateur et contrôle par P actions A1, A2, A3.

Etape 2 : Elaboration commande SFT par P actions A4.

Etape 3 : Contrôle SFT reçu par Q actions A5, A6, A7, A8, A9.

Etape 4 : Elaboration RPOS par Q action A10.

Etape 5 : Contrôle RPOS reçu par P action A11, A12, A13, A14.

La négociation initiale suppose qu'à la fin de la négociation les tables REMOTE (P) et REMOTE (Q) soient strictement équivalentes.

L'exemple suivant très simplifié, montre que cette méthode conduit bien à ce résultat.

Par souci de clarté les attributs seront au nombre de 4 (A, B, C, D) et les opérateurs associés limités à 3 (ANY, LE et EQU). Leur type sera supposé être "T" et leur format "integer".

5 - Exemple de négociation schématisée.

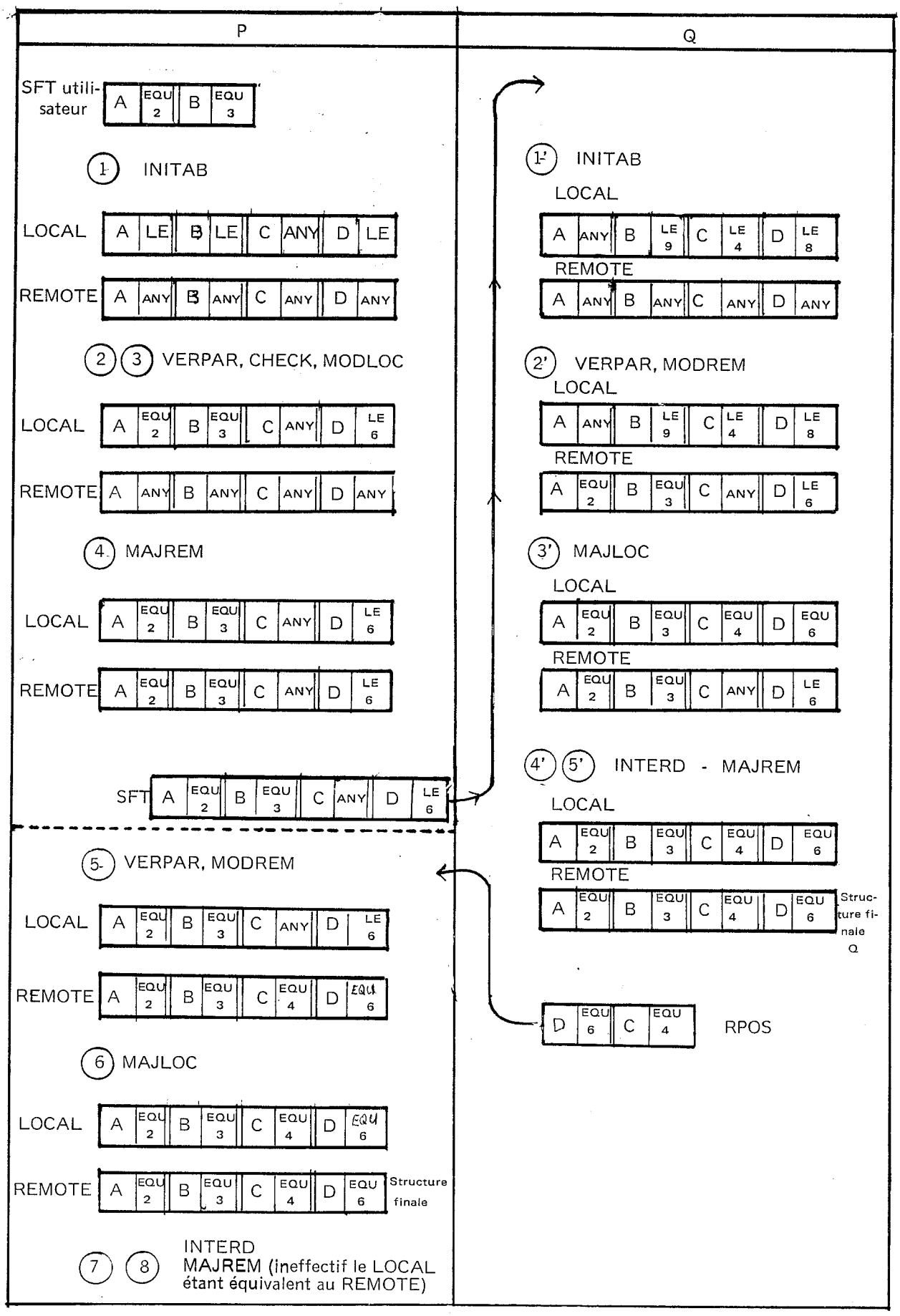


Fig 27 - Exemple de négociation initiale.

NB : A la fin de la négociation les deux structure REMOTE (P) est REMOTE (Q) sont bien équivalentes.

4.3.7.9 Gestion des marques

PROTOC va gérer les marques en collaboration avec les modules GESLEC et GESECR conformément au mécanisme décrit dans le protocole NIFTP (Voir 2-4-2-2).

Outre les attributs "marque initiale de reprise" et "fenêtre maximum d'acquittement des marques" négociés dans la phase initialisation par PROTOC, ces modules ont à leur disposition pour chaque transfert quatre compteurs de marques sauvegardés dans les dossiers de transferts:

- deux compteurs "émission"
  - LMS : numéro de la dernière marque émise.
  - LMAS : numéro de la dernière marque acquittée en émission.
- deux compteurs "réception".
  - LMR : numéro de la dernière marque reçue.
  - LMAR : numéro de la dernière marque acquittée en réception.

Ces quatre compteurs sont initialisés avec la valeur de l'attribut "marque initiale de reprise" après la phase de négociation.

La matérialisation des marques au niveau protocole pendant la phase donnée sera effectuée à l'aide des commandes MS et MR (voir 2-4-2-1).

Alors qu'un numéro de marque tient sur un mot de seize bits, l'argument d'une commande de contrôle relative à la gestion des marques est limité à un octet (voir annexe 7-3). Cet argument transmis sera constitué par l'octet de poids faible du numéro de marque à émettre (numéro de marque réel modulo 256).

Nous donnons ci-après en exemple le traitement " protocole" d'une marque et de son acquittement effectué par les modules GESLEC, PROTOC et GESECR.

Le traitement annexe effectué par GESLEC et GESECR (gestion d'une table des marques associant les marques et les adresses de reprises disques dans les fichiers) a déjà été évoqué dans les paragraphes 4-5-4 et 4-5-5.

Coté émetteur

GESLEC élabore la marque à poser par PROTOC (voir paragraphes 4-5-4) pour un transfert donné. Puis il met à jour le compteur LMS de la façon suivante :

LMS = LMS + 1

Et émet un ICB "marque à poser" vers PROTOC contenant LMS en paramètre.

Ensuite il vérifie que la fenêtre des marques n'est pas fermée à l'aide de la formule:

LMS+1 >= LMAS + fenetre d'acquittement des marques.

Si c'est le cas GESLEC interromp la lecture des enregistrements (bit LECT de la table ACTLEC à 0(voir 4-4-2-1)).

Sur réception de l'ICB "marque à poser", PROTOC émet une commande MS ayant comme argument, le numéro de marque contenu dans l'ICB modulo 256.

- Coté récepteur

La réception de la commande MS par PROTOC va donner lieu à la mise à jour du numéro de marque reçue:

LMR = LMR + 1

Et transmission à GESECR d'un ICBR "marque à poser" ayant comme paramètre LMR. GESECR va traiter cet élément (voir 4-5-5) et répondre à PROTOC par un ICB "OK pour l'acquittement" avec comme paramètre le numéro de marque MRK à acquitter.

Sur réception de cet ICB PROTOC met à jour LMAS (dernière marque acquittée en réception):

LMAS = MRK reçu de GESECR

Et émet une commande MR ayant comme argument LMAS module 256.

- Coté émetteur

Lorsque PROTOC reçoit la commande MR il va mettre à jour LMAS (dernière marque acquittée en émission) de la façon suivante :

LMAS = LMAS + ((argument (MR) - LMAS) mod 256)

Cette formule permettant de traiter le cas ou plusieurs marques sont acquittées en même temps.

Si GESLEC était en attente de fenêtre de marque, PROTOC réveille ce module en positionnant le bit LECT correspondant dans la table ACTLEC.

La gestion des marques associé à celle des dossiers de transferts, va être à la base du mécanisme de reprise que nous avons décrit par ailleurs (2-4-2).

4.3.7.10 Illustration de la gestion d'un transfert complet par PROTOC

Dans les chapitres précédents nous avons décrit les principaux mécanismes utilisés par PROTOC pour gérer les transferts.

Nous donnons ci-après le schéma simplifié des différentes opérations effectuées par PROTOC lors du déroulement d'un transfert complet (sans possibilité de reprise) dans le cas ou P est émetteur. On retrouvera dans ce schéma:

- les différentes phases du transfert (initialisation, données, terminaison) .
- les différentes étapes de la négociation initiale évoquées précédemment dans la phase initialisation (voir Fig 25 et 26).

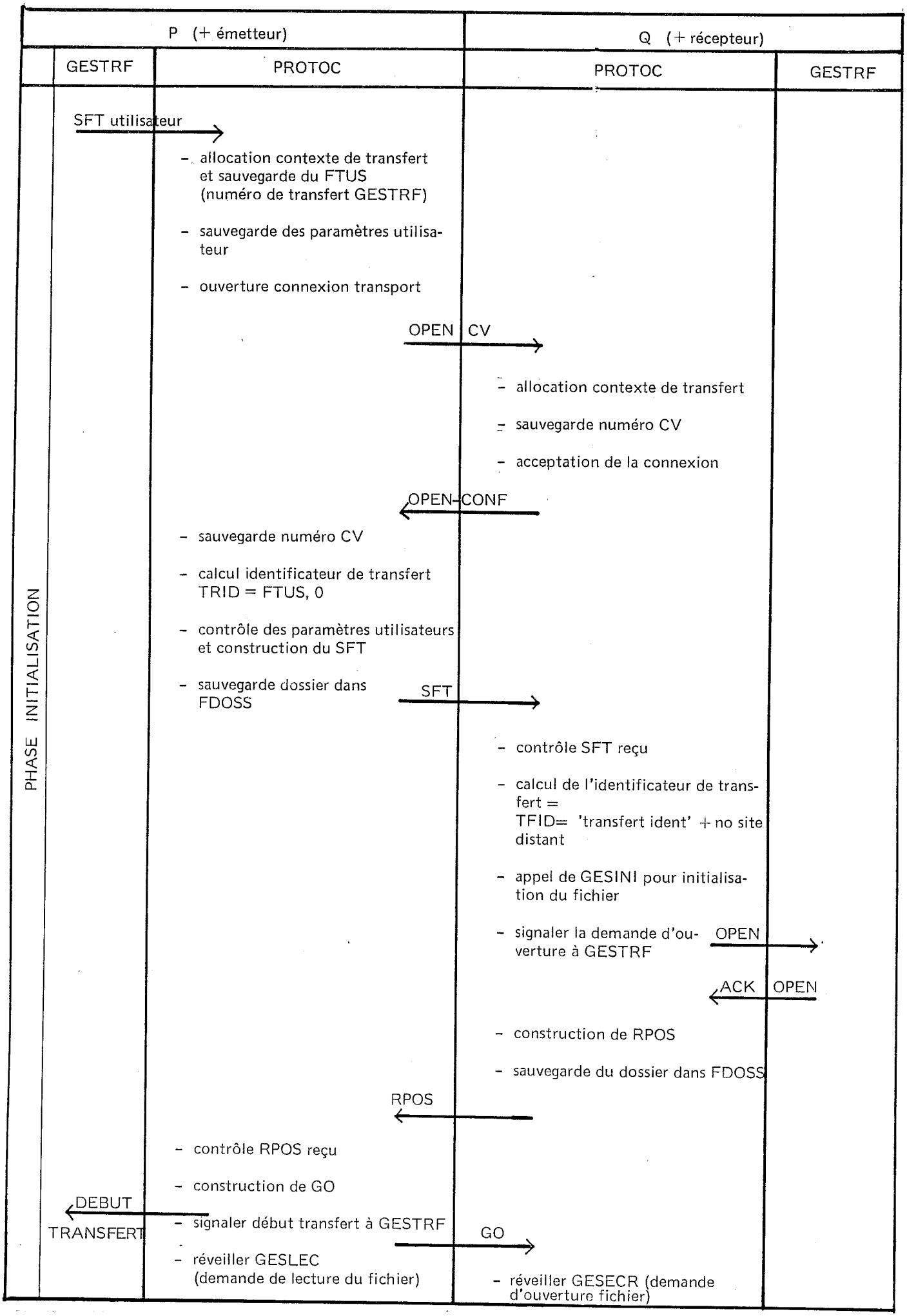


Fig 29 - Schéma de la gestion d'un transfert complet par PROTOC.

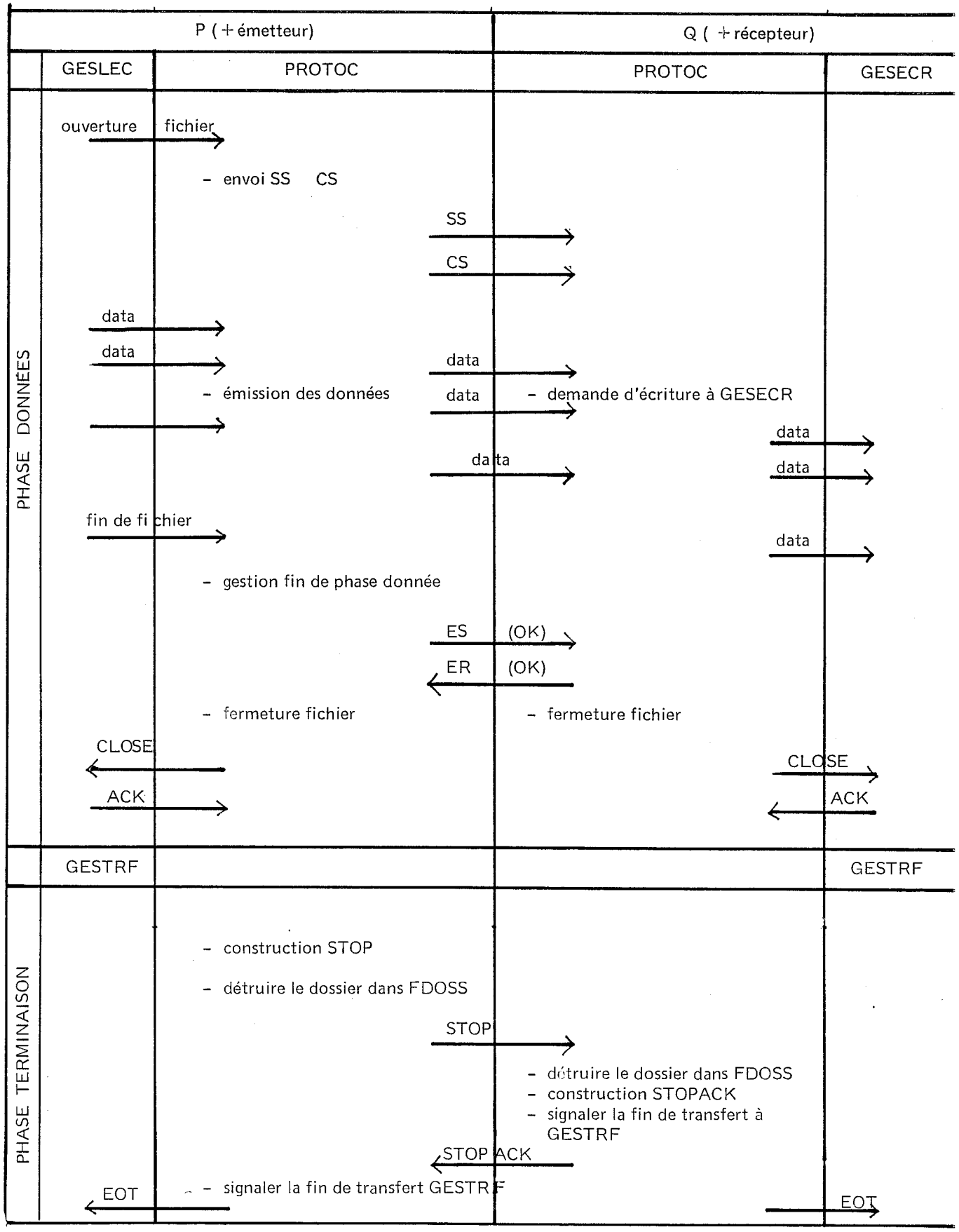


Fig 29-bis Schéma de la gestion d'un transfert complet par PROTOC.

5 REALISATION DU PROJET FTS-NIFTP LOT2

5.1 LES OBJECTIFS

Le projet LOT 2 a eu pour objectif de transformer le produit FTS-NIFTP LOT 1 qui est un service de transfert de fichier en service de transfert et de manipulation de travaux.

Pour ce faire les fonctionnalités suivantes ont été rajoutées à celles du LOT 1.

Ces fonctionnalités se divisent en quatre types d'opérations qui vont être chacune initialisées depuis le site "Agent" vers le site "Serveur" :

a) Récupération d'un fichier chez le Serveur et exécution en tant que job chez l'Agent.

B) Récupération d'une liste d'exécution chez le Serveur et impression chez l'Agent.

C) Soumission d'un fichier au Serveur, exécution de ce fichier en tant que job chez celui-ci, récupération de la liste d'exécution chez le Serveur et impression chez l'Agent.

D) Soumission d'une liste d'exécution au Serveur et impression chez celui-ci.

La version LOT 2 du produit FTS-NIFTP va utiliser les fonctionnalités du LOT 1 concernant l'émission et la réception de fichiers ainsi que les services offerts par le moniteur MMT2 pour le traitement des trains de travaux et en particulier les processeurs BATCH2 (batch), SPOOL2 et SUB2 (Soumission), dont nous donnons une description en annexe 7-9.

5.2 DESCRIPTION DES NOUVELLES FONCTIONNALITES.

Les quatre opérations décrites en 5-1 vont être chacune initialisées par l'utilisateur côté agent à travers le processeur NIFTP à l'aide d'une nouvelle commande qui va gérer les modes d'accès suivants.

- Give Job Input : opération a)
- Give Job Output : opération b) et opération c) (2ème partie)
- Take Job Input : opération c) (1ère partie)
- Take Job Output : opération d) .

Chacune de ces quatre opérations peut se définir en termes de modes d'accès 'READ' (lecture) ou 'MAKE'(écriture) du STEP 1 côté processus P ou Q auxquels on associe des fonctions de soumission au SPOOL ou au BATCH côté Agent ou Serveur.

Nous en donnons ci-après une description suivant ce principe.

A) Récupération d'un fichier chez le serveur et exécution en tant que job chez l'Agent.

Coté Agent :

- Initialisation de l'opération "GIVE JOB INPUT".
- Récupération du fichier (équivalent au mode P + READ).
- Soumission par SUB2 du fichier au BATCH local.

Coté Serveur :

- Emission du fichier vers l'Agent (équivalent au mode Q + READ).

B) Récupération d'une liste d'exécution chez le serveur et impression chez l'Agent.

Coté Agent :

- Initialisation de l'opération "GIVE JOB OUTPUT".
- récupération du fichier (équivalent au mode Q + READ).
- Soumission par SUB2 du fichier au SPOOL.

Coté Serveur :

- Emission du fichier vers l'Agent (équivalent au mode Q + READ).

C) Soumission d'un fichier au Serveur, exécution de ce fichier en tant que job chez celui-ci, récupération de la liste d'exécution chez le Serveur et impression chez l'Agent.

L'opération va se dérouler en une seule phase, le serveur émettant la liste résultat automatiquement une fois le job exécuté.

Coté Agent :

- Initialisation de l'opération 'TAKE JOB INPUT'.
- Emission du fichier vers le Serveur (équivalent au mode P + MAKE).
- .
- .
- .
- .
- Réception du fichier résultat (équivalent au mode P+MAKE).
- Soumission du fichier résultat au SPOOL.

Coté Serveur :

- Réception du fichier (équivalent au mode Q + MAKE).
- Soumission du fichier au BATCH local par SUB2.
- .
- .
- .
- Initialisation de l'opération 'TAKE JOB OUTPUT'.
- Emission du fichier.

D) Soumission d'une liste à imprimer au Serveur et impression chez celui-ci.

Coté Agent :

- Initialisation de l'opération 'TAKE JOB OUPUT'
- Emission du fichier (équivalent au mode P + MAKE).

Coté Serveur :

- Réception du fichier (équivalent au mode Q + MAKE).
- Soumission du fichier au SPOOL pour impression.

5.3 IMPLEMENTATION SUR MITRA .

L'architecture du produit FTS-NIFTP ne va pas être remise en cause par l'implémentation des nouvelles fonctionnalités. Seules des modifications vont être apportées aux modules principaux.

Elles vont être de deux types :

1. Modifications liées à la gestion des attributs et concernant :

   - la prise en compte de ces attributs au niveau des différentes interfaces (commandes ou macro-instructions).

   - La négociation initiale de ces attributs.

   - Le traitement de ces attributs une fois la négociation effectuée.

2. Modifications liées à la soumission de fichiers au SPOOL ou au BATCH .

   Ces opérations vont être réalisées à l'aide du service de soumission SUB2 du système MMT2,conformément au mécanisme précédemment décrit.

   Le produit SUB2 se présente comme un processeur. Pour les besoins du LOT 2, SUB2 va être modifié de façon à pouvoir être utilisé comme une tâche de service. La soumission d'un fichier au BATCH ou au SPOOL étant une opération "longue", elle ne sera pas assurée par le module PROTOC (pour éviter de bloquer les autres transferts en cours) mais par le module GESINI qui en plus des fonctions d'initialisation assurera donc des fonctions de soumission.

   Lorsque PROTOC désirera faire une soumission ,il sollicitera GESINI par un ICB et attendra que celui-ci le prévienne de sa bonne ou mauvaise exécution avant de continuer le transfert.

Le produit FTS-NIFTP LOT 2 en offrant un service de gestion de train de travaux à distance tout en conservant les possibilités d'être utilisé comme un service normal de transfert de fichier permettra de résoudre la plupart des problèmes concernant l'échange de fichiers dans un réseau de calculateurs hétérogènes et en particulier celui de la répartition des fonctions de traitement de l'information qui constitue une des exigences de l'informatique actuelle.

A ce stade de l'étude nous avons décrit l'implémentation du système FTS-NIFTP sur matériel MITRA dans son ensemble : architecture globale, interfaces et modules principaux. Nous allons maintenant exposer brièvement la méthode utilisée pour effectuer la qualification du produit.

5.4 LA QUALIFICATION DU PRODUIT FTS-NIFTP

Compte-tenu des nombreuses possibilités du protocole NIFTP (nombre élevé d'attributs utilisés, variétés des mécanismes supportés) et de son contexte d'utilisation (réseau hétérogène), un effort particulier a été réalisé pour la qualification dont la durée a représenté environ 50 % de la durée totale du projet.

Cette qualification s'est déroulée en deux étapes successives subdivisées elles-même en différentes phases :

- L'étape de qualification dans un contexte homogène d'ordinateurs MITRA.

- L'étape de qualification dans un contexte d'ordinateurs hétérogènes.

La qualification en milieu homogène.

Elle s'est déroulée en trois phases :

a) tests sous le "debugger" système d'un système libre-service MITRA des différents modules du produit .

Ces tests ont permis d'éliminer les erreurs 'grossières' de programmation.

B) tests en rebouclé sur une machine de test .

Pour effectuer ces tests, j'ai réalisé un module miroir simulant la couche transport (ouverture, fermeture des CV, émission réception de données, gestion des acquittements).
Cet outil implémenté à la place de la couche transport, a permis d'effectuer des tests en rebouclé sur une seule machine de test en utilisant toutes les fonctionnalités du produit sans passer par le réseau avec tous les avantages que cela implique (gain en coût, rapidité de mise en oeuvre, économie de machines utilisées).

C) Tests en réel en milieu homogène .

Ces tests ont été de même nature que les tests précédant mais effectués en réel entre deux sites MITRA reliés par le réseau TRANSPAC.
Au cours de ces tests toutes les possibilités de l'implémentation ont été vérifiées (nombre maximum de transfert simultanés, rupture de CV, gestion des reprises, etc...).

La qualification en milieu hétérogène

Les tests en milieu hétérogène ont été divisés en deux étapes :

a) test en utilisant un outil de validation du PROTOCOLE NIFTP .

Cet outil implémenté sur un site de la COMMUNAUTE EUROPEENNE à ISPRA en ITALIE est accessible via TRANSPAC et EURONET pour la FRANCE et permet à un utilisateur connecté via une console PAD de créer des scénarios de tests du protocole NIFTP à distance, permettant de simuler tous les évènements des automates NIFTP.

Une fois ses scénarios crées, l'utilisateur initialise le système NIFTP sur son site et lance l'exécution du scénario de test qui prendra l'initiative du transfert si le MITRA a été déclaré de type Q ou qui attendra que l'opérateur MITRA lance le transfert si celui-ci est de type P.

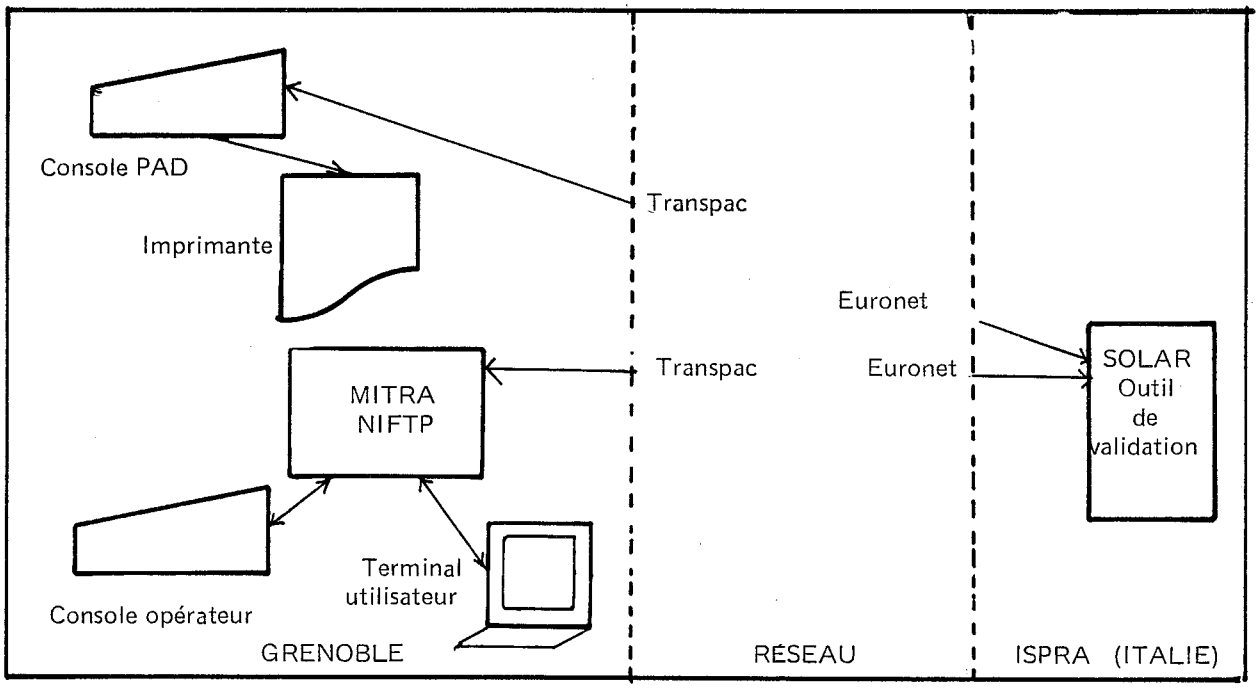


Fig 30 - utilisation de l'outil de validation du protocole NIFTP.

Ces tests ont permis de vérifier le bon fonctionnement du protocole en milieu hétérogène avant d'effectuer la dernière étape de la qualification dans un contexte d'ordinateurs hétérogènes.
Tous les constructeurs (ICL, SIEMENS, SEMS) ont dû, avant de passer aux tests finaux d'intégration, soumettre leur implémentation à des tests de ce type avec le même jeu de scénarios, ce qui a permis d'avoir des produits de même niveau de finition avant d'entreprendre cette phase finale.

B) test en liaison avec un site NIFTP non MITRA .

Ce site utilisé était un SIEMENS situé en BELGIQUE et accédé via TRANSPAC et DCS.
Les tests déroulés ont été identiques à ceux de la phase c) des tests homogènes.

L'ensemble des tests réalisés à permis non seulement d'obtenir un produit de bonne qualité sur le plan du fonctionnement général mais aussi d'améliorer sensiblement les performances initiales en permettant d'établir les valeurs optimales des paramètres de génération du système (valeurs des seuils entre les modules GESLEC-PROTOC et PROTOC-GESECR, fenêtre d'acquittement des marques, fenêtre PROTOC-TRANSPORT) assurant le meilleur compromis entre l'encombrement mémoire et les vitesses de transfert.

6 CONCLUSION.

Après l'étude du protocole NIFTP et de son implémentation sur matériel MITRA il est difficile d'apporter une conclusion définitive sur les mérites d'un tel protocole vis à vis des autres protocoles existants.

La question qui se pose est plutôt de savoir si ce produit a rempli les objectifs pour lequel il a été conçu.

Il est utile de rappeler que le protocole NIFTP a été developpé en marge des instances de normalisation en vue de pallier à une carence réelle dans le domaine des protocoles de transfert de fichier en milieu hétérogène, afin de répondre à la demande d'un grand nombre d'utilisateurs de réseaux publics ou privés.

Sur ce point il a atteint son but, la meilleure preuve étant le nombre d'implémentations réalisées à base de ce protocole sur le matériel de différents constructeurs (IBM, SIEMENS, ICL, SEMS ,etc...) pour le compte de nombreux clients sur le plan européen et en particulier la CCE.

L'inconvénient majeur vient en fait de son architecture qui en le plaçant au-dessus de la couche transport rend par la même, impossible l'implémentation directe du protocole NIFTP sur un système de communication basé sur le modèle OSI, contrairement aux protocoles de transfert de fichiers ISO et ECMA qui sont implémentés dans les couches application et présentation .

Une implémentation NIFTP n'est qu'un système de transfert de fichier alors qu'une implémentation ISO ou ECMA fournit un système de communication avec une application particulière de transfert de fichier.

En ce qui concerne les autres caractéristiques, souplesse d'utilisation, efficacité, sécurité, variétés dans les paramètres de transfert le protocole NIFTP soutient très bien la comparaison avec les autres protocoles avec sans doute un net avantage dans le domaine des performances, dû à sa structure et au format des données échangées.

De plus la possibilité d'implémenter le protocole NIFTP avec des degrés de sophistication différents a pour effet de rendre ces implémentations possibles sur des machines allant du microprocesseur aux plus puissants calculateurs existants.

Enfin la deuxième partie du projet, en fournissant un service de transfert et de manipulations de travaux a permis d'accroitre largement le domaine d'utilisation de ce système.

Néanmoins, il est certain que si le protocole NIFTP a influencé les travaux de l'ISO et l'ECMA pour la définition d'un standard de transfert de fichier en milieu hétérogène, ceux-ci reprenant les mêmes concepts mais implémentés de facon différente, dans un avenir plus ou moins proche comme pour les systèmes de communication, les protocoles définis par ces instances lui seront préférés, essentiellement pour des raisons de standardisation et non pour des raisons de performance ou de qualité des services rendus.

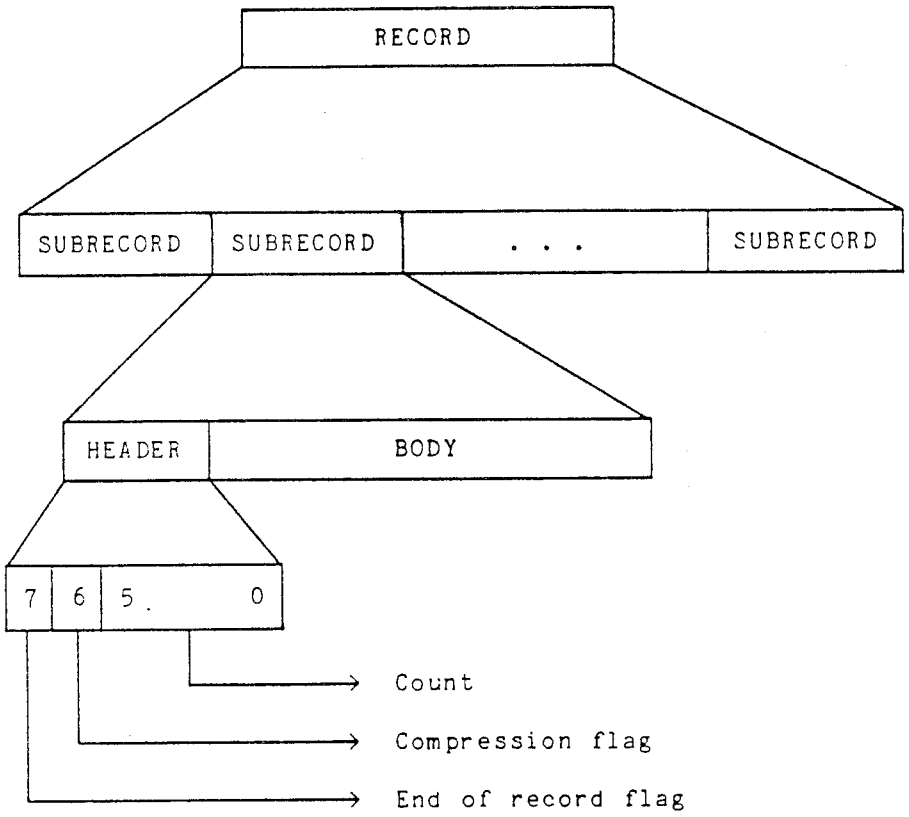
En ce qui me concerne, ce projet m'a apporté, outre une plus grande connaissance des problèmes posés par la communication entre matériels hétérogènes, une expérience enrichissante sur le plan des contacts humains (travail en équipe,négociations avec les clients, dialogue avec d'autres constructeurs) et de la conduite d'un projet pendant la deuxième partie de l'affaire.

7 ANNEXES

7.1 LES ATTRIBUTS NIFTP

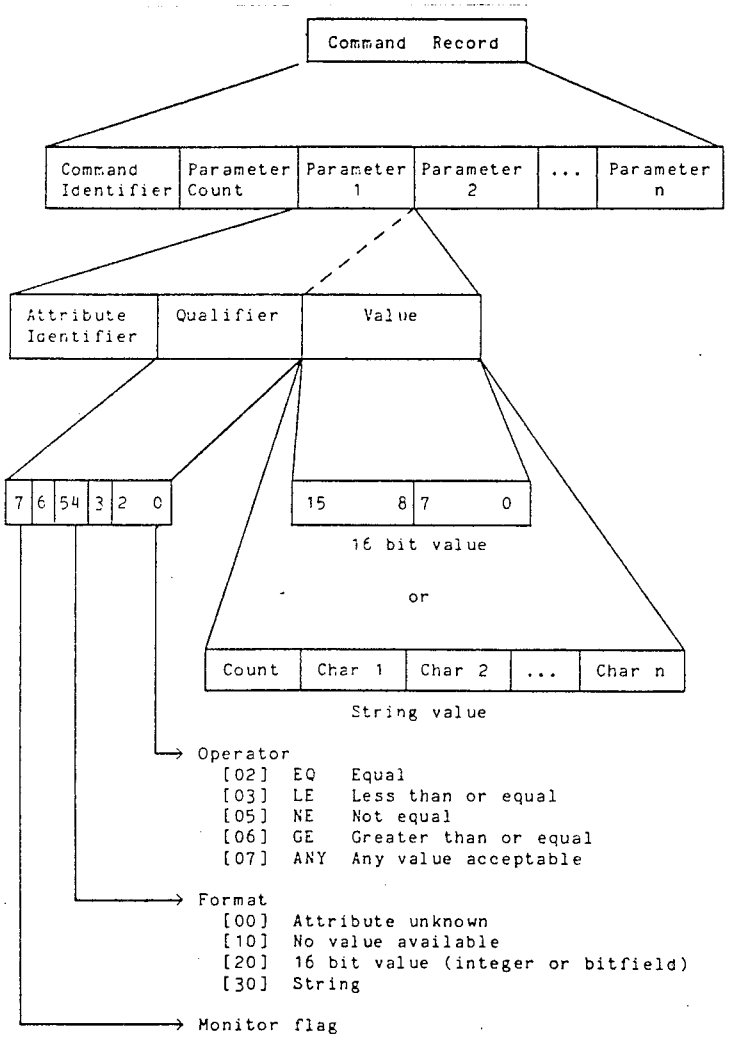
Id	Attribute Name	Default value	Format	Type	Section
00	Protocol Identification	—	bitfield	I	7.1
01	Mode of Access	none	bitfield	T	7.2
02	Text Transfer Code	[0001]	bitfield	T	7.3
03	Text Formatting	[0001]	bitfield	T	7.4
04	Binary Format	[8002]	bitfield	T	7.5
05	Maximum Transfer Record Size	[FFFF]	integer	T	7.6
06	Transmission Limit	[FFFF]	integer	T	7.7
07	Data Estimate	none	integer	I	7.8
08	Transfer Identifier	none	integer	T	7.9
09	Private Transfer Code Name	""	string	T	7.10
0A	Acknowledgement Window	[00FF]	integer	T	7.11
0B	Initial Restart Mark	[0000]	integer	T	7.12
0D	Minimum Timeout	[0258]	integer	T	7.13
0E	Facilities	[0000]	bitfield	T	7.14
0F	State of Transfer	--	bitfield	I	7.15
20	Data Type	[0001]	bitfield	T	7.16
21	Delimiter Preservation	[0000]	bitfield	T	7.17
22	Text Storage Code	none	bitfield	Q	7.18
23	Horizontal Tabs	"X"	string	T	7.19
24	Binary Word Size	[0008]	integer	T	7.20
25	Maximum Storage Record Size	none	integer	Q	7.21
26	Page Width	none	integer	T	7.22
27	Page Length	none	integer	T	7.23
29	Private Storage Code Name	""	string	Q	7.24
40	Filename	none	string	Q	7.25
42	Username	none	string	Q	7.26
44	Username Password	none	string	Q	7.27
45	File Password	none	string	Q	7.28
4A	Account	none	string	Q	7.29
4B	Account Password	none	string	Q	7.30
50	Output Device Type	LP	string	Q	7.31
51	Device Type Qualifier	""	string	Q	7.32
60	File Size	none	integer	Q	7.33
70	Action Message	—	string	I	7.34
71	Information Message	--	string	I	7.35
80	Special Options	--	any	any	7.36

7.2 LES FORMATS DES ENREGISTREMENTS NIFTP



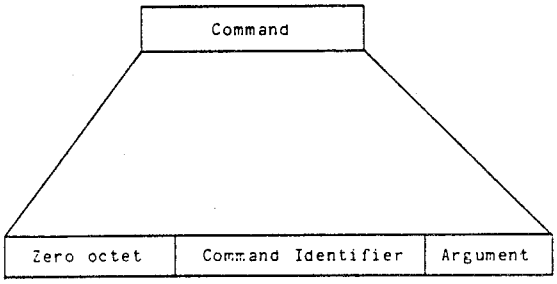
7.3 LES COMMANDES NIFTP

Les commandes d'Initialisation et de Terminaison



Command (Hex)	Purpose	Name	Used by	Section
00	Request Termination	STOP	P	5.5
01	Start Data Transfer	GO	P	5.4
02	Positive Reply	RPOS	Q	5.2
03	Negative Reply	RNEG	Q	5.3
04	Start File Transfer	SFT	P	5.1
05	Acknowledge Termination	STOPACK	Q	5.6

Les commandes de contrôle du transfert.



Command (Hex)	Description	Argument	Section
40	SS - Start of Data	mark number	6.1
41	MS - Mark Point	mark number	6.2
42	CS - Code Select	new code	6.3
43	ES - End of Data	status code	6.4
44	RR - Restart Request	mark number	6.5
45	MR - Mark Acknowledge	mark number	6.6
46	QR - Quit	status code	6.7
47	ER - End Acknowledge	status code	6.8

7.4 EXEMPLE D'UTILISATION DE LA "NOTIONAL CHARACTER MATRIX".

L'utilisation de la 'Notional Character Matrix' va dépendre des trois attributs NIFTP :

- largeur de page PW.
- tabulation horizontale T.
- longueur de page.

Nous allons voir sur un exemple l'utisation de cette matrice .

Soit:

- PW = 30 caractères la valeur de l'attribut "largeur de page".
- Ch = ' X ' la valeur de l'attribut "tabulation horizontale".
- n = 6 la longueur de la chaine ch.
- k = 4 la position du premier caractère non blanc dans ch, numérotés de 1 à n.

On va considérer qu'il y a un caractère de fin de tabulation pour chaque caractère dans la ligne de position p = (n*x)+k pour 1 =<p =<pw et x >=0

Soit dans le cas présent:

P = (6*x) + 4 pour 1 =<p =<30 et x >=0

Si x = 0 p = 4

Si x = 1 p = 10

Si x = 2 p = 16

Si x = 3 p = 22

Si x = 4 p = 28

La rencontre d'un caractère de tabulation va avoir pour effet d'inclure des espaces dans la ligne en construction jusqu'à la position du prochain caractère de tabulation déterminé.

Le résultat de ce traitement appliqué à l'enregistrement initial est le suivant:

Enregistrement initial : 'ab "tab" cdef "tab" gh'

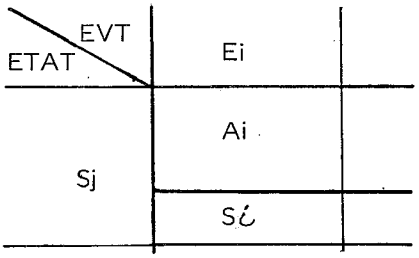
Enregistrement final : 'ab c def gh'

7.5 DESCRIPTION DES AUTOMATES DU PROTOCOLE NIFTP

Ces automates vont décrire les actions à effectuer en fonction des évènements survenus au cours des différentes phases (initialisation, transfert de données et terminaison) d'un transfert de fichier.

Dans la phase transfert de données la distinction est faite entre les évènements "normaux" et "anormaux" (commande invalide, time-out etc...).

Dans ces automates la notation suivante est utilisée :



Avec

E = Evénement

S = Etat courant

A = Action à exécuter

S_{ℓ} = Etat suivant

Les différentes commandes utilisées dans les trois phases se retrouvent dans les noms d'évènements d'états et d'actions utilisés dans ces automates (ex : STOPACK, STOP...).

- Phase Initialisation : Automate INITP (côté P)

Event / State	RPOS OK	RPOS NOK	RPOS EOF	RNEG	Inv Com	TS reset	Start FTP	TS open	End	Timed out	P error
IDLE	*	*	*	*	- IDLE	- -	- -	- OPEN	- -	- -	- -
OPEN	*	*	*	*	CLOSE IDLE	- OPEN	SFT SFT	- -	CLOSE IDLE	- -	CLOSE IDLE
SFT	GO GO	STOPr STOP	STOPt STOP	STOPr STOP	CLOSE IDLE	CLOSE IDLE	- -	- -	CLOSE IDLE	CLOSE IDLE	CLOSE IDLE

- Phase Initialisation : Automate INITQ (côté Q)

Event / State	SFT OK	SFT NOK	GO	STOP	Inv Com	TS reset	TS open	Timed out	Q error
IDLE	*	*	*	*	- IDLE	- -	- OPEN	- -	- -
OPEN	RPOS RPOS	RNEG RNEG	*	*	CLOSE IDLE	CLOSE IDLE	- -	CLOSE IDLE	CLOSE IDLE
RPOS	*	*	- GO	STOP-ACK OPEN	CLOSE IDLE	CLOSE IDLE	- -	CLOSE IDLE	CLOSE IDLE
RNEG	*	*	*	STOP-ACK OPEN	CLOSE IDLE	CLOSE IDLE	- -	CLOSE IDLE	CLOSE IDLE

- Phase terminaison : Automate TERMP (côté P)

Event / State	STOP-ACK	Inv Com	TS reset	Entry	Timed out	P Error
OK	-	-	-	STOPt	-	STOPa
	-	-	-	STOP	-	STOP
FAIL	-	-	-	STOPa	-	CLOSE
	-	-	-	STOP	-	IDLE
STOP	-	CLOSE	CLOSE	-	CLOSE	CLOSE
	OPEN	IDLE	IDLE	-	IDLE	IDLE

- Phase terminaison : Automate TERMQ (côté Q)

Event / State	STOP	Inv Com	TS reset	Timed out	Q Error
OK	STOPACK	CLOSE	CLOSE	CLOSE	CLOSE
	OPEN	IDLE	IDLE	IDLE	IDLE
FAIL	STOPACK	CLOSE	CLOSE	CLOSE	CLOSE
	OPEN	IDLE	IDLE	IDLE	IDLE

- Phase transfert de donnée : automate SENDER

Event / State	ER (OK)	ER (E)	QR (E)	QR (A)	Inv Com	TS reset	Timed out	S error
GO	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	- -	ESe ESe
DATA	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	- -	ESe ESe
RR	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	- -	ESe ESe
HOLD	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	ESa FAIL	ESe ESe
HCRR	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	ESa FAIL	ESe ESe
MR	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	ESa FAIL	ESe ESe
ESok	- OK	*	ESe ESe	- FAIL	ESe ESe	- WAIT	ESa FAIL	ESe ESe
WAIT	*	*	ESe ESe	- FAIL	ESe ESe	- WAIT	ESa FAIL	ESe ESe
ESe	- OK	- FAIL	- ESe	- FAIL	- ESe	- WAIT	ESa FAIL	- ESe

Event / State	MR	RR	ER (H)	QR (OK)	QR (H)	Data End	Mark Pnt	Rest Pnt	Ack W End	Data OK	Set Code
GO	*	*	*	SS+ ESok ESok	SS+ ESh HOLD	- - -	SS DATA	- -	- -	- -	- -
DATA	AWT DATA	RWL RR	*	ESok ESok	ESh HOLD	ESok ESok	MS+ AWL DATA	- -	- MR	DATA DATA	CS DATA
RR	AWT RR	*	*	*	ESh HCRR	- -	- -	SS DATA	- -	- -	- -
HOLD	AWT HOLD	RWL HCRR	- DATA	- HOLD	- HOLD	- -	- -	- -	- -	- HOLD	- -
HCRR	AWT HCRR	*	- RR	*	- HCRR	- -	- -	- HCRR	- -	- -	- -
MR	AWT DATA	RWL RR	*	ESok ESok	ESh HOLD	- -	- MR	- -	- -	- MR	- -
ESok	AWT ESok	RWL RR	*	- ESok	ESh HOLD	- -	- -	- -	- -	- -	- -
WAIT	AWT WAIT	RWL RR	*	ESok ESok	ESh HOLD	- -	- -	- -	- -	- WAIT	- -

- Phase transfert de données : automate RECEIVER

Event / State	SS	MS	CS	ES (OK)	ES (H)	Data	OK for Ack	Hold up	End hold up	End OK	Data NOK
GO	- DATA	*	*	*	*	*	- -	QRh PERR	- -	QRok QRok	- -
DATA	*	AWL DATA	Save DATA	- ESok	- DATA	Keep DATA	MR DATA	QRh PEND	- -	QRok QRok	RR RR
RR	RWL DATA	- RR	- RR	- RR	- RR	- RR	MR RR	QRh PERR	- -	- -	- -
PEND	*	AWL PEND	Save PEND	- PEND	- HOLD	Keep PEND	MR PEND	- -	ERh DATA	ERh QRok QRok	RR PERR
HOLD	*	*	*	*	- HOLD	*	MR HOLD	- -	ERh DATA	ERh QRok QRok	RR HOPR
PERR	RWL PEND	- PERR	- PERR	- PERR	- HORR	- PERR	MR PERR	- -	ERh RR	- -	- -
HORR	*	*	*	*	- HORR	*	MR HORR	- -	ERh RR	- -	- -
ESok	*	*	*	- ESok	- ESok	*	MR ESok	QRh PEND	- -	ERok OK	RR RR
QRok	- QRok	- QRok	- QRok	ERok OK	- QRok	- QRok	- QRok	- QRok	- -	- -	- -

Event / State	ES (E)	ES (A)	Inv Com	TS reset	Timed out	R error
GO	ERe FAIL	- FAIL	QRe QRe	RR RR	QRa FAIL	QRe QRe
DATA	ERe FAIL	- FAIL	QRe QRe	MR RR RR	QRa FAIL	QRe QRe
RR	ERe FAIL	- FAIL	QRe QRe	MR RR RR	QRa FAIL	QRe QRe
PEND	ERe FAIL	- FAIL	QRe QRe	MR RR RR	QRa FAIL	QRe QRe
HOLD	ERe FAIL	- FAIL	QRe QRe	MR QRh PEND	- -	QRe QRe
PERR	ERe FAIL	- FAIL	QRe QRe	MR RR RR	QRa FAIL	QRe QRe
HORR	ERe FAIL	- FAIL	QRe QRe	MR RR RR	- -	QRe QRe
ESok	ERe FAIL	- FAIL	QRe QRe	QRok QRok	- -	QRe QRe
QRok	ERe FAIL	- FAIL	QRe QRe	QRok QRok	QRa FAIL	QRe QRe
QRe	ERe FAIL	- FAIL	- QRe	QRe QRe	QRa FAIL	- QRe

7.6 LES JOURNAUX DE TRANSFERTS

Il existe deux types de journaux associés aux transferts sur un site :

- un journal opérateur de site contenant des messages concernant les transferts de type Q et deux destinés à l'opérateur de site,
- des journaux utilisateur contenant les messages associés aux différents utilisateurs de type P.

En dehors de ces journaux, les listes de la console opérateur et des utilisateurs fournissent l'historique des commandes effectuées.

Nous donnons ci-après un exemple des journaux de transfert et des listes opérateur et utilisateurs associées.

Dans cet exemple nous avons utilisé une configuration de test en rebouclé (voir 5-5) schématisée de la façon suivante:

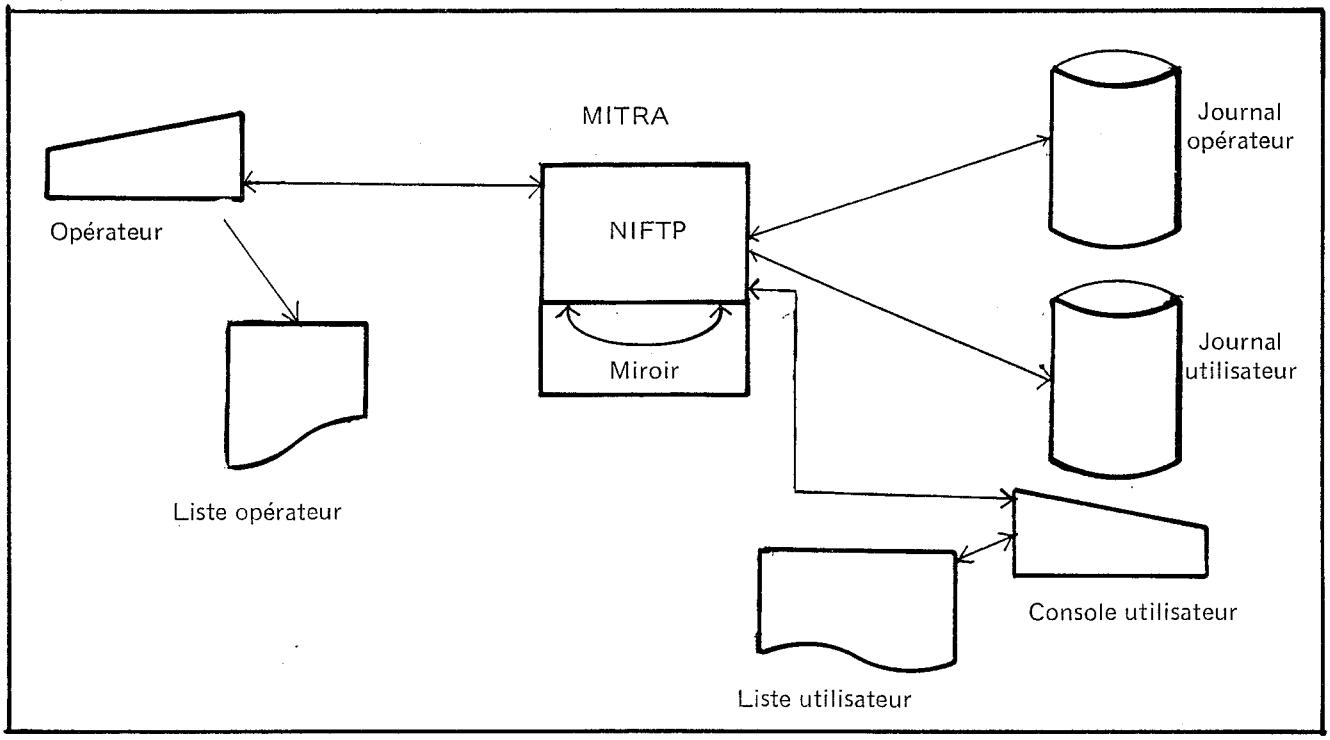


Schéma de la configuration de test rebouclée

Les commandes suivantes ont été effectuées:

1 - lancement d'un transfert par un utilisateur.

2 - demande d' arrêt du transfert avec possibilité de reprise.

3 - demande de reprise du transfert.

4 - lancement d'un autre transfert et arrêt du système.

A) Liste console opérateur

```
*
*/CM/ROSCS/PFN=MESCNIF,GENE=4,PVN=NIFTP, ---> lancement NIFTP
**/TKN = 0010
*
*/MS/EDIT,,LP00,,LP.
*
==/SWITCH TERMINATED
=
==/GROUP &04 CONNECTION : "S"
S
SS/ 1.17 SI:  2  TNS:GTP1  CV:00FF  LM:KEPR  E   RAISON:0700
SS/ 1.17 SI:  2  LN:LN00 CONNECTEE:21000000
SS/ 1.17 SI:  2  SCS ACTIF
SS/ 1.17 SI:  0  CV:&E003  SIGNAL DU REMOTE HORS ECHANGE

SS/ 1.18 SI:  0  CV:&0006  OUVERT

S/H^..
S/AFT/FT01/REP,..
S/DFT/ALL, ----------> état des transferts
S
SS/ 1.18 SI:  2  COMMANDE  DFT TERMINEE
SS/SI:  2  ,USER,IDEN,PRO,SEN,  FNAME  ,PRI,SITE,JJ,MM,AA,HH,MI,SE,STAT,
SS/SI:  2  01  FT01  P  S  HETERUP3  00  02  1/ 1/ 1  1H18'10"TRAN
SS/SI:  2  00  FT02  Q  R  HETERUP3  00  02  1/ 1/ 1  1H18'20"TRAN

S/AFT/FT01/REP, ----------> arrêt du transfert avec
SS/ 1.19 SI:  0  CV:&0006  DECONNECTE      possibilité de reprise.
S
SS/ 1.19 SI:  2  COMMANDE  AFT TERMINEE

S/DFT/ALL,
S
SS/ 1.19 SI:  2  COMMANDE  DFT TERMINEE
SS/SI:  2  ,USER,IDEN,PRO,SEN,  FNAME  ,PRI,SITE,JJ,MM,AA,HH,MI,SE,STAT,
SS/SI:  2  01  FT01  P  S  HETERUP3  00  02  1/ 1/ 1  1H18'10"WRES
SS/SI:  2  00  FT02  Q  R  HETERUP3  00  02  1/ 1/ 1  1H18'20"WRES

S/RFT/FT01, ----------> reprise du transfert
SS/ 1.19 SI:  0  CV:&0006  OUVERT
S
SS/ 1.19 SI:  2  COMMANDE  RFT TERMINEE

S/DFT/ALL,
S
SS/ 1.19 SI:  2  COMMANDE  DFT TERMINEE
SS/SI:  2  ,USER,IDEN,PRO,SEN,  FNAME  ,PRI,SITE,JJ,MM,AA,HH,MI,SE,STAT,
SS/SI:  2  01  FT01  P  S  HETERUP3  00  02  1/ 1/ 1  1H18'10"TRAN
SS/SI:  2  00  FT02  Q  R  HETERUP3  00  02  1/ 1/ 1  1H18'20"TRAN
SS/ 1.20 SI:  0  CV:&0006  DECONNECTE

S/DFT/ALL,
S
SS/ 1.20 SI:  2  PAS DE TRANSFERT CONNU

SS/ 1.20 SI:  0  CV:&0006  OUVERT

S/HSI, ----------> arrêt du système après la fin des
SS/ 1.22 SI:  0  CV:&0006  DECONNECTE      transferts en cours.
SS/ 1.22 SI:  2  LN:LN00 FERMEE:21000000
SS/ 1.22 SI:  2  SCS DESACTIVE
```

B) Liste des commandes utilisateurs (console utilisateur)

```
 +PAR1/AM=RM,MSRS=80,FAC=1E,STCN=PR,TTC=PR.
 +PAR2/FORM=80.
 +PAR3/FILE=(HETEROF0DM04  DM),SIZE=20,USN=(SEMS),USNP=(C'SEMS'),ACN=(01).
 +PAR4/FILE=(HETEROF0NIFTP DM),RECL=80.
 +PAR5/INFO=TEST DU MODE REPLACE OR MAKE .
 +SFT/DEST=2.                                    lancement 1er transfert
     1  H  8FILE TRANSFER  FT01  SUBMITTED.
 +END.
0303 END NIFTP  V0.0   LEV. 0
 +PAR1/AM=RM,MSRS=80,FAC=1E,STCN=PR,TTC=PR.
 +PAR2/FORM=80.
 +PAR3/FILE=(HETEROF3DM04  DM),SIZE=20,USN=(SEMS),USNP=(C'SEMS'),ACN=(01).

 +PAR4/FILE=(HETEROF3NIFTP DM),RECL=80.
 +PAR5/INFO=TEST DU MODE REPLACE OR MAKE .
 +SFT/DEST=2.                                    lancement 2eme transfert
     1  H 18FILE TRANSFER  FT01  SUBMITTED.
 +END.
0303 END NIFTP  V0.0   LEV. 0
 +PAR1/AM=RM,MSRS=80,FAC=1E,STCN=PR,TTC=PR.
 +PAR2/FORM=80.
 +PAR3/FILE=(HETEROF4DM04  DM),SIZE=20,USN=(SEMS),USNP=(C'SEMS'),ACN=(01).
 +PAR4/FILE=(HETEROF4NIFTP DM),RECL=80.
 +PAR5/INFO=TEST DU MODE REPLACE OR MAKE .
 +SFT/DEST=2.                                    lancement 3eme transfert
     1  H 21FILE TRANSFER  FT01  SUBMITTED.
```

C) Liste du journal opérateur

```
 1    1  1  1  1  8 13   FT02 USER 00 STARTED                    debut 1er transfert
 2    1  1  1  1  8 17   FT02 USER 00 71 32 TESTDUMODEREPLACEORMAKE
 3    1  1  1  1  8 33   FT02 USER 00 TERMINATED
 4    1  1  1  1 18 20   FT02 USER 00 STARTED                    debut 2eme transfert
 5    1  1  1  1 18 24   FT02 USER 00 71 32 TESTDUMODEREPLACEORMAKE
 6    1  1  1  1 19 13   FT02 USER 00 71 32 ** INF004 ** CR= 0000 USE
 7   R'S AFT RECEIVED
 8    1  1  1  1 19 14   FT02 USER 00 ABORTED (RES.)              reception abort
 9    1  1  1  1 19 40   FT02 USER 00 RESTARTED BY USER
10    1  1  1  1 19 42   FT01 USER 01 RESTARTED BY OPER           restart
11    1  1  1  1 20  1   FT02 USER 00 TERMINATED                  fin de transfert
12    1  1  1  1 21  7   FT02 USER 00 STARTED                    debut 3eme transfert
13    1  1  1  1 21 11   FT02 USER 00 71 32 TESTDUMODEREPLACEORMAKE
14    1  1  1  1 22  5   FT02 USER 00 TERMINATED
```

D) Liste du journal utilisateur

```
 1    1  1  1  1  8  0   USER 01 CONNECTED.                      connexion utilisateur
 2    1  1  1  1  8  4   FT01 USER 01 NEGOCIATING.
 3    1  1  1  1  8  6   USER 01 DISCONNECTED.
 4    1  1  1  1  8 15   FT01 USER 01 STARTED
 5    1  1  1  1  8 31   FT01 USER 01 TERMINATED
 6    1  1  1  1 18  7   USER 01 CONNECTED.
 7    1  1  1  1 18 11   FT01 USER 01 NEGOCIATING.
 8    1  1  1  1 18 12   USER 01 DISCONNECTED.
 9    1  1  1  1 18 22   FT01 USER 01 STARTED
10    1  1  1  1 19  6   FT01 USER 01 ABORT ASKED BY OPER (RES).
11    1  1  1  1 19 10   FT01 USER 01 71 32 ** INF004 ** CR= 0000 USE
12   R'S AFT RECEIVED
13    1  1  1  1 19 11   FT01 USER 01 ABORTED (RES.)     (DB58).
14    1  1  1  1 19 37   FT01 USER 01 RESUMPTION ASKED BY OPER.
15    1  1  1  1 19 43   FT01 USER 01 RESTARTED BY OPER
16    1  1  1  1 20  0   FT01 USER 01 TERMINATED
17    1  1  1  1 20 54   USER 01 CONNECTED.
18    1  1  1  1 20 58   FT01 USER 01 NEGOCIATING.
19    1  1  1  1 21  0   USER 01 DISCONNECTED.
20    1  1  1  1 21  9   FT01 USER 01 STARTED
21    1  1  1  1 22  3   FT01 USER 01 TERMINATED
```

7.7 LE TRANSPORT ECMA-72 CLASSE 0

L'objectif de ce standard, défini par la CCITT est d'offrir la forme la plus simple de protocole de transport en vue d'une utilisation avec des réseaux à commutation par paquet.

Pour ce faire il supporte un minimum de facilités qui sont :

- l'établissement de connexion avec négociation des paramètres.
- le transfert de données avec segmentation et transmission des erreurs à l'utilisateur.

Par contre les facilités de multiplexage, de déconnexion, de contrôle de flux, de reprise sur erreur ne sont pas supportées.

Dans le cadre de NIFTP, ce transport est utilisé avec des réseaux à commutation par paquets (TRANSPAC, EURONET, DCS) qui eux assurent ces services.

7.8 DESCRIPTION DE MINI-ORDINATEUR MITRA

Le MITRA M 625 est un miniordinateur 16 bits, orienté vers les applications de gestion et de télécommunications, utilisant le logiciel d'exploitation MMT2 (moniteur multitâches multifonctions).

7.8.0.1 ARCHITECTURE

Le MITRA 625 possède trois bus spécialisés :

- Le bus mémoire (débit de 2,2 M octets/seconde) qui supporte les modules mémoires (jusqu'à 1024 K octets au total - technologie MOS).

- Le bus périphérique asynchrone (2,2 M octets/seconde) qui permet la connexion de la périphérie MITRA en accès direct à la mémoire.

- Le bus opérateur synchrone qui intègre au niveau de l'U.C. Les opérateurs hautes performances optionnels.

L'U.C. Utilise le microprocesseur 10800 de MOTOROLA, microprogrammable, travaillant avec 14 registres rapides.

Le MITRA 625 possède une anté-mémoire de type semi-associatif, synchrone avec l'U.C. (le cycle de l'U.C. Est divisé par 2,5 environ par rapport au cycle propre de la mémoire centrale).

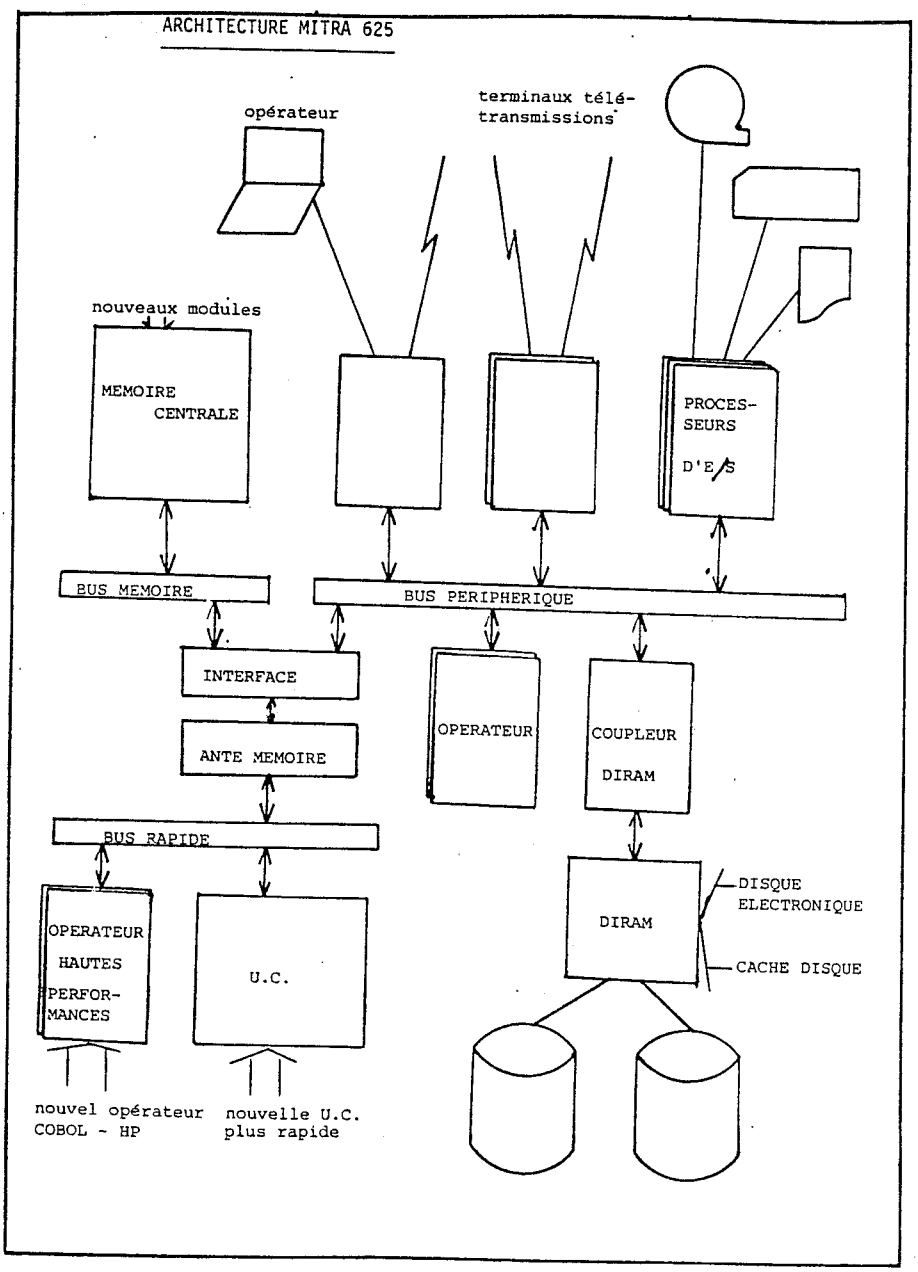
L'U.C. Est construite en technologie ECLA à base de MACRO CELLS.

De nouveaux modules mémoire plus intégrés sont proposés : 512 K octets ou 1024 K octets.

Un nouveau type de coupleur disque, le DIRAM, peut être raccordé sur le bus périphérique. Il permet d'accélérer l'accès aux données stockées sur disque grace à son anté mémoire incorporée et à son système de recherche associative de l'information.

La gestion des terminaux interactifs peut être déportée sur une machine spécialisée : le GT 25 gestionnaire de grappe de terminaux THEMIS.

Enfin, un nouvel opérateur hautes performances COBOL peut être raccordé au bus rapide.


ARCHITECTURE MITRA 625
opérateur
terminaux télé-transmissions
nouveaux modules
MEMOIRE CENTRALE
PROCES-SEURS D'E/S
BUS MEMOIRE
BUS PERIPHERIQUE
INTERFACE
ANTE MEMOIRE
OPERATEUR
COUPLEUR DIRAM
BUS RAPIDE
DISQUE ELECTRONIQUE
DIRAM
CACHE DISQUE
OPERATEUR HAUTES PERFOR-MANCES
U.C.
nouvel opérateur COBOL - HP
nouvelle U.C. plus rapide

LE CODE D'ORDRE

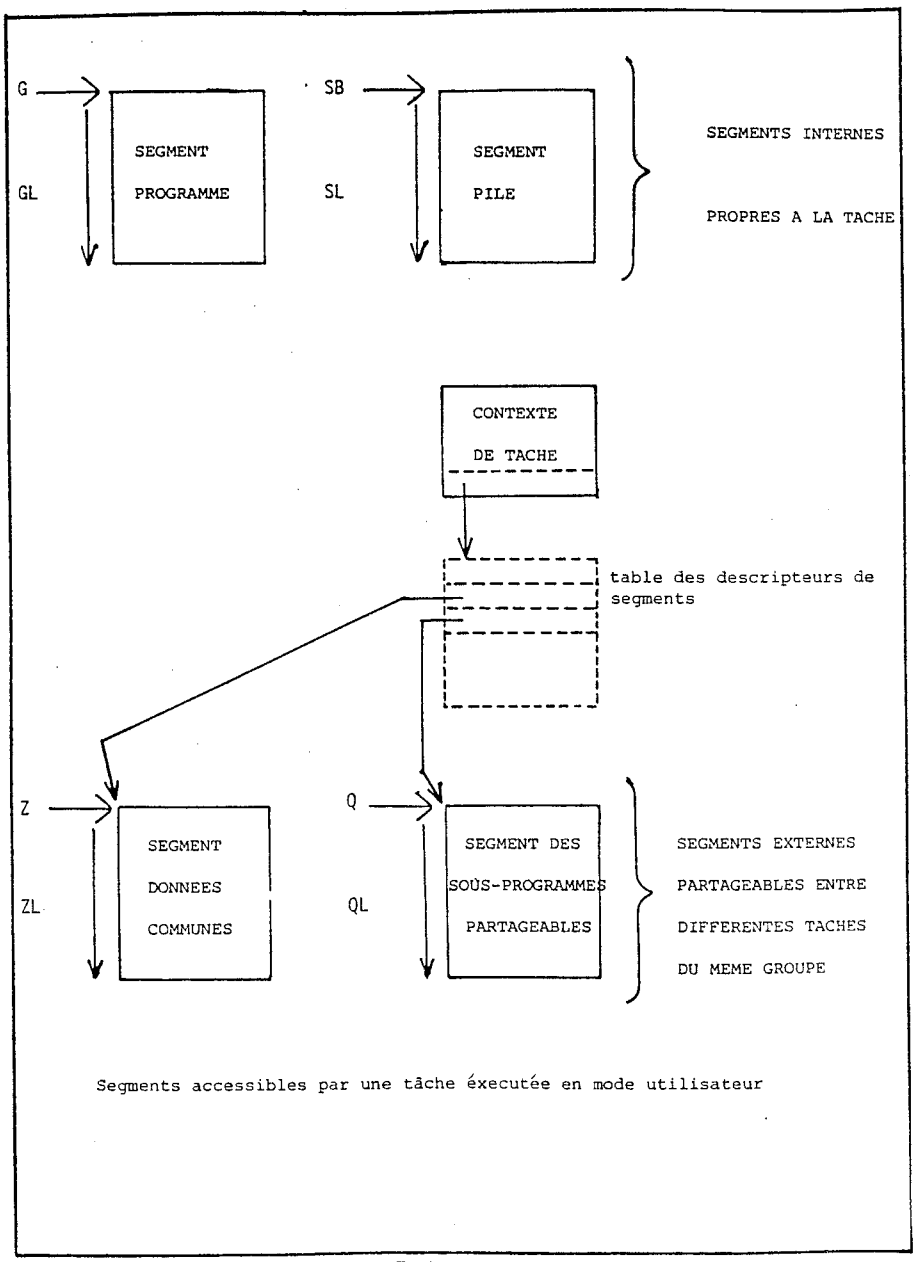
Les 152 instructions du code d'ordre donnent accès au bit, à l'octet, au mot, au double mot et à la chaîne d'octets.
Le code d'ordre s'appuie sur des possibilités d'adressage très développées:

- immédiat et immédiat indexé,
- relatif au pointeur ordinal (P) avant, arrière,
- relatif à la base locale (L) direct, indirect, indexé,
- relatif à la base générale (G) direct, indirect, indexé.

LA PROTECTION DES PROGRAMMES

La protection des programmes est assurée par les entités base et déplacements accessibles au travers des registres rapides (un couple de registres définissant une base et une taille).
A tout instant, tout programme peut accéder à

32 K mots pour le programme principal (base G)
32 K mots pour les sous-programmes communs (base Q)
32 K mots pour les données communes (base Z)



Segments accessibles par une tâche éxecutée en mode utilisateur

LE SYSTEME D'INTERRUPTIONS

Le système d'interruption comporte 63 niveaux, classés par priorité croissante.

Les priorités les moins importantes sont laissées pour les erreurs logicielles.

LES MODES DE COUPLAGE

Le MITRA offre deux modes d'échanges donnant accès aux périphériques :

- Le mode programmé,
- Le mode accès direct à la mémoire (D.M.A.) géré par les processeurs périphériques capables d'exécuter les commandes canaux préparées par l'U.C.

MEMOIRES AUXILIAIRES

Les mémoires auxiliaires du MITRA sont :

- Des unités disques de capacité 20,52 ou 53 M octets
- Des dispacks RDD 300 d'une capacité de 300 M octets
- Des dérouleurs de bandes magnétiques
  - . De densité 800 bpi, de format et de mode d'enregistrement conformes au standard ANSI, compatibles I.B.M.,
  - . Ou de densité 1600 bpi.

PERIPHERIQUES

Au MITRA, peuvent être connectés:

- deux types d'imprimantes (300 ou 600 lignes/minute),
- ainsi qu'une imprimante matricielle (100 caractères/seconde ou 132 colonnes),
- deux types de lecteurs de cartes (300 ou 600 cartes/minute) pour cartes de format 80 colonnes/12 bits,
- un module de surveillance de bon fonctionnement "chien de garde" constitué d'une autre U.C.

AUTOMATE X 25

C'est un processeur de télécommunications gérant le niveau 2 de la norme X 25, qui permet de gérer quatre interfaces V 24 (débit global maximum de 200 000 bps). Le raccordement à TRANSPAC est assuré soit direct en V 24, jusqu'à 19 200 BAUDS soit avec adaptateur en V 35 (au-delà de 19 200 bps).

DISPOSITIFS DE TELECHARGEMENT

Ce dispositif est implanté sur la console de service et permet d'amorcer une télécommande de l'ordinateur par l'intermédiaire d'une liaison de transmission (ligne synchrone), quel que soit l'état du système télécommandé.

TERMINAUX

Des terminaux de visualisation DT 15 peuvent être raccordés au MITRA pour les Entrées/Sorties d'information alphanumériques.

7.9 RAPPELS CONCERNANT LES SERVICES STANDARDS MMT2 UTILISES DANS LE LOT 2.

7.9.1 BATCH2.

BATCH2 est un exécutif chargé d'enchaîner des trains de travaux. Chaque travail, ou JOB, est défini puis soumis par un utilisateur et correspond à une des fonctions suivantes :

- catalogage d'une procédure.
- exécution d' un traitement.

L'exécutif BATCH2 s'intègre parfaitement dans la philosophie générale du système MMT2 en assurant une protection totale entre chaque travail; cette protection est obtenue par deux procédés différents et complémentaires :

- séparation des commandes et des données : les commandes sont exécutées par BATCH2, les données sont traitées par les PROCESSEURS ou les TACHES utilisateurs,
- l'exécutif BATCH2 met en oeuvre deux groupes : le groupe BATCH2 proprement dit et le groupe JOB qui est créé par une tache du groupe BATCH2. Les traitements relatifs à un job s'exécutent dans le groupe JOB et ne peuvent pas franchir les frontières du domaine qui lui a été attribué.

L'exécutif BATCH2 a la possibilité de fonctionner avec l'exécutif SPOOL2 afin d'optimiser l'acquisition et l'édition d'informations, sur des périphériques lents. L'exécution des programmes est asynchrone par rapport à la gestion des entrées-sorties.

L'utilisateur a la possibilité de choisir le catalogue, des fichiers spoolés vers lequel sera dirigée la liste produite par le job : SPOOL, USER ou REMOTE. Dans les deux derniers cas cette liste ne sera pas éditée par le SPOOL.

Plusieurs exécutifs BATCH avec leurs utilisateurs correspondants peuvent coexister sur la même machine.

7.9.2 SPOOL2 .

SPOOL2 est un exécutif chargé de réguler l'acquisition et l'édition d'informations, sur des périphériques lents.

Le SPOOL évite :

- que la vitesse des périphériques soit modulée par l'exécution des programmes,

- que l'exécution des programmes soit dépendante de la vitesse de ces périphériques.

On peut distinguer trois catégories de tâches dans le groupe SPOOL :

- Le SPOOLER : tache RACINE du groupe SPOOL :

  . Crée les symbionts et leur transmet des commandes,

  . Contrôle le dialogue avec l'opérateur.

- Les SYMBIONTS D'ENTREE :

  . Constituent les fichiers SPOOL IN sur disque à partir d'informations acquises sur périphérique lent.

- Les SYMBIONTS DE SORTIE :

  . Editent sur les périphériques lents les informations stockées dans les fichiers SPOOL OUT sur disque par des exécutifs d'exploitation.

On distingue quatre catégories de fichiers spoolés :

- les fichiers à destination d'un exécutif, type BATCH,

- les fichiers à destination d'un exécutif, type REMOTE,

- les fichiers à destination des périphériques de sortie,

- les fichiers à destination d'un exécutif, type USER.

Un fichier CATALOGUE est associé à chaque catégorie de fichier spoolé présent.

7.9.3 SUB2.

SUB2 est un processeur qui permet de :

- soumettre un fichier job au BATCH travaillant sous SPOOL,
- soumettre un fichier au SPOOL pour édition,
- soumettre un fichier au REMOTE,
- cataloguer une procédure,
- enregistrer des fichiers pour une destination "USER".

Seules les trois premières fonctionnalités seront utilisées dans le cadre du LOT2.

- SOUMISSION D'UN FICHIER JOB AU BATCH :

L'utilisateur peut définir la sous-destination BATCH (en cas de multi-batch) vers laquelle sera aiguillé son job.
Une commande JOB est élaborée en tête du fichier et éventuellement une commande LIMIT si l'utilisateur la spécifie. Ce fichier est ensuite envoyé au BATCH.

- SOUMISSION D'UN FICHIER AU SPOOL POUR EDITION :

Un utilisateur peut soumettre un fichier séquentiel alphanumérique pour édition, soit avec un octet de format, soit sans octet de format.
La destruction du fichier après édition peut être demandée par l'utilisateur.

- SOUMISSION AU REMOTE :

La tâche SUB2 enregistre le fichier à destination du REMOTE et il lui est soumis sans aucune transformation.

- CATALOGAGE DE PROCEDURES :

La tâche SUB2 permet d'exécuter une action sur les procédures cataloguées associées au numéro de compte de l'utilisateur (INC, REP, DEL, NEW, PAK).

- ENREGISTREMENT DES FICHIERS POUR UNE DESTINATION "USER" :

La tâche SUB2 enregistre le fichier à destination USER. L'utilisateur pourra récupérer ce fichier en exécutant une sélection pour la destination USER.

8 GLOSSAIRE NIFTP

- Attributs : Ensemble des caractéristiques définissant un fichier ou un transfert de fichier.

- Commandes de contrôle de transfert : Utilisées dans la phase donnée pour signaler les erreurs, reprises à chaud, suspensions, du transfert ainsi que sa terminaison.

- Commandes d'initialisation et de terminaison : utilisés dans la phase initialisation et terminaison d'un transfert pour envoyer des informations entre les processus "P" et "Q".

- Phase initialisation : Phase du transfert chargée d'établir l'identité du fichier à transférer et de définir la forme dans laquelle il doit être transféré.

- Phase donnée : phase dans laquelle les données contenues dans un fichier sont transférées de l'émetteur vers le récepteur.

- Phase terminaison : phase finale du transfert dans laquelle l'état final du transfert est transmis aux deux processus.

- Emetteur : Coté du transfert (P ou Q) depuis lequel les données sont émises pendant la phase donnée.

- Récepteur : Coté du transfert (P ou Q) vers lequel les données sont émises pendant la phase donnée.

- Négociation Initiale : Méthode suivie dans la phase initialisation afin d'obtenir un jeu de valeurs d'attributs acceptés par les processus P et Q pour un transfert donné.

- Paramètre : Les paramètres sont échangés dans les commandes d'initialisation et de terminaison. Un paramètre est constitué de trois parties (voir chapitre).

    - un numéro d'attribut.

    - un "qualifieur" : contenant le type, le format et un opérateur de négociation.

    - une partie optionnelle donnant la valeur de l'attribut.

- Dossier de transfert : Ensemble d'information sauvegardées par les processus P et Q afin de permettre la reprise d'un transfert interrompu .

- Processus P : Processus initiateur du transfert.

- Processus Q : Processus acceptant le transfert.

- "Record" NIFTP : unité logique des données échangées dans le protocole NIFTP de longueur quelconque et découpé en "subrecord" pour les besoins de la transmission.

- "Subrecord" NIFTP : unités de 64 octets maximum contenant un "header" de un octet et 63 octets d'information maximum. Un "record" est constitué par un ou plusieurs "subrecord".

- Fichier réel : nom d'une collection d'information possédant un ensemble de propriétés communes décrites de façon spécifique à un système donné.

- Fichier virtuel : Représentation standardisée d'un fichier en terme d'attributs de façon à le rendre "compréhensible" par des systèmes différents.

- Systèmes : au sens OSI un système est constitué par un ensemble d'un ou plusieurs ordinateurs du logiciel associé de périphériques, de terminaux, d'opérateurs humains, de moyens de transfert de l'information formant un tout autonome capable d'effectuer le traitement de l'information.

- Mapping : procédé permettant de transformer un fichier réel en fichier virtuel et inversement.

- Modèle de référence de l'OSI (objectif) : L'objectif du modèle de référence pour l'interconnexion des systèmes ouverts est de définir un ensemble de procédures standardisés qui vont permettre l'échange d'information entre des "systèmes" qui seront "ouverts" les uns aux autres du fait du support et de la reconnaissance mutuelle de ces procédures, sans préjuger d'une technologie particulière de ces systèmes ni de moyens d'interconnexion particuliers .

- I.S.O : Organisation Internationale de Standardisation : groupe 87 pays, le marché français étant l'AFNOR. Dans le domaine informatique ses travaux sont effectués par le comité TC97 (et le sous-comité 16 pour la partie communication).

- ECMA : association européenne des constructeurs de matériel informatique. Le but de cette association est d'établir des standards applicables à la conception fonctionnelle et à l'utilisation de matériels informatiques. Ses principaux adhérents sont CII-HB, Philips, Siemens, Burroughs, SEMS, IBM EUROPE, ICL.

9 TABLE DES FIGURES

10 BIBLIOGRAPHIE

- AFNOR - Interconnexion de Systèmes Ouverts

"Modèle de Référence"

PARIS, AFNOR, Avril 80.

- ECMA - Final Draft of "An Introduction to Open-System Interconnexion"

GENEVE, ECMA, 17/06/80.

- DATA COMMUNICATION PROTOCOLE UNIT

National Physical Laboratory

"A Network Independant File Transfer Protocol, FTP-B(80)"

Teddington - Middlesex (ENGLAND), 5/2/81.

- R.W.S. Hale Division of Numerical Analysis and Computer Science

National Physical Laboratory

"File Transfer Protocols - Comparison and Critique"

Teddington - Middlesex (ENGLAND), Juillet 81.

- ECMA.

"Standard ECMA-85, VIRTUAL FILE PROTOCOL"

GENEVE, ECMA 7, 8 juillet 82.

- ISO - SC16

"The File Service" ISO /TC97/SC16 N373

ISO, BERLIN, 7-11 Novembre 80.

" Le système MMT2 "

- Manuel de référence

Documentation SEMS

" Le système de communication SCS2 "

- Manuel de référence

Documentation SEMS .

" Le produit FTS NIFTP "

- Spécifications fonctionnelles

Documentation SEMS .