



HAL
open science

Système et réseaux informatiques : fonctions frontales dans un environnement repartitionné sur un réseau

Michelle Cart

► **To cite this version:**

Michelle Cart. Système et réseaux informatiques : fonctions frontales dans un environnement repartitionné sur un réseau. Réseaux et télécommunications [cs.NI]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1979. Français. NNT : . tel-00829888

HAL Id: tel-00829888

<https://theses.hal.science/tel-00829888>

Submitted on 4 Jun 2013

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

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

N° d'ordre 18 11

THÈSE

présentée par

Michèle CART

pour obtenir

LE TITRE DE DOCTEUR-INGÉNIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

**FONCTIONS FRONTALES
DANS UN ENVIRONNEMENT REPARTI
BATI SUR UN RESEAU LOCAL**

Soutenue à Saint-Etienne le 11 Décembre 1979 devant la commission d'examen :

MM. L. BOLLIET

Président

J.F. CHAMBON

S. GUIBOUD-RIBAUD

B. LE BIHAN

J. LE BIHAN

} Examineurs



N° d'ordre 18 II

THÈSE

présentée par

Michèle CART

pour obtenir

LE TITRE DE DOCTEUR-INGÉNIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

**FONCTIONS FRONTALES
DANS UN ENVIRONNEMENT REPARTI
BATI SUR UN RESEAU LOCAL**

Soutenue à Saint-Etienne le 11 Décembre 1979 devant la commission d'examen :

MM. L. BOLLIET

Président

J.F. CHAMBON

S. GUIBOUD-RIBAUD

B. LE BIHAN

J. LE BIHAN

} Examineurs

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur M. G. ARNOUIL
Directeur Scientifique M. C. GOUX
Directeur Administratif M. A. COINDE

PROFESSEURS DE 1ère CATEGORIE

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

PROFESSEURS DE 2ème CATEGORIE

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

DIRECTEUR DE RECHERCHE

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

MAITRES DE RECHERCHE

MM. BISCONDI	Michel	Métallurgie
COUEIGNOUX	Philippe	Informatique
DAVOINE	Philippe	Géologie
Mle FOURDEUX	Angeline	Métallurgie
MM. KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Métallurgie
MATHON	Albert	Mines Organisation
PERRIN	Michel	Géologie
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M. Philippe TRAYNARD Année Universitaire
1978-1979

Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS TITULAIRES

MM. BENOIT	Jean	Electronique - Automatique
BESSON	Jean	Chimie Minérale
BLOCH	Daniel	Physique du Solide - Cristallographie
BONNETAIN	Lucien	Génie Chimique
BONNIER	Etienne	Métallurgie
*BOUDOURIS	Georges	Electronique - Automatique
BRISSONNEAU	Pierre	Physique du Solide - Cristallographie
BUYLE-BODIN	Maurice	Electronique - Automatique
COUMES	André	Electronique - Automatique
DURAND	Francis	Métallurgie
FELICI	Noel	Electronique - Automatique
FOULARD	Claude	Electronique - Automatique
LANCIA	Roland	Electronique - Automatique
LONGEQUEUE	Jean-Pierre	Physique Nucléaire Corpusculaire
LESPINARD	Georges	Mécanique
MOREAU	René	Mécanique
PARIAUD	Jean-Charles	Chimie-Physique
PAUTHENET	René	Electronique - Automatique
PERRET	René	Electronique - Automatique
POLOUJADOFF	Michel	Electronique - Automatique
ROBERT	André	Chimie Appliquée et des Matériaux
TRAYNARD	Philippe	Chimie - Physique
VEILLON	Gérard	Informatique Fondamentale et Appliquée

* en congé pour études.

PROFESSEURS SANS CHAIRE

MM. BLIMAN	Samuel	Electronique - Automatique
BOUVARD	Maurice	Génie Mécanique
COHEN	Joseph	Electronique - Automatique
GUYOT	Pierre	Métallurgie Physique
JOUBERT	Jean-Claude	Physique du Solide - Cristallographie
LACOUME	Jean-Louis	Electronique - Automatique
ROBERT	François	Analyse Numérique
SABONNADIÈRE	Jean-Claude	Electronique - Automatique
ZADWORNÝ	François	Electronique - Automatique

MAITRES DE CONFERENCES

MM. ANCEAU	François	Informatique Fondamentale et Appliquée
CHARTIER	Germain	Electronique - Automatique
Mme CHERUY	Arlette	Automatique
MM. CHIAVERINA	Jean	Biologie, Biochimie, Agronomie
IVANES	Marcel	Electronique - Automatique
LESIEUR	Marcel	Mécanique
MORET	Roger	Physique Nucléaire - corpusculaire
PIAU	Jean-Michel	Mécanique
PIERRARD	Jean-Marie	Mécanique
Mme SAUCIER	Gabrielle	Informatique Fondamentale et Appliquée
M. SOHM	Jean-Claude	Chimie Physique

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

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

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

Personnalités habilitées à diriger des travaux de recherche (Décision du Conseil Scientifique)

E.N.S.E.E.G.

MM. BISCONDI	Michel	Ecole des Mines ST ETIENNE (Dépt Métallurgie)
BOOS	Jean-Yves	Ecole des Mines ST ETIENNE (Métallurgie)
DRIVER	Julian	Ecole des Mines ST ETIENNE (Métallurgie)
KOBYLANSKI	André	Ecole des Mines ST ETIENNE (Métallurgie)
LE COZE	Jean	Ecole des Mines ST ETIENNE (Métallurgie)
LESBATS	Pierre	Ecole des Mines ST ETIENNE (Métallurgie)
RIEU	Jean	Ecole des Mines ST ETIENNE (Métallurgie)
SAINFORT		C.E.N Grenoble (Métallurgie)
SOUQUET	Jean-Louis	U.S.M.G.
CAILLET	Marcel	E.N.S.E.E.G. (Chimie Minérale Physique)
COULON	Michel	E.N.S.E.E.G. (Chimie Minérale Physique)
GUILHOT	Bernard	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
LALAUZE	René	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
LANCELOT	Francis	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
SARRAZIN	Pierre	E.N.S.E.E.G. (Chimie Minérale Physique)
SOUSTELLE	Michel	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
THEVENOT	François	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
THOMAS	Gérard	Ecole des Mines ST ETIENNE (Chim.Min.Ph)
TOUZAIN	Philippe	E.N.S.E.E.G. (Chimie Minérale Physique)
TRAN MINH	Canh	Ecole des Mines ST ETIENNE (Chim.Min.Ph)

E.N.S.E.E.G.

MM. BOREL	Joseph	Centre d'Etudes Nucléaires de GRENOBLE
KAMARINOS	Georges	Centre National Recherche Scientifique

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

MM. BORNARD	Guy	Centre National Recherche Scientifique
DAVID	René	Centre National Recherche Scientifique
DESCHIZEAUX	Pierre	Centre National Recherche Scientifique

E.N.S.I.M.A.G.

MM. COURTIN	Jacques	Université des Sciences Sociales
LATOMBE	Jean-Claude	Institut National Polytechnique GRENOBLE
LUCAS	Michel	Université Scientifique et Médicale GRENOBLE.

Je tiens à remercier:

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

Monsieur Bernard LE BIHAN, Ingénieur à la Régie Renault, qui a accepté de juger ce travail.

Monsieur Jean LE BIHAN, Directeur du projet SIRIUS, qui a bien voulu accepter de participer à ce jury.

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

Monsieur Jean François CHAMBON, responsable scientifique à l'Ecole des Mines de Saint Etienne, qui est l'initiateur de cette étude et qui ne m'a jamais ménagé ni ses conseils ni ses encouragements.

J'exprime également toute mon amitié aux membres de l'équipe Centre de Calcul Réparti, Messieurs Yannis ANGELIDES, Merval JUREMA et Amadou NIANG qui ont contribué directement à ce projet.

Enfin, je remercie Madame Geneviève LALLICH qui m'a aidée avec beaucoup d'efficacité à établir la bibliographie de cette étude.

SOMMAIRE

CHAPITRE I:	INTRODUCTION	1
	1. Préliminaires.....	3
	2. L'environnement initial de l'étude.....	5
	3. Motivations du projet "Centre de Calcul Réparti"..	5
	4. Approche technique.....	7
	4.1 Aspect utilisateur.....	7
	4.2 Aspect système.....	8
	5. Idées principales ayant guidé la conception.....	8
	6. Plan de la thèse.....	10
CHAPITRE II:	LE CENTRE DE CALCUL REPARTI DANS LE CADRE DE L'INFORMATIQUE REPARTIE	13
	1. Configuration d'un système réparti.....	15
	1.1 Les réseaux d'ordinateurs.....	16
	1.2 Les systèmes multi-processeurs.....	17
	1.3 Les systèmes hybrides.....	18
	2. Architecture d'un système réparti.....	19
	2.1 Le niveau matériel.....	19
	2.2 Le niveau système d'exploitation.....	20
	2.3 Le niveau utilisateur.....	21
	3. Le Centre de Calcul Réparti.....	22
	3.1 Configuration.....	22
	3.2 Aspect système.....	23
	3.2 Aspect utilisateur.....	23
	3.3 Communication par échange de messages.....	23
CHAPITRE III:	PREMIER NIVEAU DE FONCTIONS	25
	1. Le service de tranport.....	27
	1.1 Fonctionnalité du service de transport.....	27
	1.2 Approche par niveaux.....	29
	2. Banalisation des terminaux.....	31

2.1	Terminal virtuel.....	32
2.1.1.	Définition d'un terminal virtuel.....	33
2.2.2.	Emulateur de terminal virtuel.....	35
2.2	Introduction d'un réseau entre un terminal et une application.....	36
3.	Plaque tournante.....	37
4.	Réseau de communication.....	39

CHAPITRE IV: DEUXIEME NIVEAU DE FONCTIONS43

1.	Pourquoi un système réseau ?.....	45
2.	Quelques exemples de systèmes réseau.....	48
2.1.	Le projet M2.....	48
2.2.	Le projet SOC.....	48
2.3.	Le projet IGOR.....	50
2.4.	UNIX.....	50
3.	Approche pour la conception du système réseau du C.C.R.....	51
3.1	Approche descendante.....	51
3.2	Approche ascendante.....	53
4.	Le langage de commande externe (LCE).....	55
4.1	Choix d'un langage.....	56
4.2	Interprétation du LCE.....	58
5.	Langages de commande internes (LCI).....	63
5.1	Coopération d'applications.....	63
5.2	Interprétation du LCI.....	65
6.	Le protocole "Centre de Calcul Réparti".....	66
6.1	Protocole de haut niveau.....	66
6.2	Fonctionnalité du protocole CCR.....	67
6.3	Gestion du protocole CCR.....	69

CHAPITRE V: ARCHITECTURE LOGIQUE73

1.	Présentation des concepts utilisés.....	75
1.1	Architecture par niveaux.....	75
1.2	Système intégré.....	77
2.	Rappel du rôle de la fonction frontale.....	79
3.	Les voies logiques de communication.....	80
4.	Dynamique des échanges.....	83
4.1.	L'émulateur de terminal virtuel.....	84

4.2.	Le niveau supérieur de la méthode d'accès des ordinateurs connectés.....	84
4.3.	Le correspondant.....	84
5.	Nature des dialogues.....	85
5.1.	L'utilisateur.....	86
5.2.	L'interprète du LCE.....	87
5.3.	L'application réseau.....	88
CHAPITRE VI:	REALISATION	91
1.	Historique de l'étude.....	93
2.	Le FRONTAL.....	94
3.	Localisation des fonctions frontales.....	97
3.1.	Service de transport.....	97
3.2.	Dynamique des échanges.....	98
3.3.	Interprétation.....	99
4.	Architecture du CCR.....	99
4.1.	Architecture du frontal.....	99
4.2.	Architecture globale.....	107
CHAPITRE VII:	CONCLUSION	109

CHAPITRE I

INTRODUCTION

1. PRELIMINAIRES
2. L'ENVIRONNEMENT INITIAL DE L'ETUDE
3. MOTIVATIONS DU PROJET "CENTRE DE CALCUL REPARTI"
4. APPROCHE TECHNIQUE
 - 4.1. Aspect utilisateur
 - 4.2. Aspect système
5. IDEES PRINCIPALES AYANT GUIDE LA CONCEPTION
6. PLAN DE LA THESE

1. PRELIMINAIRES

L'étude des réseaux a jusqu'à maintenant été abordée par l'aspect télé-communication en repoussant à plus long terme les points de vue système et externe.

L'aspect externe concerne l'utilisateur qui se voit confronté à une multitude de systèmes d'exploitation différents. Or, très peu d'utilisateurs peuvent prétendre connaître parfaitement deux ou plusieurs systèmes; ce que l'on pensait être un avantage des réseaux généraux devient donc un inconvénient. Avec leur développement croissant et notamment l'entrée dans l'ère de la télématique, il semble que le moment soit venu de chercher à faciliter l'utilisation de l'outil informatique dans le cadre d'une informatique répartie.

Quant à l'aspect système, on constate, malgré le caractère opérationnel des réseaux et l'expérience déjà acquise dans ce domaine, qu'une quantité importante de programmation système doit être investie chaque fois que l'on désire [METCALF: GEN-4] :

- soit raccorder un nouvel ordinateur à un réseau,
- soit permettre une plus large utilisation des ressources d'ordinateurs distants.

Les difficultés à surmonter sont de plusieurs ordres:

* tout d'abord, il s'agit de gérer une liaison avec un autre ordinateur. On se heurte souvent à un comportement à tendance "paternaliste" de chaque ordinateur, sous le contrôle de son système d'exploitation, vis à vis de son environnement extérieur:

- terminaux
- autre ordinateur auquel on veut faire jouer le rôle de terminal virtuel.

De plus, on constate encore un manque évident de normes dans ce domaine.

- * la difficulté suivante est de mettre en place les programmes de contrôle réseau et de gestion de protocoles dont l'intégration n'est pas toujours prévue dans un système d'exploitation.

- * enfin, se pose le problème de la communication entre processeurs, problème encore non résolu à l'intérieur d'un système d'exploitation et à fortiori lorsqu'il s'agit de faire communiquer des processeurs sous des systèmes d'exploitation différents. Ce problème est crucial notamment en ce qui concerne le partage de bases de données.

Cet exemple tendrait à prouver une certaine inadéquation des systèmes d'exploitation aux fonctions réseau, d'où la difficulté d'intégration de celles-ci. La difficulté est encore accrue par le fait que les ordinateurs installés sur un réseau représentent toute une variété de standards et de systèmes d'exploitation.

Notre étude se propose d'apporter une contribution relativement à ces deux aspects liés à l'utilisation de réseaux et plus largement au développement de l'informatique répartie. Nous définirons des fonctions réalisant, en quelque sorte, une extension des systèmes d'exploitation permettant leur intégration dans un environnement réseau en donnant une part importante à l'aspect utilisation. Ces fonctions, interposées entre les systèmes d'exploitation existants et ce qui constitue leur monde extérieur, seront regroupées sous le nom de fonctions frontales.

Dans un premier temps, nous allons décrire le projet qui est le support de cette étude, ses motivations, l'approche et la philosophie adoptées.

2. L'ENVIRONNEMENT INITIAL DE L'ETUDE

L'environnement initial est celui d'un centre de calcul disposant d'un certain nombre de mini-calculateurs isolés, chacun d'eux étant spécialisé dans le traitement d'un petit nombre de fonctions. Le fait d'affecter à un ordinateur une ou un nombre restreint de fonctions présente des justifications à la fois économiques et techniques [FARBER: GEN-1, GEN-2]. En effet, les progrès très rapides de la technologie de l'intégration font apparaître sur le marché des matériels de plus en plus performants à des prix de moins en moins élevés. Ce phénomène entraîne une remise en cause de l'outil informatique. Remplacer un gros ordinateur, capable de tout faire, par plusieurs mini-calculateurs spécialisés devient financièrement intéressant, d'une part parce que la somme des coûts des mini-ordinateurs n'atteint généralement pas le prix d'un gros ordinateur, d'autre part parce que l'investissement peut être progressif et s'effectuer au fur et à mesure des besoins. D'un point de vue technique, l'aspect spécialisation est séduisant dans la mesure où l'écriture d'un logiciel spécifique est plus facile à développer et à maintenir qu'un logiciel spécialisé. Dans le contexte d'un centre de calcul, la spécialisation des calculateurs a pour corollaire leur hétérogénéité puisque chacun d'eux est adapté aux fonctions qu'il traite. Un calculateur conçu pour des applications temps-réel est en effet très différent d'un calculateur spécialisé dans le calcul numérique.

3. MOTIVATIONS DU PROJET "CENTRE DE CALCUL REPARTI"

L'argumentation en faveur de la répartition des services offerts par le centre de calcul sur plusieurs calculateurs vient d'être présentée. Malheureusement, la décentralisation et l'hétérogénéité comportent des aspects négatifs

pour les utilisateurs du centre de calcul; il devient indispensable pour eux de connaître le schéma des ressources du centre, c'est à dire, pour chaque calculateur du centre:

- les fonctions ou applications qu'il traite,
- ses accès possibles c'est à dire les terminaux qui lui sont connectés,
- le langage de contrôle mis en oeuvre par son système d'exploitation.

Ces considérations constituent le fondement du projet et ont permis d'en définir les objectifs. Ceux-ci sont de deux ordres:

1. interconnexion des mini-calculateurs pour bénéficier des avantages apportés par les réseaux, à savoir:
 - la mise en commun des ressources (terminaux,...),
 - le partage des données (fichiers).

2. donner à l'utilisateur l'apparence d'un seul calculateur qui matérialiserait pour lui le centre de calcul avec:
 - le catalogue des services offerts par le centre,
 - les ressources périphériques du centre,
 - un langage de commande unique et simple.

Le but n'est pas de définir de nouvelles techniques réseau mais d'utiliser celles qui existent, au besoin en les particularisant. L'essentiel de notre effort porte en effet sur le deuxième point qui présente un important aspect expérimental et dont nous souhaitons montrer la "faisabilité", l'intérêt et, par la suite, la viabilité. Nos travaux vont dans le sens de l'amélioration de la relation entre utilisateurs et services; l'accent sera mis sur l'accessibilité des services aux utilisateurs.

4. APPROCHE TECHNIQUE

Elle vise à établir un cahier des charges regroupant les contraintes et les choix GUIBOUD RIBAUD: CCR-3 . Nous distinguons deux aspects.

4.1 Aspect utilisateur

Il concerne tous les facteurs associés à la qualité de service rendu à l'utilisateur:

- accessibilité des applications en insistant sur l'interactivité dont l'importance n'a cessé de s'accroître au cours de ces dernières années.
- perception simple du système à travers le langage de commande d'où simplicité d'utilisation pour des non informaticiens.
- transparence de réalisation: l'utilisateur doit ignorer l'architecture physique et logique mise en place.
- fiabilité: l'ensemble doit être construit de telle sorte que la défaillance de l'un des éléments n'entraîne pas la disparition totale des services mais permette tout de même un fonctionnement dégradé.
- extensibilité c'est à dire possibilité de développement des services offerts sans devoir procéder à un arrêt de l'exploitation.
- niveau de performance exprimé par le temps de réponse du système.

4.2 Aspect système

Il doit refléter autant que possible la simplicité extérieure. Il contient les points énoncés côté utilisateur avec toutefois une signification nuancée pour certains.

- accessibilité des applications: il s'agit de mettre en relation des utilisateurs et des applications quelles que soient leurs localisations respectives.

- extensibilité: on désire garder à l'ensemble un caractère d'extensibilité relativement au nombre de mini-calculateurs qui le composent et aux services que ceux-ci offrent. A cet aspect est également lié le point suivant.

- maintien du logiciel existant: ceci permet de bénéficier de la maintenance et des extensions de logiciel assurées par les constructeurs. L'interface avec le logiciel existant doit donc être aussi simple et clair que possible.

- architecture générale et flexible qui permette la diversité des éléments interconnectés, éléments aussi bien matériels (ordinateurs, terminaux) que logiciels (systèmes d'exploitation, applications).

5. IDEES PRINCIPALES AYANT GUIDE LA CONCEPTION

Les objectifs fixés conduisent à un certain nombre de principes directeurs dans la conception de l'architecture proposée. Le premier de ces principes consiste en une approche fonctionnelle qui se propose de distinguer des entités caractérisées par leur(s) fonction(s) et de déterminer ensuite

les interactions entre elles. Il est important de souligner l'autonomie des entités, les unes vis à vis des autres. Cette autonomie signifie la possibilité du déroulement en parallèle de leurs activités, celles-ci se synchronisant, s'il le faut, par le biais des interactions. Cette notion est d'autant plus importante que ces entités sont définies dans un environnement réparti et qu'aucune hypothèse préalable ne doit être faite sur leur localisation. Cette démarche nous conduit à distinguer et à séparer très nettement les deux aspects:

- traitement des données pour produire une action prévue dans la description des fonctions.
- interaction ou communication pour permettre l'acheminement d'informations vers d'autres entités.

L'application de ce principe vise la modularité de l'architecture proposée permettant ainsi des extensions qui ne remettent pas constamment en cause l'ensemble tout entier.

La transparence de réalisation vis à vis des utilisateurs nous a également paru un point important. Cet aspect nous amène à dégager:

- la partie fonctionnelle incluant la logique interne; cette partie doit dans la mesure du possible être ignorée de l'utilisateur.
- la partie interaction qui est portée à la connaissance des utilisateurs et qui comprend les possibilités du langage d'une part et les réponses envoyées par le système d'autre part.

6. PLAN DE LA THESE

Le but de cet ouvrage n'est pas de présenter en détail les mécanismes mis en jeu par notre réalisation mais plutôt de mettre en évidence certains aspects qui nous sont apparus essentiels au cours de nos travaux.

Le chapitre II se propose de donner un aperçu de ce qu'est l'informatique répartie afin de mieux situer la place de nos travaux.

Les chapitres III et IV sont des essais de généralisation et tentent de définir des fonctions réseau, c'est à dire liées à l'existence d'un réseau. Le premier expose les fonctions mises en oeuvre pour permettre une communication entre les éléments matériels du réseau (ordinateur, terminaux) et donner l'illusion d'une certaine homogénéité de ce matériel à la fois aux utilisateurs et aux concepteurs de systèmes répartis. Le chapitre IV présente des fonctions de niveau supérieur, utilisant celles du précédent chapitre en tant qu'outils; leur ambition est de tirer parti de l'existence d'un réseau sans pour autant pénaliser les utilisateurs. Ce chapitre nous paraît être le sujet central de la thèse par son aspect expérimental.

Le chapitre V présente une architecture logique c'est à dire une articulation structurée des fonctions décrites. Il définit la structure globale du système, en précise les principaux constituants et la communication entre ces constituants. Il constitue un préliminaire pour la réalisation du Centre de Calcul Réparti. Les principaux choix effectués au moment de la réalisation sont explicités dans le chapitre VI qui tente d'appliquer la structure logique sur la structure physique.

En conclusion, nous nous interrogeons sur les perspectives d'avenir de notre étude et sur des stratégies éventuelles pour la réalisation de systèmes orientés réseau.

Nous tenons à faire une remarque sur la terminologie employée dans cet ouvrage: les définitions de certains concepts utilisés peuvent être contestées ou paraître incomplètes; il ne s'agit pas de les imposer mais de parler un langage commun avec le lecteur.

CHAPITRE II

LE CENTRE DE CALCUL REPARTI
DANS LE CADRE DE L'INFORMATIQUE REPARTI

1. CONFIGURATION D'UN SYSTEME REPARTI
 - 1.1. Les réseaux d'ordinateurs
 - 1.2. Les systèmes multi-processeurs
 - 1.3. Les systèmes hybrides
2. ARCHITECTURE D'UN SYSTEME REPARTI
 - 2.1. Le niveau matériel
 - 2.2. Le niveau système d'exploitation
 - 2.3. Le niveau utilisateur
3. LE CENTRE DE CALCUL REPARTI
 - 3.1. Configuration
 - 3.2. Aspect système
 - 3.3. Aspect utilisateur
 - 3.4. Communication par échange de messages

1. CONFIGURATION D'UN SYSTEME REPARTI

On constate une certaine confusion dans les diverses définitions qui ont pu être données d'un système réparti. Cette confusion provient en partie du fait que les aspects à la fois matériel et logiciel sont impliqués dans la signification de l'adjectif réparti. Le terme "informatique répartie" exprime à la fois la décentralisation et l'interconnexion des moyens de traitement de l'information. Un environnement informatique devient réparti ou distribué dès lors que les repères de temps et d'espace des éléments le constituant sont indépendants.

L'intérêt actuel pour l'informatique répartie découle largement des tendances technologiques et économiques. L'informatique répartie semble être un moyen d'exploiter au mieux la potentialité du matériel sans cesse accrue par les spectaculaires développements de la micro-électronique.

Deux démarches distinctes permettent de passer d'une informatique centralisée à une informatique répartie. La première consiste à opérer un éclatement du système centralisé en un certain nombre de composants, chacun ayant son autonomie propre, pour obtenir du parallélisme. La seconde vise à procéder à une extension du système en le reliant à d'autres systèmes par l'intermédiaire de réseaux dans le but de rompre son isolement.

Ces deux approches ont constitué jusqu'à maintenant deux axes de recherche:

- les systèmes multi-processeurs
- les réseaux d'ordinateurs

Bien qu'initialement la justification et les objectifs visés aient été très distincts, ces deux tendances se rejoignent actuellement pour donner naissance à des systèmes hybrides cherchant ainsi à réaliser un compromis. La mise en oeuvre d'une informatique répartie fait appel à un vaste

éventail de techniques et de concepts dont les réseaux d'ordinateurs et les systèmes multi-processeurs constituent les deux extrêmes.

1.1 Les réseaux d'ordinateurs

La justification de regrouper des ordinateurs en un réseau s'exprime avant tout par le désir de rompre l'isolement de ces ordinateurs. Le terme "réseau d'ordinateurs" évoque l'interconnexion de gros systèmes isolés et indépendants; il sous-entend que ces éléments communiquent par l'intermédiaire de lignes de transmission. Les réseaux ont évolué à partir des télécommunications terminal-ordinateur, l'objectif initial étant la connexion d'un terminal distant à un ordinateur central. Puis, le besoin de connexion entre ordinateurs s'est fait sentir, besoin justifié par le désir d'un partage de ressources (fichiers, applications, etc...). Dans le même temps, l'utilisation d'ordinateurs pour gérer les communications, dont les techniques se développaient et devenaient plus complexes, apparaissait. Ce cheminement a permis, à la fois, le développement des techniques de communication et le développement de la coopération entre systèmes d'exploitation.

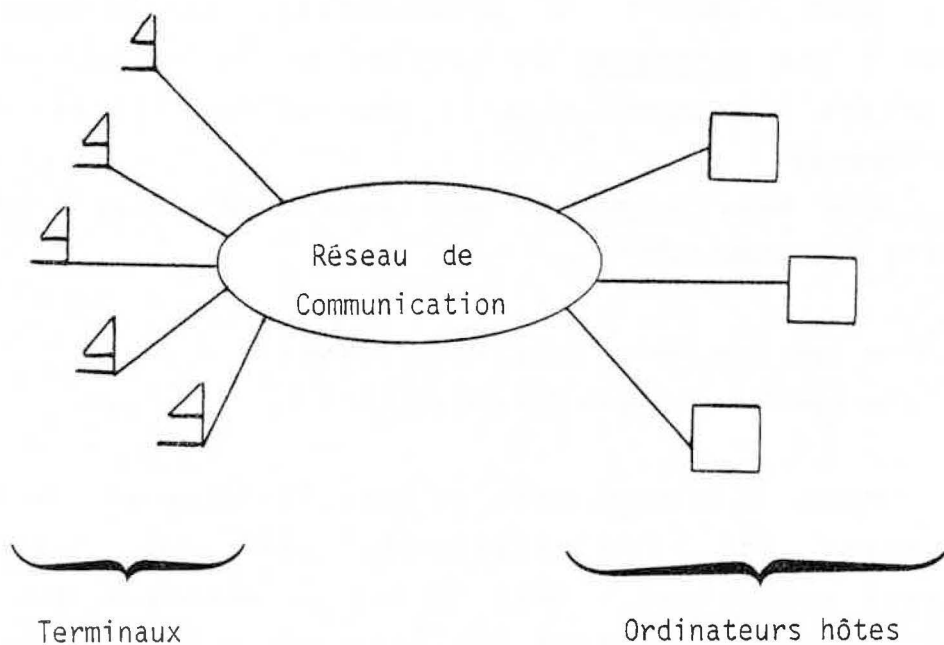


Figure 1: Vision d'un réseau d'ordinateurs

Après s'être faite dans le sens des dimensions croissantes (réseaux généraux de gros ordinateurs géographiquement dispersés) l'évolution des réseaux s'oriente actuellement vers l'interconnexion de mini-ordinateurs le plus souvent locaux.

1.2 Les systèmes multi-processeurs

Ces systèmes consistent en l'assemblage d'éléments capables de travailler en parallèle [ENSLOW: RES-11]. On vise ainsi à la fois la décentralisation et le parallélisme. Les premières réalisations consistaient, par exemple, en une connexion d'un contrôleur d'entrée-sortie à un gros ordinateur. Ensuite, elles ont évolué vers la connexion de plusieurs processeurs à une mémoire commune, comportant quelquefois l'utilisation de mini-ordinateurs connectés dans des configurations de multi-processeurs. Cette tendance à la décentralisation visait un gain de modularité et de fiabilité. On espère en effet obtenir une fiabilité accrue par la distribution du matériel, du logiciel et du contrôle; ceci suppose toutefois qu'il n'existe pas de ressource centrale qui doive fonctionner pour que l'ensemble fonctionne.

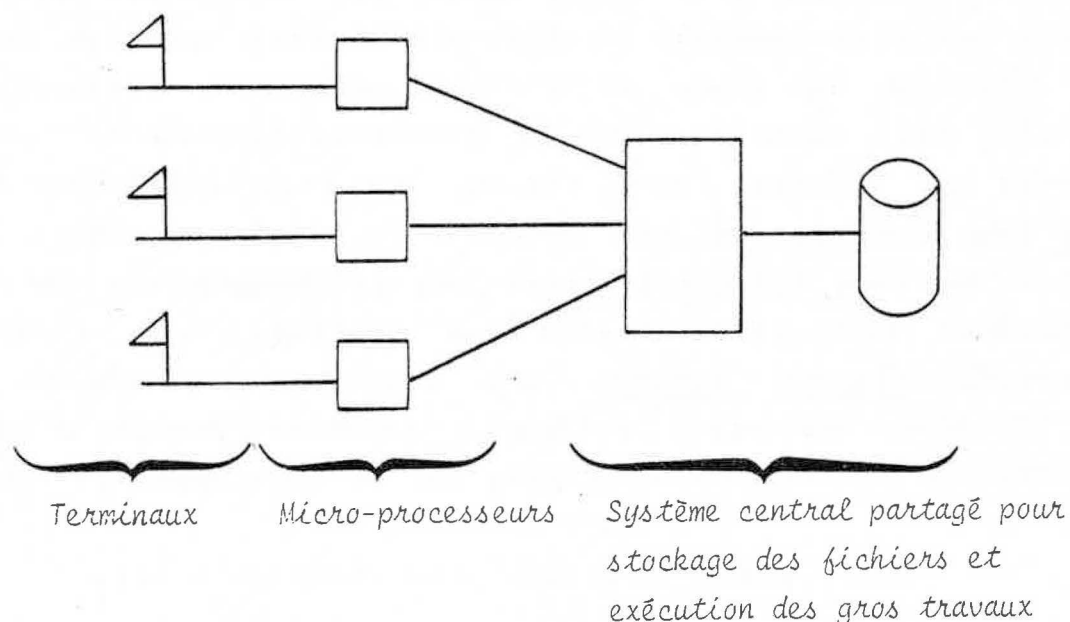


Figure 2: Exemple de système multi-processeurs

Actuellement, on constate qu'il y a moins de systèmes avec partage de mémoire centrale; par contre la communication entre processeurs basée sur une isolation accrue des composants et sur le principe de l'existence de lignes de transmission imaginaires entre eux se développe.

Bien que les services offerts par un système multi-processeurs soient très similaires à ceux d'un système mono-processeur, la présence de plusieurs processeurs introduit une dimension nouvelle dans la conception des systèmes d'exploitation et dans les techniques de conception de programmes.

La place, de plus en plus grande, occupée par les micro-processeurs est un élément déterminant dans le développement des systèmes multi-processeurs.

1.3 Les systèmes hybrides

Entre les deux catégories présentées ci-dessus, on trouve les systèmes hybrides dont l'objectif est de combiner le partage des ressources offert par les réseaux et le travail en parallèle caractéristique des multi-processeurs. La réalisation de ces systèmes peut faire appel aux techniques développées dans les deux domaines et apparaît souvent comme un compromis s'inspirant de l'une ou l'autre catégorie. Quelquefois, il s'agit tout simplement de la particularisation de techniques liées aux réseaux d'ordinateurs. Une propriété assez générale de ces systèmes est une certaine unité géographique, c'est à dire que les distances entre les différents composants sont faibles; pour cette raison ces systèmes sont généralement appelés réseaux locaux. Les essais de classification à l'intérieur de cette catégorie relèvent plutôt d'un désir d'établir une échelle de degrés que de distinguer des types.

Voici quelques exemples de réseaux locaux:

- DCS: "Distributed Computer System" [FARBER: RES-5, RES-6]; il s'agit d'un réseau de mini-ordinateurs hétérogènes reliés entre eux par un anneau de communication.

- C.mmp: "Multi-Mini-Processor" [WULF: RES-4, RES-10]; la configuration de ce système comprend d'une part des processeurs et d'autre part des blocs de mémoire: les processeurs se partagent la mémoire par l'intermédiaire d'une boîte de commutation et peuvent s'interrompre les uns les autres grâce à un bus d'interruption commun.

- M2: "système Multi-Mini-ordinateurs" [HEBENSTREIT: RES-9]; ce système sera présenté ultérieurement (chapitre IV § 2.2.).

- ARAMIS [CABANEL: RES-8]; il s'agit d'un réseau de mini-ordinateurs homogènes supportant des systèmes d'interrogation de fichiers.

- ETHERNET [METCALF: RES-2].

- MININET [MANNING: RES-12].

2. ARCHITECTURE D'UN SYSTEME REPARTI

Dans l'organisation d'un système réparti on peut distinguer au moins trois niveaux:

2.1 Le niveau matériel

Il dépend du moyen d'interconnexion choisi; il peut s'agir d'une mémoire, d'un bus, d'un anneau, d'une structure à maillage complet ou partiel, de liaisons radio/satellite, de lignes multipoint, etc...

Le choix de ce moyen met en jeu des considérations de coût, de distance entre les éléments à relier, de fiabilité, de débit d'information à échanger, etc...

Ce niveau exprime la structure physique du système c'est à dire le degré de couplage des différents éléments ANDERSON:RES-1 . On distingue:

- les systèmes faiblement couplés dans lesquels les composants sont reliés par des lignes de transmission et communiquent par des messages transmis suivant un protocole.

- les systèmes fortement couplés qui utilisent tout autre moyen de connexion; c'est le cas lorsque les composants d'un système ont des espaces d'adressage "mémoire" communs et communiquent en lisant et mettant à jour des variables partagées.

Dans tous les cas, ce niveau met en oeuvre un service de transport qui permet le mouvement de l'information entre les différents éléments du système. La réalisation de ce service de transport peut faire appel à des techniques variées:

- utilisation de lignes dédiées
- commutation de circuit
- commutation de paquets
- partage de mémoire

Les protocoles de bas niveau utilisés pour le transfert d'information entre les différents points du réseau sont gérés par ce niveau; ils sont en effet étroitement dépendants du type d'interconnexion réalisé.

2.2 Le niveau système d'exploitation

A ce niveau la distinction s'effectue selon qu'il existe un seul système d'exploitation réparti contrôlant l'ensemble ou qu'il s'agit d'un couplage de machines, chacune conservant son propre système d'exploitation.

Dans le premier cas, chaque élément du système est spécialisé dans l'une des fonctions du système d'exploitation: gestion de fichiers, gestion des terminaux, enchaînement des travaux (c'est le cas du projet M2 [HEBENSTREIT: RES-9]); on

distingue un rôle maître et des rôles subordonnés d'où un contrôle centralisé et hiérarchisé. Après une décentralisation au niveau matériel, on opère une sorte de recentralisation au niveau du logiciel.

Dans le second cas chaque sous-système (sous-système - ordinateur + système d'exploitation) est capable de fonctionner indépendamment des autres; ces sous-systèmes sont des entités équivalentes logiquement et qui coopèrent à un niveau égal; on ne distingue pas de hiérarchie dans le contrôle; celui-ci est distribué.

On peut également prendre en compte le fait que l'exécution des travaux des utilisateurs soit confiée à des éléments fixés à l'avance ou bien déterminés selon la charge du système. Ce problème devient vite complexe surtout si le système est hétérogène, et met en cause de multiples aspects parmi lesquels la transférabilité de programmes.

2.3 Le niveau utilisateur

Pour l'utilisateur le système peut apparaître:

- soit comme un seul système avec un ensemble de commandes permettant l'accès à toutes les facilités offertes; le degré de distribution lui est invisible.

- soit comme une collection de différentes machines ayant chacune son propre langage de commande, avec en plus, un langage réseau qui permet l'accès à des services qui ne sont pas situés sur la machine à laquelle l'utilisateur est connecté. L'utilisateur est conscient de la distribution et en conséquence exprime explicitement cette notion dans les travaux qu'il soumet au système. Si une ressource est accédée à travers le réseau, elle est nommée différemment que si elle était locale.

L'élément de distinction est donc le degré de transparence du réseau pour l'utilisateur.

Dans le premier cas, l'utilisateur peut s'en

remettre au système d'exploitation pour ce qui est de l'acquisition et de la gestion des ressources nécessaires à ses travaux.

Dans le second cas, cette responsabilité incombe plus ou moins à l'utilisateur. C'est lui qui décide quelles ressources locales ou distantes doivent être allouées pour son travail. Si ce travail met en jeu des fichiers, il va, éventuellement, devoir procéder de lui-même à leur transfert. Enfin, il détermine le processeur qui va exécuter le travail.

3. LE CENTRE DE CALCUL REPARTI

Le thème du projet "Centre de Calcul Réparti" (CCR) est centré sur la conception et la réalisation d'un système réparti et de ses composants fonctionnels au sein d'un réseau de mini-calculateurs. Nous allons tenter de caractériser ce système par ses différents aspects.

3.1. Configuration

La configuration se rapproche de celle des réseaux d'ordinateurs dans la mesure où il s'agit d'interconnecter des mini-ordinateurs indépendants pour réaliser un ensemble multi-mini-ordinateurs. Dans un premier temps tout au moins, il ne s'agit en aucune façon de modifier la structure matérielle de chacun de ces mini-ordinateurs et l'objectif est d'altérer le moins possible la structure logicielle.

Néanmoins, il est toujours possible de se tourner vers une orientation multi-processeurs. En effet, l'introduction de micro-processeurs peut être envisagée pour la réalisation de certaines fonctions. On peut également envisager l'introduction de liaisons bus à bus.

Dans la suite, nous plaçons cette étude dans le cadre des réseaux mais les concepts dégagés s'appliquent à toute structure répartie.

3.2. Aspect système

Le contrôle du système global n'est pas centralisé. Chaque ordinateur participant conserve son système d'exploitation qui lui permet de fonctionner en dehors d'un environnement réparti et d'assurer, en cas de défaillance un service dégradé. Ce choix comporte d'autres avantages; d'une part il minimise l'investissement en écriture de logiciel; d'autre part il permet de bénéficier des évolutions du système apportées par le constructeur et d'être libéré d'une partie des problèmes de maintenance. L'intégration dans le système est assurée par un ensemble de fonctions supplémentaires qui vont être définies dans la suite.

3.3. Aspect utilisateur

L'utilisateur dispose d'un moyen d'interaction unique, le langage de commande, avec l'ensemble du système.

D'autre part on lui offre un ensemble de ressources qu'il nomme et utilise de la même façon que dans un environnement mono-calculateur. Ces ressources peuvent être des applications, des fichiers, des appareils (imprimante, etc...).

La structure distribuée lui est donc totalement invisible.

3.4. Communication par échange de messages

Les exemples montrent que les mécanismes de communication à l'intérieur d'un système (réparti ou non) sont étroitement dépendants de la structure matérielle. Ainsi, dans un environnement où les éléments qui communiquent partagent une mémoire centrale commune, la communication s'effectue généralement par le biais d'un adressage dans cette mémoire, permettant l'accès aux données et leur modification. Ce mécanisme est adapté à la configuration mais, en contrepartie, il ne permet pas d'y apporter des modifications sans de profondes transformations. Il est donc important d'utiliser des

mécanismes tout à fait généraux qui puissent être mis en oeuvre à partir de configurations et de types de connexion très différents. En contraignant toutes les entités d'un système qui désirent communiquer à s'échanger explicitement des messages comme si elles étaient reliées par une ligne de transmission, on apporte une certaine uniformisation dans les mécanismes d'échanges de données. Il en découle une grande adaptabilité à toutes sortes de configurations de système. Un message est à la fois un véhicule d'information et un élément de synchronisation et il constitue le support de toutes les transactions. Ce mécanisme de messages est utilisable à tous les niveaux de conception.

Le service de transport d'un système réparti apparaît avec ce mécanisme comme un commutateur de messages. Sa fonction est de recueillir les messages des émetteurs et de les véhiculer jusqu'aux destinataires. Quant aux entités qui dialoguent, elles synchronisent leurs actions par le biais des messages échangés.

La recherche sur les langages de programmation s'oriente vers une communication entre modules au moyen de protocole de messages asynchrones plutôt que par des appels de procédure où un module doit toujours attendre la réponse de l'autre [FELDMAN: LAN-3, BALL: LAN-4]. Cette option va dans le sens du développement de langages de programmation pour des applications réparties.

Cette notion de message nous semble essentielle car elle paraît être la clef de nombreux problèmes de communication et de synchronisation. Elle a, de plus, l'avantage d'avoir un effet de clarification quant à la compréhension du fonctionnement du système.

CHAPITRE III

PREMIER NIVEAU DE FONCTIONS

1. LE SERVICE DE TRANSPORT
 - 1.1. Fonctionnalité du service de transport
 - 1.2. Approche par niveaux
2. BANALISATION DES TERMINAUX
 - 2.1. Terminal virtuel
 - 2.2.1. Définition d'un terminal virtuel
 - 2.2.2. Emulateur de terminal virtuel
 - 2.2. Introduction d'un réseau entre un terminal et une application
3. PLAQUE TOURNANTE
4. RESEAU DE COMMUNICATION

Dans ce chapitre et le suivant nous essayons de dégager certaines fonctions mises en évidence au cours de notre étude et qui nous paraissent essentielles dès qu'il est question d'informatique répartie. Dans ce chapitre nous nous limitons à un premier niveau de fonctions:

* service de transport

* plaque tournante

* banalisation des terminaux

qui permettent la mise en place d'un réseau de communication. Ces outils constituent un ensemble minimal pour l'utilisation d'un réseau.

1. LE SERVICE DE TRANSPORT

La variété des moyens de transmission entraîne également une grande variété de mise en oeuvre d'un service de transport. Néanmoins la diversité de ces réalisations n'affecte pas l'aspect fonctionnel que nous nous proposons de mettre en lumière.

1.1 Fonctionnalité du service de transport

Le service de transport décharge les utilisateurs d'un réseau de la complexité plus ou moins grande d'accès à leurs homologues. Les utilisateurs sont connus de celui-ci par une identification ou adresse. C'est grâce à cette adresse qu'ils peuvent se désigner les uns les autres. Les fonctions assumées, en général, sont:

a. un service de messages ou boîte aux lettres.

Les messages sont délivrés à leur destinataire:

- corrects et complets. Le contenu d'un message est valide et il n'y manque aucune information. Le service de

transport a les moyens de détecter et d'éliminer l'information erronée. De plus, un message n'est jamais délivré plus d'une fois.

- ordonnés. Les messages sont délivrés dans l'ordre où ils ont été confiés au service de transport.

b. un service d'événements ou télégrammes.

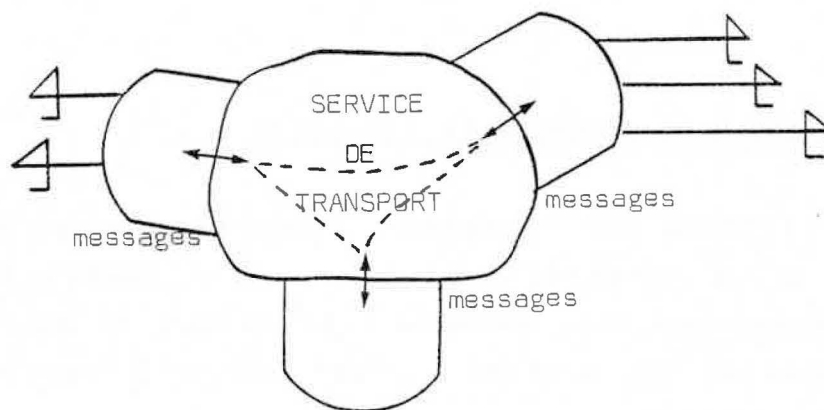
Des événements peuvent être acheminés par le service de transport à tout moment, sans tenir compte du séquençement des messages. La signification de ces événements n'est pas connue du service de transport.

c. un contrôle de flux.

Ce contrôle assure que le débit des émetteurs est asservi à la capacité des récepteurs.

d. la maintenance du réseau

Le service de transport sait à tout moment quels sont les utilisateurs disponibles ou absents; en effet le maintien d'un trafic oisif lui permet de découvrir l'absence ou la panne d'éléments du réseau même en cas de non utilisation du service.



Figure_3: Le service de transport

1.2. Approche par niveaux

Les systèmes orientés vers les communications sont souvent présentés en terme de niveaux de fonctions. Un niveau est défini quant à sa position par rapport aux autres par un certain critère. Ce critère peut être le degré de sophistication ou d'indépendance vis à vis d'un élément (le domaine matériel, ligne, etc...) mais aussi bien tout autre critère permettant une hiérarchisation. Cette approche sera explorée et justifiée par la suite; elle nous permet d'introduire certains termes.

a. Protocole

Un protocole est défini par l'ensemble des conventions qui règlent les échanges entre entités de même niveau. Ces entités ont des activités qui peuvent éventuellement s'exécuter en parallèle sur des ordinateurs différents.

La spécification d'un protocole comporte trois aspects:

- syntaxique: structure des commandes et des réponses.
- sémantique: ensemble des requêtes avec les actions qu'elles engendrent et les réponses qu'elles demandent de la part du vis à vis.
- spécification et séquençement des événements pour permettre la synchronisation.

Quant aux conventions d'échange, elles contiennent:

- a. le transfert d'information pour prévenir la perte, la duplication ou la modification des messages transmis.
- b. des fonctions de contrôle pour initialiser et synchroniser l'information de contrôle des deux côtés de la connexion et éviter les blocages et les conséquences d'erreurs de protocole.

b. Interface

Un interface est défini par les relations entre entités de niveaux adjacents.

Dans l'architecture d'un service de transport quatre niveaux sont généralement identifiés, chacun correspondant à la mise en oeuvre d'un protocole.

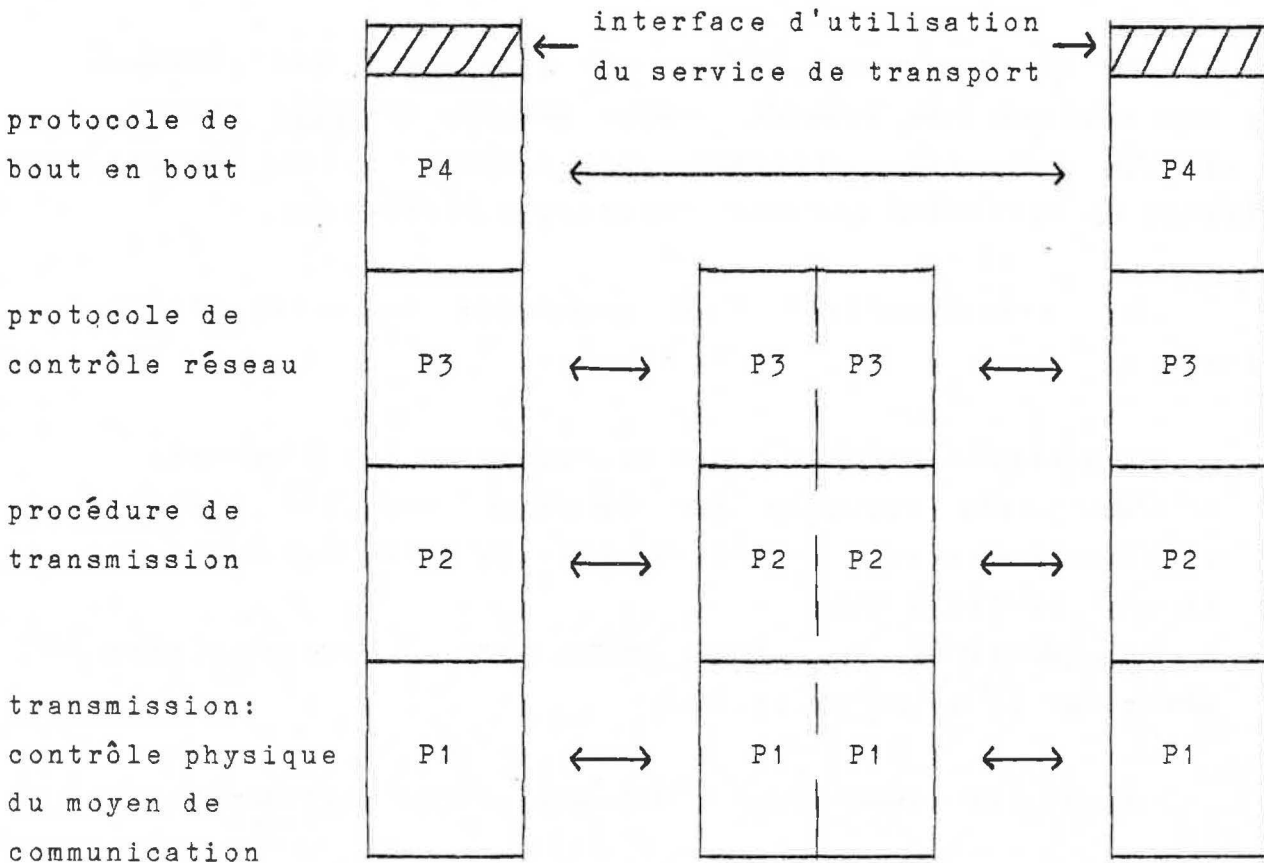


Figure 4: Les différents niveaux de protocole gérés par le service de transport

NIVEAU P1: ce niveau prend en compte les caractéristiques physiques et électriques des équipements qui assurent la liaison physique entre les éléments du réseau. Seul ce niveau est dépendant du moyen de connexion physique; cette notion est totalement transparente pour les autres niveaux.

NIVEAU P2: assure la fiabilité des données échangées entre les équipements gérés par le niveau P1.

NIVEAU P3: assure les fonctions requises pour les opérations internes au réseau telles que: adressage, routage... Il s'agit plus de fonctions de contrôle du réseau que de transfert de données.

NIVEAU P4: assure aux utilisateurs du service de transport un interface indépendant des caractéristiques du réseau.

Le service de transport met en place des voies logiques de communication entre les différents éléments du système. Il offre effectivement un service qui peut être utilisé par des protocoles de niveau supérieur:

- protocole de service de traitement par lots,
- protocole d'accès à un système de base de données distribuées,
- protocole de transfert de fichiers,
- protocole entre un terminal et une application.

2. BANALISATION DES TERMINAUX

Le terminal est l'élément qui permet à l'utilisateur de dialoguer avec un système. Dans la suite, nous utilisons également le terme accès utilisateur pour désigner le terminal.

La variété des terminaux existant sur le marché est impressionnante; on distingue, d'une part, les terminaux classiques avec des possibilités plus ou moins restreintes (type télétype), et, d'autre part, les terminaux dits intelligents pour lesquels la présence de microprocesseur augmente considérablement les possibilités.

Malgré cette diversité, il serait souhaitable de présenter une certaine homogénéité aux utilisateurs; cette homogénéité a une double signification:

- d'une part, banalisation des terminaux distants et des terminaux locaux. Vis à vis de chaque mini-ordinateur participant, le but poursuivi est lui-même double:

.permettre d'accéder à des services distants à partir de terminaux locaux;

.permettre d'accéder à des services locaux à partir de terminaux distants;

Il s'agit, par conséquent, de résoudre le problème de l'introduction d'un réseau entre un terminal et une application.

- d'autre part, homogénéité dans l'utilisation c'est à dire nivellement ou suppression des distinctions. Cette recherche conduit à la définition d'un terminal virtuel dans laquelle on s'efforce d'introduire des standards.

Les deux aspects sont très étroitement liés; nous les distinguons ici pour les facilités de l'exposé.

2.1 Terminal virtuel

La justification d'un terminal virtuel n'est pas seulement l'homogénéité pour l'utilisateur du terminal mais également, et c'est fondamental, l'homogénéité pour les applications qui ont à communiquer avec des terminaux. Cette approche va dans le sens d'une méthode d'interaction uniforme pour une grande variété de terminaux [BARBER: GEN-16].

2.1.1. Définition d'un terminal virtuel

Plutôt que d'en donner une définition purement abstraite, nous préférons le décrire comme un terminal standard concret, qui pourrait être réalisé par un constructeur.

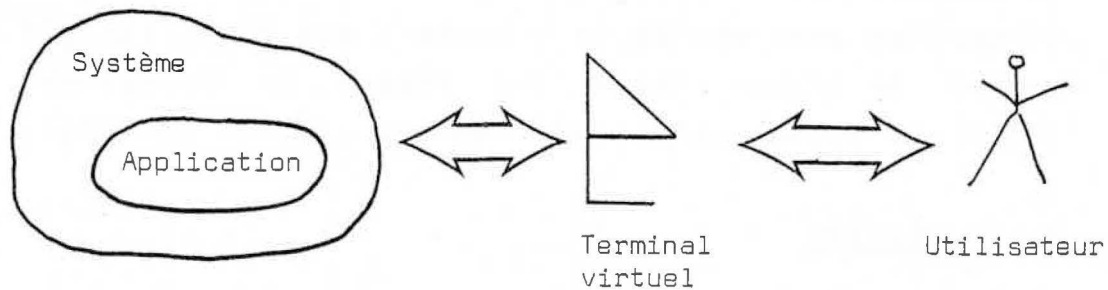


Figure 5: Le terminal virtuel, élément de communication entre l'utilisateur et le système

a. l'unité de visualisation

Matérialisée par un écran ou du papier, elle visualise une composition de symboles graphiques aux yeux de l'utilisateur. Il s'agit d'une structure de données représentée par un tableau à deux dimensions qui correspond à une séquence de positions (caractères) dans une ligne et à une séquence de lignes dans la structure de données (pages). Un pointeur, manipulé par le système ou à partir du clavier, se déplace de manière séquentielle dans cette structure de données. (Pour des terminaux virtuels plus sophistiqués, on peut envisager la présence de deux pointeurs différents)

Le format de l'unité de visualisation définit les paramètres de la structure de données: nombre de caractères par ligne et nombre de lignes par page.

Cette unité de visualisation est partagée par

l'utilisateur du terminal et le système pour visualiser les informations qu'ils échangent. Pour permettre un partage correct, elle ne peut être accédée que sur la base de messages; c'est à dire que l'écriture du texte d'un message sur l'unité de visualisation ne peut être interrompue pour procéder à l'écriture du texte d'un autre message. Ainsi, si le terminal reçoit du système le texte d'un message pendant que l'utilisateur est en train de composer lui-même un message, le message en provenance du système sera stocké en attendant que l'utilisateur ait achevé le sien. Quant aux règles de formation des messages, elles sont dictées par le système lui-même.

b. le clavier

C'est l'unité d'entrée pour l'utilisateur. Les touches peuvent être classées comme suit:

1. touches de composition de messages

Elles représentent les moyens d'accès de l'utilisateur à la structure de données; elles correspondent:

- . soit à des actions sur le pointeur dans la structure de données en accord avec le format en vigueur: passage à la ligne, saut de page, etc...
- . soit à des symboles graphiques c'est à dire à des caractères visualisés aux emplacements courants dans la structure de données. Ces caractères sont représentés de façon interne dans un code déterminé.

2. touches de fonction

Citons:

- . la touche "fin de message" qui provoque la transmission du texte du message vers le système.
- . les touches "êtes-vous là ?" qui permettent à l'utilisateur de savoir si le système avec lequel il travaille est toujours en ligne, lorsque celui-ci manifeste un silence inattendu et prolongé.

c. l'unité de contrôle

- Elle représente l'intelligence du terminal virtuel:
- . elle se charge de la construction des messages, selon les spécifications voulues.
 - . elle interprète les touches de fonction.
 - . elle contrôle le flux d'information à l'intérieur du terminal.
 - . elle gère l'interface de communication défini entre le système et le terminal virtuel.

2.1.2. Emulateur de terminal virtuel

L'émulateur permet au système d'ignorer la diversité des terminaux qui lui sont connectés et de ne connaître qu'un seul type de terminal: le terminal virtuel.

Son rôle est de simuler un terminal virtuel à partir du terminal réel. Il consiste à s'adapter aux caractéristiques physiques du terminal réel: type de connexion, vitesse, format, codes, représentation des données, etc... . L'émulateur se présente à la fois comme adaptateur d'interfaces et gestionnaire de terminal. Selon l'"intelligence" du terminal réel, ses fonctions sont plus ou moins complexes.

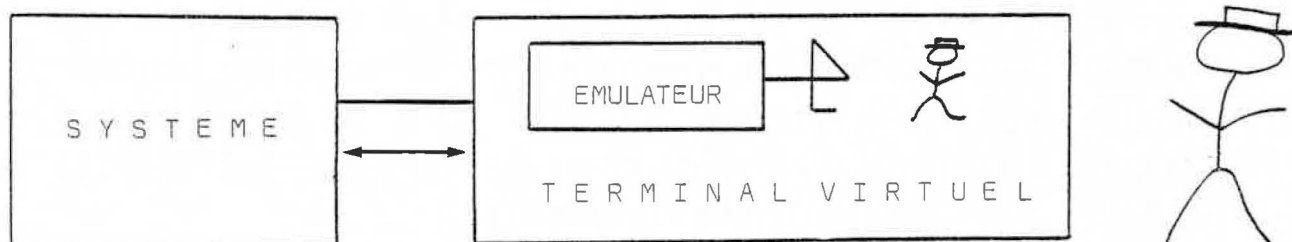


Figure 6

2.2 Introduction d'un réseau entre un terminal et une application

Il s'agit de permettre à un système d'exploitation de prendre en charge la connexion de terminaux distants, si possible comme s'il s'agissait de terminaux locaux.

L'existence d'un service de transport résoud les problèmes de connexion physique mais ne supprime pas toutes les difficultés d'ordre logiciel. Chaque système d'exploitation a sa propre méthode de gestion des terminaux et d'interaction entre terminaux et applications. Il en résulte que l'intégration de la gestion de terminaux distants est parfois très complexe. L'exemple qui suit donne un aperçu des difficultés.

Souvent, dans les systèmes en temps partagé, le contrôle et la gestion des terminaux sont liés au démarrage et à l'achèvement d'une session de travail d'un utilisateur: si un utilisateur frappe un caractère particulier sur un terminal oisif, ce caractère est interprété comme un signal de démarrage d'une nouvelle session. Ce signal doit donc être également généré quand une demande de connexion distante est reçue. Ce travail nécessite l'introduction de logiciel dans le système d'exploitation, ce qui n'est pas toujours aisé; on aborde en effet le problème de l'intégration dans un système d'exploitation de fonctions non prévues. Ce problème ne possède pas de solution standard et il est souvent résolu au coup à coup en traitant les différentes liaisons séparément et en apportant à chaque fois une solution ad hoc. Nous ne développons pas davantage ce paragraphe qui fait, par ailleurs, l'objet d'une étude approfondie [ANGELIDES: CCR-6].

L'accès à une application à partir d'un terminal distant peut être schématisé par la figure suivante.

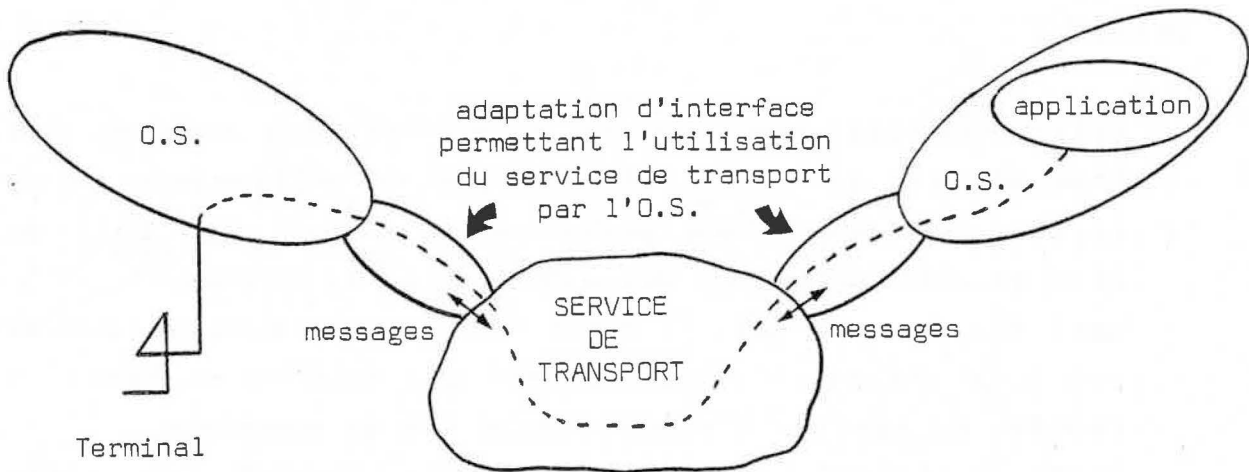


Figure 7

3. PLAQUE TOURNANTE

Les fonctions service de transport et banalisation des terminaux introduisent des possibilités, très vastes il est vrai, mais ne constituent en fait que des outils (indispensables) dans la conception du Centre de Calcul Réparti. L'utilisation de ces outils doit répondre à deux besoins qui sont:

- mettre en relation des utilisateurs, à partir d'accès utilisateurs c'est à dire de terminaux locaux ou distants, et des applications ou d'autres utilisateurs.
- relier des applications entre elles pour leur permettre de coopérer.

Ces deux points nécessitent de gérer des liaisons; chaque liaison s'effectue entre deux éléments qui sont, soit deux applications, soit deux utilisateurs, soit un accès utilisateur et une application, et constitue un lien logique

entre eux. Cette gestion inclut un certain nombre de fonctions que l'on peut englober sous le terme de plaque tournante:

- a. fonction d'établissement de liaison ou de mise en contact.

Cette fonction est plus ou moins complexe suivant que l'on choisit de mettre en relation un utilisateur et un système d'exploitation particulier, choisi par lui, ou bien un utilisateur et une application directement.

Dans le premier cas, l'accès à l'application elle-même est à la charge de l'utilisateur qui doit en exprimer la requête au système d'exploitation qui la supporte.

Dans le second cas, l'accès est effectué par cette fonction de mise en contact. Cet aspect déborde du rôle de plaque tournante et sera développé dans le chapitre suivant. Il en est de même pour les liaisons entre applications. Nous désirons seulement mettre en évidence ici des fonctions dans le cadre d'une liaison d'un utilisateur à un système d'exploitation, remplissant ainsi le rôle de concentrateur de terminaux.

- b. fonction de rupture de liaison, que ce soit sur l'initiative de l'utilisateur ou sur celle du système.
- c. fonction de transfert de données sur les liaisons, c'est à dire aiguillage entre voies de communication et gestion des échanges.

Ces fonctions impliquent l'existence d'un langage minimal à la disposition de l'utilisateur pour lui permettre d'exprimer ces trois aspects d'une communication:

- création
 - utilisation
 - destruction
- } du lien logique unissant les deux interlocuteurs

Ce langage contient les commandes nécessaires à l'utilisateur pour pouvoir se servir du réseau:

- "je demande ma connexion à tel système"
- "je demande ma déconnexion"
- "je veux transmettre cette ligne à l'application à laquelle je suis connecté"

Ce langage nécessite l'existence d'un interprète qui ait une vision globale des utilisateurs et des systèmes interconnectés et qui effectue les aiguillages nécessaires.



Figure 8

4. RESEAU DE COMMUNICATION

Ce paragraphe a pour but de donner un aperçu de l'apport réalisé par la réunion des fonctions présentées plus haut. Ces fonctions mettent en oeuvre des outils qui, mis en commun, modifient déjà notablement pour les utilisateurs la vision du Centre de Calcul Réparti.

Nous appelons réseau de communication l'ensemble des outils proposés; ce réseau de communication constitue un aspect important de la fonction frontale car il résoud tous les problèmes de transmission, tout en apportant une apparence d'homogénéité des moyens offerts par des matériels hétérogènes. Il permet la mise en commun des ressources que sont les terminaux pour plusieurs systèmes. Un terminal n'est plus dédié

à un système donné, il permet également l'accès à tous les systèmes interconnectés. On dispense ainsi l'utilisateur d'avoir à connaître les connexions des terminaux aux ordinateurs. Il n'a pas à se soucier d'une quelconque compatibilité de son terminal avec les applications auxquelles il désire accéder. Un terminal peut être connecté soit à l'un des systèmes pré-existants soit directement au réseau de communication. Les terminaux sont banalisés dans le sens où ils ne se distinguent pas les uns des autres par les possibilités (nombre et nature des services accessibles ...) qu'ils offrent.

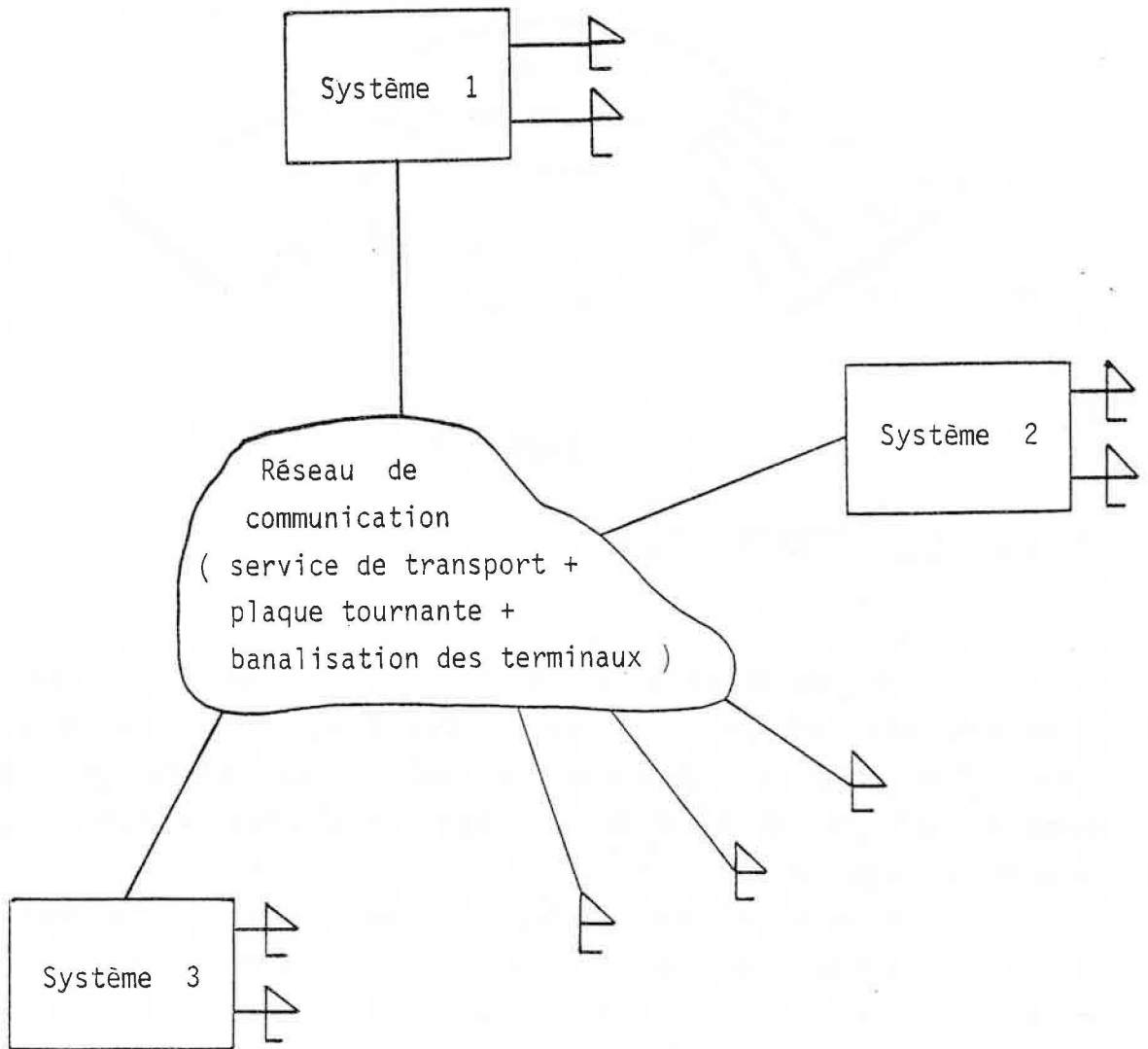


Figure 9

Parmi les systèmes présents sur le réseau, l'utilisateur doit explicitement choisir celui qui supporte l'application qu'il désire. Une fois réalisée la connexion, à travers le réseau, au système choisi, l'utilisateur peut communiquer avec celui-ci comme s'il était installé à un terminal local.

En conclusion, avec ce premier niveau de fonctions on propose une vision homogène des outils matériels alors que les outils logiciels restent diversifiés aux yeux des utilisateurs.

CHAPITRE IV

DEUXIEME NIVEAU DE FONCTIONS

1. POURQUOI UN SYSTEME RESEAU ?
2. QUELQUES EXEMPLES DE SYSTEMES RESEAU
 - 2.1. Le projet M2
 - 2.2. Le projet SOC
 - 2.3. Le projet IGOR
 - 2.4. UNIX
3. APPROCHE POUR LA CONCEPTION DU SYSTEME RESEAU DU C.C.R.
 - 3.1. Approche descendante
 - 3.2. Approche ascendante
4. LE LANGAGE DE COMMANDE EXTERNE (LCE)
 - 4.1. Choix d'un langage
 - 4.2. Interprétation du LCE
5. LANGAGES DE COMMANDE INTERNES (LCI)
 - 5.1. Coopération d'applications
 - 5.2. Interprétation du LCI
6. LE PROTOCOLE "CENTRE DE CALCUL REPARTI"
 - 6.1. Protocole de haut niveau
 - 6.2. Fonctionnalité du protocole CCR
 - 6.3. Gestion du protocole CCR

1. POURQUOI UN SYSTEME RESEAU ?

La vision du centre de calcul donnée à l'utilisateur par le réseau de communication présenté au chapitre précédent est celle d'une collection d'ordinateurs ou systèmes offrant des services et des ressources variés accessibles à partir de terminaux banalisés. L'utilisateur doit cependant connaître les langages de commande de tous les systèmes auxquels il veut accéder ainsi que le schéma de localisation des ressources sur les différents ordinateurs. En effet, les procédures de connexion logique aux différentes applications et leur mode d'utilisation sont dépendantes des systèmes hôtes.

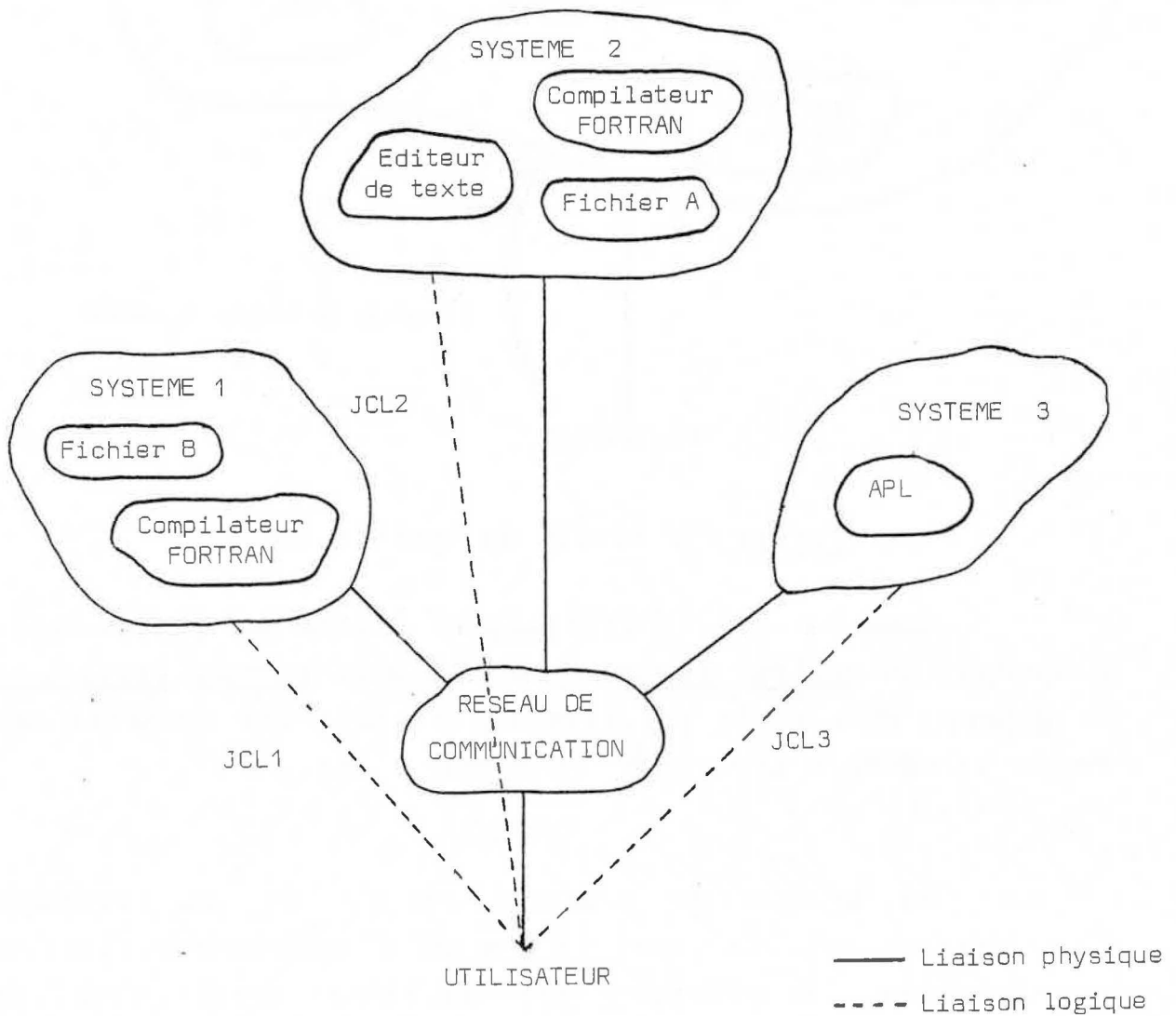


Figure 10

On sent poindre la nécessité d'un système réseau [ELOVITZ: GEN-19] qui ne prenne pas seulement à sa charge les aspects transmission et homogénéisation du matériel mais qui joue également un rôle d'intermédiaire plus actif entre les utilisateurs et les applications. Ce rôle est subordonné à la mise en oeuvre d'un langage permettant de s'adresser au "système réseau" tout entier.

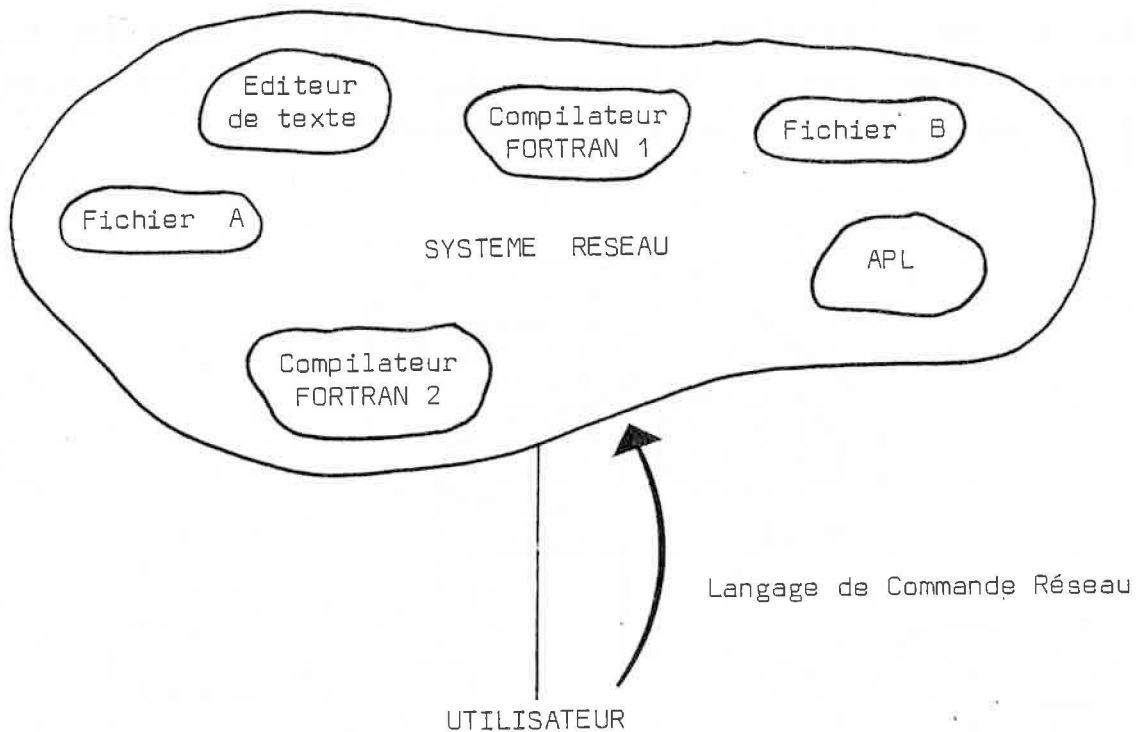


Figure 11: Vision du système réseau

Dans ce cas, l'utilisateur accède au réseau en se connectant au système réseau plutôt qu'à un système particulier du réseau. Il a accès aux services de tous les systèmes sans devoir procéder à un "LOGIN" de chacun d'eux.

Un langage de commande réseau est un instrument d'unification qui, du point de vue de l'utilisateur, remplace les langages de commande des systèmes accessibles. Les fonctions qu'il exprime sont très dépendantes de l'option

choisie quant à la vision que l'on souhaite offrir du réseau aux utilisateurs. Deux choix sont possibles:

-on veut laisser à l'utilisateur la conscience de l'existence d'un réseau; le langage de commande est l'instrument qui permet à l'utilisateur de "parler" au réseau. Dans ce cas il représente beaucoup plus qu'un langage de commande commun à tous les ordinateurs car il exprime des fonctions supplémentaires qui sont des requêtes réseau.

Exemples de requêtes réseau:

- transfert de fichier d'un site à un autre
- demande d'exécution d'un travail sur un site déterminé

La connaissance du réseau s'exprime par l'identification des sites impliqués qui figurent dans une commande en tant que paramètres. Cette connaissance peut être restreinte en rendant dans certains cas ces paramètres implicites, ou en donnant au "système réseau" la responsabilité de leur choix.

-on veut assurer la transparence complète du réseau pour l'utilisateur; le langage de commande paraît très proche d'un langage de commande "local". Dans ce cas, c'est le système qui prend entièrement en charge l'aspect réseau:

- localisation des éléments mis en jeu par une commande,
- détermination des sites impliqués,
- décision des transferts à effectuer,
- etc ...

Quel que soit le choix effectué, le système réseau gère un environnement réseau; suivant le cas, l'interprétation du langage de commande implique des actions plus ou moins complexes.

2. QUELQUES EXEMPLES DE SYSTEMES RESEAU

2.1. Le projet M2 (Multi-Mini-ordinateurs) [HEBENSTREIT: RES-9]

Ce projet a pour but la réalisation d'un système multi-mini-ordinateurs transparent pour l'utilisateur et dans lequel les différentes fonctions du logiciel sont confiées à des mini-ordinateurs spécialisés. On distingue ainsi:

- les "minis exécutants",
- le "mini maître" auquel sont connectés tous les périphériques d'entrée-sortie et qui supervise le fonctionnement global du système,
- le "mini fichier".

Tous les minis sont reliés entre eux par un bus unique.

Les minis peuvent être d'origines diverses et d'autre part un travail utilisateur (décrit dans un langage de scénario) doit pouvoir être exécuté sur n'importe quel mini exécutant. Pour permettre cette possibilité, tout programme utilisateur, écrit en langage évolué fait l'objet d'une compilation pour laquelle le langage objet est un langage commun LOC (Langage Objet Commun). Il existe, sur chaque mini, un transposeur c'est à dire un traducteur de LOC en langage machine.

2.2. Le projet SOC (Systèmes d'Ordinateurs connectés)

Ce projet [SOC: GEN-20] s'intéresse plus particulièrement aux problèmes de type logiciel et à l'utilisation du réseau par les utilisateurs classiques d'ordinateurs. Son ambition est néanmoins limitée par l'aspect homogène du réseau: les ordinateurs connectés sont identiques tant du point de vue matériel (IBM 360 ou 370) que logiciel (OS 360).

Le réseau est géré par le Système Réseau constitué d'un ensemble de programmes répartis exécutés sur chacun des ordinateurs du réseau, distincts de l'"operating system" local et formant sur chacun d'eux un tout dont les principes et les fonctions sont les mêmes. Nous n'aborderons pas ici l'aspect intégration des constituants du Système Réseau à un "operating system"; nous dégagerons plutôt les fonctions du Système Réseau:

- communication entre les ordinateurs du réseau,

- mise en oeuvre d'un langage de commande appelé langage externe (LE) pour les utilisateurs. Un ensemble de commandes exprimées en LE constitue un Travail Réseau dont l'exécution implique la compilation du LE en un langage plus élémentaire appelé langage interne (LI). Le LI est également utilisé pour les dialogues entre les différents ordinateurs. Le compilateur génère autant de blocs de LI qu'il y a d'ordinateurs mis en jeu par le travail réseau. Les caractéristiques du LI sont les suivantes:

- . chaque ordre LI est exécuté sur une seule machine.

- . si des échanges sont nécessaires entre deux ordinateurs pour exécuter les commandes d'un travail réseau, le compilateur génère des ordres qui sont exécutés sur chacune des deux machines.

- . l'exécution de chaque bloc de LI est assurée sur la machine concernée par un interpréteur.

Les possibilités offertes à l'utilisateur sont:

- le transfert de données d'un site à un autre,
- l'exécution de travaux à distance.

Lorsqu'il prépare un travail réseau, l'utilisateur est tenu de préciser les ordinateurs mis en jeu; on constate donc un manque de transparence du réseau.

2.3. Le projet IGOR (Interpréteur Général Orienté Réseau)

Cette étude entre dans le cadre du développement d'un système réseau pour le réseau Cyclades. De même que pour SOC, deux langages ont été développés [SERGEANT: LAN-1]:

- un langage externe LE qui permet la description des ressources et des actions désirées par l'utilisateur quel(s) que soi(en)t le(s) site(s).
- un langage intermédiaire produit par la compilation du LE. Sur chaque ordinateur du réseau, il existe un interpréteur de LI qui exécute le code LI.

2.4. UNIX

Dans cet exemple [MANNING: RES-13], le langage des requêtes est une extension d'un JCL d'une machine locale par addition de commandes réseau; cette solution reste liée au choix de la machine et son changement implique le changement du langage de commande.

3. APPROCHE POUR LA CONCEPTION DU SYSTEME RESEAU DU CCR

Du point de vue utilisateur le terme "Centre de Calcul Réparti" est impropre dans la mesure où l'on désire que le centre de calcul apparaisse comme un calculateur unique avec son langage de commande et ses ressources; l'aspect réseau est complètement ignoré de l'utilisateur.

Par contre, d'un point de vue système, il évoque un certain nombre de concepts nécessaires pour permettre cette transparence.

Pour aboutir finalement à la mise en évidence de fonctions à l'intérieur du système réseau du CCR, notre

approche va s'effectuer dans deux directions convergentes :

1. prendre en compte les besoins et les désirs des utilisateurs et définir des fonctions pour les résoudre; l'approche est alors descendante.

2. profiter de l'expérience acquise dans le domaine des communications et tenter d'élargir les concepts déjà développés pour rejoindre et améliorer l'aspect utilisation d'un système réseau; dans ce cas, l'approche est ascendante.

Cette phase de travail va permettre de préciser les objectifs initiaux.

3.1. Approche descendante

Elle a pour point de départ les aspirations des utilisateurs et la façon dont on désire que le centre de calcul soit perçu par eux.

. les utilisateurs:

ils désirent accéder à des applications en utilisant éventuellement des fichiers selon une procédure qui soit la plus simple possible. A cet effet, on va mettre à leur disposition un langage de commande qui substitue, à leurs yeux, tous les langages existant jusqu'alors dans le centre de calcul, principalement dans leurs fonctions de gestion de ressources (fichiers, appareils) et d'appel de processeur (lancement d'une application). De plus, ce langage masquera totalement la présence d'un réseau.

L'application sera la seule notion accessible de l'extérieur du système; le rôle du système réseau est de délivrer l'utilisateur des contraintes spécifiques aux hôtes constituant le système.

Dans ce contexte, et sans préciser davantage, le système réseau se présente à la fois comme:

-interlocuteur des utilisateurs, c'est à dire interprète du langage de commande, analysant les requêtes des utilisateurs et réalisant la connexion de chaque utilisateur avec l'application choisie par celui-ci,

et

-interlocuteur des applications ou plus exactement des systèmes d'exploitation supportant les applications.

Les échanges utilisateur-application s'effectuent par l'intermédiaire de cet interlocuteur et sous son contrôle.

Cette description très superficielle est un premier élément pour appréhender le système; elle introduit cinq rôles; chaque rôle est tenu par une entité indépendante des autres, la localisation géographique de chacune d'elles étant quelconque (répartie, localisée sur un site) et leur fonctionnement étant asynchrone. Ces rôles sont:

1. utilisateur
2. adaptateur entre le système d'exploitation supportant les accès utilisateurs et le reste du système
3. interprète du langage de commande
4. application
5. adaptateur entre le système d'exploitation supportant l'application et le reste du système; en effet, les applications situées sur les différents hôtes interconnectés sont accessibles via le système d'exploitation de ces hôtes et le langage de commande associé, une des hypothèses de notre étude étant de conserver le logiciel spécialisé existant sur chaque hôte.

Ces rôles ne sont pas indépendants les uns des autres; il y a donc nécessité d'introduire des conventions de communication et de dialogue. C'est à ce niveau qu'intervient l'expérience et les enseignements tirés des recherches en matière de réseau.

Schématiquement, les entités définies auront les relations indiquées par les traits pleins dans la figure ci-dessous.

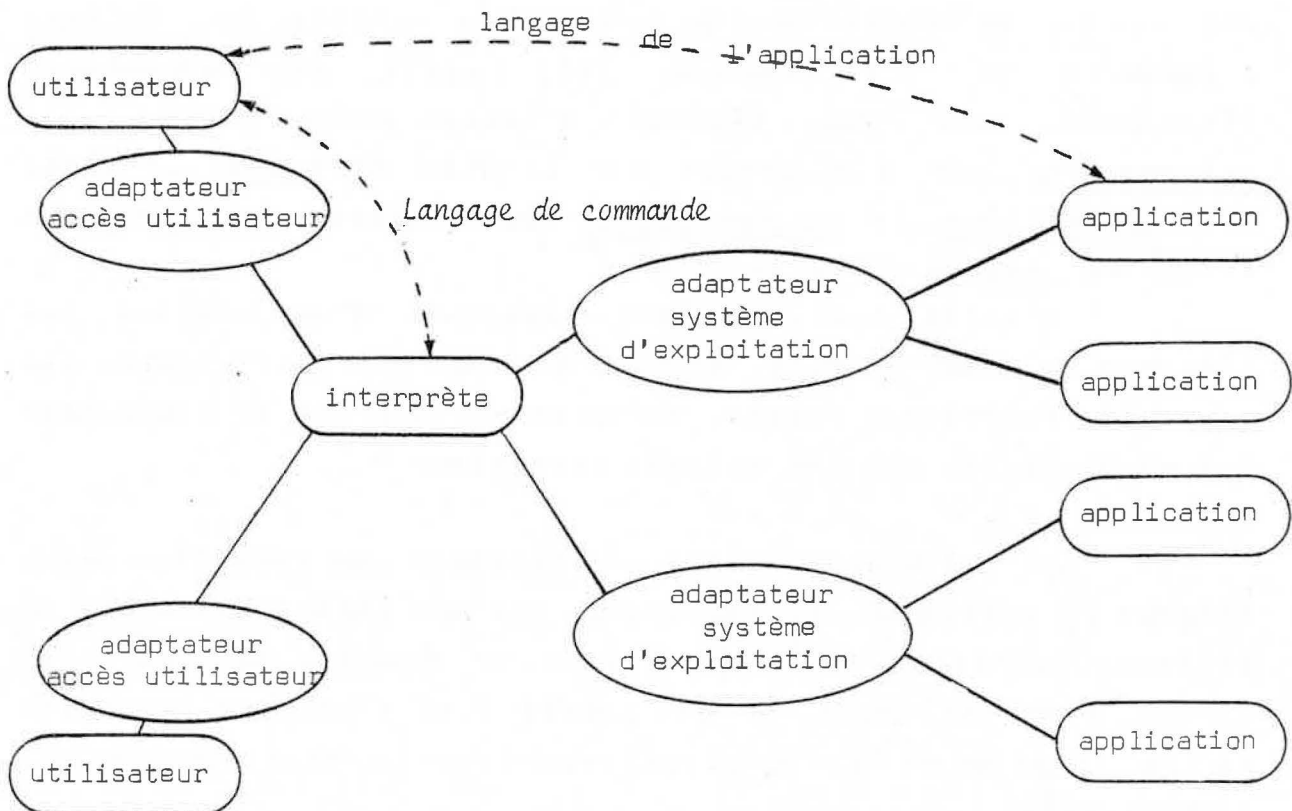


Figure 12

3.2. Approche ascendante

Cette approche est une approche réseau; il s'agit en effet de gérer un environnement réparti. Le propre d'un système réparti est qu'il n'existe pas à l'intérieur de ce système une entité qui ait connaissance de l'état global du système: en d'autres termes, on ne distingue pas de "maître" au sein d'un

système réparti [LE LANN: GEN-3]. En particulier, deux entités qui désirent communiquer ne peuvent pas s'en remettre à un "maître" qui contrôle leurs échanges. En l'absence de supérieur hiérarchique commun qui fixe des conventions d'échange et serve d'intermédiaire dans leurs rapports, ces entités qui disposent par ailleurs d'une autonomie complète doivent se définir des règles de comportement mutuel; ces règles définissent un protocole.

Dans ce contexte une information n'est pas seulement constituée de données destinées à des applications, mais aussi d'une "étiquette" qui définit sa provenance, sa destination, etc La communication entre les entités qui doivent s'échanger de l'information fait appel aux mécanismes développés pour les réseaux d'ordinateurs; toutes les interactions vont s'effectuer sur la base de messages. Quant aux conventions de communication elles vont s'exprimer en termes de langage et protocole.

D'autre part, il est important d'uniformiser les méthodes de communication; même si des entités sont locales les unes par rapport aux autres, elles communiqueront en respectant les mêmes règles que les entités distantes.

Il est bien mal aisé de discerner une frontière entre langage et protocole. Un protocole, qui est déjà en lui-même un langage, constitue un support pour un langage de plus haut niveau, assurant, dans les meilleures conditions, le transport de ce langage et la synchronisation des interlocuteurs qui l'utilisent.

Le contenu des messages véhiculés peut être schématisé ainsi:

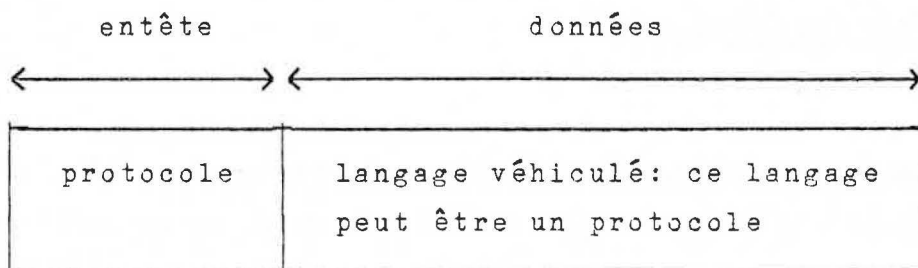


Figure 13

Dans la suite, nous réserverons les termes :

- langage pour les dialogues entre entités terminales :
utilisateur, application, interprète,

et

- protocole pour les conventions d'échanges servant au
transport de ces langages.

4. LE LANGAGE DE COMMANDE EXTERNE

Les utilisateurs ont accès aux applications via le système réseau par l'intermédiaire d'un langage leur permettant de formuler des requêtes. Ce langage de commande représente la vision que les utilisateurs ont du CCR; pour cette raison il est appelé langage de commande externe (LCE).

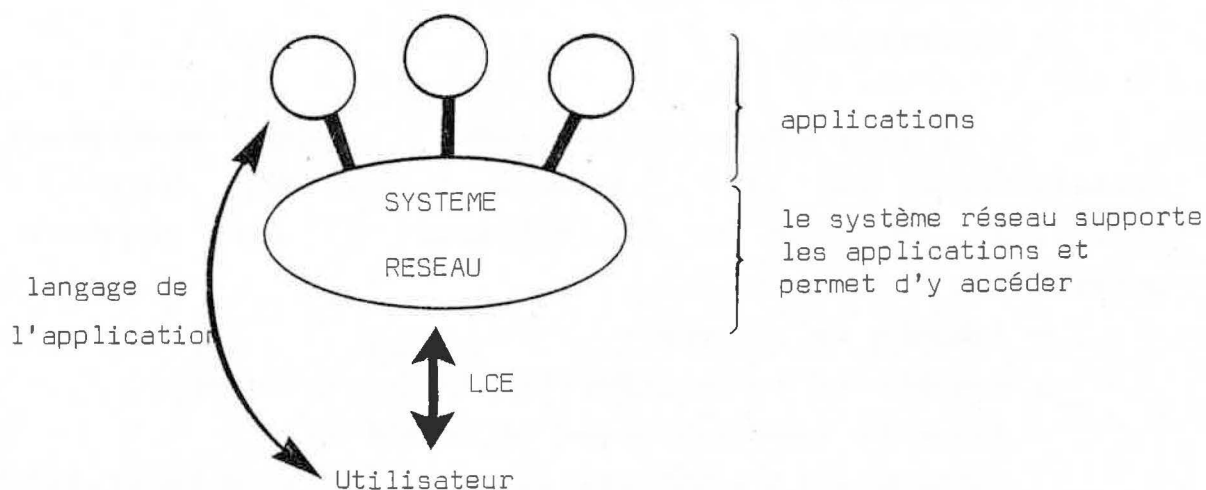


Figure 14

Le LCE intervient donc comme langage de dialogue entre les utilisateurs et le système réseau. Sa raison d'être

est de masquer aux utilisateurs la variété et l'ésotérisme des langages de contrôle existant sur les ordinateurs du réseau et de leur offrir une vision plus agréable du centre de calcul grâce à un gain de simplicité et d'homogénéité.

Pour faciliter la compréhension du système et permettre à l'utilisateur de s'en servir aisément, il est important de lui donner une structure simple et facile à comprendre. Il ne s'agit en aucun cas de réaliser un super-langage de commande convertible dans les langages propres à chaque ordinateur mais plutôt des éléments indispensables pour permettre à l'utilisateur d'accéder à l'application de son choix.

4.1. Choix d'un langage

Ce langage doit être le plus proche possible d'un langage naturel pour la facilité d'apprentissage et d'emploi. D'autre part, comme il s'agit d'un outil de dialogue, il doit être à la fois interprétatif et interactif [RAYMOND:LAN-7]. Son objectif essentiel étant de permettre l'accès à des applications, il doit offrir les requêtes de:

- connexion à une application
- déconnexion

Il doit également permettre d'exprimer simplement les enchaînements les plus fréquents (compilation suivie d'une exécution). En plus de ces requêtes, il doit contenir des utilitaires tels que:

- demande de l'heure,
- demande de renseignements sur un utilisateur,
- envoi de messages entre utilisateurs,
- fabrication et utilisation de procédures cataloguées,
- utilisation des ressources (fichiers, appareils),
- etc ...,

Parmi cet éventail de commandes on pressent déjà une distinction entre:

- celles qui nécessitent des échanges entre les entités du système et qui, de ce fait, doivent prendre en compte l'environnement réseau. L'exemple le plus typique est la demande de connexion à une application qui, pour être satisfaite, exige un certain nombre d'accords préalables.

- celles pour lesquelles aucun échange n'est nécessaire parce que l'interprète possède dans son propre environnement d'exécution tous les éléments dont il a besoin.

Il existe un certain nombre de langages de commande réseau:

- NJCL (Network Job Control Language) [RAYMOND: LAN-7] dont la forme est dérivée de PASCAL.

- NCL développé dans le cadre du projet SOC [DU MASLE: LAN-8].

- LE étudié pour une utilisation sur CYCLADES [SERGEANT: LAN-6].

Ces langages de commande sont des langages très puissants, dont les fonctions dépassent largement le cadre du simple accès à des services. Ce sont de véritables langages de programmation, dont le but est beaucoup plus l'écriture d'application réseau que la mise à disposition de services à des utilisateurs non initiés. Ils tentent de résoudre de façon globale le problème du langage de commande dans un réseau général.

Le choix d'un langage implique un compromis entre sa facilité d'emploi et ses possibilités d'exprimer les fonctions désirées. Dans notre cas, nous avons opté pour un langage simple, créé de toutes pièces à partir des besoins des utilisateurs et extensible [TOSAN: CCR-2].

4.2. Interprétation du LCE

L'interprète du LCE est le véritable interlocuteur des utilisateurs. Toutes les commandes des utilisateurs lui parviennent car il est le seul à connaître les actions qu'elles induisent.

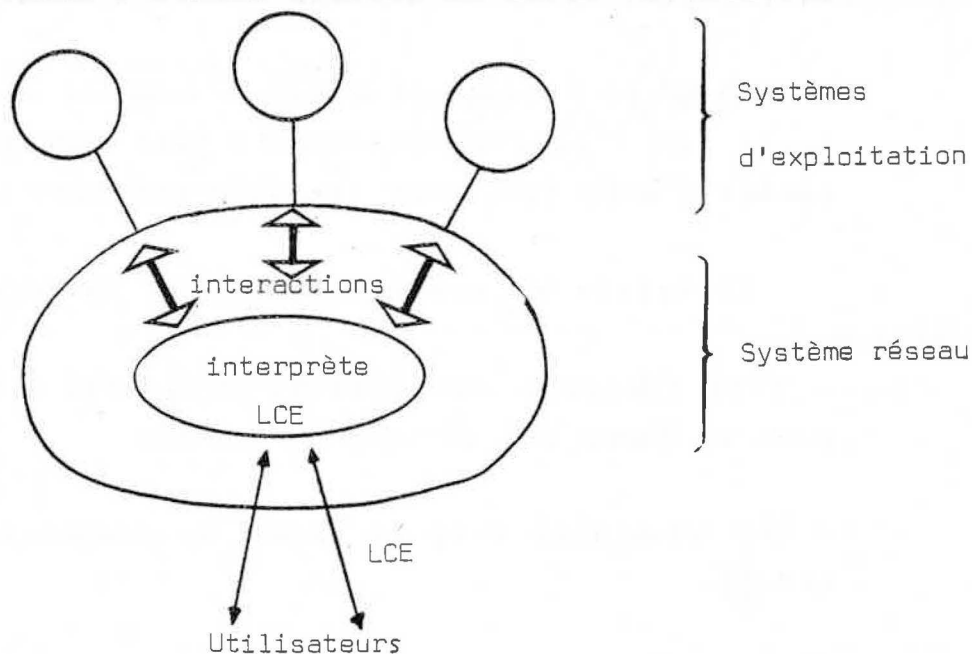


Figure 15

Il joue un rôle centralisateur vis à vis des utilisateurs. C'est lui qui détient la liste de tous les utilisateurs potentiels du centre de calcul. A chacun d'eux est associé un environnement qui le personnalise vis à vis de l'interprète; toute commande est interprétée en fonction de l'utilisateur qui l'a émise, c'est à dire en fonction de son environnement. En d'autres termes, la syntaxe du LCE est générale mais sa sémantique est personnalisée pour chaque utilisateur, qui dispose ainsi d'un LCE qui lui est propre et qui définit sa vision du centre de calcul (les applications auxquelles il a accès, ses prérogatives, etc ...). Les environnements des utilisateurs constituent donc des données essentielles pour l'interprète.

Il est également nécessaire pour l'interprète qu'il connaisse un minimum d'informations concernant les applications; un aspect important de son rôle est en effet d'interpréter des requêtes d'accès à des applications. A son niveau, sont donc recensées à un instant donné toutes les applications disponibles dans le centre.

Ces deux types d'information reflètent bien le rôle de médiateur que joue l'interprète dans l'initialisation des dialogues des utilisateurs avec les applications.

Après avoir esquissé le contexte de travail de l'interprète nous allons aborder quelques aspects de ses fonctions.

Nous n'insisterons pas sur l'analyse syntaxique qui ne pose pas de problème majeur car il semble évident que la syntaxe du langage doive être entièrement connue de l'interprète.

Par contre, au niveau de l'analyse sémantique on peut choisir des orientations bien différentes:

- a. L'interprète contrôle toute la sémantique du langage.

Il possède donc toutes les informations utiles relatives aux entités (utilisateurs ou applications) mises en cause par la commande. Prenons l'exemple d'une commande de connexion à un service avec comme paramètre un nom de fichier. Avec l'option choisie, le nom du fichier doit être contrôlé par l'interprète avant l'exécution de la commande, à savoir la connexion proprement dite. Cela implique que l'interprète connaisse, au moins en partie, la gestion de fichier qu'utilise l'application. Cette solution semble démesurée et peu extensible car l'introduction d'une nouvelle application entraîne de profondes modifications de l'interprète.

- b. L'interprète contrôle en partie seulement la sémantique du langage.

Certains paramètres, bien que faisant partie du LCE, sont transparents pour lui, leur analyse sémantique étant reportée sur d'autres entités du système. L'idée est de localiser l'interprétation de ces paramètres aux applications elles-mêmes, puisque ce sont elles qui les utilisent et qui, par voie de conséquence, en connaissent mieux que quiconque la signification. Il s'avère alors nécessaire d'introduire des éléments supplémentaires dans le système, ces éléments étant chargés d'un complément d'interprétation en tenant compte à la fois des applications et des systèmes d'exploitation supportant ces applications; ces éléments constituent des enveloppes pour les applications; ces enveloppes servent d'interface entre le système réseau, le système d'exploitation local et les applications, d'où leur qualificatif d'"enveloppes réseau".

Notre choix s'est porté sur la deuxième solution car elle limite l'intrusion de l'interprète dans ce que l'on peut considérer comme le "domaine privé" des applications, et elle est facilement extensible. De plus, elle permet une certaine décentralisation du contrôle au niveau du système réseau.

Le schéma fonctionnel se précise: il est représenté par la figure 16.

Le dernier volet de l'interprétation est l'exécution proprement dite de la commande; deux cas peuvent se présenter:

- a. l'exécution ne nécessite pas la collaboration d'autres entités.

Un exemple est donné par la commande HELP qui permet à l'utilisateur d'obtenir des renseignements tels que: applications disponibles, rappels sur la syntaxe du LCE, etc Toutes ces informations sont contenues dans l'environnement de l'utilisateur; l'interprète qui dispo-

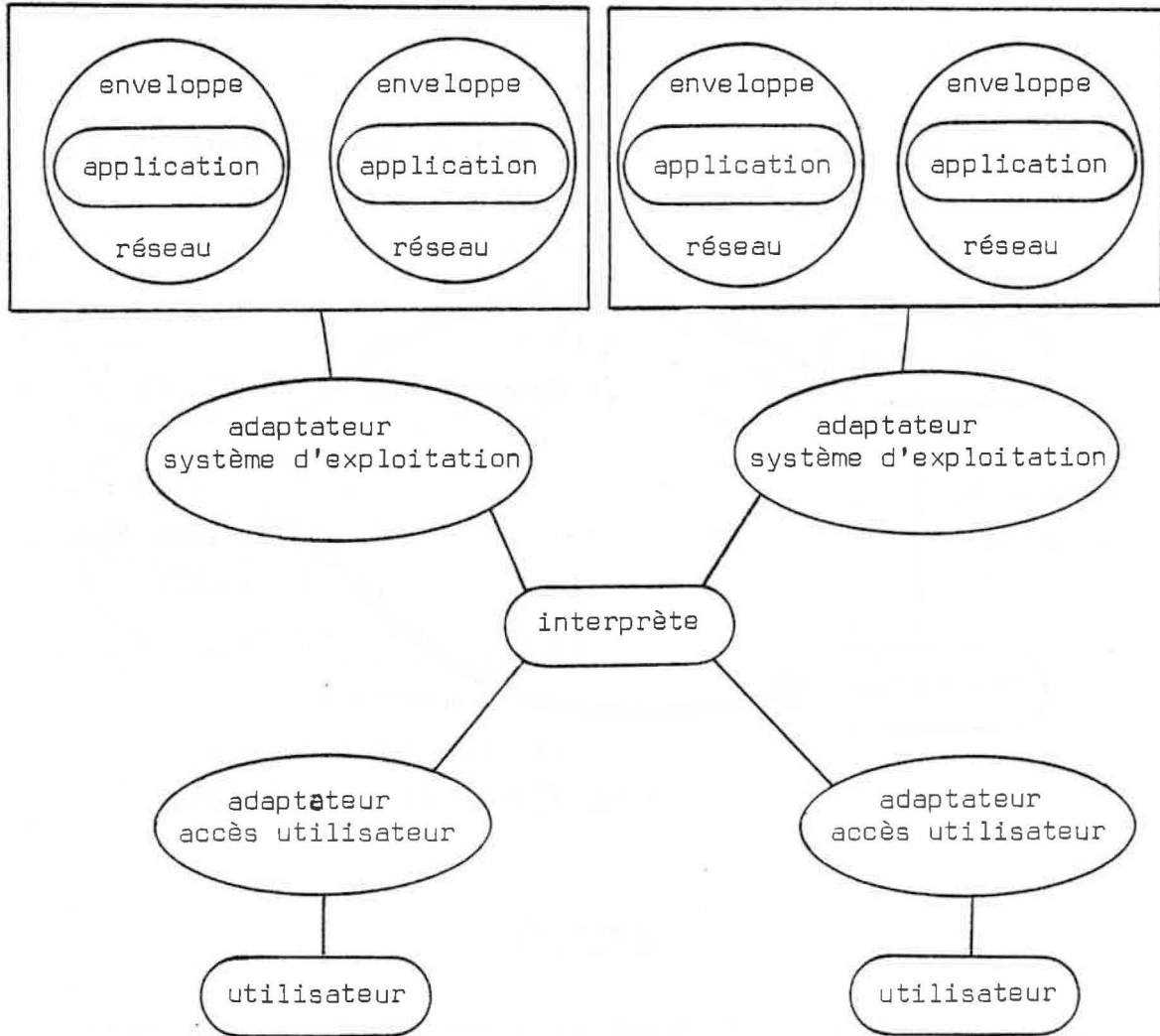


Figure 16: Schéma fonctionnel

se de cet environnement n'a besoin du concours d'aucune autre entité pour répondre à la demande de l'utilisateur.

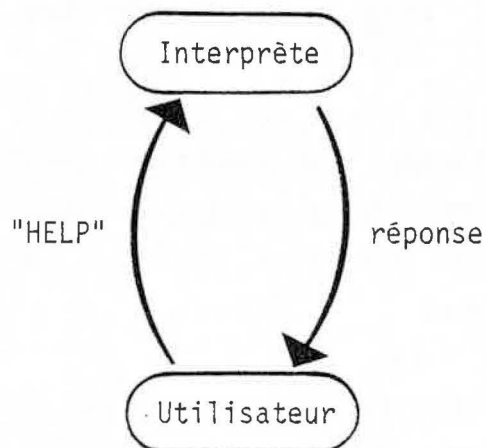


Figure 17

b. l'exécution nécessite des interactions entre certaines entités.

Le cas typique est celui de la demande de connexion à une application.

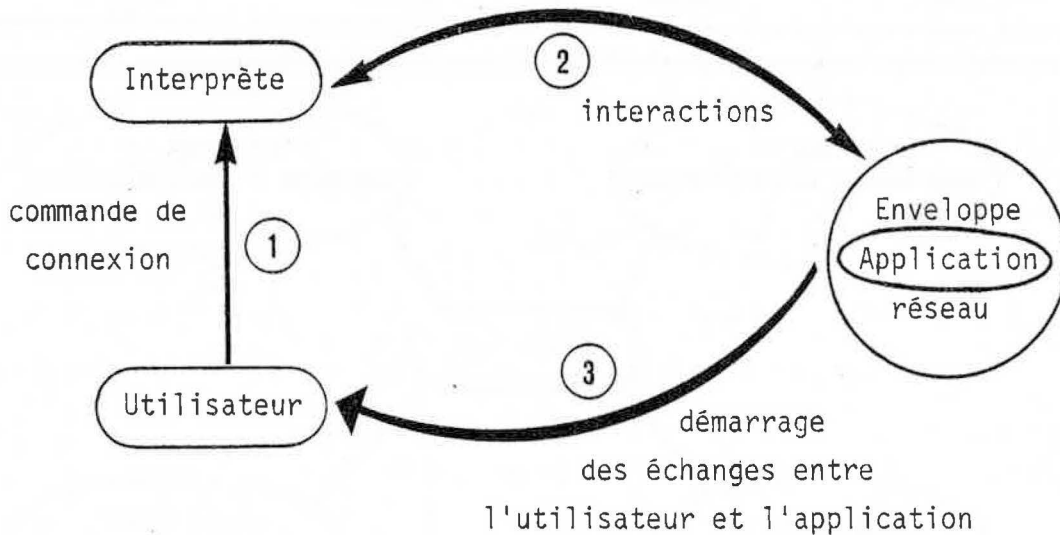


Figure 18

Les phases 1, 2, 3, vont se succéder dans le temps.

Les interactions entre l'interprète et l'enveloppe réseau se traduisent par des échanges d'informations entre eux. Ces informations engendrent de part et d'autre des actions dont le séquençement doit être respecté (mise à jour d'environnement, etc ...); une synchronisation entre les deux entités est donc indispensable: elle s'effectue sur la base des messages échangés.

A un moment donné, la commande devra parvenir au système d'exploitation supportant l'application sous la forme qu'il attend, c'est à dire dans son langage de contrôle. Deux orientations sont possibles:

- l'interprète connaît le JCL de chaque hôte et s'adresse à celui-ci dans ce langage. Il distille au système d'exploitation les commandes nécessaires en JCL pour

l'exécution de la commande comme s'il s'agissait d'un utilisateur local.

- l'interprète s'adresse aux systèmes d'exploitation dans un langage prédéfini, commun à tous, la traduction étant à la charge de l'enveloppe réseau qui sert de médiateur entre l'application, le système d'exploitation local et le système réseau.

Quel que soit le choix effectué, le rôle d'adaptateur disparaît du schéma fonctionnel (figure 16), la gestion de ses fonctions étant supportée, soit par l'interprète, soit par l'enveloppe réseau des applications, soit par les deux.

5. LANGAGES DE COMMANDE INTERNES

5.1 Coopération d'applications

L'existence de langages de commande internes est liée à la mise en oeuvre d'applications réparties. Une application répartie consiste en un ensemble de composants (modules ou éléments de logiciel) inter-agissant à travers des moyens de communication dans un but de coopération, chaque composant étant associé à un site d'exécution. Dans le cadre de notre étude, le terme coopération d'applications paraît mieux choisi car il suggère que ce sont des applications déjà existantes et autonomes qui inter-agissent alors que pour une application répartie chaque composant n'offre pas nécessairement un service disponible sans le concours des autres composants.

La coopération d'applications est une étape supplémentaire dans la conception du CCR. Elle s'avère indispensable, dans certains cas, pour assurer à l'utilisateur la parfaite transparence de localisation des ressources. Le besoin le mieux pressenti se situe au niveau de la gestion de fichiers. En

effet, si l'on ne veut pas limiter l'utilisation d'un fichier à son site de création, il devient nécessaire d'introduire une coopération entre les différents systèmes de gestion de fichiers pour réaliser une gestion globale des fichiers. Cela suppose donc des interactions entre les différents systèmes de gestion de fichiers, et ceci entraîne l'existence de conventions de communication c'est à dire d'un langage de dialogue. Le débat, déjà engagé au sujet du LCE, réapparaît. Faut-il:

- que chaque système de gestion de fichiers connaisse le JCL des autres systèmes, pour pouvoir exprimer ses échanges avec eux dans leur propre langage immédiatement compris par eux. Chaque système se conforme à l'interface que les autres imposent.

- définir un langage intermédiaire commun à tous les systèmes de gestion de fichiers, suffisamment puissant pour exprimer toutes les fonctions désirées et dans lequel seraient exprimées toutes les interactions.

La deuxième solution est plus élégante car elle est générale et permet l'addition ou la modification de l'un des systèmes de gestion de fichiers sans requérir la modification de chacun des autres. D'autre part, elle va dans le sens d'une homogénéisation au niveau des langages véhiculés par le réseau.

Chaque fois que des applications désirent coopérer, elles utilisent le langage intermédiaire défini entre elles. Ce langage peut être unique et général au niveau du CCR ou bien particulier et donc adapté à chaque cas de coopération. Il est formalisé et peut être aussi complexe qu'on veut puisqu'il est invisible à l'utilisateur. Il sert au dialogue entre des entités internes d'où son nom de langage de commande interne (LCI).

5.2. Interprétation du LCI

Cette interprétation consiste à traduire le langage de commande interne dans les différents langages de commande des systèmes d'exploitation qui supportent les applications qui coopèrent. Inversement, il s'agit également de traduire le JCL local dans le langage commun.

Ces fonctions sont remplies par l'enveloppe réseau qui joue donc un rôle d'interprète des différents langages utilisés pour les dialogues non prévus par le système d'exploitation. Dans le cadre des échanges entre un utilisateur et une application, le rôle de l'enveloppe réseau est sans effet puisque ces deux entités s'expriment dans un même langage.

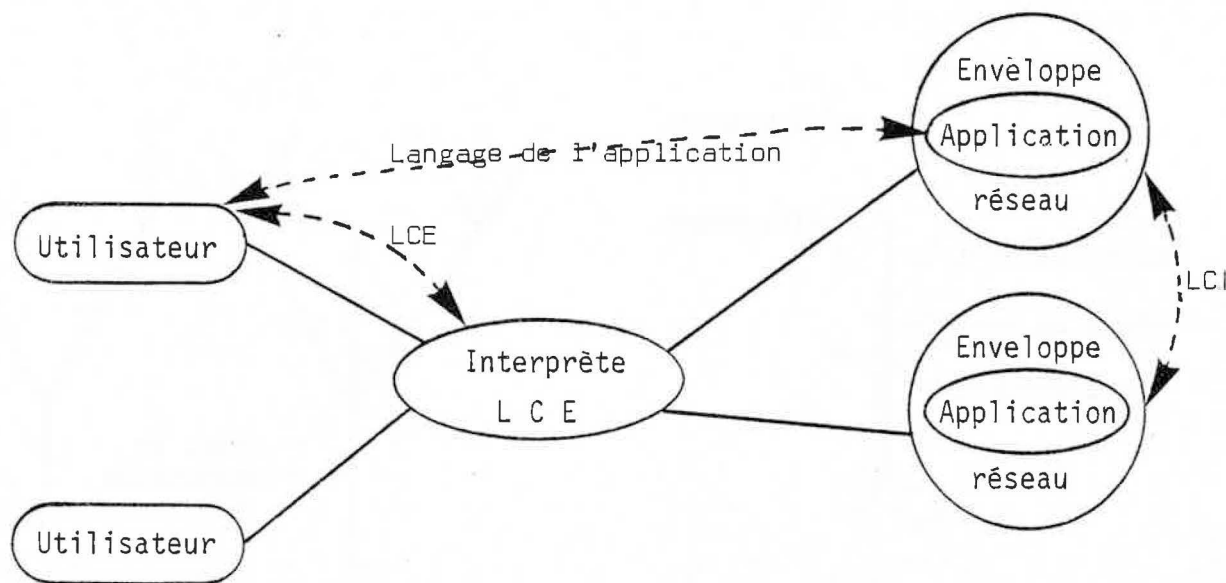


Figure 19: Schéma des langages de dialogue

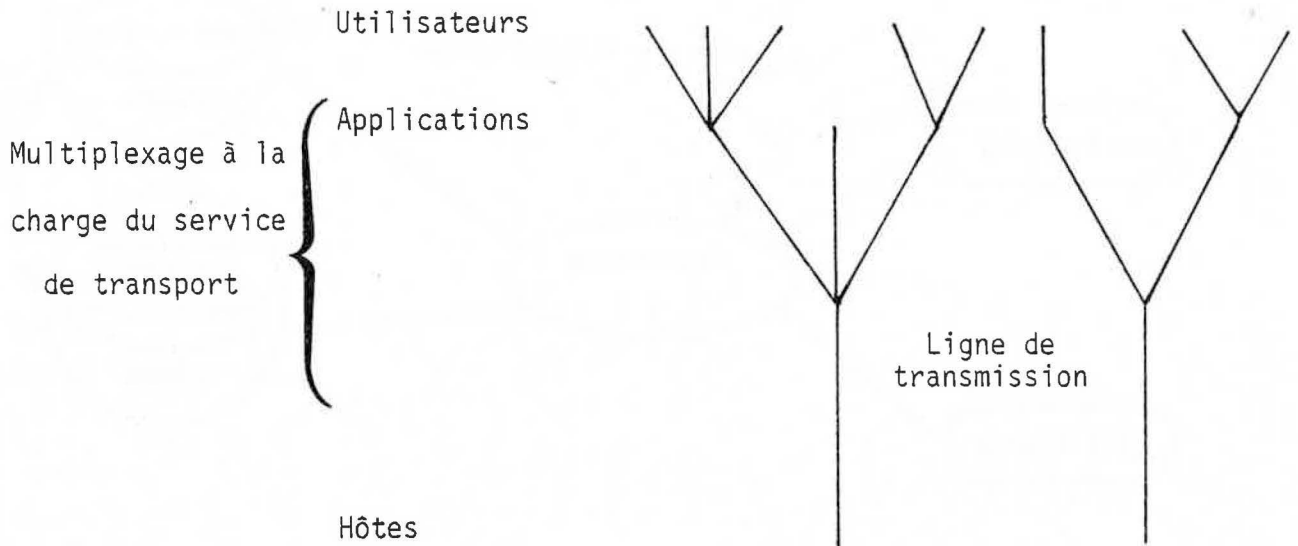
L'ensemble application et enveloppe réseau de cette application sera appelé, dans la suite, application réseau.

6. LE PROTOCOLE "CENTRE DE CALCUL REPARTI"

Pour régir les dialogues entre les différents interlocuteurs du CCR, nous avons vu qu'il était nécessaire d'introduire un protocole d'échange; ce protocole appelé "protocole centre de calcul réparti" permet à la fois le transfert d'informations et la synchronisation entre toutes les entités. Toute entité qui désire communiquer avec d'autres à l'intérieur du CCR doit respecter ce protocole.

6.1. Protocole de haut niveau

Par rapport aux protocoles déjà présentés dans le chapitre III, celui-ci offre une vision plus fine du CCR; il introduit en effet un niveau de multiplexage supplémentaire ainsi que le montre la figure.



Figure_20: Le multiplexage et les différents niveaux de protocole

Le niveau protocole CCR est un niveau intermédiaire entre le niveau langage (utilisateur \leftrightarrow interprète, utilisateur \leftrightarrow application, interprète \leftrightarrow enveloppe réseau, enveloppe réseau

←→ enveloppe réseau) et le service de transport dont les fonctions ne sont pas suffisamment sophistiquées pour permettre des communications entre entités de haut niveau. Il intervient comme support pour le transport de LCE, de LCI (ou de JCL) et des langages des applications. Il met en place une structure de dialogue alors que les langages expriment le dialogue.

Il s'agit d'un protocole de niveau utilisateur; tout échange d'information s'effectue pour le compte d'un utilisateur. L'identification de l'utilisateur fait partie intégrante de l'entête; cette caractéristique rapproche le système réseau d'un système transactionnel.

6.2. Fonctionnalité du protocole CCR

Les services offerts par le protocole CCR recouvrent de multiples aspects:

- . mécanisme d'adressage des entités qui inter-agissent
- . établissement et rupture de dialogue
- . transfert de données sur les liaisons établies
- . contrôle d'intégrité
- . contrôle de sécurité
- . aspect administratif

1. Mécanisme d'adressage des entités qui inter-agissent

Les entités qui inter-agissent sont, nous le rappelons:

- les utilisateurs (avec l'interprète ou avec les applications)
- l'interprète du LCE
- les applications réseau

Le protocole CCR offre un moyen de nommer ces entités: ce moyen permet de spécifier les interlocuteurs impliqués dans les différents dialogues.

2. Etablissement et rupture de dialogue

Pour pouvoir démarrer, un dialogue doit être initialisé par l'établissement d'un lien logique reliant les deux interlocuteurs, c'est à dire l'association de deux entités. Ce lien ne peut être créé que si les deux interlocuteurs y consentent; il est maintenu pendant tout le temps des échanges et rompu à la demande de l'un des deux interlocuteurs. Ce service réalise un contrôle de connexion.

3. Transfert de données sur les liaisons établies

Une fois qu'un lien a été créé entre deux interlocuteurs, l'un et l'autre peuvent effectuer des demandes d'émission et de réception de données à destination ou en provenance de leur vis à vis. Ces échanges de données sont accompagnés d'un contrôle de dialogue dans le but de synchroniser l'émetteur et le récepteur. En effet, dans la gestion d'un dialogue, il n'y a pas seulement le besoin d'un établissement et d'une rupture de dialogue, mais également de savoir à tout moment quel est l'interlocuteur qui peut parler et par voie de conséquence quel est celui qui doit écouter.

4. Contrôle d'intégrité

Le protocole prend en charge de multiples contrôles apportant ainsi une plus grande fiabilité. Il assure un contrôle de flux qui adapte le débit de l'émetteur à la capacité du récepteur. Il gère également des mécanismes

de rattrapage pour protéger le système contre des défaillances éventuelles (pas de réponse d'un service lorsque l'utilisateur a demandé une déconnexion) et lui permettre de se retrouver dans un état cohérent.

5. Contrôle de sécurité

Il s'agit d'effectuer une sorte de supervision des droits attachés à chacun des interlocuteurs afin de vérifier que ceux-ci ne vont pas être outrepassés par une action quelconque.

6. Aspect administratif

Il est facile d'intégrer à la gestion du protocole des fonctions de comptabilisation de diverses valeurs (temps de connexion à un service, nombre d'échanges effectués, nombre de connexions) permettant un certain contrôle administratif des utilisateurs.

6.3. Gestion du protocole CCR

Toutes les entités qui désirent échanger de l'information à l'intérieur du CCR doivent respecter le protocole CCR. Cependant, il est important de distinguer les fonctions remplies par ces entités et la gestion du protocole. C'est pourquoi nous introduisons pour chaque entité un niveau supplémentaire assumant cette gestion.

a. Gestion du protocole pour le compte de l'utilisateur.

L'utilisateur ne doit pas percevoir l'existence du protocole; ce protocole est plus exactement un protocole d'échange entre le terminal virtuel de l'utilisateur et le système. Il est donc naturel qu'il soit géré par l'émulateur du terminal virtuel. (voir chapitre III § 2.1.2).

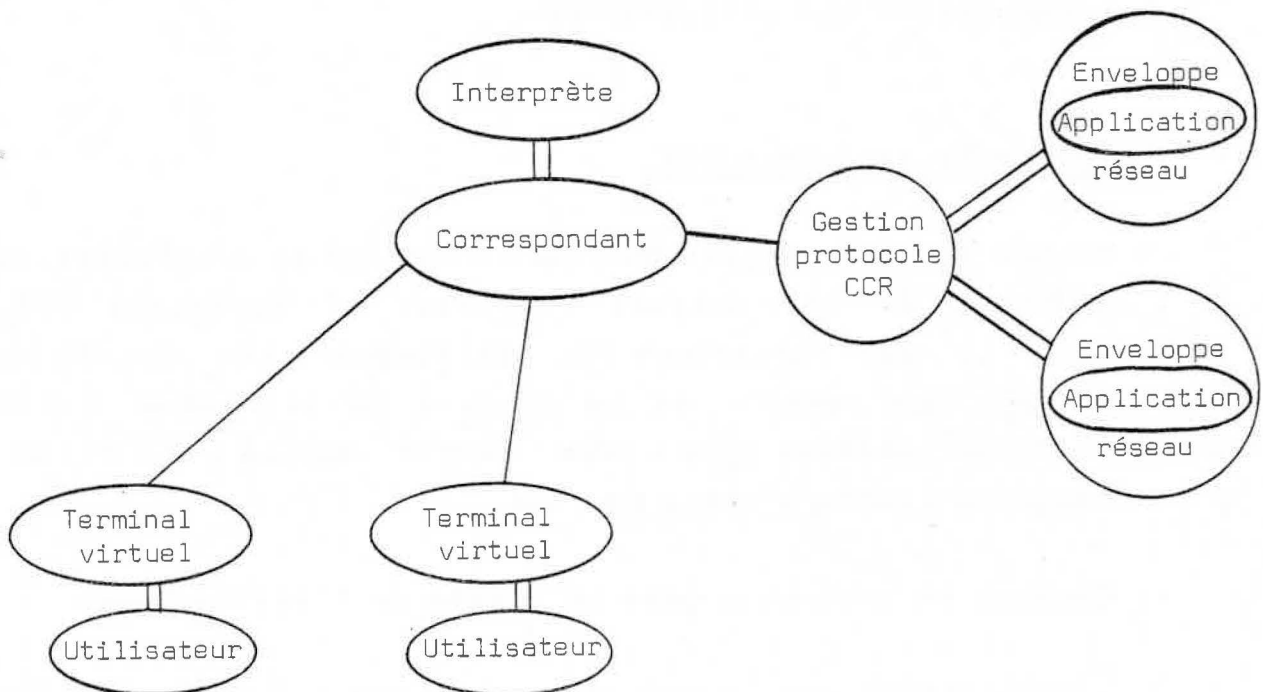
b. Gestion du protocole pour le compte des applications et des enveloppes réseau.

Les applications sont situées sur les différents hôtes. Il existera donc sur chaque hôte un module de gestion du protocole CCR

c. Gestion pour le compte de l'interprète du LCE.

Nous avons vu que l'interprète est un élément centralisateur dans le système CCR et que certaines de ses actions concernent la gestion des liaisons. Nous allons déporter sur l'entité qui gère le protocole pour le compte de l'interprète, la gestion de ces liaisons. Cette entité sera également chargée de l'aiguillage entre tous les éléments du CCR. Nous lui donnons le nom de correspondant.

L'ensemble des entités est représenté par la figure suivante:



== interface
— voie logique de communication

Figure_21

Seul le correspondant a une vision globale des utilisateurs et des applications.

Le gestionnaire de terminal ne connaît qu'une entité: le correspondant. Quant aux enveloppes réseau, elles s'adressent toujours au correspondant pour envoyer des plis aux utilisateurs ou aux autres applications dans le cas de coopération.

CHAPITRE V

ARCHITECTURE LOGIQUE

1. PRESENTATION DES CONCEPTS. UTILISES
 - 1.1. Architecture par niveaux
 - 1.2. Système intégré
2. RAPPEL DU RÔLE DE LA FONCTION FRONTALE
3. LES VOIES LOGIQUES DE COMMUNICATION
4. DYNAMIQUE DES ECHANGES
 - 4.1. L'émulateur de terminal virtuel
 - 4.2. Le niveau supérieur de la méthode d'accès des ordinateurs connectés
 - 4.3. Le correspondant
5. NATURE DES DIALOGUES
 - 5.1. L'utilisateur
 - 5.2. L'interprète
 - 5.3. L'application réseau

1. PRESENTATION DES CONCEPTS UTILISES

1.1 Architecture par niveaux

Cette technique consiste en un assemblage hiérarchisé de niveaux comme le montre la figure 22.

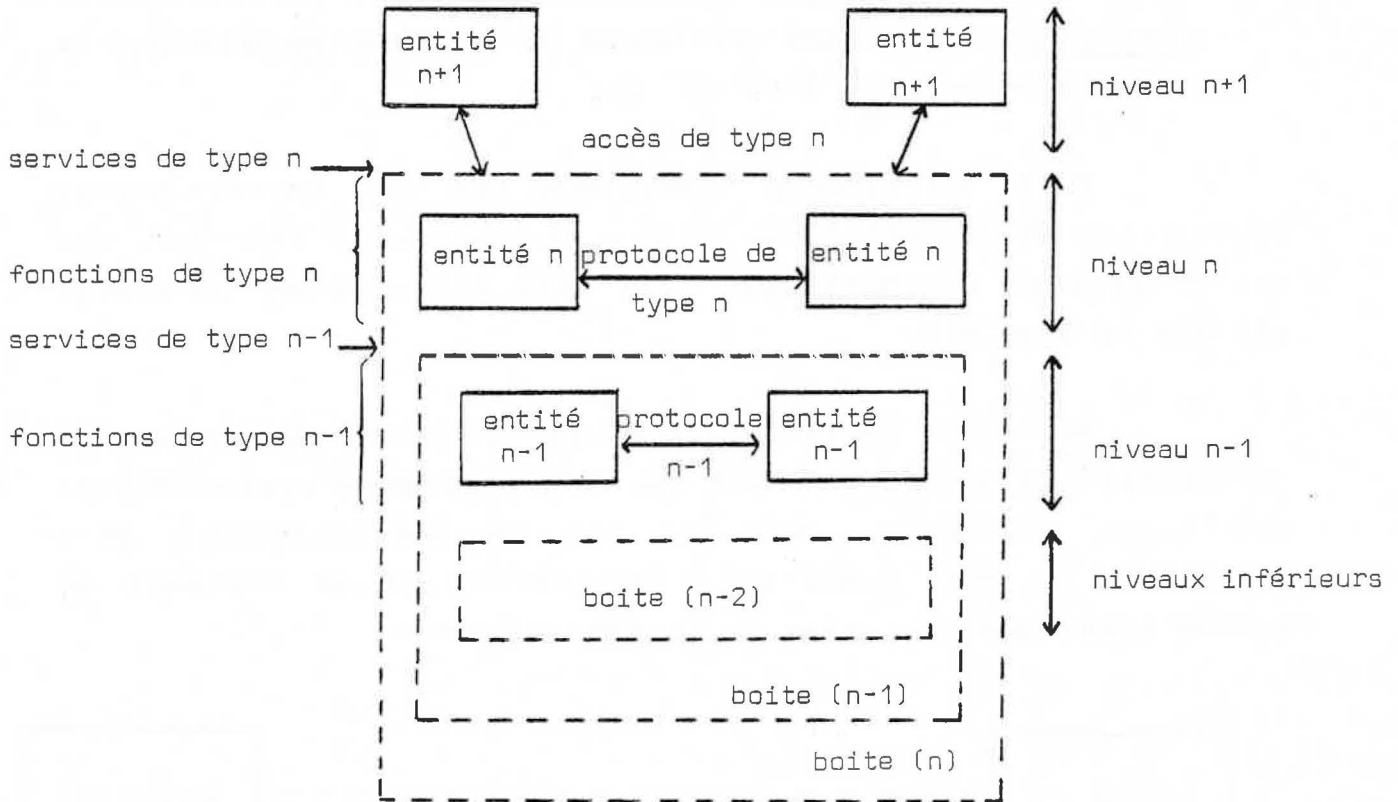


Figure 22: Assemblage hiérarchisé

On peut en donner la définition formelle suivante:

- * le niveau n de la structure peut utiliser les services de type n-1 offerts par le niveau n-1 à travers un accès de type n-1.
- * la structure des niveaux inférieurs n'est pas connue par le niveau n qui considère seulement les services rendus par la boîte de type n-1.
- * le niveau n est composé d'entités de type n qui coopèrent selon un protocole de type n.

- * les entités de type n réalisent des fonctions de type n en utilisant les services de type n-1 pour offrir des services de type n au niveau n+1.
- * les spécifications de l'architecture d'un niveau font référence à l'ensemble de services offerts par les niveaux inférieurs, par l'intermédiaire des fonctions d'accès. Cet ensemble de fonctions d'accès qui définit un interface peut être vu comme la manière de décrire la structure logique d'un réseau.

Cette définition n'implique pas une correspondance bi-univoque entre entités de type n et entités de type n-1. Une entité de type n-1 peut très bien être utilisée par plusieurs entités de type n.

Le modèle de la figure 22 suffit pour représenter des configurations simples telles que deux systèmes connectés par une ligne dédiée. Dans le cas de configurations plus complexes, on peut combiner à cet assemblage un chainage de boîtes ainsi que le montre la figure suivante:

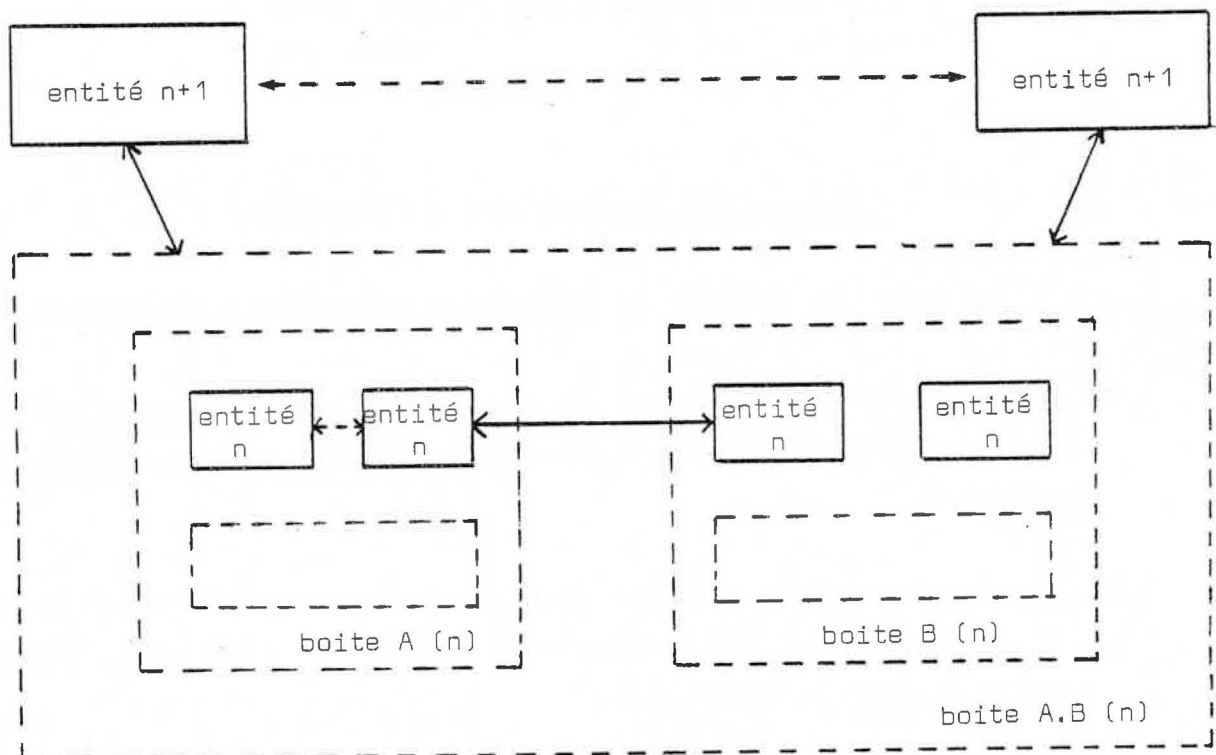


Figure 23

Le fondement [PARNAS: GEN-14] visant à structurer un système en niveaux répond à un désir de partitionner les fonctions qui doivent assurer:

- * la séquentialité dans le temps de niveau en niveau.
- * l'indépendance des activités des différents niveaux: un changement à l'intérieur d'un niveau ne doit affecter que ce niveau.
- * la transparence de l'implémentation de chaque niveau pour ses utilisateurs.
- * le partage de services communs par différents utilisateurs.

1.2 Système intégré

Il existe différentes approches possibles en vue de la réalisation d'un système réparti. Ces approches prennent en compte les trois aspects:

- langage de commande
- système réseau
- systèmes existants

En fonction de chacune de ces approches, il est possible de caractériser le système ainsi obtenu [NEWMAN: RES-14].

a. Système centralisé

On ignore les systèmes existants et on en crée un de toutes pièces, qui répond aux spécifications. Cette approche nécessite un gros investissement de programmation mais présente un avantage: le système est bien spécialisé; le langage de commande est compris par toutes les entités du système sans qu'il y ait besoin de traduction; il en découle donc un gain en rapidité d'exécution. Toutefois, la panne du système central est fatale.

b. Système standardisé

On conserve les différents systèmes existants; on adopte l'un d'entre eux et on impose aux autres systèmes de s'y conformer. L'intérêt de cette solution est de ne pas entraîner l'arrêt de fonctionnement de tout le système en cas de panne de l'un des hôtes. Par contre, elle peut entraver des développements ultérieurs.

c. Système intégré

Chaque hôte conserve le système existant; on doit donc réaliser un interface qui réponde aux spécifications réseau définies. Cet interface assure deux fonctions:

- . en sortie: traduire les requêtes dans le langage commun à tous les hôtes.
- . en entrée: traduire dans le langage local les requêtes exprimées dans le langage commun.

L'interface de chaque système peut être réalisée sur l'hôte considéré ou bien tous les interfaces peuvent être déportés sur un hôte qui effectue les traductions vues précédemment.

Les avantages d'un système intégré sont divers:

- il permet de continuer à utiliser les données déjà existantes et, en plus de les partager sur le réseau.
 - il permet une approche "évolutive" c'est à dire par étapes successives, mieux acceptée par les utilisateurs.
- En cas de développement du système, on ne modifie pas le logiciel des différents hôtes, excepté celui qui est chargé de réaliser les interfaces.

2. RAPPEL DU RÔLE DE LA FONCTION FRONTALE

Nous allons procéder à un bref rappel des divers éléments fonctionnels mis en évidence dans les deux précédents chapitres.

- * service de transport mettant en oeuvre des voies logiques de communication.
- * gestionnaire de terminal virtuel.
- * gestionnaire du protocole CCR avec un rôle particulier pour le correspondant qui assure l'aiguillage entre entités.
- * interprète du LCE
- * enveloppe réseau qui réalise l'adaptation des interfaces "application - système d'exploitation local - système CCR" et dans laquelle on inclut l'interprétation du LCI.
- * l'application proprement dite

La mise en place d'une architecture logique va consister à distinguer des niveaux de hiérarchie parmi ces éléments fonctionnels et ensuite, à préciser les interactions entre ces niveaux et également les relations entre éléments de même niveau.

Nous considérons trois niveaux qui sont apparus naturellement au cours de la définition des fonctions.

Le premier correspond à la mise en oeuvre du service de transport; nous n'entrerons pas dans le détail de sa structure mais nous insisterons sur son aspect externe, c'est à dire sur l'aspect outil qu'il offre.

Le second niveau est constitué par la mise en place d'une structure de dialogue utilisant le service de transport, ce qui correspond à la gestion du protocole CCR. Nous attribuerons à ce niveau la fonction de dynamique des échanges.

Enfin, le niveau supérieur réalise le dialogue proprement dit, c'est à dire un dialogue de bout en bout à l'aide des différents langages.

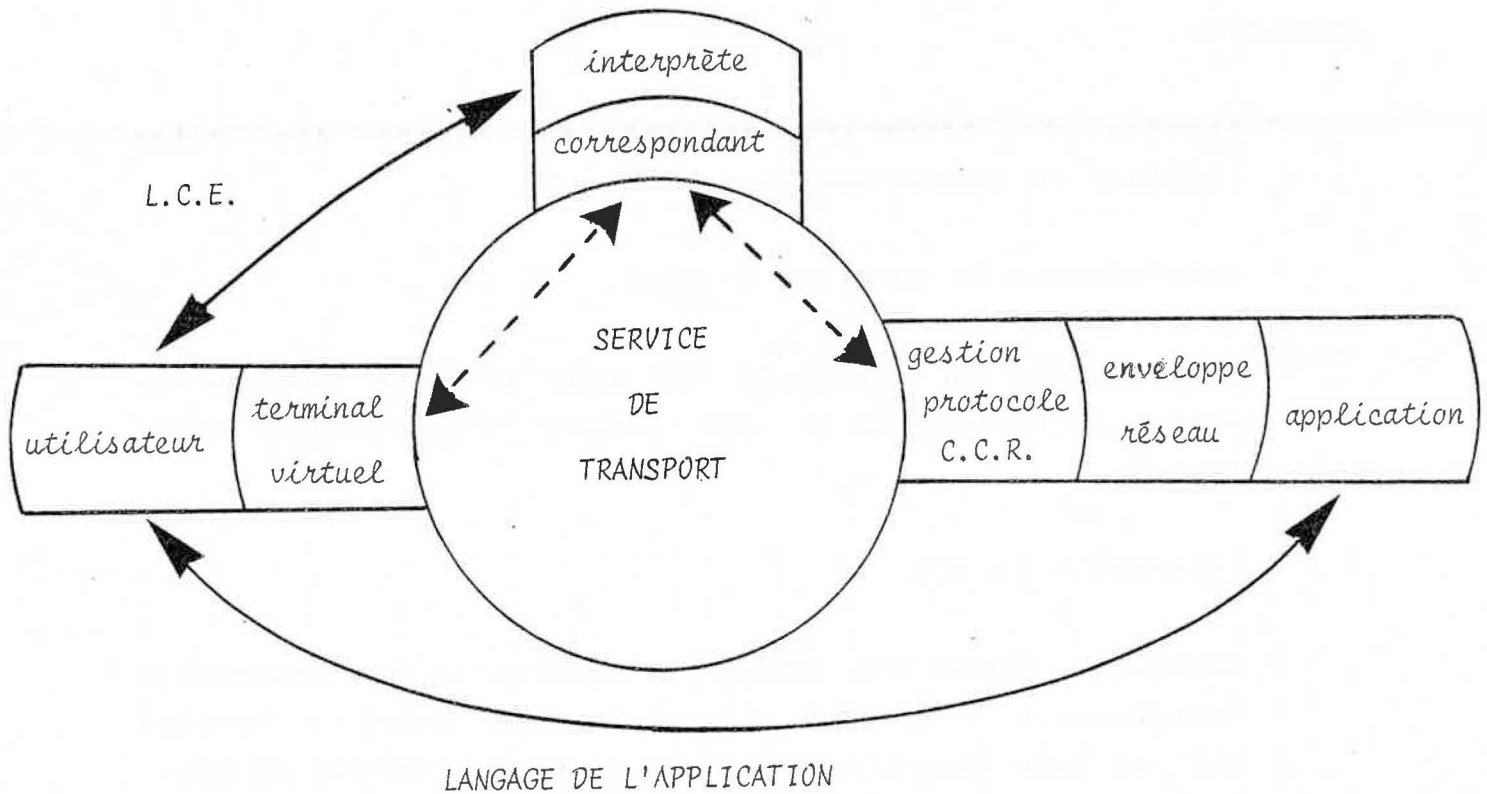


Figure 24

3. LES VOIES LOGIQUES DE COMMUNICATION

Ce sont des chemins logiques reliant, soit des appareils virtuels au correspondant, soit des applications (au sens application réseau) au correspondant. En fait, la nature de l'entité (appareil ou application) qui se trouve à l'extrémité d'une voie logique de communication (VLC) importe peu. Par le biais de ces VLC, le correspondant a une vision globale des accès utilisateurs et des applications.

Ces liaisons sont mises en oeuvre par le service de transport en tenant compte de la localisation des deux entités à relier.

Une voie logique de communication se décompose en un ou plusieurs ensembles, chacun comprenant TOSAN: CCR-2 :

1. le support de transmission: ligne, bus, ensemble modem ligne - modem.
2. la gestion de transmission c'est à dire gestion physique du support et gestion de la procédure de transmission.
3. la gestion des protocoles de communication.

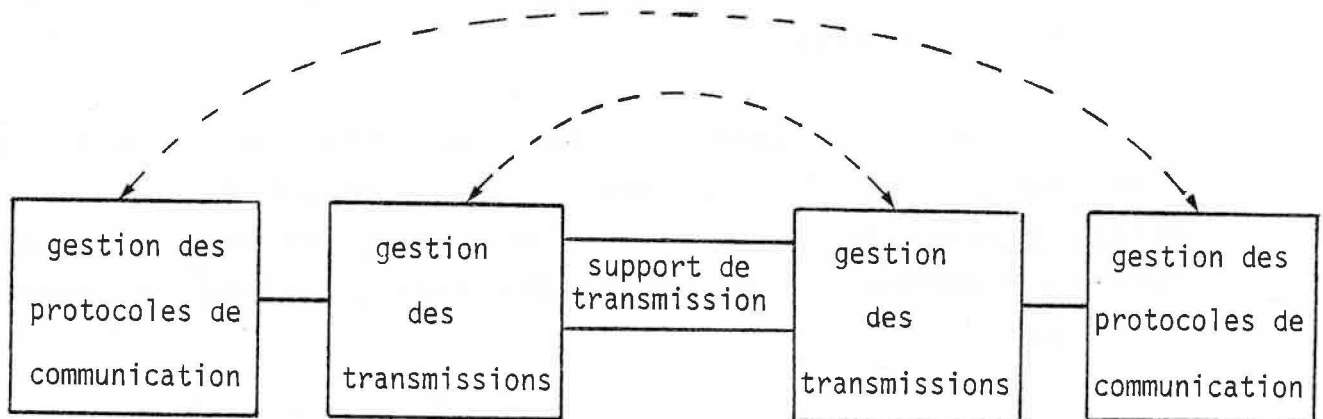


Figure 25: Schéma général d'une VLC

C'est par l'intermédiaire des VLC que les communications entre entités du CCR peuvent s'effectuer. Les informations qu'elles permettent d'échanger leur sont transparentes. Leur rôle est de garantir que ces informations arrivent à destination avec un taux d'erreur minimal et d'assurer un contrôle de flux des données échangées.

Les VLC constituent des objets sur lesquels un certain nombre d'opérations sont possibles. Ces opérations peuvent être de deux types:

- utilisation
- maintenance

a. Utilisation

L'utilisation des voies logiques de communication s'effectue par l'intermédiaire d'une primitive d'échange (émission / réception) sur une VLC. Il existe également une primitive de demande d'état qui permet d'obtenir des informations concernant la voie (ouverte ou fermée: voir § suivant). Une VLC se comporte comme un subordonné vis à vis de ses utilisateurs, exécute leurs ordres et en contrepartie les informe d'un certain nombre d'événements:

- fin d'échange
- signal d'appel au correspondant (interaction à partir d'un terminal)

Le correspondant joue un rôle privilégié dans l'utilisation des VLC: il est la seule entité à en avoir une vision globale et il possède l'initiative de leur utilisation dans de nombreux cas (par exemple dans les échanges avec un terminal).

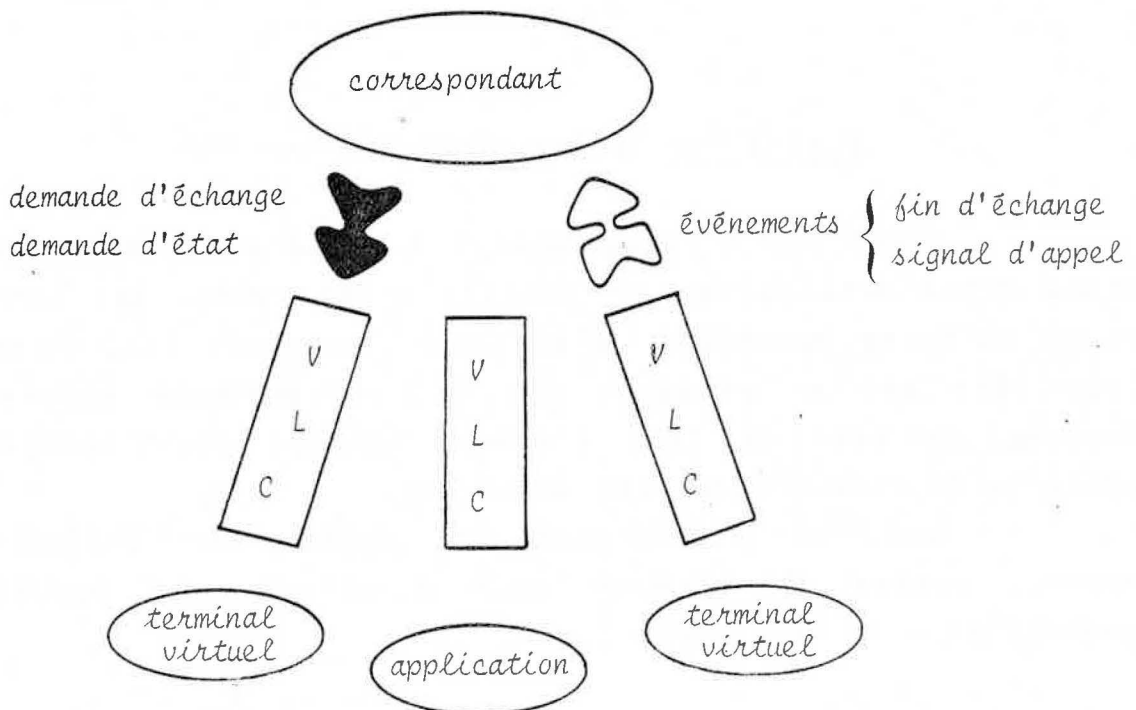


Figure 26

b. Maintenance

A partir de la configuration physique du système, c'est à dire des éléments matériels le constituant, on réalise une configuration logique, c'est à dire un schéma montrant clairement quels sont les éléments logiques qui sont disponibles. Ainsi, bien qu'un accès utilisateur soit effectivement connecté, on peut très bien en refuser l'utilisation, c'est à dire ne pas l'inclure dans la configuration logique. Il en est de même pour une application. C'est pourquoi il existe des commandes d'ouverture et de fermeture de voie logique de communication.

La commande d'ouverture introduit l'élément situé à l'extrémité de la voie dans la configuration logique du système et fait démarrer la gestion de cet accès. Ce peut être un accès utilisateur, un appareil ou une application. Inversement, la commande de fermeture l'en retire.

La maintenance inclut également la prise en compte de défaillances au niveau du fonctionnement.

4. DYNAMIQUE DES ECHANGES

Les supports des échanges, à savoir les voies logiques de communication, ayant été définis, il reste à préciser la dynamique de ces échanges. Celle-ci repose sur le protocole CCR qui sert en quelque sorte de véhicule pour l'information qui est échangée entre les entités qui dialoguent. Les éléments concernés par cet aspect sont ceux qui ont à supporter et à gérer le protocole. Il s'agit:

1. de l'émulateur du terminal virtuel qui se charge des échanges entre le terminal virtuel et le système,
2. du niveau supérieur de la méthode d'accès de chaque ordinateur connecté qui va gérer le protocole pour le compte de toutes les entités localisées sur cet hôte,
3. du correspondant qui joue le rôle d'intermédiaire entre ces entités.

Ces éléments vont assurer un contrôle de dialogue entre toutes les entités. Ce contrôle est supervisé par le correspondant qui, pour chaque utilisateur, prend en compte les deux types de dialogue:

- utilisateur - interprète,
- utilisateur - application.

Nous allons maintenant apporter quelques précisions sur chacun de ces éléments.

4.1. L'émulateur de terminal virtuel

C'est lui qui donne au système l'impression d'avoir en face de lui un terminal standard. Il est relié au correspondant par l'intermédiaire d'une voie logique de communication sur laquelle vont circuler les messages échangés entre l'utilisateur et l'interprète ou les applications. En plus de son rôle défini au chapitre III § 2.2, il prend en charge la gestion du protocole CCR. La seule entité qu'il connaisse (par son adresse réseau) est le correspondant logiquement situé à l'autre extrémité de la VLC.

4.2. Le niveau supérieur de la méthode d'accès des ordinateurs participants

Il constitue l'interface entre toutes les entités situées sur un ordinateur et le service de transport. Il gère le protocole CCR pour tous les messages émis (reçus) par les entités locales à destination (en provenance) d'entités distantes.

4.3. Le correspondant

Il gère les liaisons (création, destruction, utilisation) et sert de relais entre les voies logiques de communication. Il joue un rôle de carrefour central dans tous les échanges. Nous pouvons discerner dans son rôle deux aspects:

a. le correspondant utilisateur

Sa fonction est de créer, maintenir et détruire les liens logiques avec les utilisateurs. Il prend en charge les messages en provenance des utilisateurs et destinés à l'interprète du LCE ou aux applications. Il introduit la gestion du protocole CCR pour les messages émis par l'interprète vers les utilisateurs.

b. le correspondant application

Sa fonction est de créer, maintenir et détruire les liens logiques avec les applications. Il prend en charge l'aiguillage des messages en provenance des applications et les dirige:

- soit vers des utilisateurs (dialogue utilisateur - application),
- soit vers d'autres applications (cas de la coopération d'applications),
- soit vers des appareils (fichiers, imprimante),

Il introduit la gestion du protocole CCR pour les messages émis par l'interprète du LCE vers des enveloppes réseau.

5. NATURE DES DIALOGUES

Les dialogues dont nous parlons ici sont les dialogues de bout en bout c'est à dire entre entités terminales. A l'intérieur du CCR, on distingue trois types de dialogue:

- dialogue utilisateur - interprète,
- dialogue utilisateur - application,
- dialogue application - application.

Chacun de ces dialogues est réalisé dans un langage particulier, respectivement:

- LCE,
- langage de l'application; ce sera, par exemple, APL, le langage d'un éditeur de textes, etc,
- LCI.

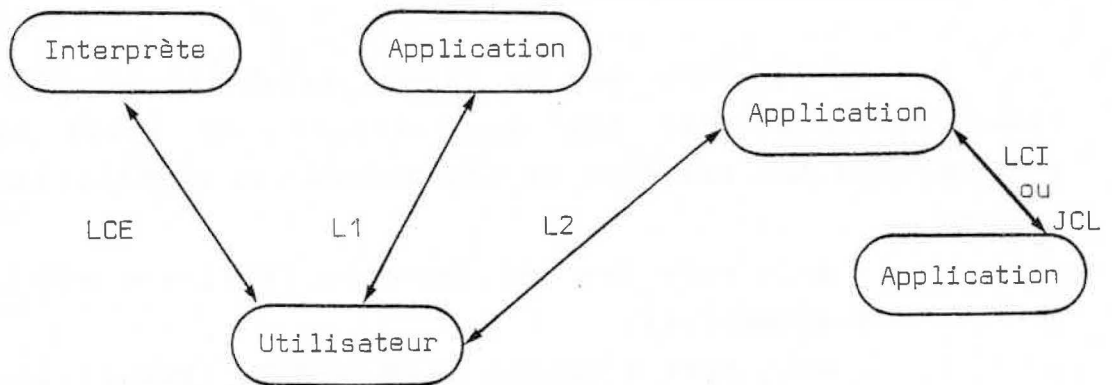


Figure 27: Les différents types de dialogue

Nous allons maintenant passer en revue les entités participant à ces dialogues.

5.1. L'utilisateur

C'est l'entité qui utilise les services du CCR. Un utilisateur du CCR est connu par:

- son nom d'utilisateur qui le caractérise en propre en dehors de toute relation physique avec le CCR;
- le point d'accès au CCR (terminal) qu'il utilise et qui lui permet d'inter-agir avec le système.

Toute utilisation du CCR est liée à la notion de session de travail. Une session commence nécessairement par une séquence dans laquelle l'utilisateur se fait connaître du

ystème par son identification (nom et mot de passe). Cette phase de "LOGIN" permet au système d'associer un accès utilisateur et l'utilisateur concerné. Cette association est rompue par une commande de "LOGOUT" qui met fin à la session en cours. A l'intérieur d'une session, les interactions possibles sont de deux types:

- celles qui sont exprimées en LCE; elles constituent le moyen par lequel les utilisateurs peuvent faire connaître leurs désirs et imposer leur volonté au système: ce sont des requêtes;

- celles qui sont adressées à une application du CCR et font partie du dialogue utilisateur - application. Elles ne peuvent intervenir qu'après que le système ait établi un lien à la suite d'une requête de connexion à une application, exprimée en LCE.

Dans la suite, nous réservons le terme commandes pour les interactions des utilisateurs avec l'interprète du LCE et le terme échanges pour les interactions des utilisateurs avec les applications. Les deux types d'interactions ne sont pas exclusifs l'un de l'autre; ils ont lieu en parallèle au gré de l'utilisateur.

5.2. L'interprète

Il s'agit de l'interlocuteur des utilisateurs dans le système CCR. Il symbolise le système CCR tout entier vis à vis de l'utilisateur. A travers le LCE, c'est à lui que l'utilisateur s'adresse au cours de sa session de travail. Sa fonction est d'interpréter le langage de commande externe et réaliser les actions qui en découlent. Lorsque ces commandes ne mettent pas en jeu d'applications, les interactions restent limitées au correspondant utilisateur et au terminal virtuel. Par contre, lorsqu'il s'agit de commandes impliquant une (ou des) application(s), le correspondant application intervient et se charge des interactions avec les enveloppes réseau.

c. L'application réseau

Une application réseau est constituée de trois niveaux:

- service local: c'est l'application offerte par le calculateur hôte (compilateur, éditeur de texte, etc...). C'est avec ce niveau que l'utilisateur a l'impression de dialoguer lorsqu'il est connecté à une application.
- environnement service local: ce sont les éléments du système d'exploitation du calculateur hôte nécessaires à l'exécution du service local. Ce niveau reçoit des requêtes dans le langage de commande local.
- enveloppe réseau: il s'agit du niveau qui effectue le complément d'interprétation du LCE et qui prend en charge les dialogues en LCI ou en JCL avec d'autres applications dans le cas de coopération d'applications.

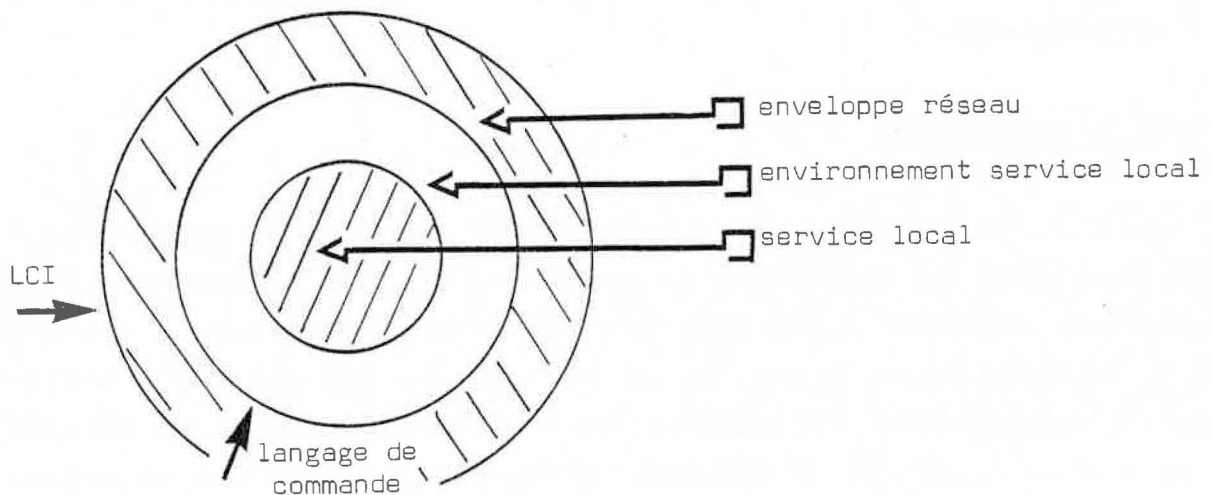


Figure 28

Les fonctions de l'enveloppe réseau dépendent du choix effectué quant au langage de dialogue utilisé entre des applications qui coopèrent:

1. chaque enveloppe réseau qui veut transmettre des données à une application doit se conformer à l'interface que celle-ci impose. Ceci implique que l'enveloppe connaît le JCL de toutes les applications avec lesquelles elle désire coopérer.
2. les applications réseau qui désirent coopérer se définissent un langage commun qu'elles utilisent lors de leurs interactions. L'enveloppe n'est chargée que de faire correspondre le JCL local et le langage commun défini.

CHAPITRE VI

REALISATION

1. HISTORIQUE DE L'ETUDE
2. LE FRONTAL
3. LOCALISATION DES FONCTIONS FRONTALES
 - 3.1. Service de transport
 - 3.2. Dynamique des échanges
 - 3.3. Interprétation
4. ARCHITECTURE DU C.C.R.
 - 4.1. Architecture du FRONTAL
 - 4.1. Architecture globale

L'architecture logique recouvre la définition des éléments fonctionnels et de leurs rapports; elle prévoit le contrôle des éléments physiques et de l'information qui va être véhiculée et traitée par eux. Quant à la réalisation, elle comprend la spécification, la localisation et la réalisation en elle-même des modules de logiciel qui répondent à la fonctionnalité désirée.

Dans ce chapitre, nous cherchons à faire correspondre au graphe logique défini dans les précédents chapitres un graphe physique, c'est à dire prenant en compte les connexions physiques.

Pour expliquer certains des choix effectués dans la réalisation, nous allons faire un bref historique de quelques aspects de la recherche dans le domaine de la télé-informatique à l'Ecole des Mines de Saint Etienne.

1. HISTORIQUE DE L'ETUDE

Le point de départ de cette recherche s'inscrit dans le cadre du développement du réseau CYCLADES. L'objectif initial était l'accès au réseau CYCLADES à partir des matériels du centre de calcul (terminaux, ordinateurs Philips P1175 et Télémécanique T1600). Puis, dans un contexte plus large il s'agissait de permettre:

1. aux terminaux locaux d'accéder aux services distants disponibles sur le réseau.
2. aux terminaux distants connectés au réseau d'accéder aux services locaux disponibles sur le P1175.
3. aux terminaux locaux d'accéder indifféremment aux divers services locaux (banalisation des terminaux).

Ces objectifs furent réalisés en utilisant l'un des ordinateurs (le T1600 en l'occurrence) comme FRONTAL, la gestion de tous les accès étant déportée sur ce frontal.

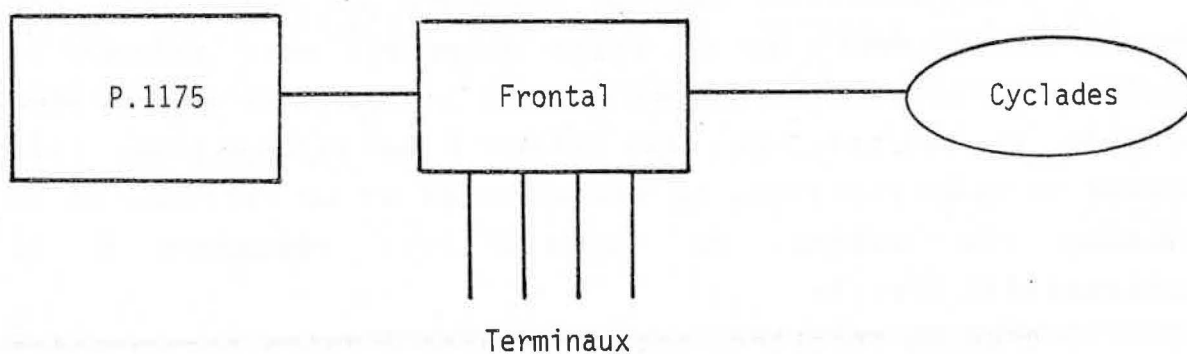


Figure 28

D'autres ordinateurs ayant été acquis par l'Ecole, on envisagea de généraliser le rôle du FRONTAL. Il concentrerait les logiciels de communication avec chaque ordinateur local et avec le réseau externe et serait utilisé:

- par chaque ordinateur local en frontal du réseau,
- par le réseau en frontal de chaque ordinateur local,
- par chaque ordinateur local en frontal des autres ordinateurs du site,
- par les terminaux pour accéder au réseau et aux ordinateurs locaux.

Ainsi se développa l'idée du Centre de Calcul Réparti bâti autour du concept de FRONTAL.

2. LE FRONTAL

Pour satisfaire les fonctions du frontal, un système spécialisé et adapté a entièrement été conçu et réalisé sur le T1600. Ce système "orienté entrées-sorties" réalise à partir du calculateur de base un "calculateur étendu" qui fournit un sur-ensemble des possibilités du T1600. Ce "calculateur étendu"

offre les fonctions du calculateur de base et, par logiciel, des fonctions plus évoluées:

- gestion des tâches et des interruptions
- gestion dynamique de la mémoire principale
- gestion d'une mémoire auxiliaire sur disque
- gestion de files d'attente
- gestion de réveils
- fonctions élémentaires diverses (décalage, positionnement de bits dans une file, etc ...).

Ces fonctions constituent des outils pour la programmation du frontal.

L'architecture logique du frontal repose sur le concept de machine [CHAMBON, LE BIHAN: GEN-8].

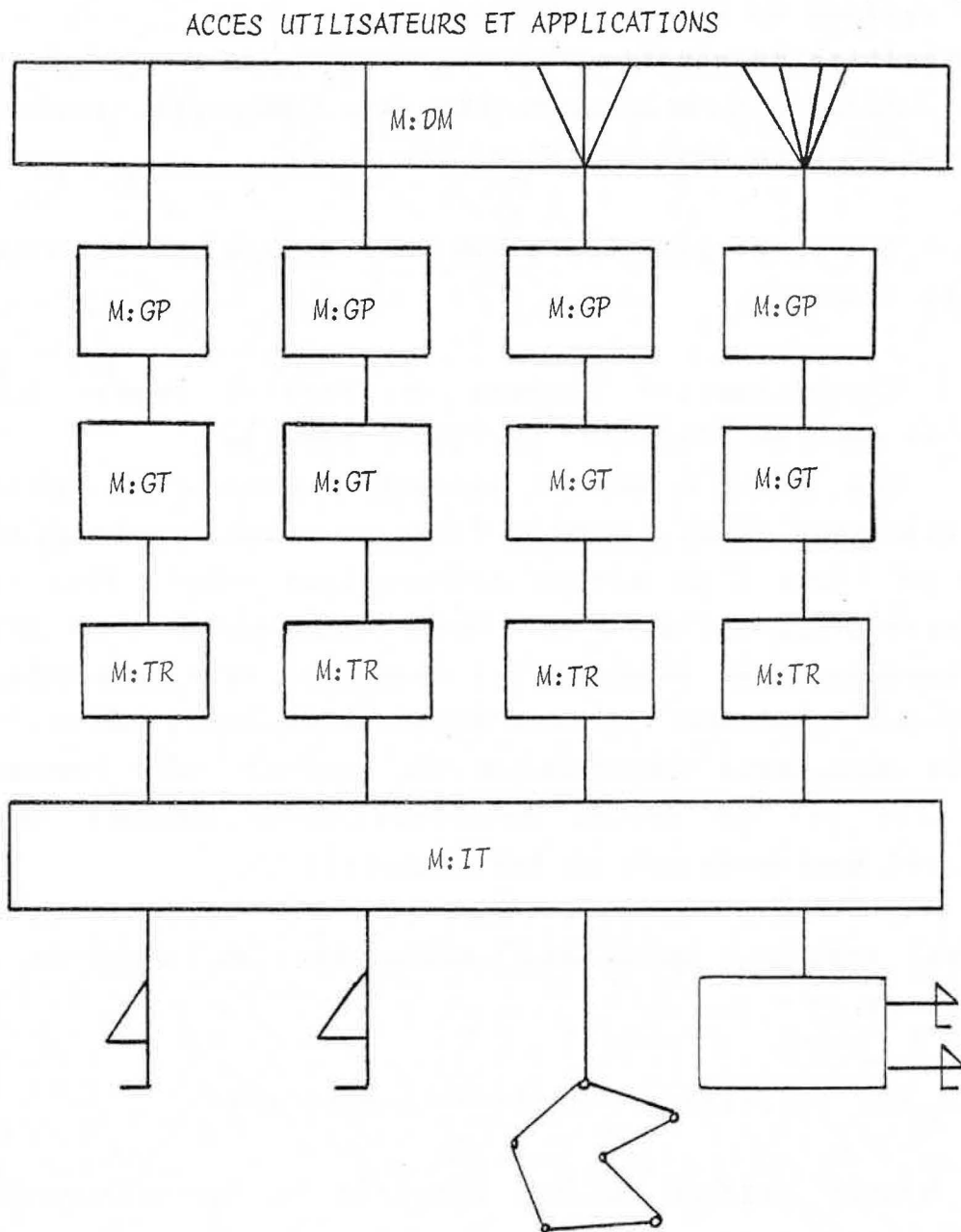
Une machine est un ensemble fonctionnel spécialisé dans l'exécution d'une fonction (plus ou moins complexe). Toute machine se situe à un niveau hiérarchique donné. Elle reçoit des commandes des machines de niveau supérieur et leur signale des évènements. Pour exécuter ces commandes elle peut elle-même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

L'élément essentiel du frontal est la machine d'entrées sorties, elle-même décomposée en machines plus élémentaires.

Les différents niveaux reconnus sont:

- niveau gestion du bus d'entrée et des interruptions (M:IT).
- niveau ligne: ensemble de machines (M:TR) qui gèrent chacune un type de coupleur d'entrée sortie.
- niveau procédure: ensemble de machines (M:GT) qui gèrent les procédures de transmission.

- niveau protocole: ensemble de machines (M:GP) gérant chacune un protocole de communication.
- niveau boîte aux lettres (M:DM) qui assure un multiplexage / démultiplexage.



Figure_29: La machine d'entrées-sorties

Le rôle du frontal est de gérer les accès aux différents éléments matériels (terminaux, autres ordinateurs, ...) d'assurer la banalisation de ces matériels et de réaliser l'aiguillage entre les accès.

3. LOCALISATION DES FONCTIONS FRONTALES

Après ces quelques préliminaires sur l'historique de l'étude, nous abordons la phase de réalisation et en premier lieu le problème de la localisation des fonctions frontales. La localisation pose le problème du choix dans la répartition des fonctions.

3.1. Service de transport

Historiquement, le frontal prenait en charge un certain nombre de fonctions:

- gestion de terminaux,
- gestion de procédures de transmission,
- gestion de protocoles: accès à d'autres ordinateurs ou même à des réseaux.

On est donc très tenté de faire du frontal le coeur du service de transport. Le réseau local a alors une configuration étoilée, dont le centre est constitué par le frontal. Les constituants du réseau sont connectés au frontal; chacun d'eux n'a donc qu'à gérer son accès au frontal.

Un tel choix comporte des inconvénients mais se justifie pour plusieurs raisons. Au titre des inconvénients, nous citerons le manque de fiabilité: le frontal est en effet un élément central, critique et vulnérable; une panne est donc fatale. Cependant, le système peut alors fonctionner en mode dégradé, chaque ordinateur reprenant son autonomie propre. On peut également objecter que le frontal crée un goulot d'étranglement dans un système par ailleurs riche en parallélisme. Quant aux justifications, elles sont également de deux ordres:

- extensibilité: l'adjonction d'un nouvel ordinateur dans le réseau limite les modifications de logiciel à cet ordinateur et au frontal.

- meilleure appréhension et maîtrise du système: ceci est un aspect important dans une première phase de réalisation et de mise au point.

Il s'agit de démontrer la faisabilité d'une étude; une fois que ce but est atteint, rien n'empêche de remettre en question le choix de la répartition des fonctions pour pallier les insuffisances (fiabilité, temps de réponse, etc, ...).

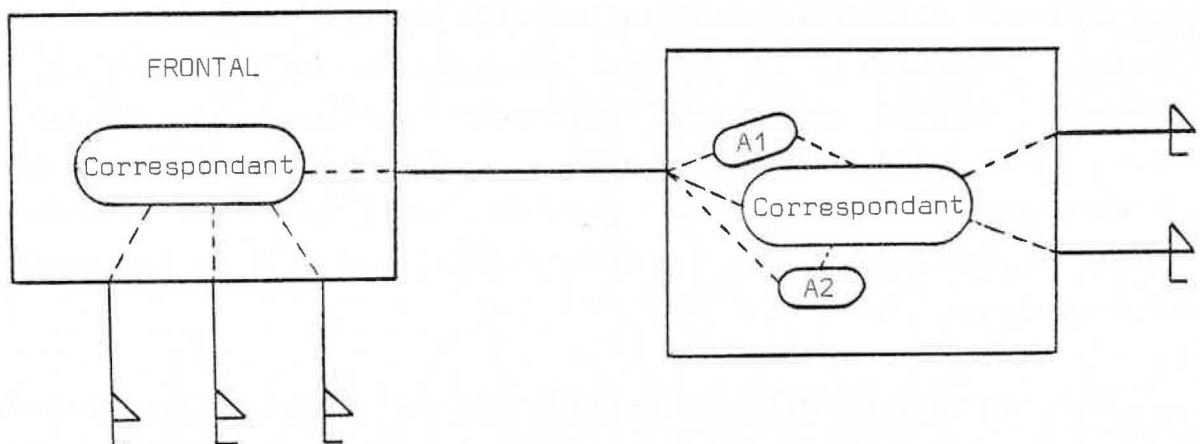
3.2. Dynamique des échanges

D'après la configuration du réseau, il est tout naturel que le correspondant soit localisé sur le frontal puisque c'est la plaque tournante de l'ensemble.

La gestion du protocole CCR doit être assurée pour tout élément (terminal ou ordinateur) désirant s'intégrer au CCR.

Un terminal peut être connecté soit directement au frontal (on dira que c'est un terminal local), soit à l'un des autres ordinateurs du réseau (terminal distant).

Pour les terminaux distants (c'est à dire non connectés directement au frontal), il semble superflu de devoir faire transiter les échanges qui les concernent par le frontal dans le cas où ceux-ci s'effectuent avec une application située sur l'ordinateur auquel les terminaux sont connectés; d'où l'idée de déporter une partie du correspondant sur les ordinateurs gérant des terminaux distants (par rapport au frontal).



A1 et A2 sont des applications

3.3. Interprétation

L'interprétation du LCE doit être réalisée le plus près possible des accès utilisateurs pour éviter de multiples transferts. En effet, on a tout intérêt à effectuer le plus possible le traitement des données à l'endroit où elles sont émises. Cependant, il est préférable de n'avoir qu'une version de l'interprète LCE, pour des raisons d'encombrement et de maintenance; d'où le choix de localiser l'interprète du LCE sur le frontal.

Quant à l'interprétation du LCI, elle est locale aux applications qui ont besoin de cet environnement pour coopérer avec d'autres applications.

4. ARCHITECTURE DU CCR

La description de l'architecture du CCR va s'effectuer en deux temps; tout d'abord nous présentons l'architecture de frontal et ensuite l'architecture de l'ensemble.

4.1. Architecture du frontal

Elle a été partiellement décrite dans le paragraphe 2. avec la présentation de la machine d'entrées sorties Celle-ci constitue une partie essentielle du service de transport; c'est l'outil utilisé par le correspondant pour joindre les autres entités du système. La vision que le correspondant a de ces entités (terminaux ou applications) est matérialisée par des unités symboliques (SU). Il existe une SU pour chaque accès utilisateur et pour chaque application; ces SU représentent les entités sur lesquelles le correspondant peut interagir. Ces SU sont toutes banalisées: qu'il s'agisse de SU terminal ou de SU application, il existe un moyen standard d'effectuer des demandes d'échange.

Un réseau externe (Cyclades, Transpac) est vu comme un ensemble de services (ensemble de SU pour le correspondant)

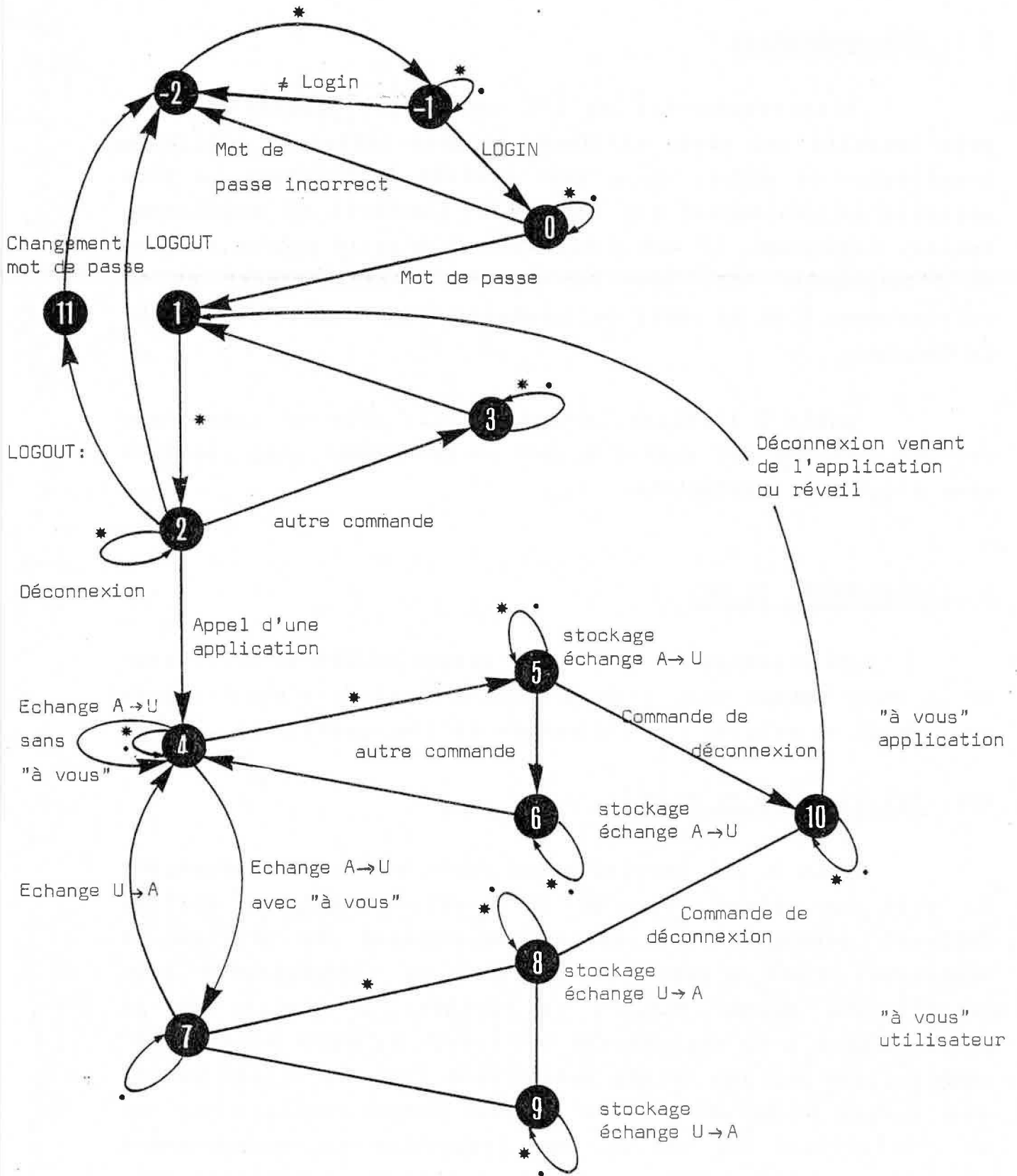


Figure 32: Automate associé à chaque utilisateur

exactement comme un ordinateur du site pour lequel le correspondant ne connaît que les services disponibles. A chacune de ces SU est associée une voie logique de communication qui relie le correspondant soit à un terminal, soit à une application. C'est la voie logique qui gère les protocoles d'accès en vigueur entre le frontal et l'hôte supportant l'entité située à l'autre extrémité de la voie.

Le correspondant est lui-même divisé en deux parties; l'une interagit avec les SU accès utilisateur, l'autre avec les SU application.

Un contrôle de dialogue entre les différents interlocuteurs (utilisateur, interprète, application) est assuré par le correspondant. Les différents types de dialogue sont modelés par un automate (figure 32) associé à chaque utilisateur; cet automate évolue sous le contrôle du correspondant et tient compte des modes de dialogue en vigueur entre les interlocuteurs. On distingue deux modes de dialogue:

- a. à l'alternat: le droit d'envoyer un message est initialement donné à l'un des interlocuteurs (dans les dialogues utilisateur - application c'est généralement l'application). Lorsque l'un d'eux envoie un message, le droit ou non d'envoyer un message est transmis à l'autre interlocuteur par l'intermédiaire d'une indication de A VOUS. Ce mode permet aux deux interlocuteurs d'avoir une vision cohérente des données échangées puisque le séquençement des messages est respecté.
- b. indéfini: le droit d'envoyer un message n'est pas contrôlé par le protocole CCR. Ce mode peut être utilisé par des applications pour lesquelles les messages émis ou reçus ne sont pas directement reliés par leur signification; c'est le cas, par exemple, d'un service de traitement par lots: à tout moment l'opérateur ainsi que le système peuvent envoyer un message. Il est donc impossible d'identifier l'interlocuteur qui enverra le prochain message sous peine de provoquer un blocage. Ce mode est

également utilisé pour permettre une anticipation dans les échanges (également dans le cas des "Gateway").

L'architecture du FRONTAL est schématisée par la figure suivante:

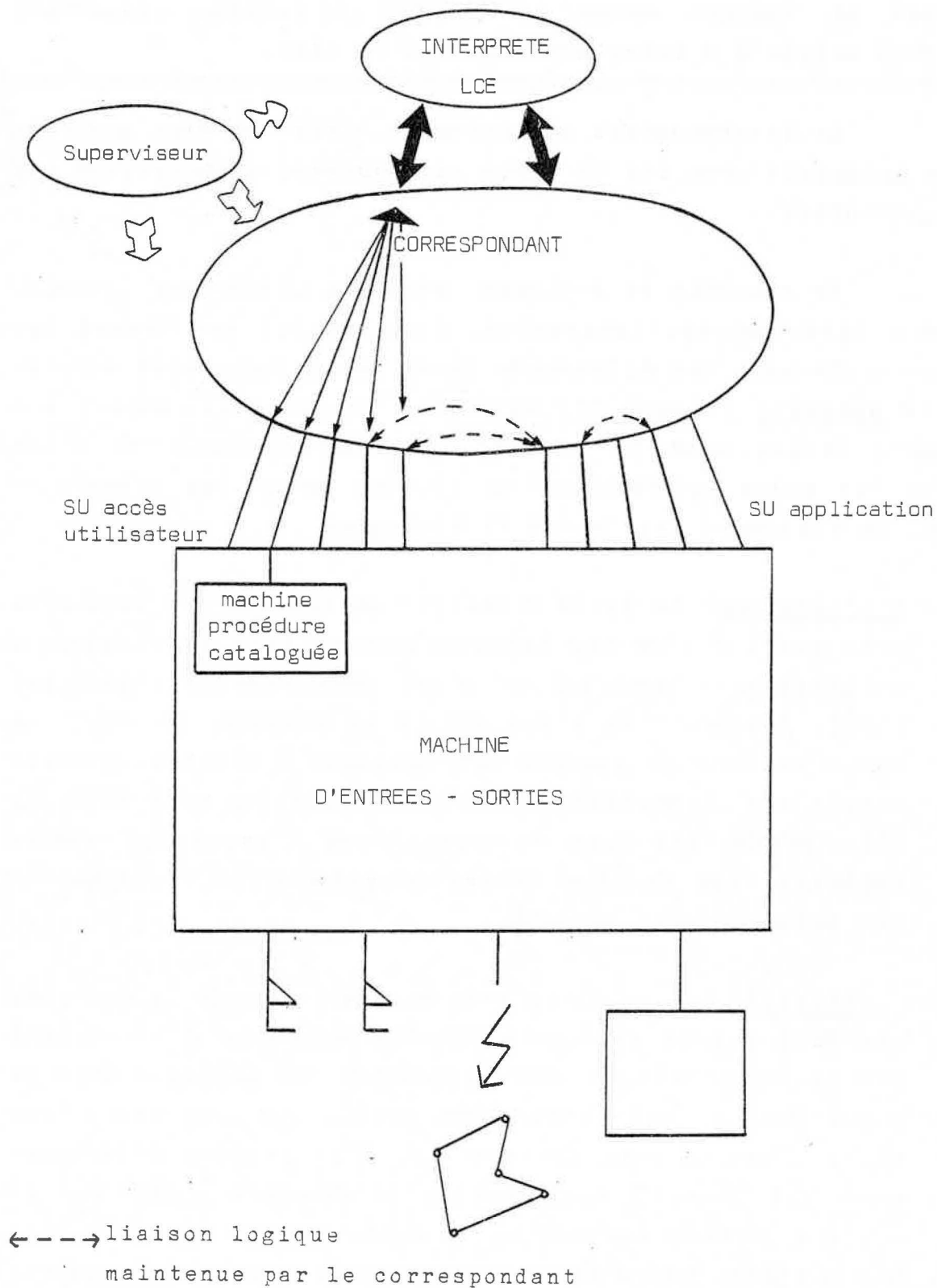


Figure 33: Architecture du FRONTAL

Le correspondant a une vision logique des différentes entités composant le CCR.

Quant au superviseur, il contrôle l'aspect matériel du frontal: contrôle de la configuration, ouverture et fermeture de SU, maintenance, détection de défaillances matérielles, etc... .

Les liaisons accès utilisateur - application sont créées à la demande de l'interprète par le correspondant application au moment de l'interprétation d'une commande de connexion. Ensuite elles sont utilisées à l'insu de l'interprète au gré des échanges effectués entre l'utilisateur et l'application.

Les langages circulant sur les VLC ont déjà été présentés et nous ne reviendrons pas sur cet aspect. Nous développerons plutôt quelques mécanismes mis en place pour réaliser les échanges.

Mécanisme de dialogue

Il est schématisé par la figure 34.

A tout moment, l'utilisateur peut adresser des requêtes au système. Il lui suffit, pour cela, d'enfoncer la touche de son terminal représentant la fonction APPEL. Ce signal est interprété comme le désir de la part de l'utilisateur d'envoyer une commande; le correspondant utilisateur effectuera donc une demande de réception sur la voie logique appelante et récupèrera la commande de l'utilisateur qui sera transmise à l'interprète.

Quant aux échanges avec l'application connectée, ils s'effectuent au rythme de l'utilisateur et de l'application.

Au niveau de la gestion du terminal, des mécanismes permettent d'éviter que des blocages ou des interférences entre les deux types de dialogue ne se produisent. Une priorité est donnée aux interactions entre l'utilisateur et l'interprète de sorte que, à tout instant, l'utilisateur puisse envoyer une commande même si son terminal est déjà en réception d'un message pour une application.

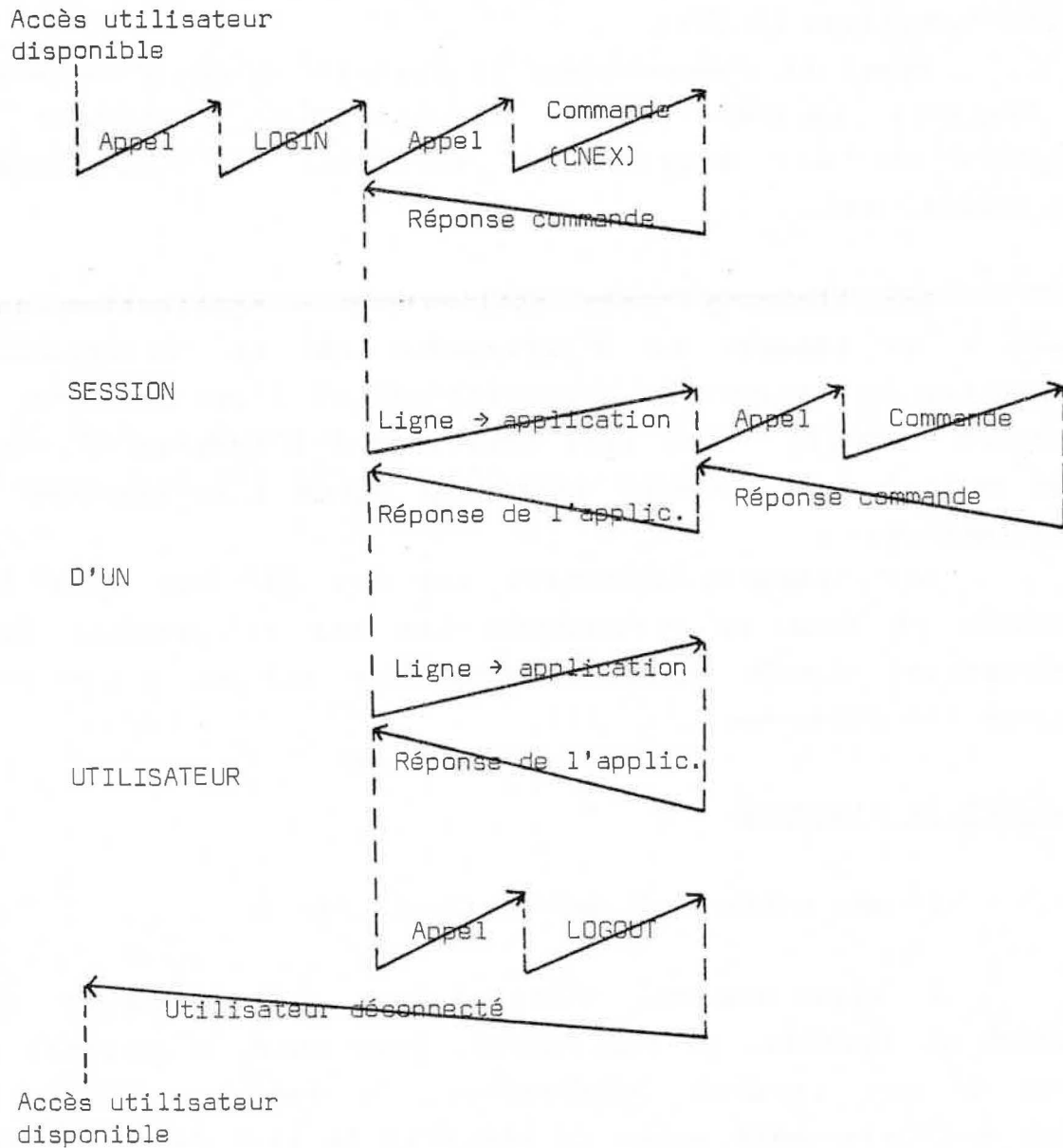


Figure 34

Procédures cataloguées

Le mécanisme de procédure cataloguée permet l'enchaînement automatique des commandes et des échanges. Il est réalisé de la façon suivante. Il existe une machine appelée "machine procédure cataloguée"; elle est située à l'intérieur de la machine d'entrées sorties et est accessible par une SU identique aux autres SU. Elle simule en entrée (monde extérieur vers le système) l'accès de l'utilisateur pour lequel une procédure cataloguée a été déclenchée; de cette façon, aucune distinction sur la provenance d'un message ne peut être

effectuée au niveau du correspondant et de l'interprète. Elle distille des messages contenant soit des commandes soit des échanges au correspondant; le premier caractère de chaque message indique son type: commande ou échange.

Il existe deux types de procédures cataloguées:

- a. les procédures utilisateurs créées par des utilisateurs qui ne contiennent donc que des commandes en LCE et des échanges avec des applications dans le langage de ces applications.
- b. les procédures système qui peuvent éventuellement contenir des commandes exprimées dans le JCL des systèmes hôtes. Ces commandes entrent dans le cadre des dialogues avec des applications; elles présentent un intérêt lors de l'interprétation d'une commande de connexion à une application. En effet, si l'interprétation de cette commande déclenche une procédure cataloguée contenant les différentes commandes nécessaires au système supportant l'application et exprimées dans son langage pour réaliser la connexion à l'application sous le système hôte, on a résolu par un moyen détourné le problème de la transparence des JCL pour les utilisateurs sans pour autant reporter sur l'interprète la connaissance de ces JCL. Ceux-ci apparaissent seulement dans des fichiers de commande.
C'est en quelque sorte le déport d'une partie de l'enveloppe réseau de l'application.

Structure de l'interprète

Elle est décrite avec précision dans un autre document [TOSAN: CCR-2]; c'est pourquoi nous n'en donnons ici qu'un bref aperçu.

L'interprète est composé de modules indépendants:

- module d'analyse syntaxique et de reconnaissance des commandes,

- module de contrôle des prérogatives de l'utilisateur,
- modules d'exécution: il en existe un par commande.

Cette structure permet l'addition aisée d'une nouvelle commande qui se traduit alors par l'introduction d'un mnémonique dans la liste des commandes et d'un module d'exécution. Un mécanisme de recouvrement permet une utilisation moindre de mémoire vive.

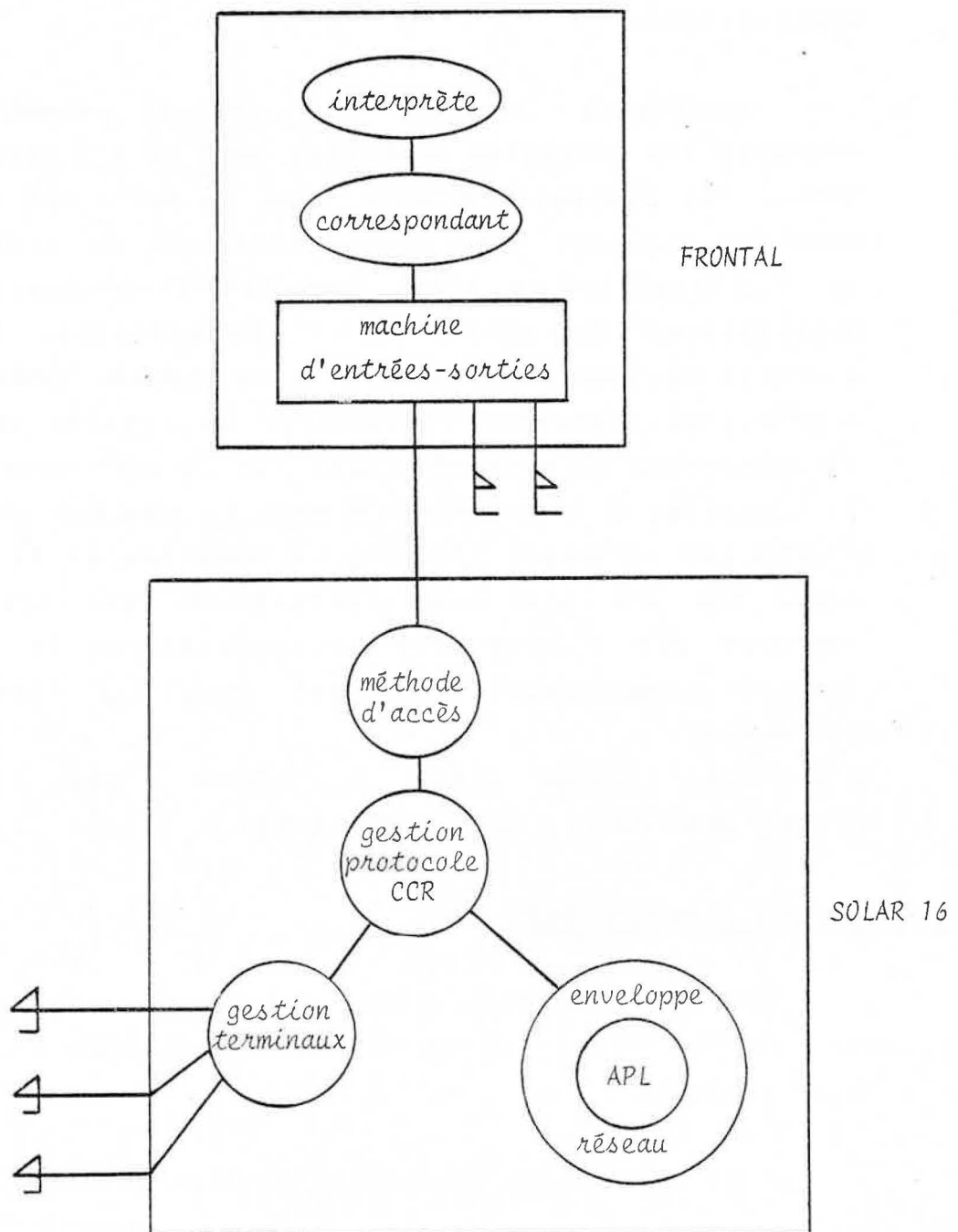


Figure 35: Architecture globale

4.2. Architecture globale

L'architecture du frontal ayant été présentée, il reste à préciser celle des calculateurs qui lui sont connectés. Nous nous limitons à l'architecture concernant les fonctions frontales réalisées sur ces calculateurs, chaque système d'exploitation étant considéré comme un tout dont seules les méthodes d'accès peuvent éventuellement être extraites.

La figure 35 montre l'exemple de la connexion au FRONTAL du SOLAR 16 qui offre le service APL et possède des terminaux.

CONCLUSION

Nous nous étions proposés dans ce travail de traiter de quelques aspects de l'informatique répartie vus sous un angle fonctionnel. Il nous reste à examiner les prolongements possibles de notre travail. Nous allons procéder à la démarche inverse de celle utilisée dans cet ouvrage. Nous donnerons l'état et les suites possibles de nos travaux, puis nous élargirons notre vision pour aborder des aspects plus généraux concernant quelques idées développées dans cette thèse.

L'architecture du frontal, telle qu'elle a été présentée avec ses spécifications au chapitre VI est actuellement mise en place. Un certain nombre de commandes LCE ont été implémentées, d'autres pourront l'être au fur et à mesure des exigences: la structure de l'interprète LCE [TOSAN: CCR-2] est en effet adaptée à une extension modulaire des commandes du langage.

Le service de transport assure la liaison entre plusieurs ordinateurs (T.1600, P.1175, SOLAR 16/65, PDP 11/40) et un réseau externe (CYCLADES).

Quant à la mise à disposition des services existant sur ces différents ordinateurs, elle n'est que partiellement effectuée: actuellement, sont accessibles les services offerts sur le réseau Cyclades, le service TSM (système d'exploitation du SOLAR), et divers services ECHO situés sur les différents ordinateurs hôtes qui permettent une validation progressive des mécanismes. A court terme, le service APL sur SOLAR doit être mis à la disposition des utilisateurs via le CCR.

Un aspect important des développements ultérieurs réside dans la validation des idées émises concernant la coopération d'applications. Ceci fait l'objet de travaux actuels dont l'illustration consiste à mettre en place un système de coopération de systèmes de gestion de fichiers [JUREMA, NIANG: CCR-5]. Cette étude se situe dans l'un des prolongements les plus importants de l'informatique répartie, à savoir la gestion de bases de données réparties.

Il serait également intéressant de procéder à des mesures d'évaluation notamment en ce qui concerne la charge du frontal, et selon les résultats obtenus, remettre en cause le schéma fonctionnel de répartition.

Trois aspects nous paraissent essentiels en ce qui concerne l'informatique répartie et ses développements.

L'architecture

Il ne fait aucun doute que la conception des futurs systèmes d'exploitation sera influencée par les exigences en matière de réseau

Dans cet ouvrage, nous avons examiné quelques unes de ces exigences pour interfacer les systèmes d'exploitation existants avec un réseau en prenant en compte d'autres aspects que l'aspect transmission.

On constate que, de plus en plus, aux architectures physiques des systèmes se substituent des structures fonctionnelles sur des équipements informatiques hétérogènes. Cependant, une uniformisation de concepts doit s'opérer pour que ces structures deviennent transparentes aux utilisateurs.

Jusqu'à maintenant, les solutions aux problèmes posés par l'informatique répartie semblent quelque peu disparates. Néanmoins, les similitudes dans les différentes implémentations (notion de hiérarchies de niveaux, concept de communication par l'intermédiaire de messages, ...) sont importantes dans la mesure où elles suggèrent les points essentiels de l'impact de la notion de réseau et permettent de dégager des stratégies pour la réalisation de systèmes orientés réseau ou plus généralement de systèmes répartis. Un modèle d'architecture de réseau informatique au sein de laquelle puissent prendre place les protocoles normalisés permettant la constitution de réseaux hétérogènes a été défini par l'ISO (Organisation Internationale de Normalisation) [ISO: GEN-21]. De plus, les techniques générales développées pour les réseaux d'ordinateurs peuvent être des solutions efficaces aux problèmes de la communication dans un système d'exploitation.

Les langages

Parallèlement aux développements concernant l'architecture des systèmes répartis, une évolution doit également s'effectuer dans le domaine des langages. Ceci concerne aussi bien les langages de commande que les langages de programmation qui sont les moyens par lesquels les utilisateurs perçoivent ces systèmes.

Un des objectifs poursuivis tout au long de nos travaux a été de simplifier l'accès aux systèmes informatiques répartis en les présentant aux utilisateurs comme un ensemble de services logiques et non plus comme un assemblage d'équipements physiques.

Quant aux langages de programmation, ils doivent être adaptés à la structure distribuée mise en place et tenir compte des possibilités de parallélisme et de transférabilité.

Les applications et les bases de données réparties

Pour profiter au maximum des avantages potentiels des systèmes répartis, il est nécessaire de développer des outils adaptés, d'une part à la coopération d'applications et, d'autre part à la gestion de données. Ce dernier point comprend la possibilité de définir et de manipuler des données réparties sur différents équipements comme un seul ensemble logique, cet ensemble étant alors considéré comme une base de données (répartie) [LE BIHAN: GEN-15].

Ce dernier aspect constitue maintenant le prolongement de notre étude.

Nous venons d'évoquer quelques uns des thèmes liés à l'avenir de l'informatique répartie. De gros efforts restent encore à accomplir pour que la structure d'un système à architecture répartie devienne transparente aux utilisateurs sans pour autant nécessiter d'importants investissements. Notre travail constitue un pas dans cette direction.

BIBLIOGRAPHIE

SYSTEMES ET RESEAUX: GENERALITES

- GEN-1 FARBER D.J.
"An overview of distributed processing aims"
Proc. 8th IEEE Computer Society International Conference
(Fev. 74)
- GEN-2 ZIMMERMANN H.
"Open systems architecture"
Eurocomp 78
- GEN-3 LE LANN G., NEGARET R.
"Une approche unifiée des systèmes informatiques répartis à
travers le concept de communication"
Congrès AFCET, Gif-sur-Yvette (Nov. 76)
- GEN-4 METCALF R.M.
"Strategies for operating systems in computer networks"
Proc. ACM Nat. Conf. 1972 pp 278-281 (Août 72)
- GEN-5 METCALF R.M.
"Strategies for interprocess communication in a distributed
computing system"
Proc. of the Conference on Computer Communications and
Teletraffic (Avril 72)
- GEN-6 POUZIN L.
"An integrated approach to network protocols"
National Computer Conference pp 701-707 (Mai 75)
- GEN-7 Digital Equipment Corporation
"DECNET: Technical Summary"
Technical publication
- GEN-8 CHAMBON J.F., LE BIHAN B.
"Architecture d'un frontal en environnement télé-
informatique"
Thèse de Docteur Ingénieur, EMSE (Oct. 76)

- GEN-9 KIMBLETON S., MANDELL R.
"A perspective on network operating systems"
AFIPS national computer conference proceedings, vol. 45
pp 551-560 (Juin 76)
- GEN-10 LUCA R., GAYRAL F.
"Etude du calculateur de support de terminaux"
Réseau Cyclades, GAL 008 (Mai 72)
- GEN-11 MANNING E.G.
"Distributed data processing"
La conception des systèmes répartis édité par l'IRIA
pp 175-184 (Juin 78)
- GEN-12 PEEBLES R.
"Distributed data management: an introduction"
La conception des systèmes répartis édité par l'IRIA
pp 197-218 (Juin 78)
- GEN-13 GUIBOUD-RIBAUD S.
"Mécanismes d'adressage et de protection dans les systèmes
informatiques- Application au noyau GEMAU"
Thèse d'Etat, Grenoble (Juin 75)
- GEN-14 PARNAS D.L., SIEWIOREK D.P.
"Use of the concept of transparency in the design of
hierarchically structured systems"
Communications of the ACM 18,7 pp 401-408 (Juil. 75)
- GEN-15 LE BIHAN J.
"La répartition des données dans les réseaux informatiques"
Congrès AFCET, Gif-sur-Yvette (Nov. 76)
- GEN-16 BARBER D.L.A.
"The role and nature of a virtual terminal"
Computer communication review, Vol. 7, No 3 (Juil. 77)
- GEN-17 BOCHMANN G.V., PROBST W.G.
"Operating system design with computer network communication
protocols"
Fifth Data Communication Symposium pp 4-19 4-25 (Sept. 77)

- GEN-18 ZIMMERMANN H.
"Proposal for a virtual terminal protocol"
Réseau Cyclades, IRIA (Janv. 76)
- GEN-19 ELOVITZ H.S., HEITMEYER C.L.
"What is a computer network?"
Compcon Fall pp 5-3 5-10 (Sept. 77)
- GEN-20 SOC Project
Report 1, Project departement, IBM France (Juin 71)
- GEN-21 ISO
"Reference model of open system architecture"
TC97/SC16/N227 (Juin 79)

RESEAUX LOCAUX

- RES-1 ANDERSON G.A., JENSON E.D.
"Computer interconnexion structures: taxonomy,
characteristics and examples"
Computing surveys 7, 4, pp 197-213 (Déc. 75)
- RES-2 METCALF R.M., BOGGS D.R.
"Ethernet: distributed packet switching for local computer
networks"
CACM vol. 19 No 7 (Juil. 76)
- RES-3 ASHENHURST R.L., VONDEROHE R.H.
"A hierarchical network"
Datamation Vol. 21, No 2 pp 40-44 (Fev. 75)
- RES-4 WULF W., LEVIN R.
"A local network"
Datamation pp 47-50 (Fev. 75)
- RES-5 FARBER D.J.
"A ring network"
Datamation pp 44-46 (Fev. 75)

- RES-6 FARBER D.J., LARSON K.C.
"The system architecture of the distributed computer system:
the communication system"
Proc. Symposium on Computer Communications Networks and
Teletraffic pp 21-27 (Avril 72)
- RES-7 FORTUNE P.J., LIDINSKY W.P., ZELLE B.R.
"Design and implementation of a local computer network"
Conference Record, International Conference on Communication
(Avril 77)
- RES-8 CABANEL J.P. and all (Equipe Aramis)
"A decentralized control method in a distributed system"
Laboratoire de Systèmes Informatiques
Université Paul-Sabatier, Toulouse
- RES-9 HEBENSTREIT J.
"Projet M2: conception et réalisation d'un système
Multi-Mini-Ordinateurs"
Congrès AFCET, Gyf-sur-Yvette (Nov. 76)
- RES-10 WULF W.A, BELL C.G.
"C.mmp: a multi-mini-processor"
Proc. AFIPS, FJCC, Vol. 41 (1972)
- RES-11 ENSLOW P.H.
"Multiprocessor organisation: a survey"
Comcon Fall 77 pp 5-3 5-10
- RES-12 MANNING E., PEEBLES R.W.
"A homogenous network for data sharing: communications"
Computer Networks Vol. 1, No 4, pp 211-224 (Mai 77)
- RES-13 MANNING E., HOWARD R., O'DONNELL C., PAMMETT K., CHANG E.
"A UNIX-base local processor and network access machine"
Computer Networks, Vol. 1, No 2, pp 139-142 (Sept. 76)

RES-14 NEWMAN I.A., SLADE A.J.
 "Making an effective user system from a collection of minicomputers"
 Department of Computer Studies
 Loughborough University of Technology, U.K.

LANGAGES DE COMMANDE

LAN-1 SERGEANT G., FARZA M.N.
 "Machine interprétative pour la mise en oeuvre d'un langage de commande sur le réseau Cyclades"
 Thèse de Docteur Ingénieur, Toulouse Paul-Sabatier (Oct. 74)

LAN-2 CHUPIN J.C.
 "Command languages and heterogenous networks"
 in Unger ed, IFIP Working Conference on command languages, Suède p 311-318 (Juil. 74)

LAN-3 FELDMAN J.A.
 "High level programming for distributed computing"
 Communications of the ACM, Vol. 22, No 6 (Juin 79)

LAN-4 BALL E., FELDMAN J., RASHID R., ROVNER P.
 "RIG. Rochester's Intelligent Gateway: System overview"
 IEEE Transactions on Software Eng. SE-2, No 4, pp 321-328 (Déc. 76)

LAN-5 PRICE R.J.
 "A language for distributed processing"
 AFIPS Conference Proceedings, Vol. 48, pp 957 967 (1979)

LAN-6 SERGEANT G., DANG Ng. X.
 "Système et langage portable pour le traitement d'applications réparties sur un réseau hétérogène"
 Séminaire IRIA: Langages et traducteurs (1977)

- LAN-7 RAYMOND J.
"NJCL un langage de commande pour réseau d'ordinateurs"
Contrat CRI, Grenoble I (Juil. 73)
- LAN-8 DU MASLE J.
"An evaluation of the LE/1 network command language designed
for the SOC network"
in Unger ed, IFIP Working conference on command languages,
Suède (Juil. 74) pp 319 326

CENTRE DE CALCUL REPARTI

- CCR-1 CHAMBON J.F., GUIBOUD RIBAUD S., LE BIHAN B., TOSAN Y.
"Intérêt et faisabilité d'un centre de calcul fondé sur un
réseau de mini-ordinateurs"
Note interne, EMSE (Octobre 76)
- CCR-2 TOSAN Y.
"Langages et protocoles dans un centre de calcul réparti"
Thèse de 3ème cycle, EMSE (Mars 78)
- CCR-3 GUIBOUD-RIBAUD S.
"Système de réseaux de mini-ordinateurs spécialisés"
EMSE (Déc. 75)
- CCR-4 CART M.
"Développement d'un logiciel de Frontal pour un centre de
calcul réparti"
Rapport DEA, EMSE (Oct. 77)
- CCR-5 JUREMA M., NIANG A
"Système de coopération de systèmes de gestion de fichiers"
Rapport interne (Juil. 79)
- CCR-6 ANGELIDES J.
"Intégration de systèmes dans un réseau local d'ordinateurs.
Application: connexion du système TSM 16 au Centre de Calcul
Réparti"
Thèse de 3ème cycle, EMSE (à paraître)

