



**HAL**  
open science

# Un modèle de fonctionnement d'une couche d'une architecture de communication

Thu Nguyen Ba

► **To cite this version:**

Thu Nguyen Ba. Un modèle de fonctionnement d'une couche d'une architecture de communication. Génie logiciel [cs.SE]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 1983. Français. NNT : . tel-00809722

**HAL Id: tel-00809722**

**<https://theses.hal.science/tel-00809722>**

Submitted on 9 Apr 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.

# **THESE**

présentée par

**NGUYEN BA THU**

pour obtenir le diplôme de

**DOCTEUR-INGENIEUR**

**EN INFORMATIQUE**

**UN MODELE DE FONCTIONNEMENT  
D'UNE COUCHE D'UNE ARCHITECTURE  
DE COMMUNICATION**

Soutenue à Saint-Etienne le 6 Décembre 1983  
devant la Commission d'Examen :

**JURY**

M. J. FERRIÉ

Président

Mlle M. CART

MM. J.P. ANSART

S. GUIBOUD-RIBAUD

M. HABIB

}  
Examineurs



# **THESE**

présentée par

**NGUYEN BA THU**

pour obtenir le diplôme de

**DOCTEUR-INGENIEUR**

**EN INFORMATIQUE**

**UN MODELE DE FONCTIONNEMENT  
D'UNE COUCHE D'UNE ARCHITECTURE  
DE COMMUNICATION**

Soutenue à Saint-Etienne le 6 Décembre 1983

devant la Commission d'Examen :

## **JURY**

M. J. FERRIÉ

Président

Mlle M. CART

MM. J.P. ANSART

S. GUIBOUD-RIBAUD

M. HABIB

Examineurs



ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur : M. M. MERMET  
Directeur des Etudes et de la Formation : M. J. LEVASSEUR  
Directeur des Recherches : M. J. LEVY  
Secrétaire Général : Melle M. CLERGUE

---

PROFESSEURS DE 1ère CATEGORIE

MM. COINDE	Alexandre	Gestion
GOUX	Claude	Métallurgie
LEVY	Jacques	Métallurgie
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
RIEU	Jean	Mécanique - Résistance des Matériaux
SOUSTELLE	Michel	Chimie
FORMERY	Philippe	Mathématiques Appliquées

PROFESSEURS DE 2ème CATEGORIE

MM. HABIB	Michel	Informatique
PERRIN	Michel	Géologie
VERCHERY	Georges	Matériaux
TOUCHARD	Bernard	Physique Industrielle

DIRECTEUR DE RECHERCHE

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

MAITRES DE RECHERCHE

MM. BISCONDI	Michel	Métallurgie
DAVOINE	Philippe	Géologie
Mle FOURDEUX	Angeline	Métallurgie
MM. KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Métallurgie
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

PERSONNALITES HABILITEES A DIRIGER DES TRAVAUX DE RECHERCHE

MM. DRIVER	Julian	Métallurgie
GUILHOT	Bernard	Chimie
THOMAS	Gérard	Chimie



Je tiens à remercier :

Monsieur Jean FERRIE, Professeur à l'Université de Montpellier, qui m'a fait l'honneur de présider le jury de cette thèse et qui m'a donné des remarques constructives sur le contenu de ce mémoire.

Monsieur Jean Pierre ANSART, Directeur du projet RHIN à l'Agence de l'Informatique, qui a accepté de juger ce travail.

Monsieur Serge GUIBOUD-RIBAUD, Ingénieur chez Hewlett Packard, qui s'est intéressé à ce travail et qui a bien voulu faire partie du jury.

Monsieur Michel HABIB, Directeur du département informatique de l'Ecole des Mines de Saint-Etienne, grâce à la vigilance duquel ce travail a pu aboutir.

Mademoiselle Michèle CART, Assistante à l'Université de Montpellier, qui a guidé ce travail lorsqu'elle était Attachée de recherche à l'Ecole des Mines de Saint-Etienne et qui ne m'a jamais ménagé ni ses conseils ni ses encouragements.

J'exprime également ma reconnaissance à :

Monsieur Jean François CHAMBON pour les remarques et les corrections qu'il a apportées à ce mémoire.

Monsieur Paul André PAYS qui m'a aidé à constituer le jury.

Messieurs LOUBET, BROSSARD, VELAY et DARLES qui ont assuré avec beaucoup de soins et de gentillesse la réalisation de cet ouvrage.





## SOMMAIRE

INTRODUCTION . . . . .	1
------------------------	---

### CHAPITRE I :

STRUCTURATION D'UNE ARCHITECTURE DE COMMUNICATION . . . . .	7
1 - PRELIMINAIRE . . . . .	9
2 - APPROCHE TOPOLOGIQUE . . . . .	10
3 - APPROCHE FONCTIONNELLE . . . . .	12
4 - CONCEPT DE CONNEXION . . . . .	16
4.1 - Communication orientée connexion . . . . .	17
4.2 - Communication sans connexion . . . . .	18
5 - INTERFACE D'UTILISATION D'UN SERVICE DE COMMUNICATION. . . . .	19
5.1 - Facilités de service . . . . .	19
5.2 - Facilités rattachées au concept de connexion . . . . .	20
5.3 - Modèle d'interaction à l'interface du service. . . . .	20
6 - STRUCTURATION D'UN SERVICE DE COMMUNICATION. . . . .	25
6.1 - Niveaux d'abstraction d'un service de communication. . . . .	25
6.2 - Emballage d'un seul service. . . . .	30
6.3 - Emballage de plusieurs services relayés. . . . .	41

### CHAPITRE II :

FONCTIONS D'UNE COUCHE (n) . . . . .	53
1 - PRELIMINAIRE . . . . .	55
2 - DIFFERENTES UNITES DE DONNEES. . . . .	55
3 - FONCTIONS A L'INTERFACE. . . . .	58
4 - GESTION DES FONCTIONS DE COMMUNICATION . . . . .	60
4.1 - Protocole. . . . .	60
4.2 - Sous-protocole . . . . .	61
4.3 - Groupage - Dégroupage. . . . .	64
4.4 - Fragmentation - Réassemblage . . . . .	64
5 - ADRESSAGE DANS UNE COUCHE (n). . . . .	65

6 - MISE EN CORRESPONDANCE DES CONNEXIONS. . . . .	.67
6.1 - Multiplexage . . . . .	.68
6.2 - Eclatement . . . . .	.69
7 - UTILISATION DU SERVICE (n-1) . . . . .	.71

**CHAPITRE III :**

<b>EXEMPLE DE REALISATION DES COUCHES TRANSPORT ET SESSION . . . .</b>	<b>.73</b>
1 - PRELIMINAIRE . . . . .	.75
2 - REALISATION D'UNE STATION DE TRANSPORT . . . . .	.75
2.1 - Fonctions mises en oeuvre. . . . .	.76
2.2 - Schéma fonctionnel . . . . .	.77
2.3 - Schéma d'exécution . . . . .	.81
2.4 - Critiques. . . . .	.86
3 - REALISATION D'UNE ENTITE SESSION . . . . .	.88
3.1 - Fonctions mises en oeuvre. . . . .	.88
3.2 - Schéma fonctionnel . . . . .	.89
3.3 - Schéma d'exécution . . . . .	.92
3.4 - Critiques. . . . .	.95

**CHAPITRE IV :**

<b>DECOMPOSITION EN SOUS-NIVEAUX FONCTIONNELS D'UNE COUCHE (n) . .</b>	<b>.99</b>
1 - PRELIMINAIRE . . . . .	101
2 - STRUCTURATION D'UNE COUCHE (n) EN SOUS-COUCHES . . . . .	101
2.1 - Services (n) et (n-1) de type sans connexion . . . .	103
2.2 - Service (n) orienté connexion, service (n-1) de type sans connexion. . . . .	105
2.3 - Service (n) de type sans connexion, service (n-1) orienté connexion . . . . .	106
2.4 - Services (n) et (n-1) orientés connexion . . . . .	107
3 - CONCEPT DE NOEUD DE CORRESPONDANCE . . . . .	110

**CHAPITRE V :**

<b>MODELE DYNAMIQUE D'UNE COUCHE (n) . . . . .</b>	<b>113</b>
1 - PRELIMINAIRE . . . . .	115
2 - CONCEPT D'ACTIVITE . . . . .	115

3 - MODELE DYNAMIQUE D'UNE ENTITE (n) . . . . .	118
3.1 - Considérations initiales . . . . .	118
3.2 - Description du modèle dynamique. . . . .	120
3.3 - Fonctionnement du modèle . . . . .	128
4 - QUELQUES CAS PARTICULIERS DU MODELE. . . . .	136
4.1 - Service (n) orienté connexion, service (n-1) de type sans connexion. . . . .	136
4.2 - Service (n) de type sans connexion, service (n-1) orienté connexion . . . . .	137
5 - RESUME DES CARACTERISTIQUES DU MODELE. . . . .	138
<b>CONCLUSION</b> . . . . .	<b>141</b>
<b>BIBLIOGRAPHIE</b> . . . . .	<b>143</b>



## INTRODUCTION

### ARCHITECTURE DE RESEAUX

Les réseaux informatiques connaissent une expansion rapide à l'heure actuelle. Il devenait nécessaire de faire évoluer l'informatique centralisée vers une informatique distribuée car les systèmes d'informations sont par nature répartis. Mais cette évolution n'a été possible que grâce à des progrès réalisés dans plusieurs domaines : la technologie (micro-électronique, transmission) et les techniques informatiques avancées nécessaires au développement des réseaux (commutation de paquets, architecture de réseaux).

La définition d'une architecture est de première importance car les réseaux sont des systèmes complexes. Déjà, des constructeurs informatiques proposent des architectures qui leur sont propres : SNA chez IBM, DSA chez Bull, DECNET chez DEC, BNA chez Burrough, etc...

Toutes ces architectures sont basées sur une technique de structuration habituellement utilisée à l'heure actuelle : la structuration en couches. Cette technique permet de séparer nettement les fonctions réseaux et donc de faciliter la conception, la réalisation, l'évolution d'un réseau.

### MODELE DE REFERENCE POUR L'INTERCONNEXION DES SYSTEMES OUVERTS

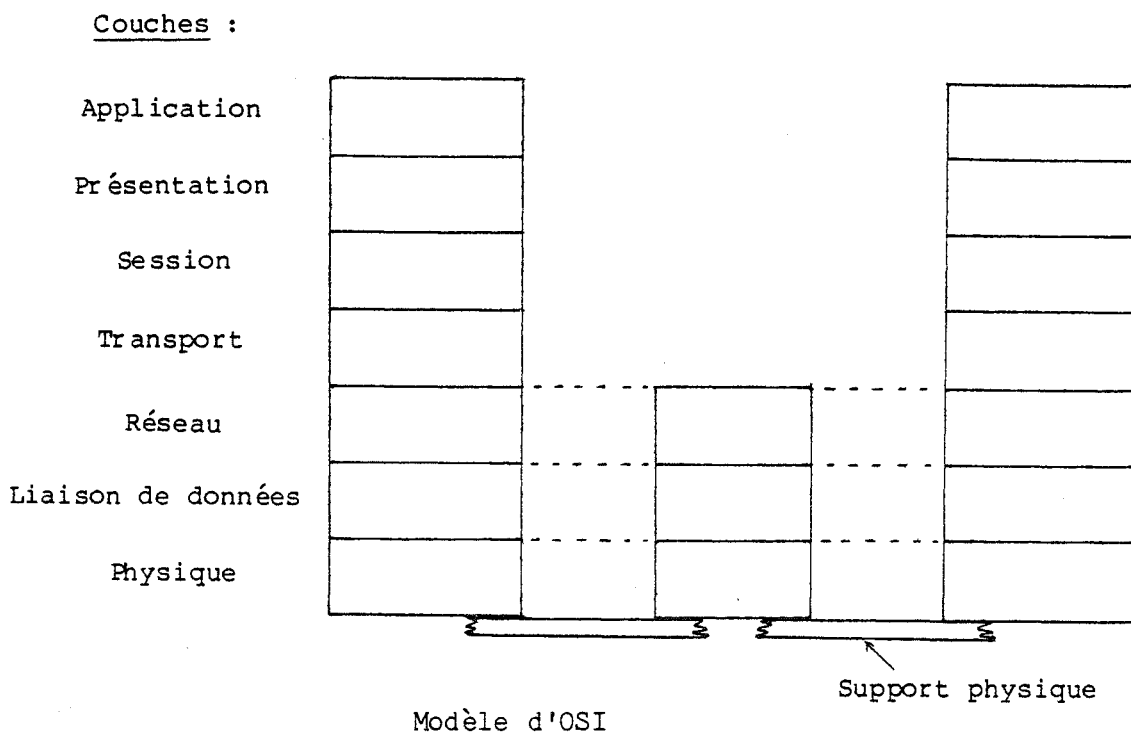
Malheureusement, les architectures de réseaux proposées par les différents constructeurs ne sont généralement pas compatibles tant sur le plan de structuration des fonctions que sur le plan de définition des protocoles de communication (\*). Ce qui exclut la possibilité d'interconnecter des équipements informatiques provenant de constructeurs différents sans un important investissement supplémentaire pour le développement des convertisseurs et adaptateurs appropriés.

---

\* i.e. des règles de dialogue entre 2 ou plusieurs systèmes

Avec la croissance très rapide du marché des réseaux informatiques, le besoin d'un ensemble de normes conçues pour les réseaux hétérogènes se fait sentir. Ainsi, en 1977 le comité technique 97 (TC 97) de l'Organisation Internationale de Normalisation (ISO) a créé un sous-comité 16 (SC 16) chargé d'étudier des normes d'interconnexion des systèmes ouverts (OSI) (\*).

Le premier résultat de travaux fut la proposition d'un modèle de référence définissant une architecture de réseau informatique hiérarchisée en 7 couches au sein de laquelle s'implantent des protocoles normalisés. Par la suite, le modèle doit être complété pour chaque couche par la définition d'un protocole normalisé.



Parallèlement à l'élaboration de ce modèle, le SC 16 coordonne les travaux de normalisation de trois autres groupes de

---

\* Open Systems Interconnection. De façon succincte, un système est dit "ouvert" s'il est apte à travailler dans un environnement hétérogène. Concrètement, il doit se conformer au modèle de référence d'OSI et aux protocoles standard.

travail : Gestion du réseau, Couche Application et Présentation, Couche Session et Transport.

Quant aux trois premières couches (Physique, Liaison de données, Réseau), la recommandation X25 du CCITT (\*) constitue déjà une norme internationale.

Selon le calendrier de l'ISO, on espère avoir tous les protocoles standard bien définis en 1984 /A1/.

#### PROJET PILOTE RHIN (\*\*) ET ORIGINE DE NOTRE ETUDE

Malgré la sortie progressive des normes internationales, "il existe encore trop d'ambiguïtés, trop de paramètres, une grande multiplicité d'options à partir desquelles on peut bâtir des réseaux à la fois normalisés et incompatibles !" (étude réalisée par INFOREP-BNI (\*\*\*) /A2/).

Consciente de ce type de problèmes, l'Agence de l'Informatique (ADI) a lancé, début 1981, le projet pilote RHIN dont l'objectif est de définir et d'expérimenter l'environnement technique nécessaire à l'élaboration, l'évaluation, la qualification, la production et la diffusion des protocoles normalisés /A3/.

Dans le cadre d'un contrat avec le projet RHIN, nous avons mené une étude sur la transformation d'un micro-ordinateur existant à vocation générale, le PCC 2000 de PERTEC, en un système "ouvert". Elle consiste, dans un premier temps, à intégrer dans un frontal destiné à être relié au micro-ordinateur, le logiciel X25 et dans un deuxième temps à mettre en oeuvre les couches Transport et Session conformément aux spécifications des protocoles proposés comme "pré-standard" par les organismes de normalisation.

---

\* Comité Consultatif International pour le Télégraphe et le Téléphone

\*\* Réseau Hétérogène Informatique Normalisé

\*\*\* INFOREP : Association générale des utilisateurs de l'informatique  
répartie

BNI : Bureau d'orientation de la normalisation en informatique



A la suite de cette réalisation, nous avons fait les remarques suivantes :

- Souvent, le concepteur doit spécifier lui-même de nombreuses fonctions locales dans la réalisation d'une couche donnée. Ces fonctions doivent être isolées des fonctions du protocole de la couche qui sont définies par des spécifications normalisées. D'une façon générale, il est nécessaire de bien séparer les différentes fonctions de la couche. On parvient ainsi à une structure modulaire qui facilite grandement la mise en oeuvre ainsi que des modifications ultérieures éventuelles.

- Il existe un certain schéma de structure de réalisation qui se répète d'une couche à une autre.

Ces remarques nous ont suggéré l'idée d'étudier un modèle de fonctionnement d'une couche qui soit à la fois général et modulaire pour en faciliter la description et la réalisation.

#### PLAN DE LA THESE

Dans le premier chapitre, nous rappelons les concepts d'interconnexion des systèmes ouverts (OSI), dans le cadre d'une présentation qui se veut pédagogique des architectures de communication. L'objectif est de donner une vue d'ensemble cohérente et non ambiguë de tous les termes et concepts de base d'architecture utilisés (la plupart du temps consacrés par les textes officiels) qui reviennent constamment dans la suite de l'étude.

Dans le chapitre II, nous exposons les fonctions internes d'une couche, tout en insistant sur leur hiérarchie de traitement et les problèmes à résoudre lors de leur mise en oeuvre.

Ces deux chapitres nous ont semblé nécessaires : d'abord comme introduction à ces notions nouvelles de la téléinformatique,

ensuite pour permettre au non spécialiste, peu habitué au "jargon" spécialisé de ces techniques, d'aborder la suite de l'étude sans se heurter à des problèmes de vocabulaire.

Le chapitre III est consacré à la présentation de la réalisation des couches Transport et Session. On y décrit la mise en oeuvre du logiciel autour d'un noyau système multitâche mis au point par nos soins. Nous avons choisi la démarche de présenter d'abord une réalisation car c'est cette expérience qui nous a permis de bien maîtriser les problèmes et de dégager quelques éléments d'une méthodologie de décomposition et de réalisation d'une couche quelconque.

Les 2 chapitres suivants proposent d'étudier un modèle de fonctionnement d'une couche de communication quelconque :

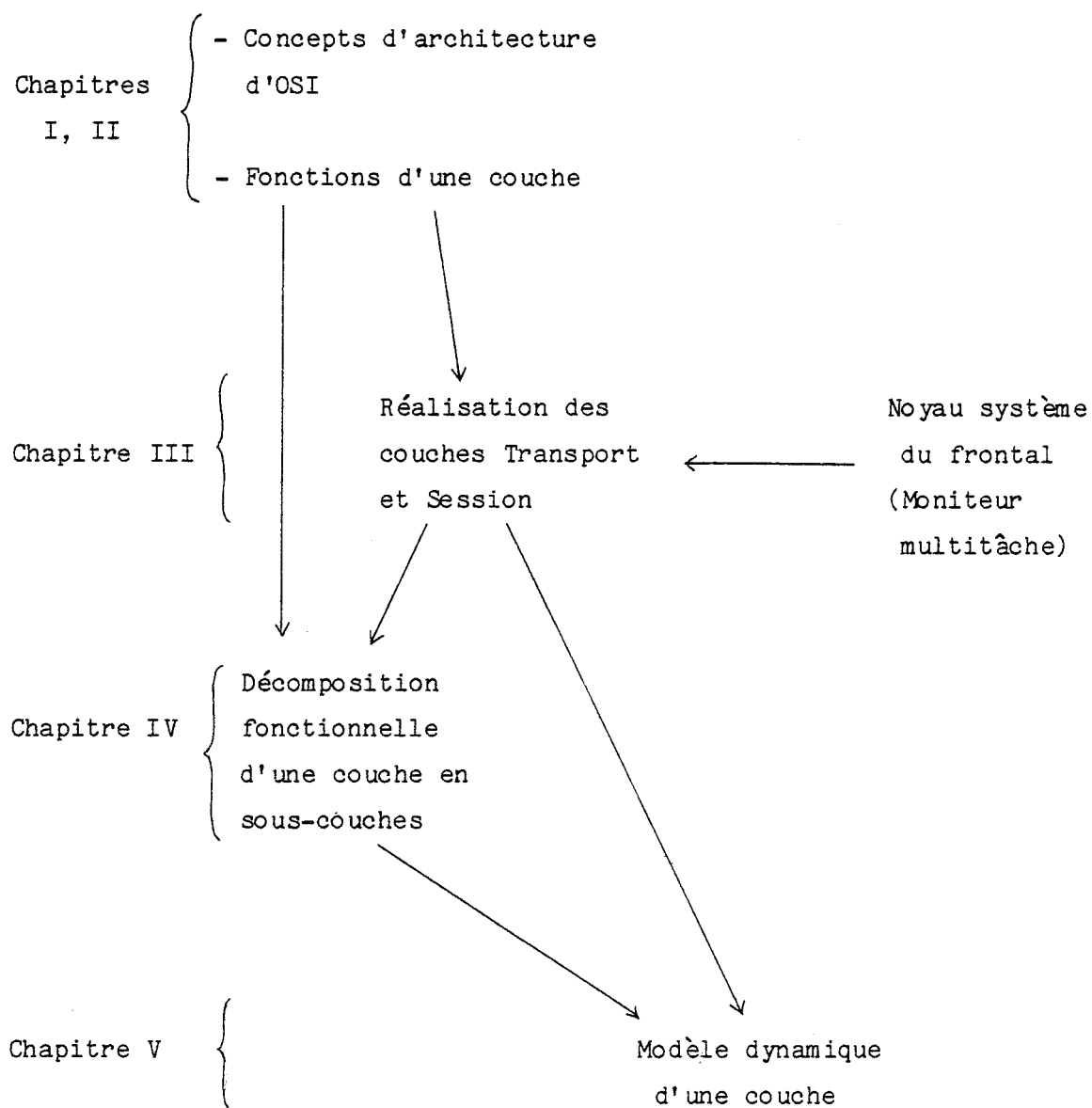
Dans le chapitre IV on envisage, d'une façon statique, la structuration et la décomposition fonctionnelle, et ceci sous forme de "sous-couches".

Le chapitre V se propose d'ajouter la composante temps et de donner ainsi un modèle dynamique de la couche.

Le tout devrait permettre de décrire avec suffisamment de précisions, le fonctionnement d'une couche et de mettre en évidence plusieurs aspects à prendre en compte lors de sa mise en oeuvre (parallélisme, fonctions d'administration locale,...).

Enfin, nous donnons dans la conclusion quelques idées qui peuvent orienter les prolongements possibles de cette étude.

Le schéma suivant donne la logique de l'enchaînement des chapitres. Les flèches indiquent la répercussion des différents éléments étudiés les uns sur les autres.



Plan de la thèse

## CHAPITRE I

### STRUCTURATION D'UNE ARCHITECTURE DE COMMUNICATION

- 1 - PRELIMINAIRE
- 2 - APPROCHE TOPOLOGIQUE
- 3 - APPROCHE FONCTIONNELLE
- 4 - CONCEPT DE CONNEXION
  - 4.1 - Communication orientée connexion
  - 4.2 - Communication sans connexion
- 5 - INTERFACE D'UTILISATION D'UN SERVICE DE COMMUNICATION
  - 5.1 - Facilités de service
  - 5.2 - Facilités rattachées au concept de connexion
  - 5.3 - Modèle d'interaction à l'interface du service
- 6 - STRUCTURATION D'UN SERVICE DE COMMUNICATION
  - 6.1 - Niveaux d'abstraction d'un service de communication
  - 6.2 - Emballage d'un seul service
  - 6.3 - Emballage de plusieurs services relayés



## 1 - PRELIMINAIRE

Il est apparu au fil des années, dans le domaine des réseaux, un effort de modélisation et de standardisation de l'architecture des systèmes de communication. Le modèle de référence de l'ISO (International Organization for Standardization) sur l'Interconnexion des Systèmes Ouverts (ou OSI, abréviation de 'Open Systems Interconnection') en est un exemple /B1/. Il fait intervenir de nombreux concepts dits d'OSI : couche, service, interface, protocole, adressage,... En outre, dans quelques articles publiés dans ce domaine, on trouve une tentative de synthèse des concepts de connexion /B2/, de structuration des systèmes de communication /B3,B4/, qui facilitent la compréhension du modèle d'OSI ou lui apportent des précisions.

Nous tentons dans ce chapitre de passer en revue tous les aspects d'une architecture de communication. Les grandes rubriques suivantes servent de cadre à la présentation :

- Une **approche topologique** qui donne une vue globale de l'environnement d'une communication à l'aide des notions : système, réseau de systèmes,...

- Une **approche fonctionnelle** qui présente les concepts utilisés pour la description du fonctionnement d'une communication : service, connexion, interface d'utilisation du service,...

- Une **structuration** de l'architecture de communication qui vise à expliciter les notions de couches et de superposition de couches.

Nous essayerons de rendre ce chapitre le plus didactique possible. Pour ce faire, nous serons amenés à rappeler des concepts connus pour certains. Le spécialiste réseau pourra donc survoler ou même sauter ce chapitre sans que cela nuise à la compréhension de la suite de la thèse.

La terminologie utilisée provient, en grande partie, des documents officiels décrivant le modèle de référence d'OSI.

## 2 - APPROCHE TOPOLOGIQUE

Nous rappelons dans ce paragraphe la définition de **système** et de **réseau de systèmes** et introduisons la notion de **système abstrait**. Ces concepts permettent de décrire l'environnement physique supportant les communications entre machines.

Voici la définition de "système" au sens d'OSI :

Système : ensemble comprenant un ou plusieurs ordinateurs, le logiciel associé, des périphériques, des terminaux, des opérateurs humains, des processus physiques, des moyens de transfert d'information etc... et constituant un tout autonome capable d'effectuer des traitements et/ou des transferts d'information /B1/.

Des systèmes s'interconnectent pour former des **réseaux** grâce à des **supports physiques** (ligne électrique, fibre optique, faisceau hertzien,...) sur lesquels transitent des informations sous des formes très élémentaires (ondes, signaux) (figure 1).

La définition donnée ci-dessus ne caractérise pas complètement le système, en particulier elle ne décrit pas le rôle assumé par ce système. Dans une communication par exemple, un système peut être une source, un puit ou un endroit de transit des informations.

Comme on ne préjuge pas de son contenu, un système peut lui-même contenir un réseau de systèmes interconnectés. On peut alors parler de **système abstrait** (figure 2). Le terme "system cut" utilisé dans /B4/ désigne ce même concept.

Prenons l'exemple d'un réseau de commutation de paquet (TRANSPAC, DATAPAC ...) : chaque noeud du réseau est caractérisé par un rôle de routage des informations et le réseau constitué de tous les noeuds peut être considéré comme un système abstrait chargé du transport des informations. De même, l'interconnexion de 2 réseaux forme un système global de transport des informations (figure 3).

Quelque soit l'abstraction réalisée, on obtient toujours une configuration d' **interconnexion de systèmes** servant de support pour les communications.

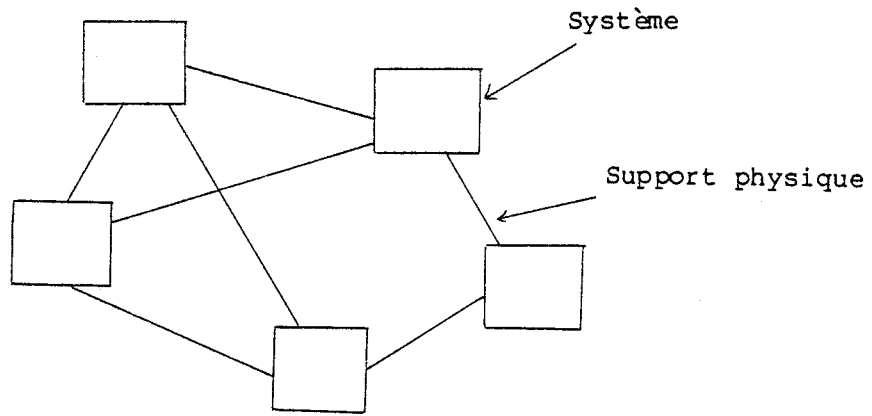


Figure 1 : Réseau de systèmes

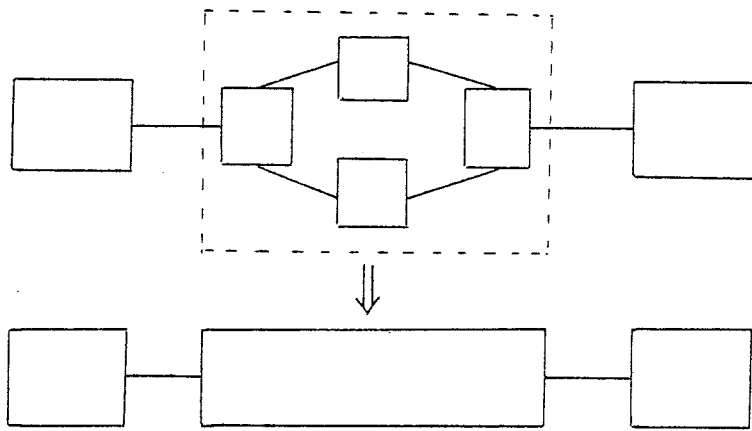


Figure 2 : Abstraction de systèmes

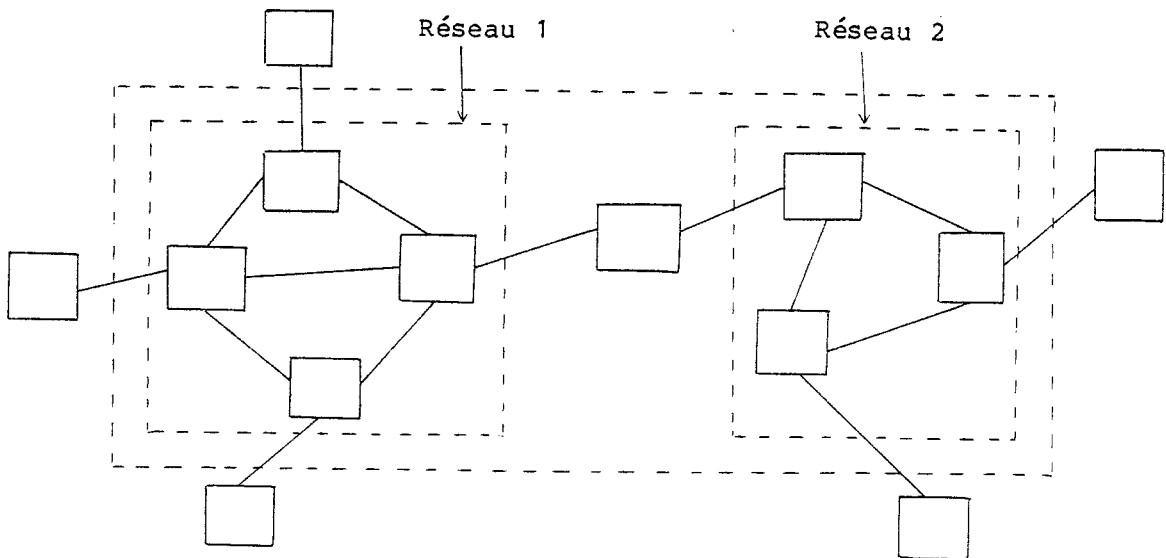


Figure 3 : Interconnexion de deux réseaux



### 3 - APPROCHE FONCTIONNELLE

Dans cette approche, la notion de systèmes se dissimule derrière le concept d'entités fonctionnelles.

Une **entité fonctionnelle (\*)** est un élément logique, indépendant de l'implantation, assumant une certaine fonction dans une architecture informatique donnée.

Dans un environnement d'informatique répartie, une application est généralement composée de plusieurs entités coopérantes qui se trouvent dispersées sur des systèmes distincts. La nécessité de communiquer est une caractéristique commune de toutes les applications réparties; c'est pourquoi, on essaie de rendre la fonction de communication indépendante des applications.

Le modèle fonctionnel proposé sur la figure 4 divise l'univers de l'informatique répartie en deux mondes :

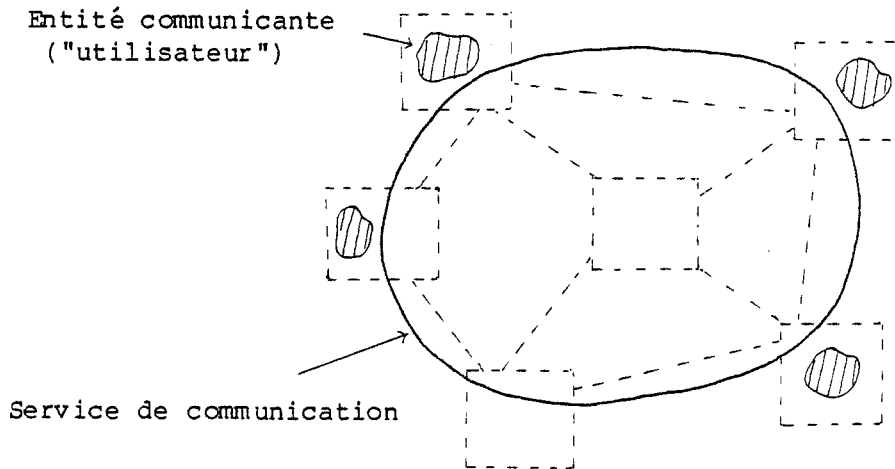


Figure 4 : Service de communication

---

\* Nous utiliserons tout au long des chapitres le terme "entité" qui est un terme consacré dans le vocabulaire d'OSI.

- le monde du **service de communication**
- le monde des **entités d'application** ou **entités communicantes**, qui constituent l'espace des utilisateurs du service de communication. Par la suite, nous appellerons les entités communicantes "**utilisateurs**".

Cette division permet aux entités d'application de se décharger de la tâche de gestion des communications et de concentrer les efforts sur leur travail spécifique.

Remarquons qu'un service de communication est lui-même une activité répartie puisqu'il résulte d'un travail commun réalisé par tous les systèmes du réseau. Si l'on utilise le modèle précédent pour décomposer la structure du service, il apparaît un nouveau service de communication à l'intérieur du premier qui assure un sous-ensemble des fonctionnalités de communication initiales (figure 5).

Le même raisonnement pourra s'appliquer de nouveau et ainsi de suite. Ceci suggère intuitivement une méthode de décomposition fonctionnelle d'un service de communication.

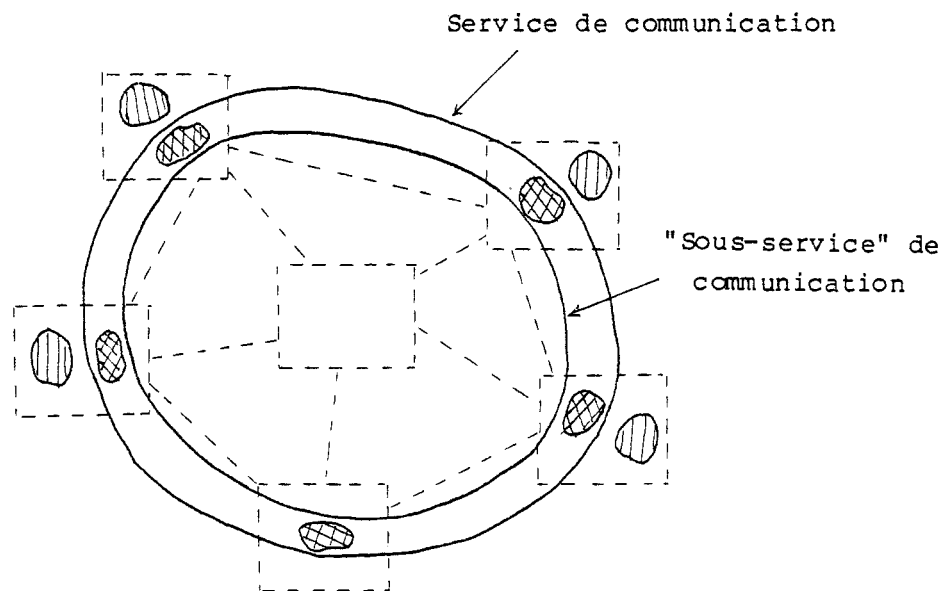


Figure 5 : Décomposition fonctionnelle d'un service de communication

Dans la suite de ce paragraphe, nous présentons brièvement les principales fonctions de **communication** généralement mises en oeuvre :

a/ **L'adressage** : Pour assurer une communication entre deux ou plusieurs utilisateurs, le service doit être capable en premier lieu de les désigner. Pour cela, une **fonction d'adressage** intégrée dans le service s'avère nécessaire.

Notons d'abord que les utilisateurs se distinguent l'un de l'autre grâce à des **appellations** (des noms) qui leur sont attribuées individuellement. L'ensemble de tous ces noms constitue le **domaine des appellations** .

La fonction d'adressage peut être réalisée par l'une des deux méthodes suivantes :

- La première consiste à mémoriser dans le service toutes les appellations des utilisateurs; chaque système du service possède alors un exemplaire du domaine des appellations. Il en résulte que le système devient très sensible à toute modification du domaine des appellations qui entraîne de lourds traitements (mise à jour des copies multiples).

- La seconde consiste à définir des "guichets" entre le service et les utilisateurs. Ces "guichets" sont figés et connus par toutes les parties concernées. Cette solution permet de rendre le service indépendant de l'espace des utilisateurs. Le concept du **point d'accès** au service donné dans le modèle d'OSI est basé sur ce principe (voir plus loin au 6.3.1).

b/ **Le transfert de données** qui est la fonction de base de la communication ; elle doit prendre en compte les besoins des utilisateurs, par exemple :

- La nature de la communication : dialogue (deux partenaires), colloque (plusieurs partenaires).

- La nature des messages échangés : messages isolés (communication du type transactionnel), suite de messages (transfert de volumes de données).

- La performance et la qualité du transfert : celles-ci varient en fonction des besoins et du type d'application (transport de la voix, transport des documents, applications temps réel,...) et font intervenir les paramètres suivants :

. Débit de la transmission : la régulation du flot de données est assurée par la fonction de **contrôle de flux** .

. Délai de transmission

. Fiabilité de la transmission : elle nécessite la mise en oeuvre des fonctions de **contrôle d'erreur** et de **récupération sur erreur** .

c/ **La gestion du dialogue** : Si cette fonction est mise en oeuvre, le service assure que le dialogue entre les utilisateurs se déroule conformément à un certain scénario. Par exemple, une communication bidirectionnelle simultanée autorise les utilisateurs à émettre les données en même temps; une communication à l'alternat interdit des émissions simultanées.

d/ **La gestion de la synchronisation** : c'est la fonction qui permet aux utilisateurs de synchroniser leurs échanges et en cas d'anomalie, de reprendre le dialogue à un certain point du passé sans avoir à tout recommencer.

e/ **L'interprétation des données** : lorsque les utilisateurs n'ont pas les mêmes conventions de codage des informations, leur communication nécessite la mise en place d'une fonction d'interprétation des données à l'intérieur du service.

On peut donc dire qu'un service de communication se comporte simultanément comme un "messenger", un "traducteur" et un "arbitre de dialogue".

Notons finalement que l'implantation de toutes les fonctions présentées précédemment à l'intérieur du service est totalement transparente aux utilisateurs.

Dans les paragraphes suivants, nous étudierons :

- Le **concept de connexion** : la connexion est un support logique de communication fourni par le service.

- L' **interface d'utilisation du service** : on définira un modèle d'interaction entre le service et les utilisateurs.

- La **structuration d'un service de communication** : on introduira les notions de couche et de superposition de couches.

#### 4 - CONCEPT DE CONNEXION

Nous étudions dans ce paragraphe deux modes de communication selon l'utilisation ou non du support logique de communication qu'est la connexion. Un article /B2/ a fait le point sur ce concept. Nous le présentons ici à notre manière.

Une **connexion** est une association entre deux ou plusieurs utilisateurs, à laquelle sont rattachées les fonctions de communication du service : le transfert de données, la gestion du dialogue, la gestion de la synchronisation, etc...

Une connexion reliant deux utilisateurs est dite connexion point à point. Une connexion qui en relie plusieurs est dite connexion multipoint (figure 6).

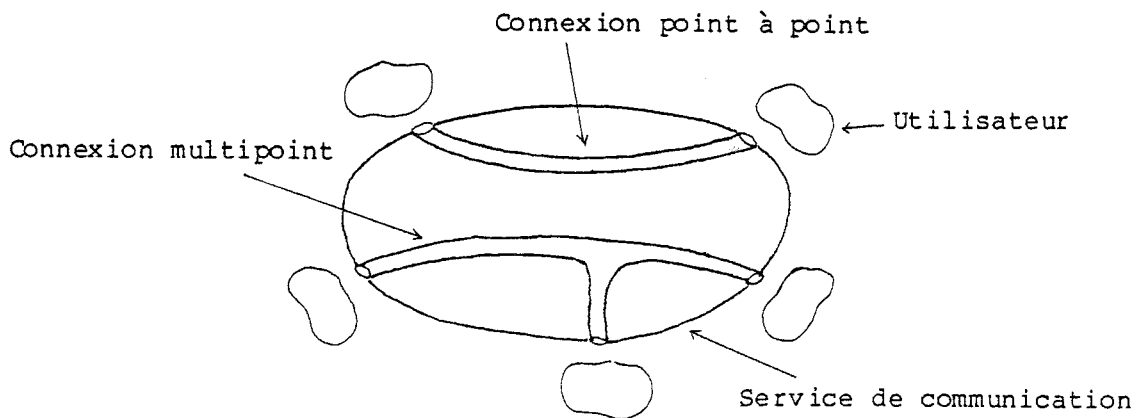


Figure 6 : Connexion

Dans un même service, on peut avoir des communications de qualités différentes d'une connexion à l'autre. Une connexion individualise par conséquent une communication.

#### 4.1 - Communication orientée connexion

Ce mode de communication est caractérisé par trois phases de fonctionnement :

- L'établissement de la connexion : le service n'établit qu'une connexion qu'après un accord mutuel des utilisateurs sur les caractéristiques de la communication et à condition que le service soit capable de la gérer. Dans cette phase, les utilisateurs peuvent négocier, par l'intermédiaire du service, la "qualité" de la communication. La connexion ainsi créée et maintenue par le service garantit cette qualité pendant toute la durée de la communication. En outre, l'identificateur attribué à la connexion au moment de sa création permet de désigner une communication parmi plusieurs au sein du service.

- L'utilisation de la connexion : c'est à dire des fonctions de communication fournies par le service et négociées à l'établissement de la connexion ; par exemple : transfert de données, octroi d'un droit pour émettre (dialogue à l'alternat), resynchronisation, etc...

- La libération de la connexion : elle s'effectue sur l'initiative soit d'un utilisateur lorsque celui-ci désire terminer la communication, soit du service lorsqu'il ne lui est plus possible de maintenir la qualité de la communication requise. Dans tous les cas, le service prend en charge la libération de la connexion, éventuellement après l'accord de tous les utilisateurs concernées.

Globalement, ce mode de communication présente les caractéristiques suivantes :

- La qualité de la communication est négociable entre les utilisateurs. Par cette raison, les fonctions de contrôle d'erreur et de contrôle de flux, ainsi que les fonctions de gestion du dialogue et de synchronisation peuvent être choisies et paramétrées indépendamment pour chaque connexion.

- L'adressage est simplifié dans les deux phases d'utilisation et de libération de la connexion grâce à son identificateur fourni au moment de son établissement.

- L'ordre d'émission des blocs de données par un utilisateur est conservé d'une extrémité de connexion à l'autre.

Du point de vue des applications, le service orienté connexion est le mieux adapté aux communications pour lesquelles s'impose un séquençement des données transférées (transfert des volumes de données) ou une bonne qualité de transfert (communication à fiabilité importante, à régulation de flux,...). Un tel service augmente en général le délai de propagation des données d'un bout à l'autre en raison d'une multitude de contrôles effectués.

#### 4.2 - Communication sans connexion

Ce mode de communication s'adapte à un autre type de besoin des utilisateurs. Pour des communications de courte durée qui ne donnent lieu qu'à des échanges de messages isolés (transactions) dont l'ordre n'est pas important et dont le transfert doit être pris en compte immédiatement, il n'est généralement pas nécessaire d'établir une connexion.

Le service opérant selon le mode "sans connexion" achemine les blocs de données entre d'un utilisateur à un autre sans qu'il faille établir au préalable un accord comme dans le cas du service orienté connexion. En général, le service n'effectue aucun contrôle spécifique sur les données à transférer et considère les blocs de données comme étant indépendants les uns des autres.

On trouve plusieurs exemples de ce mode de communication dans la littérature des réseaux informatiques : service datagramme, message /B5/, connexion libre /B6,B7/.

Du point de vue fonctionnel, un service orienté connexion est plus complexe qu'un service sans connexion car il comporte, en dehors des fonctions de contrôle plus sophistiquées de transfert de données, celles relatives à l'établissement et à la libération des connexions. C'est pourquoi dans la suite, nous nous intéresserons essentiellement à des services orientés connexion.

## 5 - INTERFACE D'UTILISATION D'UN SERVICE DE COMMUNICATION

Les règles d'interaction entre le service et les utilisateurs constituent l'interface d'utilisation du service. Leur description nécessite l'introduction de nouveaux concepts : **facilités, primitives, points d'accès au service**. Nous en rappelons les aspects essentiels dans ce qui suit.

### 5.1 - Facilités de service

Comme nous l'avons dit dans le paragraphe 3, l'implantation des fonctions de communication à l'intérieur du service est totalement transparente aux utilisateurs. A l'interface, l'utilisation du service se traduit en terme de **facilités** auxquelles les utilisateurs peuvent accéder.

Le déroulement d'une facilité de service donnée implique nécessairement des **interactions** entre le service et les utilisateurs (figure 7).

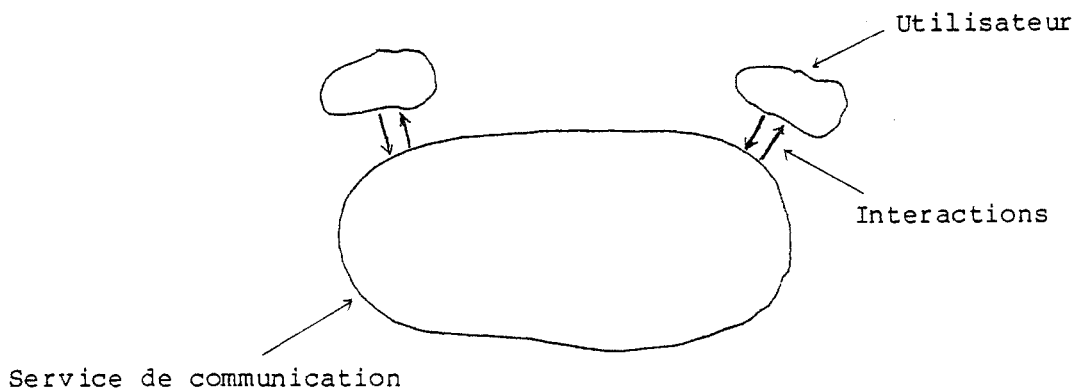


Figure 7 : Interface d'utilisation du service



## 5.2 - Facilités rattachées au concept de connexion

- Une facilité d'établissement d'une connexion : elle permet de demander l'établissement d'une connexion à l'intérieur du service en fonction de la qualité de la communication demandée par les utilisateurs.

- Des facilités d'utilisation d'une connexion : transfert de données, échange d'un droit d'émission (gestion du dialogue), pose des points de synchronisation, resynchronisation (gestion de la synchronisation), etc...

- Une facilité de libération d'une connexion : elle permet aux utilisateurs de libérer la connexion précédemment établie.

## 5.3 - Modèle d'interaction à l'interface du service

Le déroulement d'une facilité est vu de l'extérieur du service comme un enchaînement des interactions entre les utilisateurs et le service. Ainsi, l'interface d'utilisation du service sera entièrement définie par une description de toutes les interactions et de tous leurs enchaînements possibles.

Le modèle de description des interactions suivant est basé sur deux notions : **points d'accès au service** et **primitives de service** /B8/.

### 5.3.1 - Points d'accès au service

C'est le "guichet" du service par lequel passent les interactions entre le service et un utilisateur (figure 8).

Chaque point d'accès au service est identifié par un nom, appelé **adresse du point d'accès** . L'ensemble des adresses de tous les points d'accès à un service constitue le **domaine d'adressage** du service /B9/.

Le concept de point d'accès permet de rendre le service indépendant de l'espace des utilisateurs en ce qui concerne l'adressage. En effet, pour résoudre l'adressage, il suffit qu'un utilisateur soit associé à un point d'accès à un instant donné ; il sera alors l'unique destinataire des interactions issues du service et passant par ce point d'accès. De telles associations peuvent être modifiées dynamiquement pour changer le destinataire des interactions sans influencer le domaine d'adressage du service.

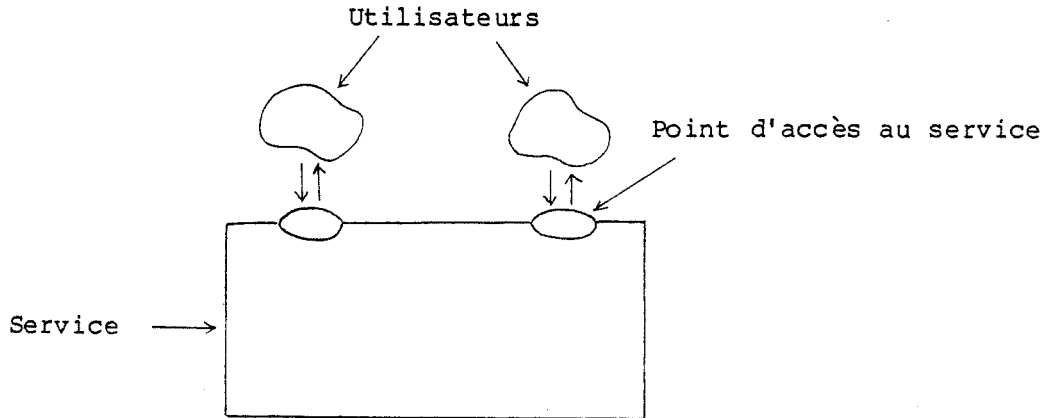


Figure 8 : Points d'accès au service

Dans l'exemple de la figure 9, les associations locales étant "entité X - point d'accès A" et "entité Y - point d'accès B", l'entité X désirant établir une communication avec l'entité Y s'adresse au point d'accès A et demande l'établissement d'une connexion entre deux adresses A et B. Le service, possédant la connaissance du domaine d'adressage pourra effectuer cette liaison sans avoir besoin de connaître l'identité des entités X et Y. Si au cours du temps, une entité Z vient remplacer l'entité X pour communiquer avec l'entité Y, elle peut utiliser la connexion précédente.

Notons que plusieurs extrémités de connexion peuvent coexister en un point d'accès donné.

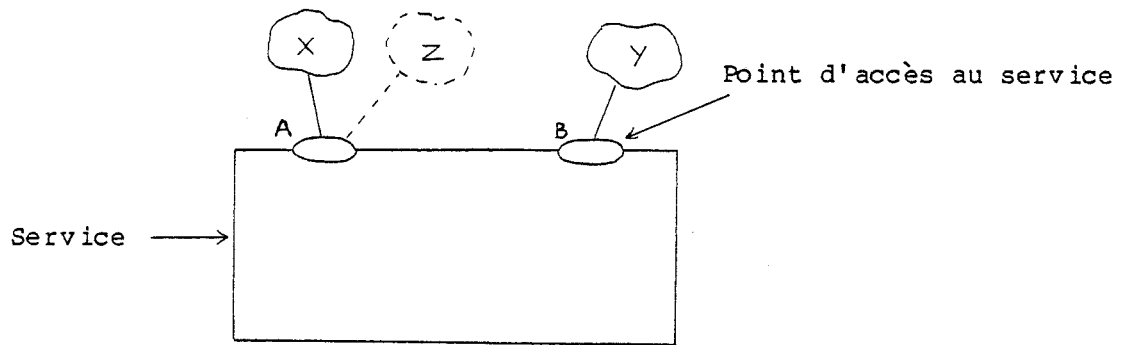


Figure 9 : Adressage par les points d'accès

### 5.3.2 - Primitives de service

Une **primitive** est un élément conceptuel, indépendant de l'implantation, décrivant une interaction relative à une facilité donnée entre le service et un utilisateur.

Selon la direction et la signification de l'interaction, on distingue 4 types principaux de primitives /B8/ :

- Requête : interaction issue d'un utilisateur et portant la demande d'une certaine facilité de service.

- Indication : interaction issue du service et destinée à un utilisateur. Le service la génère soit pour répercuter l'événement "requête d'une facilité" provenant d'un utilisateur, soit pour notifier à l'entité communicante l'état de fonctionnement du service (anomalie,...).

- Réponse : interaction issue d'un utilisateur et portant son avis, refus ou acceptation, sur l'indication reçue proposant une facilité de service. La réponse n'est pas obligatoire dans certains cas.

- Confirmation : interaction issue du service en un point d'accès où a été émise une requête, pour notifier à l'utilisateur la réponse de son correspondant.

Les primitives sont schématisées par des flèches partant du service (indication, confirmation) ou pointant vers le service (requête, réponse) en un point d'accès donné.

### 5.3.3 - Groupes de primitives

Les primitives de types différents et concernant une même facilité de service forment un groupe. Il existe par exemple le groupe de primitives d'établissement d'une connexion, celui de libération d'une connexion, ou celui de transfert de données, etc ...

Le déroulement normal, c'est à dire sans anomalie, d'une facilité de service produit un enchaînement des primitives du groupe correspondant. Citons un exemple : l'établissement d'une connexion point à point se traduit par la suite de primitives ci-dessous :

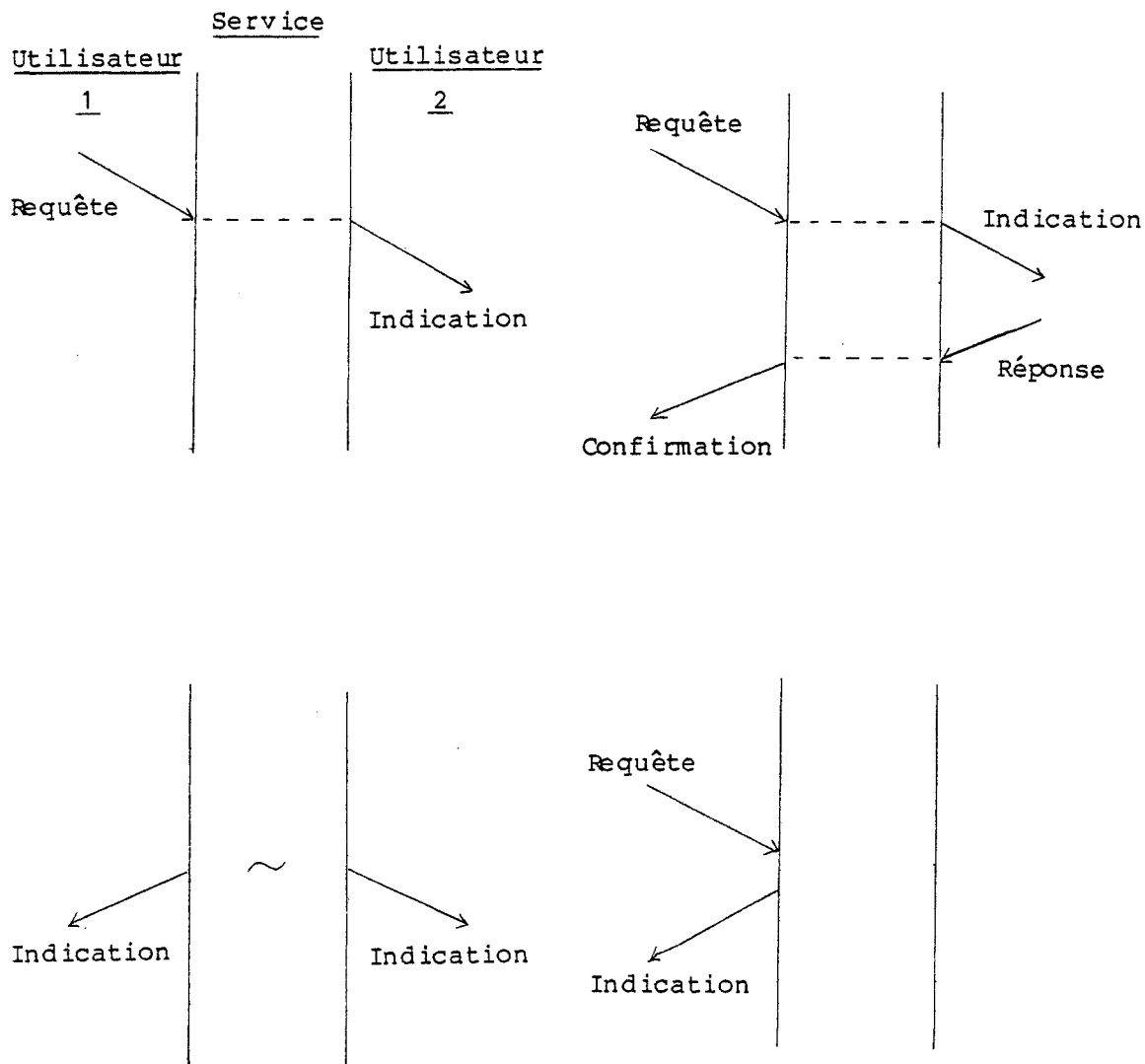
- Requête d'établissement d'une connexion : elle est émise par un utilisateur et porte des paramètres donnant les caractéristiques souhaitées de la communication.

- Indication d'établissement d'une connexion : elle est envoyée par le service à l'utilisateur destinataire pour lui demander son avis.

- Réponse d'établissement d'une connexion : après l'examen des paramètres proposés, l'utilisateur envoie une réponse au service donnant son avis (accord ou refus) avec éventuellement des contre-propositions.

- Confirmation d'établissement d'une connexion : le service répercute la réponse précédente à l'utilisateur qui a demandé l'établissement de la connexion.

Voici quelques exemples d'enchaînement des primitives du même groupe. On utilise une méthode de représentation par **diagramme temporel des séquences** /B8/.



- - - - : les pointillés signifient qu'il y a une relation d'ordre entre les 2 primitives. L'une est générée à la suite de l'arrivée de l'autre au service.

~ : le ~ signifie qu'il n'y a pas de relation d'ordre entre les primitives de part et d'autre du symbole

Figure 10 : Enchainement des primitives

## 6 - STRUCTURATION D'UN SERVICE DE COMMUNICATION

Comme nous l'avons mentionné au paragraphe 3, il est possible d'utiliser des niveaux d'abstraction successifs pour structurer un service de communication. Cette technique, déjà classique en programmation structurée, présente plusieurs intérêts :

- C'est une démarche systématique que le concepteur peut utiliser pour analyser n'importe quel service de communication.

- Elle permet de créer de nouveaux services plus puissants, plus évolués ou plus fiables à partir des services existants en leur adjoignant des fonctions supplémentaires.

- Une modification du service s'effectue généralement à un niveau d'abstraction donné et n'influence pas les niveaux d'abstraction inférieurs, un niveau d'abstraction étant formé d'un ou plusieurs services considérés comme des "boîtes noires".

Dans le domaine des réseaux informatiques, cette technique est habituellement utilisée. H. ZIMMERMANN dans /B3/ la désigne en terme d' **emballage** . On trouve des exemples d'utilisation de cette technique dans /B4/.

Dans ce paragraphe, nous montrons que cette technique appliquée à un service de communication conduit à l'étude de deux configurations de base : **emballage d'un seul service** et **emballage de plusieurs services interconnectés** (ou services relayés). Elle implique une architecture hiérarchisée en niveaux : des niveaux d'adressage, de connexions, de protocoles, etc... Plusieurs exemples tirés d'applications pratiques permettent d'illustrer nos idées.

### 6.1 - Niveaux d'abstraction d'un service de communication

Un service de communication doit gérer non seulement des fonctions de communication de haut niveau telles que la gestion du dialogue et de la synchronisation, le transfert des messages, mais

également des fonctions de très bas niveau telles que l'échange des signaux sur les supports physiques. C'est pourquoi, la décomposition du service en plusieurs niveaux fonctionnels successifs permet d'une part de donner une structure modulaire au service, et d'autre part d'y répartir ses fonctions (figure 11).

Les fonctions qui se trouvent dans le niveau le plus bas (ou le plus interne) gèrent évidemment le transfert des informations élémentaires sur le support physique.

Nous attribuons un numéro  $n$  à un niveau de service donné. Le service est alors noté service  $(n)$ , son niveau adjacent inférieur est noté service  $(n-1)$ , son niveau adjacent supérieur, service  $(n+1)$ .

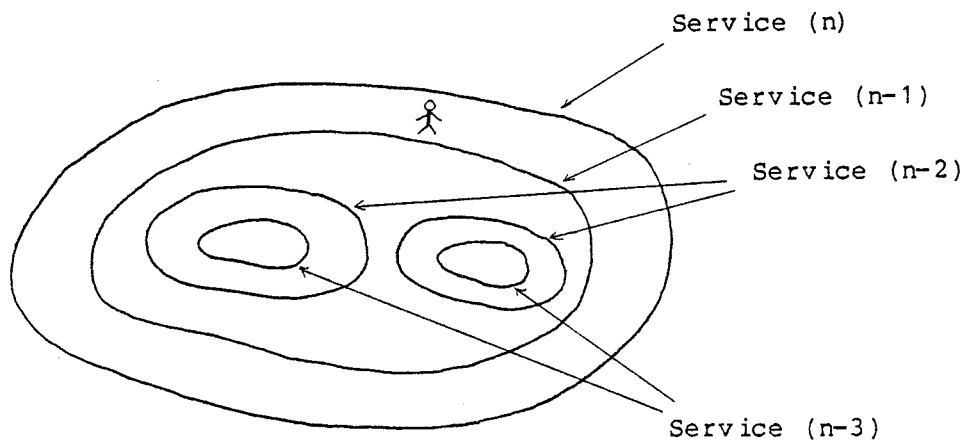


Figure 11 : Niveaux d'abstraction d'un service de communication

La caractéristique essentielle de cette structuration est la transparence complète de la structure interne d'un niveau par rapport aux niveaux supérieurs. Dans la figure 11, l'observateur qui se trouve au niveau  $(n)$  ignore complètement la réalisation du service  $(n-1)$ . Une conséquence importante de cette caractéristique est que les informations échangées entre les entités d'un certain niveau ont des significations propres à ce niveau.

En examinant les différentes réalisations des services de communication existants, nous pouvons dire que la structuration d'un service s'appuie sur les deux configurations de base suivantes :

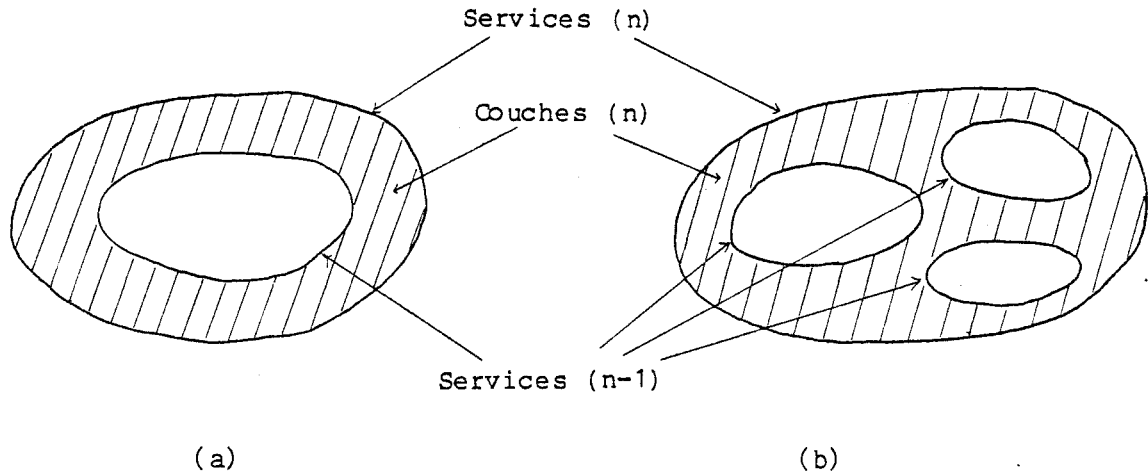


Figure 12 : Deux configurations de base

La première est l'emballage d'un seul service (fig 12.a), et la seconde est l'emballage de plusieurs services du même niveau fonctionnel (fig 12.b). On peut dire que la première configuration est un cas particulier de la seconde; toutefois leur distinction a pour but de faciliter la présentation des problèmes spécifiques à chacune d'elles. Dans la pratique, le premier cas correspond souvent à une amélioration d'un service existant alors que le deuxième cas s'applique au problème de relai des services de communication.

On voit apparaître dans les deux cas un **espace fonctionnel** entre l'interface du service (n) englobant et ceux des services (n-1) englobés, qui représente l'activité additionnelle par rapport au niveau fonctionnel (n-1). On l'appelle une **couche (n)** .

Un service de communication est alors constitué d'un empilement de couches. En voici deux illustrations :

a/ Modèle d'OSI (Interconnexion des Systèmes Ouverts) /B1/

L'ISO propose un modèle de système construit selon une architecture à 7 couches. La couche la plus haute contient les applications et les 6 couches restant forment un service de communication dont voici la décomposition fonctionnelle :



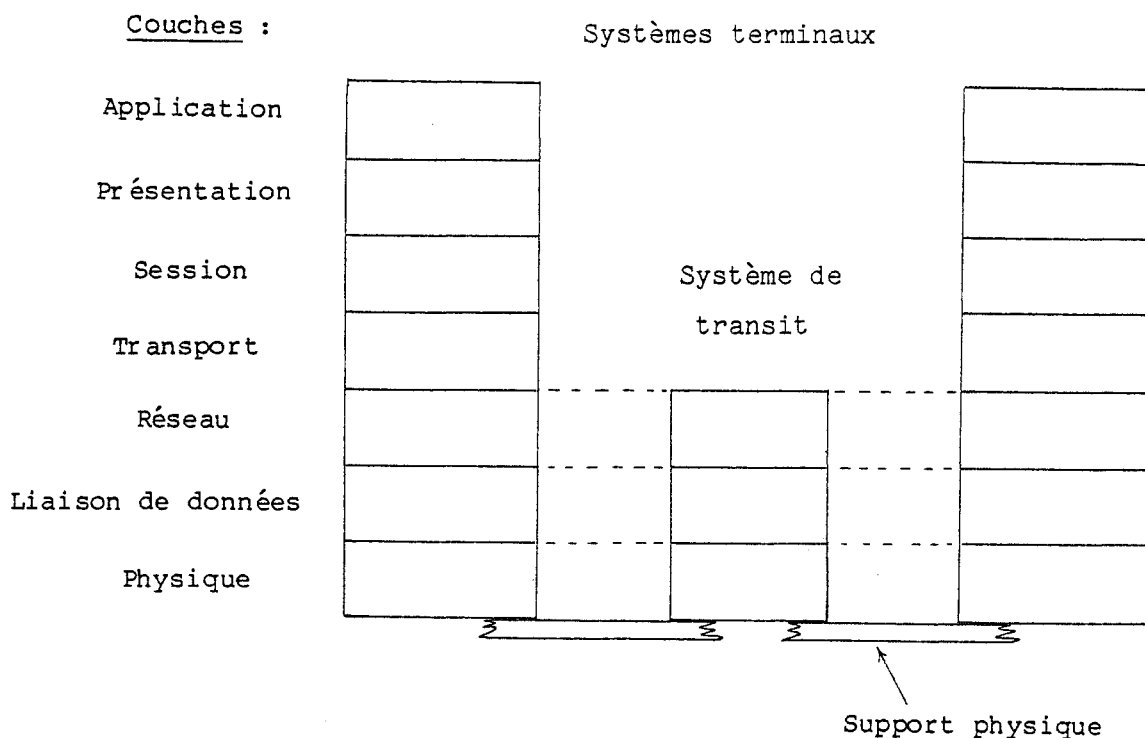


Figure 13 : Modèle d'OSI

- Couche Physique (1) : assure le transfert des informations au niveau du signal entre 2 systèmes adjacents reliés par des supports physiques.

- Couche Liaison de données (2) : assure le transfert des informations au niveau d'une chaîne de bits entre 2 systèmes adjacents reliés par des supports physiques.

- Couche Réseau (3) : assure l'établissement d'une ou plusieurs liaisons entre deux systèmes terminaux quelconques grâce à une série de liaisons fournies par la couche 2 et l'acheminement des données (paquets) sur ces liaisons.

- Couche Transport (4) : assure un contrôle de transfert de données entre deux systèmes terminaux (contrôle de bout en bout) et une optimisation de l'utilisation du service Réseau. Cette couche permet à l'utilisateur de choisir une certaine qualité de service de transport de données de bout en bout en fonction de ses besoins.

- Couche Session (5) : contient des fonctions de contrôle du dialogue et de la synchronisation entre tâches distantes.

- Couche Présentation (6) : contient des fonctions de transformation des données transférées entre des entités d'application

de la couche 7. Des formes standard de présentation des données sont définies dans cette couche.

b/ Interconnexion de deux réseaux de communication

L'objectif de l'interconnexion de 2 réseaux de communication est d'élargir le champ de communication des utilisateurs. Elle permet en effet aux utilisateurs d'un des réseaux de communiquer avec ceux de l'autre.

Du point de vue de l'utilisateur, les problèmes dus à l'interconnexion devraient lui être transparents, c'est à dire que deux réseaux interconnectés devraient apparaître globalement à l'utilisateur comme un réseau unique. Ceci signifie, du point de vue fonctionnel qu'il faut recouvrir les deux réseaux par une couche additionnelle qui assure, entre autres, les fonctions d'adressage global et de relai des facilités de service (figure 14).

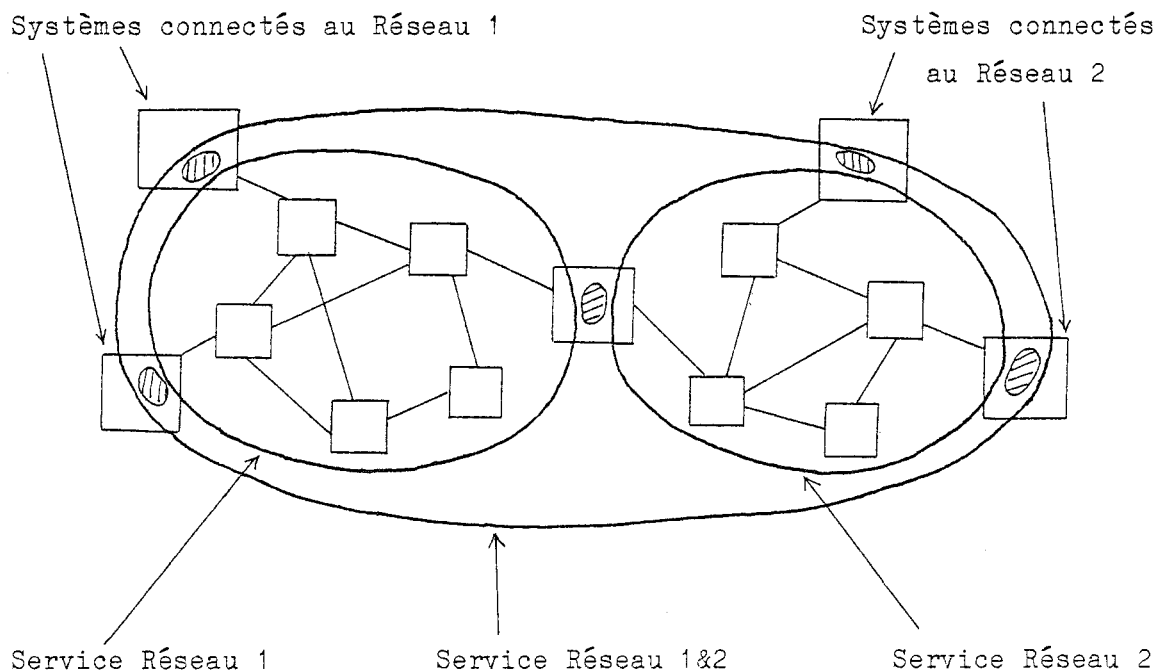


Figure 14 : Interconnexion de deux réseaux

L'ensemble des entités hachurées représente l'activité de la couche additionnelle.

Nous allons étudier dans ce qui suit les différents rôles qu'une couche doit assumer, respectivement dans les deux configurations de base.

## 6.2 - Emballage d'un seul service

Une couche est constituée, en raison de son caractère réparti, d'un certain nombre d'entités fonctionnelles situées sur des systèmes distincts. Ces entités sont en liaison d'une part avec des points d'accès du service englobant pour servir des utilisateurs qui se trouvent à l'extérieur du service, et d'autre part avec des points d'accès du service englobé, grâce auxquels elles peuvent accéder aux facilités fournies par ce dernier (figure 15). Nous les appelons **entités serveurs (n)** de la couche (n).

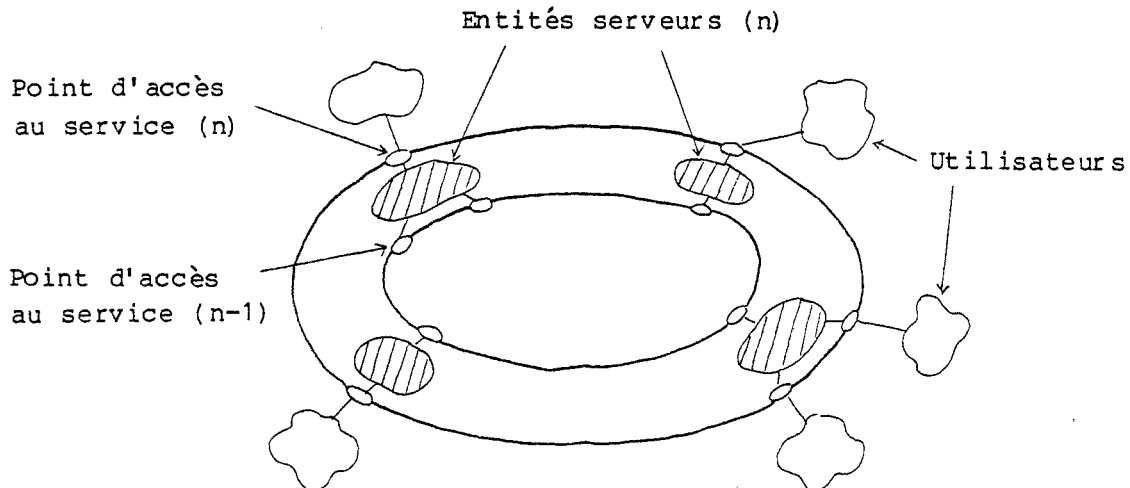


Figure 15 : Entités serveurs (n)

L'activité de ces entités est plus ou moins importante selon les fonctions qu'on veut mettre dans la couche. Il existe des fonctions qui sont de simples adaptations d'interface locales à un système donné, mais aussi des fonctions qui nécessitent une collaboration étroite entre les entités (n).

Nous les classons selon les rôles suivants :

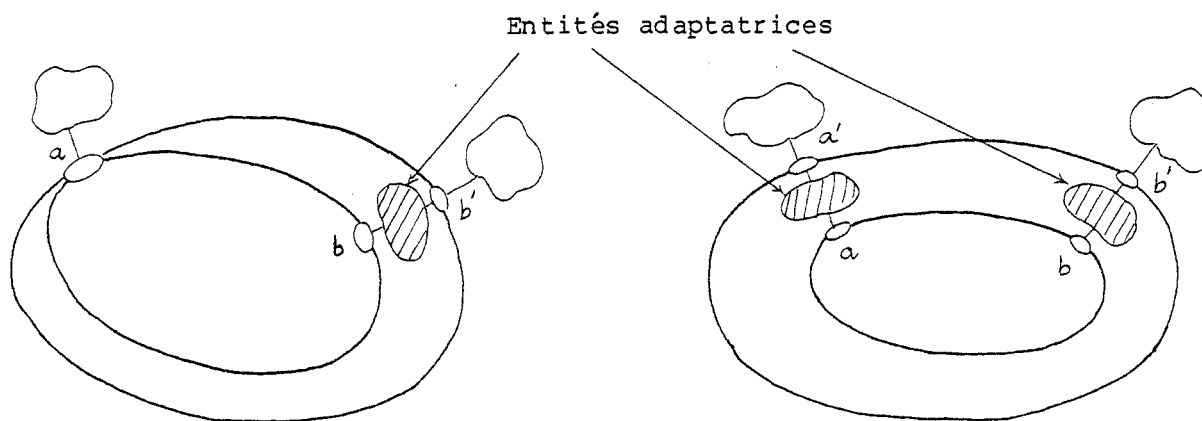
- Modification de l'interface d'utilisation du service
- Mise en correspondance de deux domaines d'adressage
- Mise en correspondance des supports de communication
- Mise en place d'une fonction de communication

qui seront présentés successivement dans ce qui suit.

### 6.2.1 - Modification de l'interface d'utilisation du service

Parfois, pour simplifier l'interface d'utilisation d'un service donné ou pour restreindre le nombre de facilités à fournir aux utilisateurs ou encore pour faciliter la conversion des interfaces de deux services à relayer (voir plus loin au 6.3.2), on est amené à modifier localement cette interface. Il s'agit de redéfinir des règles d'utilisation du service et le déroulement de ces facilités. Pour cela, on enveloppe le service par une couche additionnelle qui réalise l'adaptation nécessaire. Une telle couche présente les caractéristiques suivantes :

- L'adaptation d'interface est une fonction locale relative à un point d'accès. Par conséquent, il n'existe aucun dialogue entre les entités de la couche. Les figures 16.a et 16.b donnent les deux configurations typiques de l'adaptation.

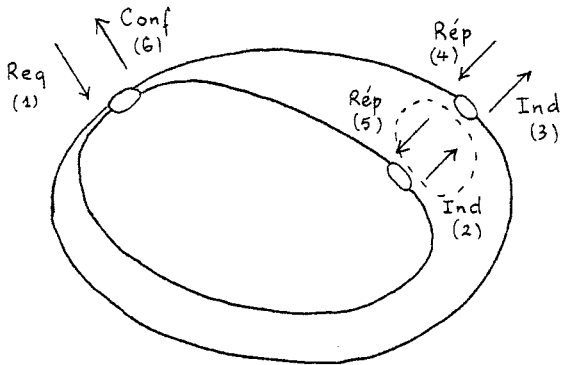


(a): une adaptation locale

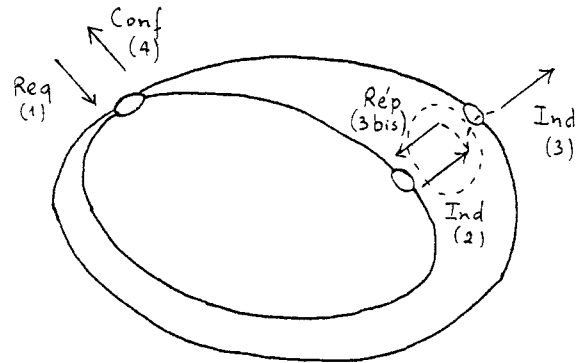
(b): deux adaptations indépendantes

Figure 16 : Adaptation d'interface

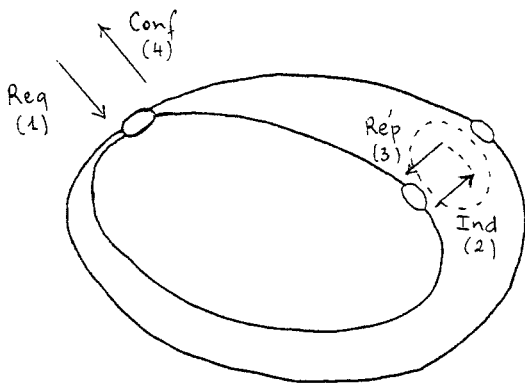
- Parmi les facilités du service initial, on trouvera des facilités dont les interactions sont reportées intégralement au nouveau service et des facilités dont le déroulement est partiellement ou complètement pris en charge par les entités adaptatrices. Les figures 17.a, 17.b, 17.c montrent quelques possibilités d'enchaînement des interactions à travers la couche d'adaptation.



(a): interactions inchangées



(b): simplification des interactions



(c): facilité disparaissant dans l'entité adaptatrice

Notation :

Req : requête

Ind : indication

Rép : réponse

Conf : confirmation

Les numéros entre parenthèses donnent l'ordre des primitives dans le temps

Figure 17 : Enchaînement des interactions à travers la couche d'adaptation

- En général, on identifie le domaine d'adressage du nouveau service à celui du service initial. Dans la figure 16.a par exemple, on a  $b=b'$  et dans la figure 16.b,  $a=a'$ ,  $b=b'$ .

- En ce qui concerne un service orienté connexion, la couche d'adaptation d'interface ne modifie pas les caractéristiques des connexions (fiabilité, fonctions rattachées à la connexion) et n'ajoute pas de niveau de connexion supplémentaire.

Exemple : PAD (Packet Assembly Disassembly) /B10/

Le PAD est une adaptation intégrée dans un noeud de raccordement du réseau de commutation de paquets TRANSPAC pour permettre à un terminal en mode caractère (ETTD-C : équipement terminal de transmission de données - caractère) d'y accéder et de se mettre en rapport avec une application située sur un matériel raccordé au réseau en mode paquet (ETTD-P : équipement terminal de transmission de données - paquet).

Voici deux exemples de fonctionnement du PAD :

- Etablissement d'un circuit virtuel (CV) par un ETTD-C : l'utilisateur désirant établir une communication avec un ETTD-P à partir d'un ETTD-C doit envoyer au préalable une requête d'établissement d'un CV de son terminal et attendre en conséquence une confirmation (figure 18).

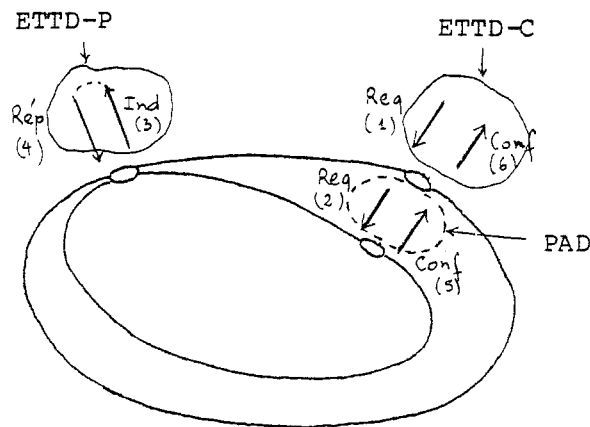


Figure 18 : Etablissement d'un CV par un ETTD-C

Dans le cas du PAD de TRANSPAC, nous avons :

. Req (1) : requête d'établissement d'un CV issue de l'ETTD-C et destinée au PAD, qui se présente physiquement sous forme d'une chaîne de chiffres donnant l'adresse de l'ETTD-P

. Req (2), Ind (3), Rép (4), Conf (5) : requête, indication, réponse, confirmation d'établissement d'un CV

. Conf (6) : confirmation d'établissement du CV renvoyée à l'ETTD-C. Physiquement, c'est une chaîne de caractères :

- COM si le résultat est positif
- LIB suivi de la raison, si le résultat est négatif

- Etablissement d'un circuit virtuel par un ETTD-P : dans ce cas le PAD prend en charge la réponse à la requête de l'ETTD-P en fonction de l'état courant de raccordement et d'interaction avec l'ETTD-C.

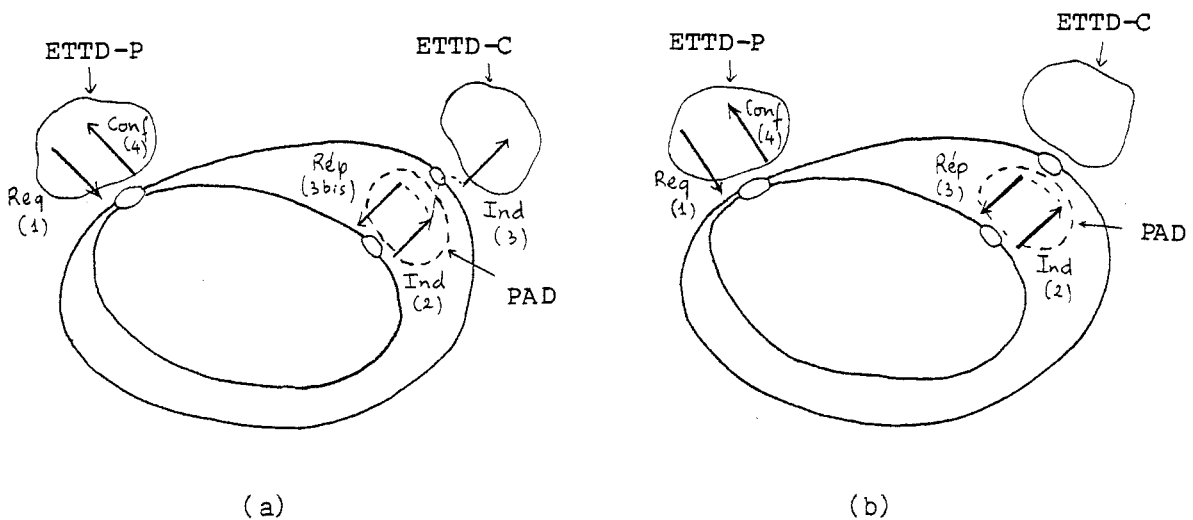


Figure 19 : Etablissement d'un CV par un ETTD-P

La figure 19.a montre le cas de l'établissement d'un CV par un ETTD-P avec succès. L'ETTD-C en est notifié par l'indication (3). Dans TRANSPAC c'est l'indication COM suivie de l'adresse de l'ETTD-P.

Dans le cas où le PAD se trouve dans l'impossibilité d'assurer la liaison avec l'ETTD-C, ou s'il y a déjà un CV en cours

d'établissement, il répondra négativement à la requête de l'ETTD-P; l'ETTD-C n'est pas averti (figure 19.b).

### 6.2.2 - Mise en correspondance de deux domaines d'adressage

Il est impossible pour un service d'adresser des entités indépendantes sans avoir au moins un nombre égal de points d'accès. C'est le cas d'un abonné d'un réseau public X25 : chaque équipement terminal de transmission de données (ETTD) relié au réseau par une ligne d'accès est identifié par une adresse unique (adresse de l'ETTD). Si l'on veut mettre en place plusieurs applications indépendantes dans cet ETTD, il est nécessaire qu'on puisse les identifier séparément. Une solution est d'adjoindre une nouvelle couche au service initial fournissant autant de points d'accès que d'applications. Cette couche assure alors le multiplexage des points d'accès (n) sur ceux du service (n-1) (figure 20).

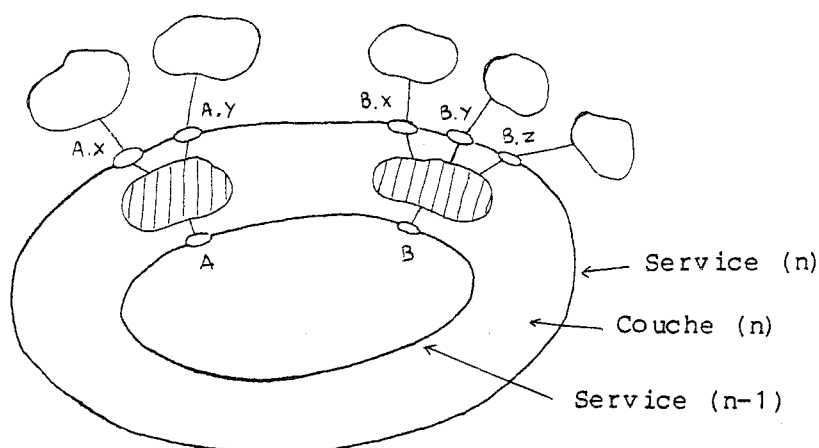


Figure 20 : Multiplexage des points d'accès

L'exemple présenté n'est qu'un cas de figure parmi plusieurs d'un problème plus général concernant l'adressage dans une couche (n): la mise en correspondance de deux domaines d'adressage de deux niveaux consécutifs. Il en existe plusieurs configurations comme le montre la figure 21 :



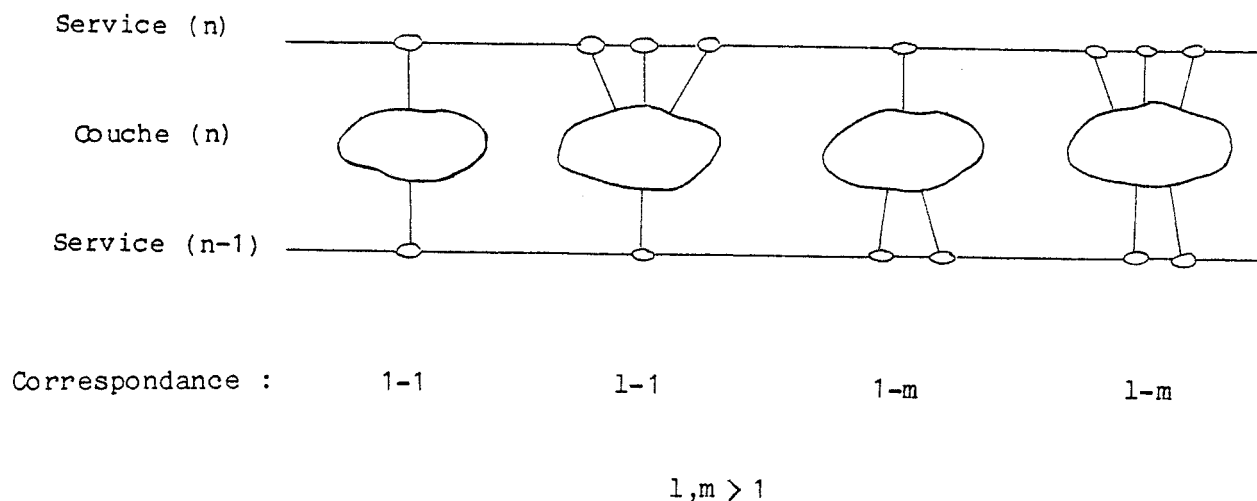


Figure 21 : Correspondance de deux domaines d'adressage

Chaque entité (n) doit avoir la connaissance des deux domaines d'adressage (n) et (n-1) et des règles qui permettent d'en faire la correspondance. C'est pourquoi, il faut définir au préalable des **conventions de correspondance** au sein de la couche (n). Des techniques de correspondance basées sur l' **adressage hiérarchisé** ou l' **adressage par tables** sont expliquées dans /B1/.

La figure 20 donne un exemple d'un adressage hiérarchisé : une adresse (n) est formée de la concaténation d'une adresse (n-1) avec une adresse définie localement dans le niveau (n).

### 6.2.3 - Mise en correspondance des supports de communication

Un support de communication sous-entend une connexion pour un service orienté connexion. Pour un service de type sans connexion, la communication s'effectue sans support particulier.

Il est possible de bâtir un service sans connexion à partir d'un service orienté connexion et réciproquement par l'adjonction d'une nouvelle couche (figures 22,23).

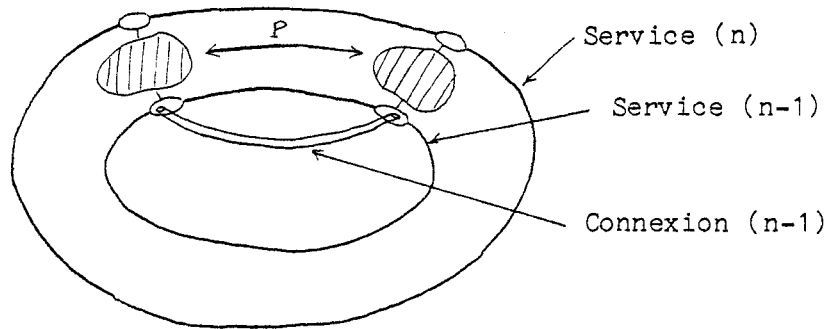


Figure 22 : Couche (n) réalisant un service de type sans connexion. Service (n-1) orienté connexion.

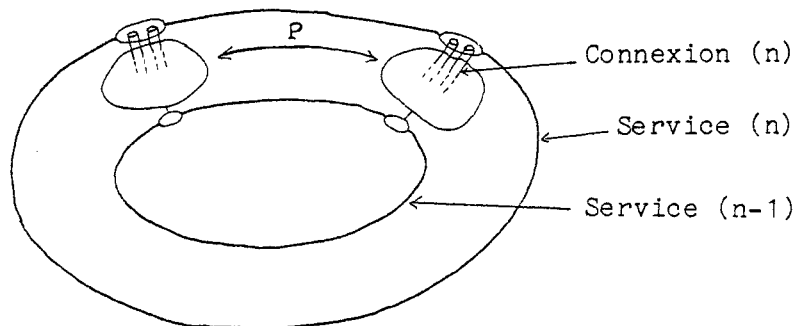


Figure 23 : Couche (n) réalisant un service orienté connexion Service (n-1) de type sans connexion.

Pour que les entités de la nouvelle couche puissent collaborer à la réalisation de nouvelles fonctions à caractère réparti (établissement, libération d'une connexion, transfert de données ...), il faut qu'elles se mettent d'accord sur un certain nombre de conventions définissant un langage commun et des règles de dialogue. Ces conventions forment un **Protocole** .

Exemple : Réalisation d'un service datagramme à partir d'un service de circuits virtuels et inversement : on met en place au dessus du service initial un protocole datagramme ou un protocole de circuits virtuels selon le cas /B11/.

Le cas le plus fréquemment rencontré est la réalisation d'un service (n) orienté connexion à partir d'un service (n-1), lui-aussi orienté connexion. Plusieurs configurations de correspondance des connexions sont possibles :

a/ Bijection : une connexion (n) est associée à une connexion (n-1) et inversement (figure 24).

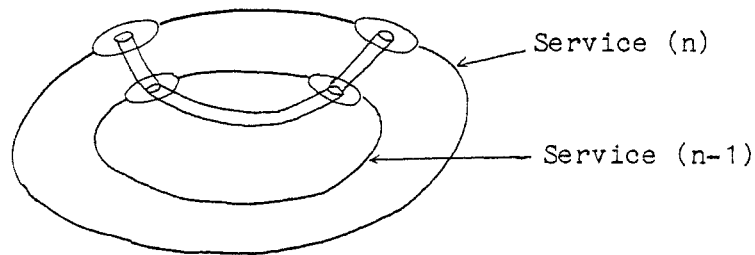


Figure 24 : Bijection de connexions

b/ Multiplexage : plusieurs connexions (n) sont multiplexées sur une connexion du service (n-1) (figure 25). Cette configuration permet d'optimiser l'utilisation des connexions (n-1), c'est le cas du multiplexage des connexions de Transport sur des circuits virtuels (ce sont des connexions des couches 4 et 3 d'OSI).

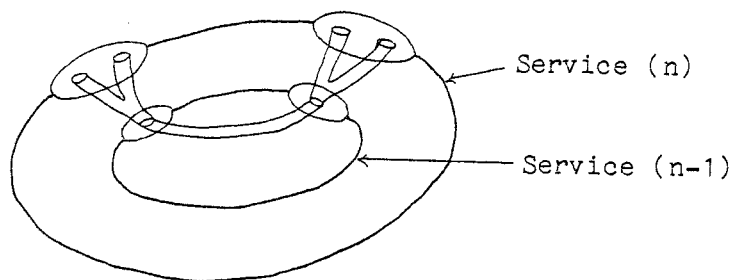


Figure 25 : Multiplexage de connexions

c/ Eclatement : plusieurs connexions (n-1) participent à la réalisation d'une connexion (n) (figure 26). Cette configuration permet d'accroître la disponibilité du service et éventuellement la vitesse de transfert des données.

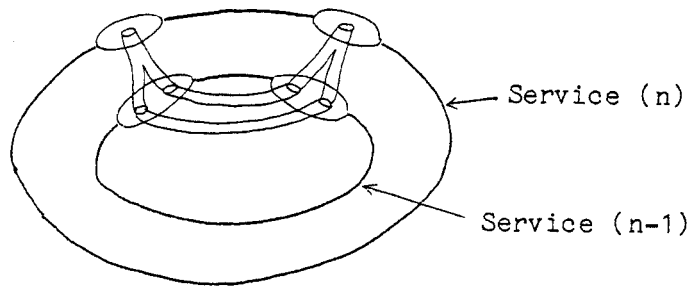


Figure 26 : Eclatement de connexions

Exemple : La liaison multiligne dans le réseau TRANSPAC. L'ETTD est relié à un noeud du réseau par plusieurs lignes physiques. Ce qui permet d'avoir une disponibilité plus élevée (une panne survenue sur une ligne physique n'arrête pas le transfert des données qui continue sur les autres lignes) et une vitesse de transfert plus grande /B10/.

On peut aussi imaginer une combinaison du multiplexage et de l'éclatement comme l'exemple de la figure 27 :

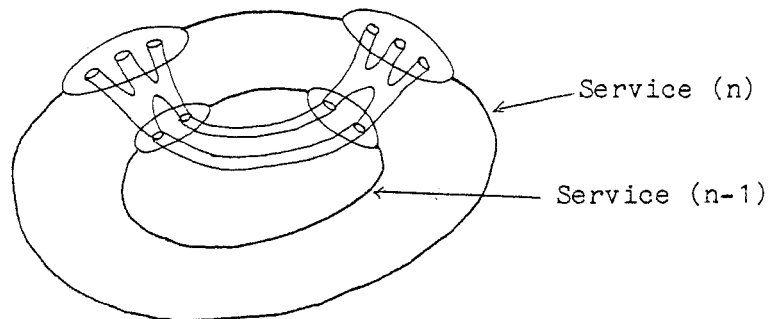


Figure 27 : Multiplexage et éclatement de connexions

La gestion de la correspondance des connexions (n) sur les connexions (n-1) est assurée par la couche (n). Il y a plusieurs façons de la réaliser :

- Soit définir des conventions d'établissement statique de la correspondance. Dans ce cas les entités (n) adoptent toujours une configuration de correspondance donnée avec des paramètres figés

(degré de multiplexage, etc...) lors de la création d'une connexion (n-1)

- Soit définir un protocole d'établissement de la correspondance. Dans ce cas les entités (n) peuvent négocier dynamiquement une configuration particulière de correspondance.

#### 6.2.4 - Mise en place d'une fonction de communication

Comme nous l'avons mentionné en 3. et en 6.1, la décomposition fonctionnelle d'un service de communication par la technique d'abstractions successives permet d'organiser les fonctions de communication qu'il faut réaliser. On classe par exemple de haut en bas les fonctions suivantes : interprétation des données, gestion du dialogue et de la synchronisation, transport de données.

En appliquant la même technique à la fonction de transport de données, cette dernière sera décomposée en fonctions de communication plus élémentaires à répartir sur des niveaux différents : transfert au niveau message, transfert au niveau bit, avec les fonctions de contrôle associées : contrôle de flux, contrôle d'erreur.

Concrètement toutes ces fonctions dites de communication sont installées dans des couches successives. La réalisation d'une ou plusieurs fonctions de communication dans une couche nécessite la définition d'un **protocole** car ces fonctions sont toutes à caractère réparti.

Ainsi, citons par exemple pour les différentes couches du modèle d'OSI, le protocole d'appareils virtuels (couche Présentation), le protocole de gestion du dialogue et de synchronisation (couche Session), le protocole de transport (couche Transport), etc...

En toute généralité, une couche donnée peut avoir en même temps tous ces rôles qui viennent d'être énoncés. Ceci suggère l'introduction de la notion de sous-niveaux fonctionnels à l'intérieur d'une couche. Nous aurons l'occasion d'en parler dans le chapitre IV.

### 6.3 - Emballage de plusieurs services relayés

Par rapport au cas précédent (emballage d'un seul service) cette configuration introduit la notion de **relai de service** comme on l'a vu dans l'exemple du 6.1.b.

Du point de vue fonctionnel, la mise en place du relai de service se traduit par l'adjonction d'une couche enveloppant tous les services initiaux et assurant la transmission des facilités de communication de service en service, de telle sorte que le service global rendu à tous les utilisateurs soit partout identique. Le relai de service doit être transparent aux utilisateurs.

Une telle couche comporte plusieurs types d'entités fonctionnelles (figure 28) :

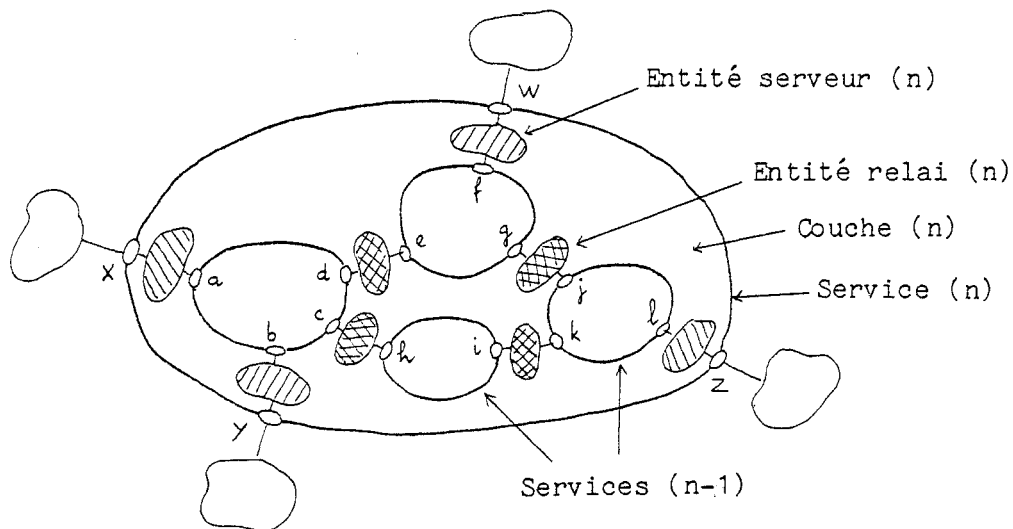


Figure 28 : Emballage de services

- les entités (n) qui interagissent avec des utilisateurs à travers des points d'accès (n) jouent le rôle de **serveur** ,
- les entités (n) dont l'activité n'est que d'interconnecter les services (n-1) jouent le rôle de **relai de service** .

Il existe également des entités (n) qui ont en même temps les deux rôles précédents, mais nous les ignorons pour simplifier la présentation.

Nous verrons successivement les problèmes liés à la mise en oeuvre de telles couches : routage, conversion des interfaces et conversion des protocoles.

### 6.3.1 - Adressage - Routage

De façon générale, chaque entité (n) connaît le domaine d'adressage (n) du service global et tous les domaines d'adressage (n-1) des services (n-1) afin de pouvoir mettre en liaison deux points d'accès (n) quelconques. Pour cela, elle met en oeuvre une fonction appelée **routage** qui est en fait une généralisation de la fonction de mise en correspondance des domaines d'adressage présentée en 6.2.2 et qui consiste à traduire un couple d'adresses (n) associées à 2 points d'accès donnés en une série d'adresses (n-1) traçant un chemin allant d'un point d'accès (n) à l'autre. La figure 28 nous donne des exemples de chemins suivants :

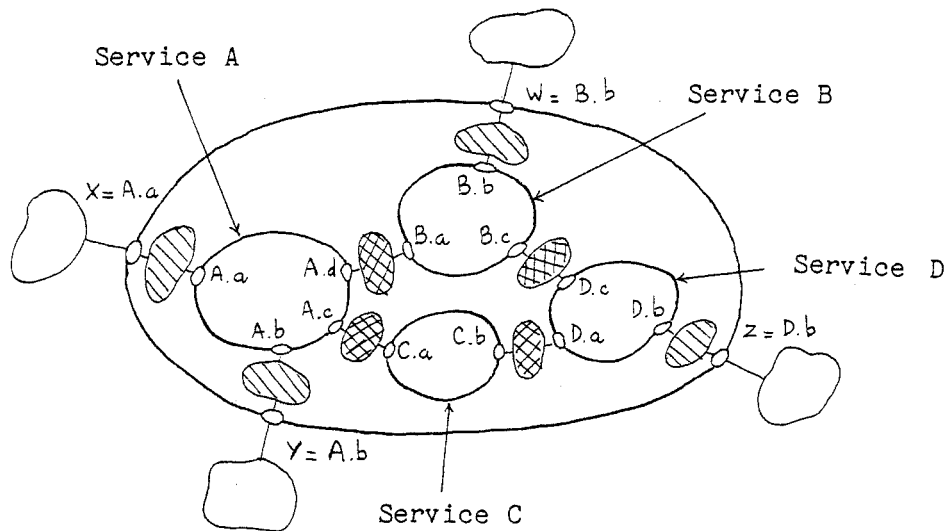
$(X, Y) \rightarrow (a, b)$

$(X, W) \rightarrow (a, d, e, f)$

$(X, Z) \rightarrow (a, d, e, g, j, l) \text{ ou } (a, c, h, i, k, l)$

Dans la pratique, il suffit à chaque entité (n) de la couche d'effectuer un **routage partiel**, c'est à dire de déterminer seulement la prochaine adresse (n-1) à atteindre. Le message à acheminer doit porter dans ce cas l'adresse globale du destinataire puisque celle-ci est décodée de proche en proche au sein des entités (n).

Généralement, les domaines d'adressage (n-1) sont indépendants les uns des autres. C'est pourquoi il peut y avoir des **conflits d'adresses** lors de l'interconnexion : des points d'accès (n-1) situés sur des services (n-1) différents peuvent porter un même nom. Pour permettre aux entités (n), et tout particulièrement aux entités relai de les distinguer on peut utiliser par exemple la technique d'adressage hiérarchisé suivante : à l'adresse (n-1) d'un point d'accès on ajoute l'identificateur du service (n-1) auquel il appartient (figure 29).



Chaque service (n-1) a un nom : A,B,C,D.

"A.a" est le point d'accès "a" du service "A"

Figure 29 : Adressage hiérarchisé

Exemple : - CATENET /B12/  
- Recommandation X121 (plan de numérotation internationale des réseaux publics) /B13/

Ce sont des propositions basées sur l'adressage hiérarchisé pour l'interconnexion des réseaux publics. Une adresse globale vis à vis de l'ensemble des réseaux est composée de deux parties :

- l'adresse d'un réseau,
- l'adresse du point d'accès local à ce réseau du destinataire.

La résolution d'une adresse globale s'opère dans un premier temps sur la partie d'adresse de réseau pour fournir un chemin conduisant jusqu'à un point d'entrée de ce réseau, ensuite le traitement de la partie d'adresse locale permet d'atteindre le destinataire final.

Nous remarquons que pour un transfert de données, il peut exister plusieurs chemins pour parvenir au destinataire. Dans la



littérature on trouve plusieurs politiques et de nombreux algorithmes de routage : méthode de l'inondation, du routage fixe, méthodes adaptatives /B14/.

### 6.3.2 - Couche comportant la conversion des interfaces

Il s'agit d'une première solution pour relayer les **facilités de communication** de différents services interconnectés. La couche à ajouter comporte en plus de la fonction de routage, des fonctions de conversion des interfaces qui se trouvent précisément dans des entités relai (figure 30).

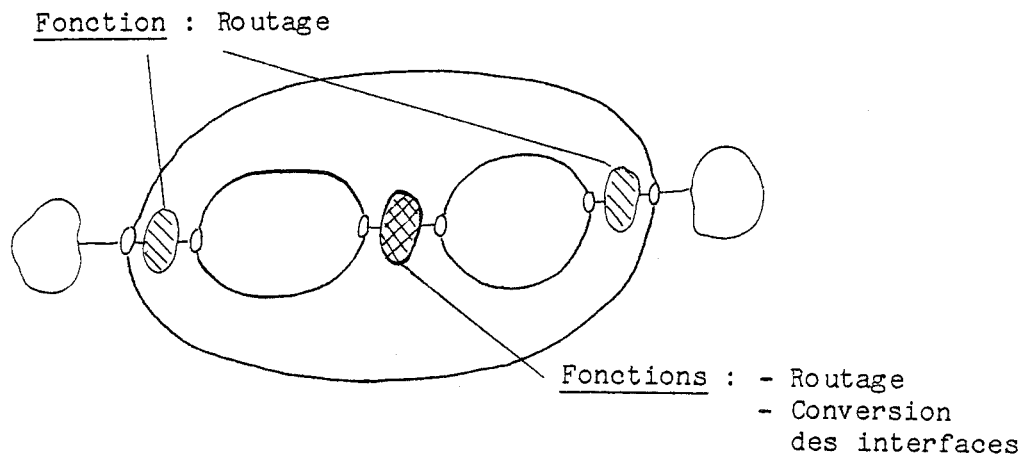


Figure 30 : Couche comportant la conversion des interfaces.

Pour simplifier la discussion, nous considérons le cas de deux services à relayer.

La conversion des interfaces suppose que les deux services à relayer fournissent les mêmes facilités de service. Dans ce cas, les deux services sont dits **équivalents** .

De plus, ces services doivent être **relayables** (\*) /B11/. Cette condition signifie que toutes les facilités d'un service, y comprises celles qui ne sont pas accessibles aux utilisateurs (par

---

\* Dans /B11/ le terme anglais utilisé est "cascadable".

exemple l'indication "service en dérangement" ou les diagnostics du réseau qui sont issus de traitements internes au service) sont **transmissibles** à l'autre et vice-versa. Elle permet de préserver la cohérence des facilités à fournir au niveau du service global.

Si ces deux conditions sont satisfaites, la mise en série de deux services (n-1) pourra se faire par une simple conversion, au sein de l'entité relai, des primitives d'interaction aux deux interfaces. Les facilités des services (n-1) sont alors reportées au niveau du service global (n) sans être modifiées. Leur qualité sera celle des services (n-1) si ceux-ci fournissent des facilités de même qualité. Dans le cas où une facilité n'a pas la même qualité sur les deux services (n-1) (par exemple, si pour la facilité de transfert de données les deux services assurent des contrôles d'erreur et de flux différents), la qualité globale n'est équivalente qu'à la moins bonne de celles des deux services (n-1).

Si au contraire les deux services (n-1) ne sont pas équivalents ou relayables, il y a deux solutions pour réaliser leur mise en série :

- soit choisir de ne relayer que les deux sous-ensembles des facilités des services (n-1), pour lesquels les deux conditions précédentes sont vérifiées (restreindre les facilités au niveau du service global)

- soit ajouter une nouvelle couche à un seul ou aux deux services (n-1) pour les rendre équivalents et relayables.

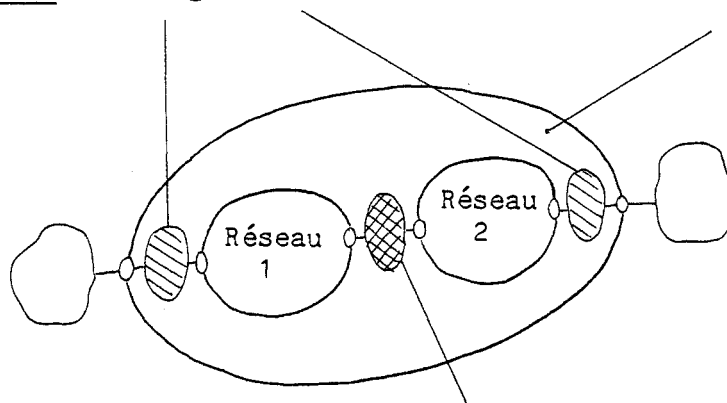
Illustrons ces propos par quelques cas pratiques :

Exemple 1 : Interconnexion de deux réseaux publics de commutation de paquets, offrant les mêmes facilités et les mêmes interfaces d'utilisation (cas rare dans la réalité) (figure 31)

L'entité relai, appelée ici **passerelle** (gateway) assure les fonctions de routage inter-réseau et de conversion des interfaces d'accès aux deux réseaux.

Fonction : Routage inter-réseau

Couche d'interconnexion de réseaux



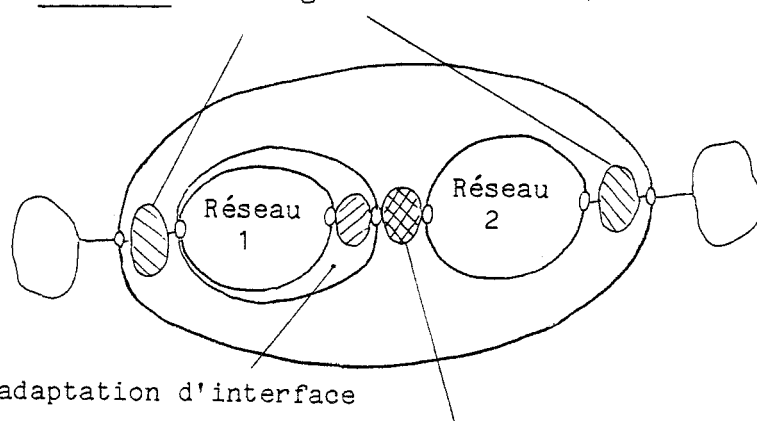
Fonctions : - Routage inter-réseau  
- Conversion des interfaces

Figure 31

Bien entendu, si l'un des deux réseaux a un meilleur contrôle d'erreur sur le transfert de données que l'autre (par exemple son protocole interne prévoit un nombre plus important de réémissions de paquets en cas d'erreur) le contrôle d'erreur résultant de la mise en série des deux réseaux est alors équivalent au moins bon des deux.

Exemple 2 : Interconnexion de 2 réseaux offrant les mêmes facilités mais pas les mêmes interfaces d'utilisation - Introduction d'une couche d'adaptation d'interface (figure 32)

Fonction : Routage inter-réseau



Fonctions : - Routage inter-réseau  
- Conversion des interfaces

Figure 32

La couche d'adaptation d'interface rend les services Réseau 1 et Réseau 2 équivalents et relayables.

Concrètement, la mise en oeuvre d'une telle interconnexion nécessite l'introduction d'une passerelle reliant les deux réseaux, dans laquelle sont implantées les fonctions de routage, d'adaptation et de conversion des interfaces. C'est la solution adoptée initialement pour l'interconnexion des réseaux DATAPAC et TELENET et aussi des réseaux TYMNET et DATAPAC /B15/.

L'avantage de cette méthode est qu'un minimum de modifications est à apporter aux différents points d'accès des deux réseaux, la passerelle se chargeant du reste des fonctions d'interconnexion, ce qui représente un moindre travail pour la liaison des deux réseaux. Cette méthode a quand même un inconvénient : dès qu'on augmente le nombre de réseaux à interconnecter, il y a une prolifération des adaptations à réaliser. Le travail devient très rapidement inextricable.

Exemple 3 : Interconnexion de deux réseaux offrant les mêmes facilités de service mais pas les mêmes interfaces d'utilisation - Définition d'une interface commune - Introduction de deux couches d'adaptation d'interface - Recommandation X75 /B16/.

Fonction : Routage inter-réseau

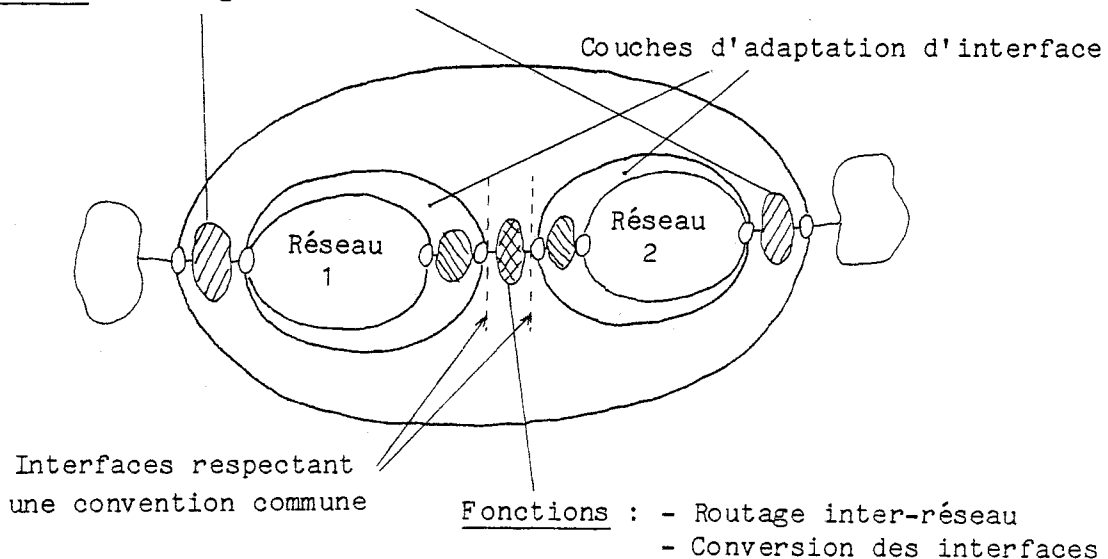


Figure 33

Il s'agit ici de définir une interface commune, un concept déjà classique en architecture matérielle des machines (bus commun), que tous les réseaux doivent respecter afin de pouvoir s'interconnecter. S'ils n'ont pas cette interface, il faut ajouter une couche d'adaptation sur chacun d'eux (figure 33).

La recommandation X75 pour l'interconnexion des réseaux publics est basée sur ce principe /B16/. L'adoption de cette recommandation pour un réseau public quelconque implique :

- une intégration du plan d'adressage hiérarchisé X121 dans tous les noeuds de commutation du réseau, appelés DSE (Data Switching Exchange)

- une extension d'un noeud particulier appelé STE (Signalling Terminal) par la mise en place d'un protocole X75 et de son interface avec X25. Cette partie constitue une **demi-passerelle**. Une liaison physique entre 2 STE de 2 réseaux à interconnecter permet le fonctionnement de la passerelle complète.

- l'adoption d'un **même ensemble de facilités** fournies à l'interface de chacun des réseaux à interconnecter pour faciliter la mise en correspondance de X25 sur X75.

La figure 34 donne le schéma fonctionnel de l'interconnexion de 2 réseaux par l'interface X75 et la figure 35, le schéma topologique correspondant.

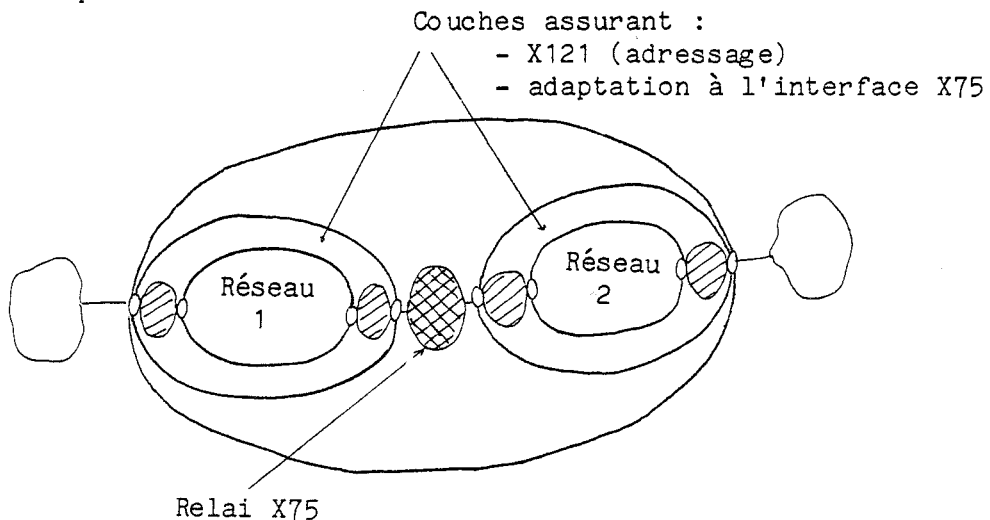
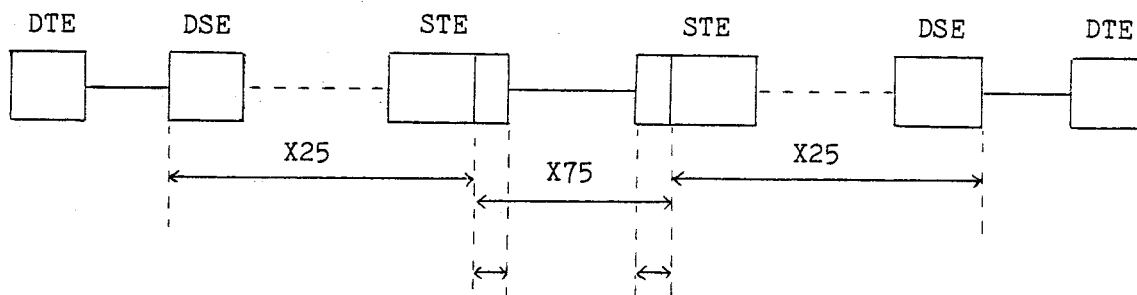


Figure 34



Demi-passerelles

DTE : Data Terminal Equipment (ETTD en français)

STE : Signalling Terminal

DSE : Data Switching Exchange

Figure 35

La solution d'utiliser une interface commune permet une interconnexion systématique des réseaux sans investissement supplémentaire, mais au prix d'un développement initial assez important (un STE) et de l'obligation de limiter les facilités d'un réseau X25 particulier à celles de l'interface X75 /B15/.

Exemple 4 : Interconnexion de deux réseaux n'offrant pas les mêmes facilités de service : un réseau en mode datagramme et un réseau en mode circuit virtuel (CV) /B11/

Dans ce cas il faut choisir un niveau d'équivalence de service pour l'interconnexion, qui peut être soit le service CV, soit le service datagramme.

- Si le niveau choisi est le service datagramme, on ajoute une couche supplémentaire au réseau en mode CV pour le rendre équivalent au réseau en mode datagramme avant la mise en place de la conversion des interfaces (figure 36).

- Si au contraire le niveau choisi est le service CV, on ajoute une couche au réseau en mode datagramme pour le rendre équivalent au réseau en mode CV (figure 37).

(Voir à ce sujet les figures 22 et 23 de ce chapitre)

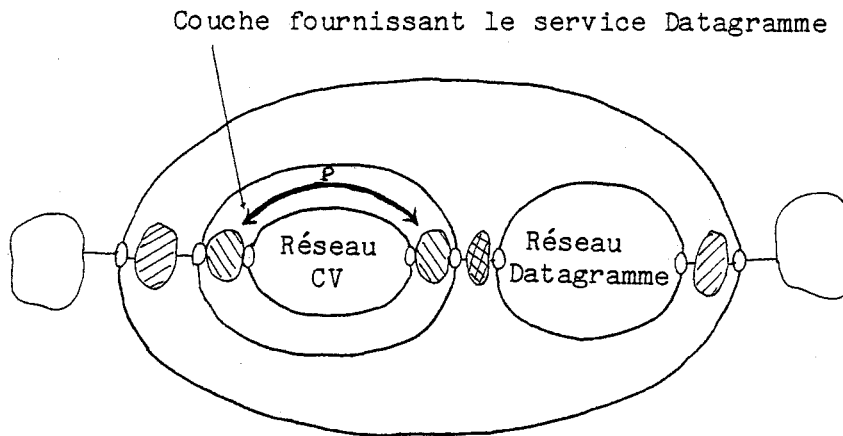


Figure 36

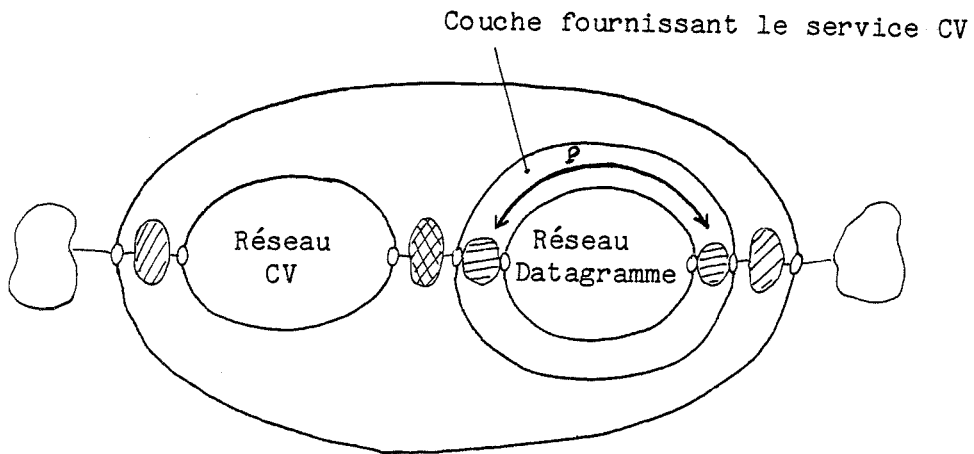


Figure 37

### 6.3.3 - Couche comportant une série de protocoles

Outre l'interconnexion des services par la conversion des interfaces, une autre façon de faire consiste à introduire au-dessus des services à interconnecter une série de protocoles. On ajoute alors un niveau supplémentaire de contrôle de communication (figure 38).

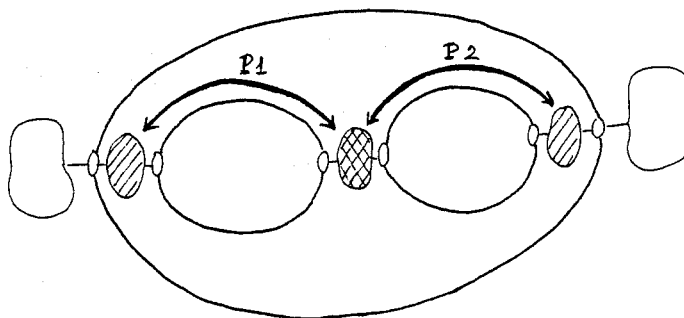


Figure 38

Cette couche possède les caractéristiques suivantes :

- Elle comporte, évidemment, un niveau de fonctions d'adressage inter-service (routage)

- Du fait qu'il y a un niveau de protocole au dessus des interfaces, les services à interconnecter ne sont pas contraints d'être rigoureusement équivalents.

- Les entités relai doivent assurer une fonction de conversion des protocoles pour enchaîner la transmission des informations. En général les protocoles à relayer sont identiques ou au moins équivalents. Lorsqu'ils ne le sont pas, la conversion se restreint à un sous-ensemble relayable de ces protocoles.

- Les protocoles permettent la mise en place d'une ou plusieurs fonctions de contrôle de la communication au dessus des services à interconnecter. Par conséquent, la qualité de la communication de **bout en bout** est entièrement définie et contrôlable, alors que dans le cas d'une conversion des interfaces, la qualité du service global est déduite de celles des services interconnectés.

\* \* \*

\*



Nous venons de passer en revue, dans les grandes lignes, les principaux types de fonctions réalisées par une couche, leurs raisons d'être, les problèmes de leur mise en oeuvre et les solutions correspondantes.

Dans le chapitre suivant, en nous limitant au cas d'une couche enveloppant un seul service, nous allons décrire les fonctions internes d'une entité serveur (n).

## CHAPITRE II

### FONCTIONS D'UNE COUCHE (n).

- 1 - PRELIMINAIRE
- 2 - DIFFERENTES UNITES DE DONNEES
- 3 - FONCTIONS A L'INTERFACE
- 4 - GESTION DES FONCTIONS DE COMMUNICATION
  - 4.1 - Protocole
  - 4.2 - Sous-protocole
  - 4.3 - Groupage - Dégroupage
  - 4.4 - Fragmentation - Réassemblage
- 5 - ADRESSAGE DANS UNE COUCHE (n)
- 6 - MISE EN CORRESPONDANCE DES CONNEXIONS
  - 6.1 - Multiplexage
  - 6.2 - Eclatement
- 7 - UTILISATION DU SERVICE (n-1)



## 1 - PRELIMINAIRE

Le chapitre précédent a mis en évidence les principales fonctions que doit assurer une couche. Nous allons détailler dans ce chapitre les traitements que subissent les informations lorsqu'elles la traversent.

Certaines fonctions (telles que multiplexage, éclatement,...) qui sont seulement mentionnées dans le modèle de référence d'OSI sont explicitées dans ce chapitre.

Nous excluons de notre étude l'aspect "administration de la couche". Le lecteur pourra trouver quelques idées préliminaires à ce sujet dans /B1/ et /C1/. En outre, la présentation se limite au fonctionnement d'une entité serveur (n). L'extension à une entité relai nécessiterait des fonctions supplémentaires de conversion des interfaces ou celles de conversion des protocoles.

Notons finalement que pour un spécialiste réseaux qui est déjà familier des concepts d'OSI et des réalisations de couches, il y a un certain nombre de concepts connus rappelés dans ce chapitre. Nous avons essayé d'y ajouter quelques précisions sur la mise en oeuvre de certaines fonctions, issues de notre expérience de réalisation en vraie grandeur des couches Transport et Session.

## 2 - DIFFERENTES UNITES DE DONNEES

Avant de détailler les différentes fonctions d'une couche, il est important de soulever le problème de la sémantique des informations qui circulent dans cette couche. Le terme "message" est souvent utilisé pour désigner une chaîne de caractères transférée d'un système à un autre ou d'un processus à un autre. Dans le contexte du modèle en couches où l'on distingue les interactions entre couches adjacentes

(interactions locales dans un système) des échanges d'informations entre deux entités homologues situées sur 2 systèmes distincts il est nécessaire de pouvoir donner aux messages des significations différentes. Un message est appelé dans le contexte d'OSI "unité de données" et pour une couche (n) donnée, il existe plusieurs types d'unités de données :

- Unités de données du service (n) (SDU (n) : Service Data Unit)

Ce sont des données fournies par une entité (n+1) émettrice et transférées de façon transparente par le service (n) à l'entité (n+1) réceptrice. Les SDU (n) sont envoyées sur les connexions (n) : elles forment des flots de données sur ces connexions. En toute généralité, on peut avoir, pour une connexion (n) donnée, des flots de données différents (par exemple un flot de données normales et un flot de données express).

- Unités de données de l'interface(n) (IDU(n):Interface Data Unit)

Pour respecter les contraintes de l'interface (n), les données dans les SDU (n) sont transmises à cette interface dans des unités de données de l'interface (n). Celles-ci comportent, en plus des données de SDU (n), des informations de contrôle de l'interface (ICI : Interface Control Information) permettant de coordonner les interactions locales entre l'entité (n+1) et l'entité (n).

- Unités de données du protocole (n)(PDU(n) : Protocole Data Unit)

Les entités (n) homologues dialoguent entre elles selon un protocole (n) dans le but de réaliser les fonctions de communication (n). Un message du dialogue défini par le protocole est appelé une unité de donnée du protocole (PDU). Les entités (n) s'échangent des PDU pour coordonner leur travail commun. Il existe plusieurs types de PDU : ouverture d'une connexion, transfert de données, fermeture d'une connexion, etc... Des PDU de transfert de données ont pour rôle de transporter les données des SDU. Dans ce cas, une PDU (n) peut contenir plusieurs SDU (n) (fonction de groupage) ou un fragment d'une SDU (n) (fonction de fragmentation). Une PDU comprend un ou deux champs :

. Le champ des informations de contrôle du protocole (PCI : Protocol Control Information) : ce sont des informations fabriquées par une entité (n) et exploitées par son homologue pour contrôler les fonctions de communication.

. Le champ des données de service (données des utilisateurs) : ces données sont transférées de façon transparente par les entités (n) car elles ne sont significatives que pour les entités (n+1) (émettrice et réceptrice). Ce champ n'existe que dans les PDU de transfert de données.

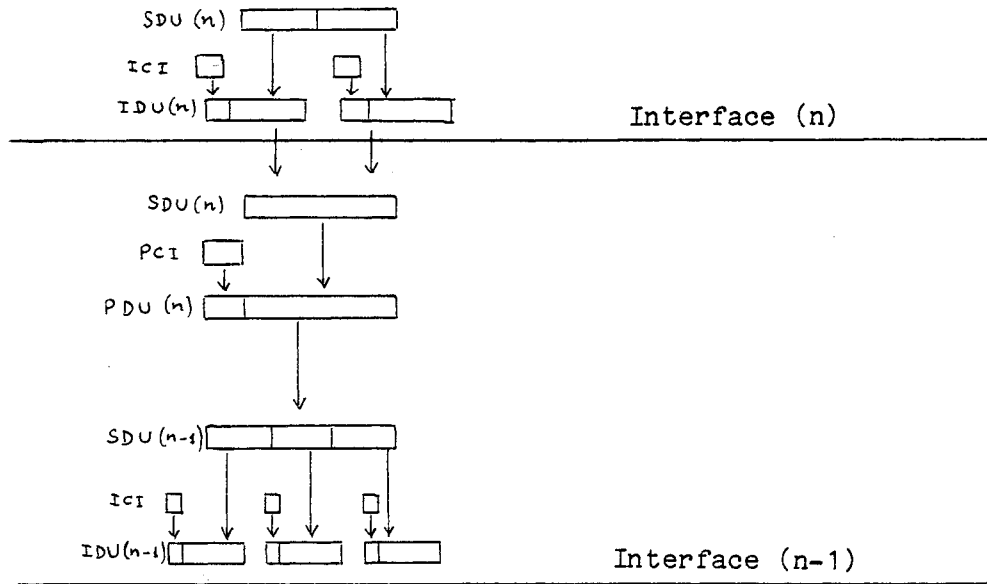


Figure 1 : Différentes unités de données

La couche (n) manipule également des SDU (n-1) et des IDU (n-1) :

- Unités de données du service (n-1) (SDU (n-1))

Les PDU émises par une entité (n) et destinées à une entité (n) homologue doivent être acheminées par le service (n-1) dans des SDU (n-1). Une SDU (n-1) peut contenir une PDU (n), un fragment de celle-ci ou plusieurs PDU (n) concaténées.

- Unités de données de l'interface (n-1) (IDU (n-1))

Elles transportent les données des SDU (n-1) au niveau de l'interface (n-1) (Interactions à l'interface (n-1) entre deux couches (n) et (n-1)).

La figure 1 donne une image de la transformation successive de ces unités de données, en partant de l'interface (n) et allant à l'interface (n-1).

Les différentes unités de données ont généralement des tailles maximales différentes et les informations à transférer sont souvent de longueurs variables. C'est pourquoi les données traversant une couche subissent des transformations de longueur telles que le **groupage**, le **dégroupage**, la **fragmentation**, le **réassemblage**, la **concaténation**, la **séparation** .... Elles seront présentées dans la suite de ce chapitre.

3 - FONCTIONS A L'INTERFACE

Il y a trois fonctions principales à l'interface. Ces fonctions se trouvent à l'interface (n) et elles se répètent à l'interface (n-1). Il est important de souligner le caractère local de ces fonctions ; elles dépendent par conséquent d'une implantation donnée.

1/ Une fonction d'émission et de réception des primitives en un point d'accès donné

2/ Une fonction de contrôle de flux qui agit sur les différents flots de données traversant l'interface pour équilibrer la charge des deux couches adjacentes. Une connexion (n) correspond généralement à un flot de données. Toutefois, on peut définir plusieurs flots de données sur une connexion (n) (flot de données normales, flots de données express). Dans ce cas, le contrôle de flux est appliqué séparément à ces flots de données. Cette fonction nécessite une collaboration entre deux entités (n) et (n+1) situées de part et

d'autre de l'interface (n). Pour cela, le champ ICI d'une IDU véhicule des informations de contrôle de flux qui sont élaborées à l'émission par une entité et exploitées à la réception par l'entité adjacente.

Le mécanisme de contrôle de flux par le **crédit** peut être utilisé ici. La valeur du crédit calculée en fonction de la charge locale en cours représente un nombre d'IDU qu'une entité accepte de recevoir de son adjacente.

3/ Une fonction d'adaptation de longueur de SDU à celle des IDU (et inversement) : une SDU est une unité de données du service, souvent de longueur variable, fournie par une entité (n+1) et destinée à une entité (n+1) distante. Or, les données doivent traverser l'interface (n) sous la forme d'IDU dont la taille maximale est fixée localement à chaque interface et pour chaque système. Lorsque la SDU est trop longue, son passage à travers l'interface nécessite la création de plusieurs IDU. Des informations de contrôle contenues dans le champ ICI des IDU permettent de reconstituer les SDU à partir des IDU reçues (figure 2).

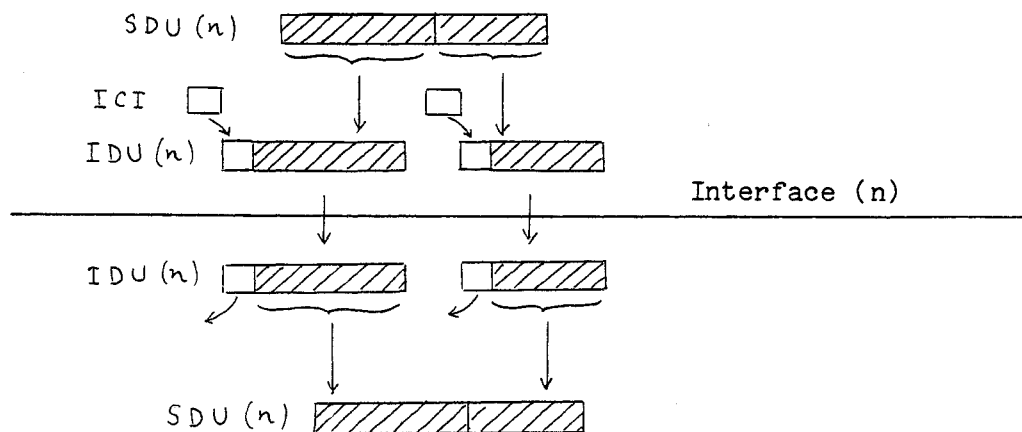


Figure 2 : SDU (n) et IDU (n)



#### 4 - GESTION DES FONCTIONS DE COMMUNICATION

Dans cette partie, nous exposons en premier lieu quelques aspects spécifiques du protocole (n) contrôlant les fonctions de communication (n), et en second lieu les fonctions de construction des PDU transportant des données de service.

##### 4.1 - Protocole

Pour réaliser des fonctions de communication, il est nécessaire de définir un protocole qui impose des règles de coopération entre deux entités (n) homologues. Les PDU sont des messages échangés d'une entité à l'autre pour permettre la mise en oeuvre du protocole. Schématiquement, nous présentons ce type d'interaction entre entités homologues par les flèches horizontales à l'intérieur de la couche (n) comme le montre la figure 3. En réalité, les PDU sont confiées au service (n-1) pour être acheminées d'un bout à l'autre. En outre, il y a des interactions entre entités adjacentes, schématisées par des flèches verticales, coordonnant localement le déroulement du protocole suivant les désirs de l'utilisateur.

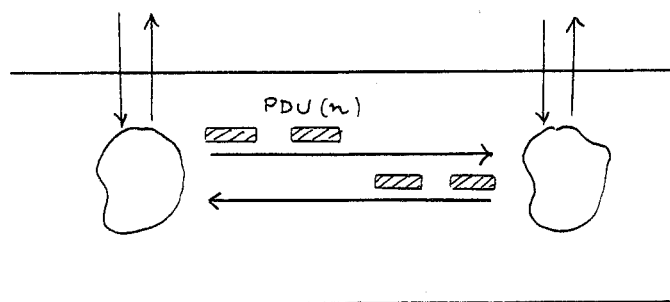


Figure 3 : Interactions relatives au protocole (n)

Le protocole est spécifique à une couche donnée et fait l'objet d'études de normalisation.

#### 4.2 - Sous protocole

En remarquant que les fonctions du protocole sont rattachées aux facilités que le service (n) propose aux utilisateurs de la couche (n+1), on peut subdiviser l'ensemble de ces fonctions en sous-ensembles, chacun formant un sous-protocole du protocole (n) et élaborant l'une des facilités.

Citons quelques exemples de sous-protocoles :

- Etablissement d'une connexion
- Libération d'une connexion
- Transfert de données sur une connexion
- Réinitialisation (Niveau Paquet de la recommandation X25 /B17/)
- Transfert de jeton (Couche Session, Protocole ECMA-75 /C3/)
- Resynchronisation (Couche Session, Protocole ECMA-75 /C3/)
- .....

Nous rappelons ci-dessous quelques aspects importants de certains de ces sous-protocoles.

##### 4.2.1 - Négociation des paramètres dans le sous-protocole d'établissement d'une connexion

Au moment de l'établissement d'une connexion au profit de deux entités (n+1), les deux entités (n) homologues doivent se mettre d'accord sur les différentes caractéristiques à attribuer à celle-ci. Par exemple :

- la classe de protocole qui est un paramètre de qualité du service. Citons le cas du protocole de Transport ECMA-72 qui possède 5 classes /C2/ :

- 0 : classe "terminal simple" (Teletex)
- 1 : classe de base

- 2 : classe "contrôle de flux"
- 3 : classe "recouvrement d'erreur"
- 4 : classe "détection et recouvrement d'erreur"

- le sous-ensemble du protocole : la couche (n) peut offrir plusieurs sous-ensembles de protocole dont chacun s'adapte à un type de besoin des utilisateurs du service (n). Par exemple, le protocole de Session ECMA-75 propose 4 sous-ensembles /C3/ :

- A : sous-ensemble de base
- B : sous-ensemble "interactif"
- C : sous-ensemble "synchronisation"
- D : sous-ensemble "TWA" (Two-Way Alternate)

- les options : utilisation d'un flot de données express,...

- les paramètres caractéristiques des PDU : longueur maximale,...

La négociation des paramètres donne lieu généralement à l'échange de "propositions" et de "contre-propositions" entre deux entités (n). Il y a un retour de "contre-proposition" lorsqu'une entité (n) n'est pas capable de satisfaire la qualité de la communication demandée par l'entité (n) homologue ou que l'utilisateur (entité (n+1)) refuse les paramètres proposés. Les paramètres dans la contre-proposition sont par conséquent de valeur moins importante par rapport à ceux proposés (longueur des PDU plus petite, classe inférieure de protocole...).

Tout ceci montre que les connexions existant dans une même couche ne fonctionnent pas forcément de la même manière.

#### 4.2.2 - Contrôle d'erreur dans le sous-protocole de transfert de données.

Les PDU échangées entre les entités (n) et acheminées par le service (n-1) peuvent :

- se perdre sur leur trajet si la fiabilité du service (n-1) n'est pas parfaite

- arriver en désordre si les PDU empruntent des chemins différents à l'intérieur du service (n-1)

- arriver avec plusieurs exemplaires de PDU identiques si l'entité (n) les a dupliquées à l'envoi (cas de l'éclatement sur plusieurs connexions (n-1), voir 6.2) ou a effectué des réémissions après réception d'acquittements négatifs des émissions de PDU précédentes.

Le protocole possède des fonctions telles que l'horloge de garde, le contrôle de séquençement, etc..., qui détectent ces situations afin d'y remédier (remise en ordre, réémission...)

#### 4.2.3. - Contrôle de flux dans le sous-protocole de transfert de données.

Le contrôle de flux du protocole a pour but d'asservir l'émission des PDU contenant des données de service effectuées par une entité (n) à la capacité de réception de son homologue et ceci dans les 2 sens de transfert de l'information.

Lorsque le contrôle de flux du protocole est mis en oeuvre au sein d'une couche (n), le service de communication doit en tenir compte dans la mise en oeuvre du contrôle de flux à l'interface (n) afin de répercuter la régulation des flots de données internes à la couche aux utilisateurs.

Citons deux notions complémentaires souvent utilisées pour réaliser le contrôle de flux et le contrôle d'erreur : **la fenêtre et le crédit** /B14/.

- Les crédits d'émission et de réception permettent la régulation de flux.

- Les fenêtres d'émission et de réception permettent le contrôle d'erreur (la taille de la fenêtre est liée à la notion de crédit et dépend de la stratégie mise en oeuvre).

#### 4.3 - Groupage - Dégroupage

Nous avons vu que les données contenues dans des SDU (n) sont transportées d'une entité (n) à l'autre dans des PDU (n). Lorsque la PDU est suffisamment grande pour contenir plusieurs SDU, la mise en oeuvre d'une fonction de groupage permet d'optimiser le transfert. A la réception de la PDU par l'entité (n) distante, l'opération inverse, c'est à dire le dégroupage, doit être effectuée pour extraire des SDU de la PDU (figure 4).

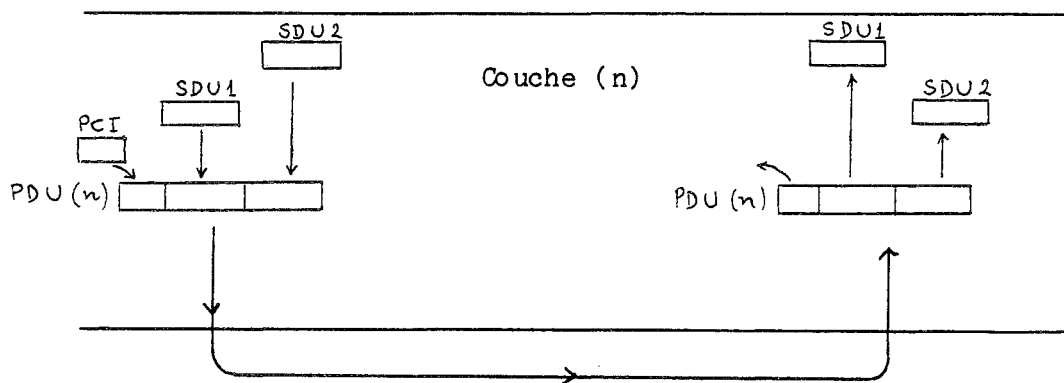


Figure 4 : Groupage et Dégroupage

#### 4.4 - Fragmentation - Réassemblage

Lorsqu'une SDU est plus grande que la taille limite d'une PDU, il faut la fragmenter à l'émission et en réassembler les différents morceaux à la réception (figure 5).

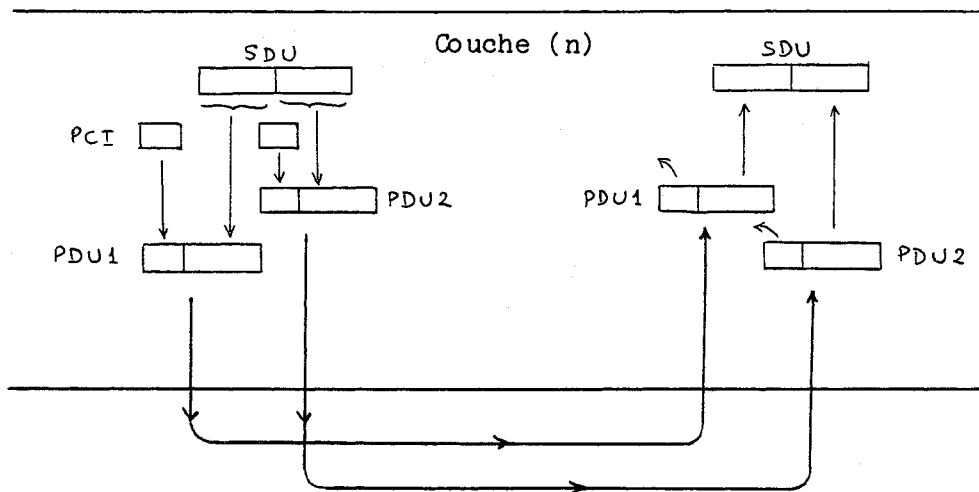


Figure 5 : Fragmentation et Réassemblage

5 - ADRESSAGE DANS UNE COUCHE (n)

La fonction d'adressage dans une couche (n) a pour but de déterminer un chemin passant par le service (n-1) pour l'acheminement des PDU (n) d'une entité (n) à une autre. Concrètement cette fonction fournit une suite d'adresses de points d'accès (n-1) par lesquels les données (n) doivent passer.

Le modèle de référence d'OSI mentionne trois fonctions : la fonction répertoire, la mise en correspondance des points d'accès et le routage :

- la fonction **répertoire** traduit les appellations globales d'entités homologues (n) en adresses (n-1) utilisées par les entités pour coopérer,

- la fonction de **mise en correspondance des points d'accès** permet d'associer à des adresses (n) desservies par une entité (n) des

adresses (n-1) utilisées par celle-ci pour accéder au service (n-1),

- la fonction de **routage** à l'intérieur d'une couche (n) permet de relayer la communication à l'aide d'une chaîne d'entités (n). Elle consiste donc à traduire une adresse (n) associée à une entité (n+1) en un chemin, ou itinéraire, permettant d'atteindre l'entité (n+1).

La figure 6 illustre l'utilisation de ces trois fonctions

Notation:

(E1, E2) : communication entre deux entités E1 et E2

A, B : adresses (n)

a, b, c, d, e, f : adresses (n-1)

e1, e2, e3, e4 : entités (n)

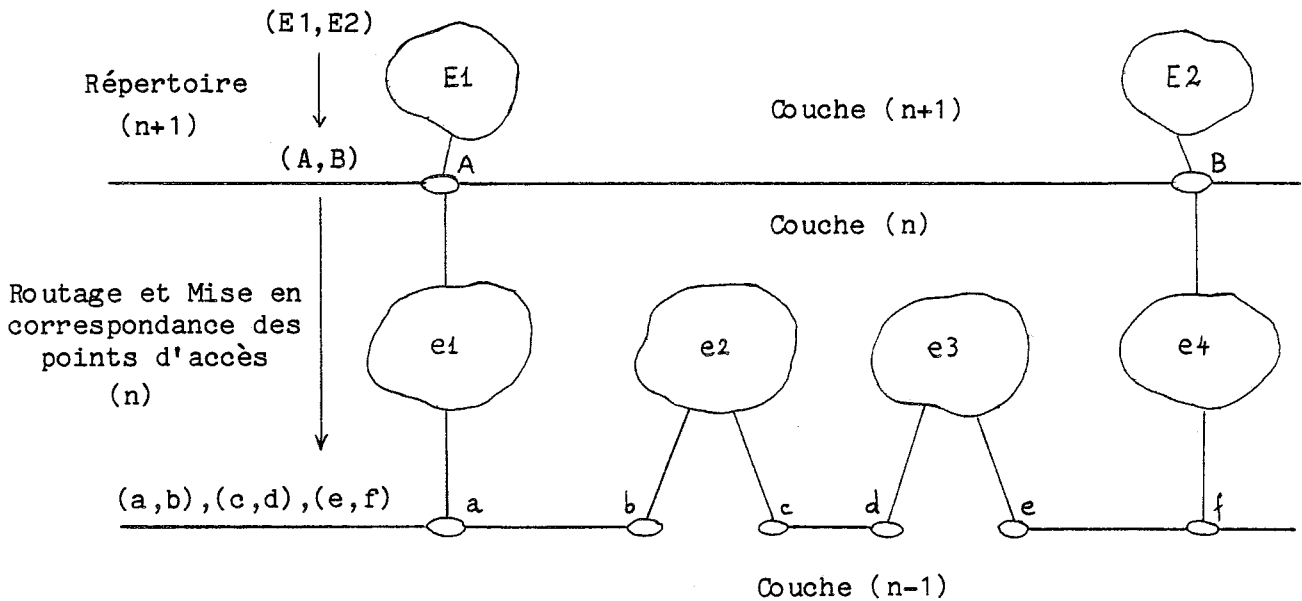


Figure 6 : Adressage dans une couche

Chaque entité (n) peut effectuer un routage partiel, c'est à dire déterminer seulement la prochaine adresse (n-1) à atteindre. Ainsi, le chemin sera constitué de proche en proche.

Nous donnons un certain nombre de remarques concernant la mise en oeuvre de ces fonctions:

- En général, pour résoudre l'adressage on utilise des tables (répertoire, table de mise en correspondance, table de routage). Dans le cas d'un adressage hiérarchisé, il suffit de connaître la structure de l'adresse (n) pour l'extraire l'adresse (n-1) /B1/.

- Une entité (n) doit posséder localement des informations sur les espaces de désignation suivants : le domaine d'appellation (n), le domaine d'adressage (n) et le domaine d'adressage (n-1). Il en résulte qu'une modification de l'un de ces espaces (ajout ou retrait d'une entité (n), d'un point d'accès (n), etc...) entraîne une mise à jour des tables d'adressage dans toutes les entités (n) de la couche. C'est une opération de reconfiguration globale d'une couche (n), assurée par des fonctions d'Administration de la couche.

- Dans le cas où les services (n) et (n-1) sont orientés connexion une association locale d'un identificateur d'une connexion (n) à celui d'une connexion (n-1) permet de faciliter le passage de la couche (n) à la couche (n-1) des informations et d'éviter de refaire à chaque transfert de données les opérations d'adressage fastidieuses déjà effectuées une fois lors de la mise en correspondance des points d'accès (opération qui oblige à explorer des tables d'adresses).

## 6 - MISE EN CORRESPONDANCE DE CONNEXIONS

Il s'agit des fonctions qui recueillent les données provenant des connexions (n) (resp. (n-1)), les traitent et les dirigent vers les connexions (n-1) (resp. (n)). Nous avons vu qu'il y a plusieurs politiques de correspondance de connexions : multiplexage, éclatement ou une combinaison des deux (cf chap I, §6.2.3).

Les données issues des connexions (n) sont des PDU (n); elles seront transformées, éventuellement après avoir subi des opérations de concaténation, éclatement, ..., en SDU (n-1) que le service (n-1) se chargera d'acheminer par la suite. Dans le sens inverse, des opérations de séparation, ou de recombinaison sont éven-



tuellement appliquées aux SDU (n-1) provenant du service (n-1) pour reconstruire des PDU (n) destinées aux différentes connexions (n).

Nous allons exposer dans ce paragraphe les différents aspects de la mise en oeuvre de ces politiques de correspondance.

### 6.1 - Multiplexage

Le multiplexage est une configuration où une seule connexion (n-1) prend en charge plusieurs connexions (n). La manière la plus simple de transférer les PDU provenant des différentes connexions (n) multiplexées est de les passer les unes après les autres à la connexion (n-1) correspondante. Dans ce cas, une PDU (n) devient une SDU (n-1). A l'autre extrémité de la connexion (n-1), le tri des PDU reçues est possible grâce à une identification de la connexion (n) que contient chaque PDU.

Lorsqu'on veut optimiser le nombre de transferts sur une connexion (n-1), plusieurs PDU (n) peuvent être concaténées (elles forment alors une SDU (n-1)) à l'émission, et séparées à la réception (figure 7), mais ceci implique les contraintes suivantes :

- Si aucune information de contrôle de la concaténation n'est ajoutée dans la SDU (n-1), il faut que chacune des PDU (n) ait une longueur connue, ou contienne dans son champ PCI (Protocol Control Information) un paramètre donnant sa longueur pour permettre la séparation des PDU à la réception de la SDU.

- Si au contraire les informations du champ PCI d'une PDU ne donnent pas la longueur de celle-ci, il faut qu'à la concaténation des PDU des informations de contrôle supplémentaires (délimiteurs,...) soient ajoutées dans la SDU afin de permettre leur séparation par la suite.

En outre pour ne pas surcharger une connexion (n-1) il faut prévoir la possibilité de limiter le nombre de connexion (n) qui y sont multiplexées.

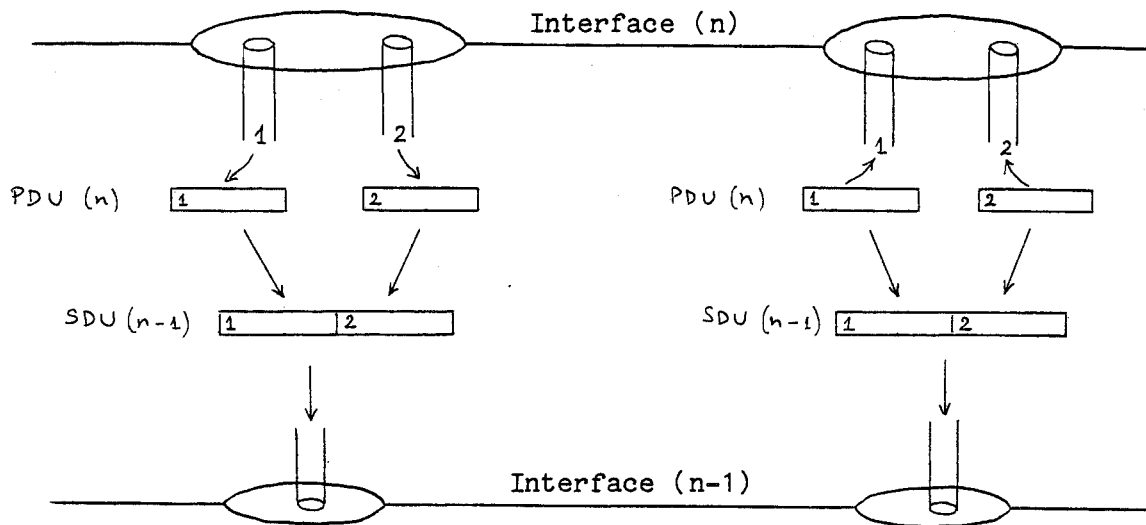


Figure 7 : Concaténation et séparation

## 6.2 - Eclatement

Dans ce cas plusieurs connexions (n-1) sont utilisées pour acheminer les informations (PDU) provenant d'une seule connexion (n). Il existe au moins trois politiques d'envoi des données :

### a/ Duplication des PDU

Chaque PDU issue d'une connexion (n) est dupliquée en autant d'exemplaires que de connexions (n-1) qui sont associées à cette connexion (n). Ces exemplaires identiques sont envoyés séparément sur les connexions (n-1). Le premier arrivé sans erreur est pris en compte, les autres seront rejetés. Ainsi, cette technique est particulièrement intéressante lorsque le service (n-1) est de mauvaise qualité car elle augmente la probabilité de réussite du transfert dans un seul envoi; cependant elle est chère en ressources mémoire et surcharge le service (n-1).

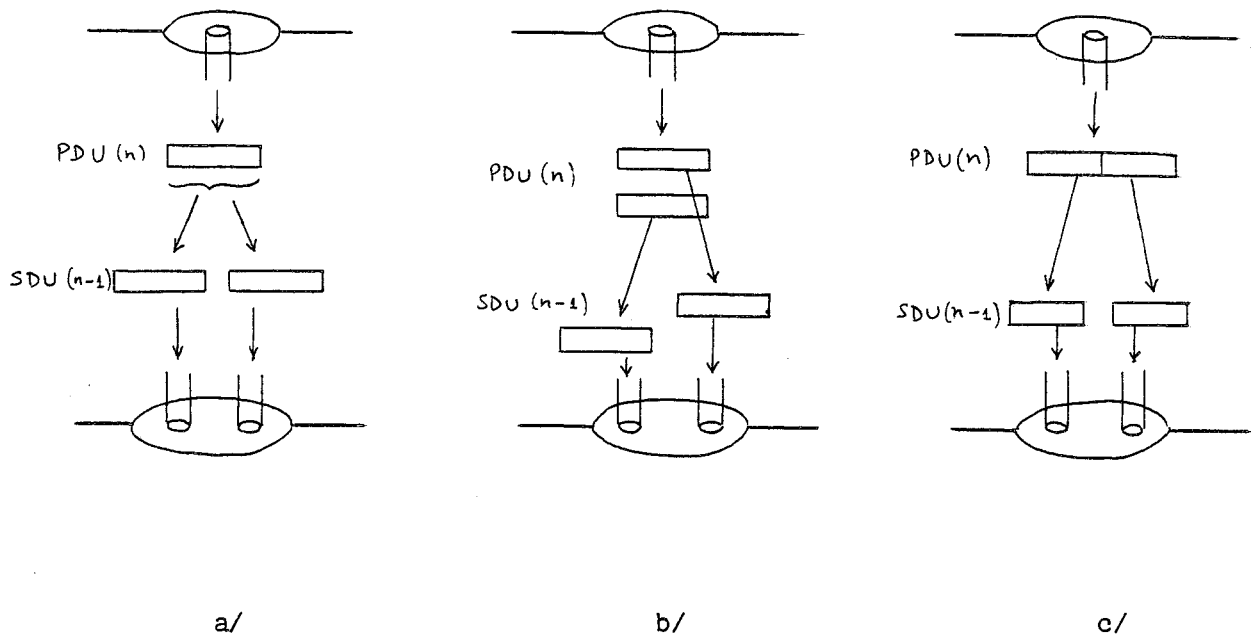


Figure 8 : Quelques politiques d'envoi de données dans une configuration d'éclatement de connexions

b/ Répartition des PDU sur les connexions (n-1)

Les connexions (n-1) reçoivent les PDU issues d'une connexion (n) dans un certain ordre fixé (tourniquet par exemple). De cette façon, le transfert à travers le service (n-1) est en principe plus rapide à condition que les connexions (n-1) fonctionnent réellement en parallèle. Par contre, cette politique n'augmente pas la probabilité de réussite du transfert d'une PDU dans un seul envoi. Par ailleurs, les PDU peuvent arriver au destinataire en désordre.

c/ Eclatement et Recombinaison des PDU

Il s'agit d'éclater la PDU à émettre en plusieurs morceaux et de les envoyer séparément sur les différentes connexions (n-1). Une SDU (n-1) est alors un fragment de la PDU (n). L'entité (n) réceptrice

doit attendre l'arrivée de tous les morceaux pour reconstituer la PDU.

Cette solution, qui permet en principe d'accroître la vitesse de transfert, est cependant assez fragile et lourde à mettre en oeuvre. En effet, elle repose sur un bon fonctionnement de toutes les connexions (n-1) faisant partie d'une même configuration d'éclatement. On voit que ce n'est pas le cas si le service (n-1) n'est pas très fiable. De plus, il faut définir des informations de contrôle de l'éclatement qui accompagnent les différents morceaux de la PDU pour en permettre la recombinaison à l'arrivée.

### 7 - UTILISATION DU SERVICE (n-1)

L'obtention d'une connexion (n-1), le transfert de données sur celle-ci, sa libération, ... sont des opérations d'accès aux facilités du service (n-1) qui doivent respecter des règles d'utilisation imposées par celui-ci. Par ailleurs, l'entité (n) doit réagir à des indications provenant du service (n-1) : établissement distant d'une connexion (n-1), mauvais fonctionnement du service (n-1), etc... Pour tout cela, il faut prévoir dans l'entité (n) un ensemble de fonctions qui gèrent les interactions avec le service (n-1). Plus précisément, elles assurent les rôles suivants :

- l'obtention et la libération des connexions (n-1) en fonction des besoins du niveau (n) : la qualité de la connexion (n-1) est choisie en fonction de celle de la connexion (n) à établir.

- le transfert des données sur les connexions (n-1) : émission et réception des SDU (n-1).

- la prise de décision en vue de répondre à des indications du service (n-1) : accord pour l'établissement distant d'une connexion (n-1), pour une demande de reprise, etc...

De même que les fonctions à l'interface vues au paragraphe 3 ces fonctions sont locales et définies par le concepteur.

\* \*  
\*

Nous avons passé en revue dans ce chapitre les différentes fonctions de traitement qui peuvent se trouver en toute généralité dans une entité (n). Avant d'introduire un modèle de fonctionnement d'une couche (n) (chapitres IV et V), dans le chapitre suivant nous décrivons une mise en oeuvre des deux couches Transport et Session. Nous inversons la démarche habituelle, en montrant d'abord une réalisation, pour que le lecteur puisse se rendre mieux compte des problèmes liés à la description et à la mise en oeuvre d'une couche. Pour cela, nous exposons notre réalisation telle qu'elle a été menée réellement. Elle donne lieu à des critiques constructives ; celles-ci nous permettront par la suite d'élaborer un modèle de fonctionnement d'une couche (n).

## CHAPITRE III

### EXEMPLE DE REALISATION DES COUCHES TRANSPORT ET SESSION

- 1 - PRELIMINAIRE
  
- 2 - REALISATION D'UNE STATION DE TRANSPORT
  - 2.1 - Fonctions mises en oeuvre
  - 2.2 - Schéma fonctionnel
  - 2.3 - Schéma d'exécution
  - 2.4 - Critiques
  
- 3 - REALISATION D'UNE ENTITE SESSION
  - 3.1 - Fonctions mises en oeuvre
  - 3.2 - Schéma fonctionnel
  - 3.3 - Schéma d'exécution
  - 3.4 - Critiques



## 1 - PRELIMINAIRE

L'origine des réalisations dont il est question dans ce chapitre est une étude menée en commun, dans le cadre du projet RHIN (Réseaux Informatiques Hétérogènes Normalisés), par l'Agence de l'Informatique, l'Ecole des Mines de Saint-Etienne et la société Périmatique. Elle consiste à démontrer la "faisabilité" de la transformation d'un micro-ordinateur existant et à vocation générale, en un système ouvert, c'est à dire apte à la communication avec l'extérieur selon le modèle d'OSI. L'Ecole des Mines a pris en charge la réalisation logicielle dont la mise en oeuvre des 5 premières couches (Physique, Liaison de données, Réseau, Transport, Session) constitue la partie essentielle.

La totalité de la réalisation est décrite dans les 4 rapports relatifs à la convention d'étude /D1,D2,D3,D4/. Nous choisissons de montrer ici l'essentiel de la réalisation des 2 couches 4 et 5 (Transport et Session) conformément aux spécifications des protocoles RHIN /D5,D6/.

Les critiques données à la fin de chaque description de réalisation visent à en tirer certains éléments d'une méthodologie de décomposition d'une couche ; ceci guidera par la suite l'étude d'un modèle de couche que nous exposerons dans les chapitres IV et V.

## 2 - REALISATION D'UNE STATION DE TRANSPORT (\*)

Cette réalisation a été inspirée des travaux effectués par l'équipe de description formelle du projet RHIN, proposant un modèle de Station de Transport /D7/. Le développement a été mené en collaboration avec une autre équipe du projet RHIN.

---

\* La Station de Transport est une entité serveur de la couche Transport



Nous allons présenter successivement :

- les fonctions de la couche Transport mises en oeuvre ;
- la décomposition fonctionnelle d'une Station de Transport en modules ;
- le schéma d'exécution qui, s'appuyant sur un moniteur multitâche utilisé comme noyau système, propose une façon de mettre en oeuvre les modules introduits et la communication entre ceux-ci.

### 2.1 - Fonctions mises en oeuvre

La couche Transport décharge les entités de la couche supérieure (couche Session) des détails d'un transfert de données de bout en bout tout en leur proposant des classes de service de qualités différentes. En plus, elle assure un accès optimisé au service de Réseau en fonction de la qualité et de la capacité globale de ce dernier : mise en oeuvre du multiplexage, de la concaténation, de la séparation,...

Nous avons mis en oeuvre la classe 2 du protocole de Transport RHIN. Cette classe gère le contrôle de flux sur chacune des connexions de Transport pour éviter la congestion dans la phase de transfert de données et la surcharge du service Réseau due au multiplexage des connexions de Transport sur les connexions Réseau.

Voici les fonctions importantes qui ont été mises en oeuvre :

- mise en correspondance des adresses de Transport et adresses Réseau;
- multiplexage de connexions de Transport sur une même connexion Réseau;
- établissement et libération de connexions de Transport;
- transfert de données sur une connexion de Transport;
- contrôle de séquence de bout en bout pour chaque connexion;
- contrôle de flux de bout en bout pour chaque connexion;
- concaténation et séparation;
- transfert d'unités de données express.

## 2.2 - Schéma fonctionnel

Le rôle de la Station de Transport est de mettre en oeuvre des connexions de Transport disponibles pour la couche Session en utilisant le service Réseau existant. Il peut se décomposer en trois groupes de fonctions :

- Etablissement, utilisation et libération de connexions de Transport conformément au protocole de Transport RHIN.
- Gestion du multiplexage de plusieurs connexions de Transport sur une même connexion Réseau.
- Utilisation du service Réseau : établissement, utilisation et libération de connexions Réseau.

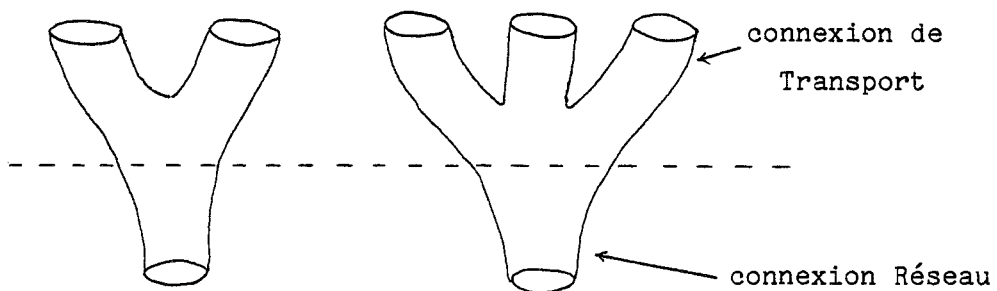


Figure 1

Ce schéma laisse apparaître deux niveaux de fonctionnalité à l'intérieur de la couche Transport (figure 2) :

- Un niveau supérieur qui gère le protocole de Transport et les interactions avec la couche Session. Ces fonctions sont relatives à des connexions de Transport.
- Un niveau inférieur qui fait correspondre les connexions de Transport à des connexions Réseau et gère les interactions avec la couche Réseau.

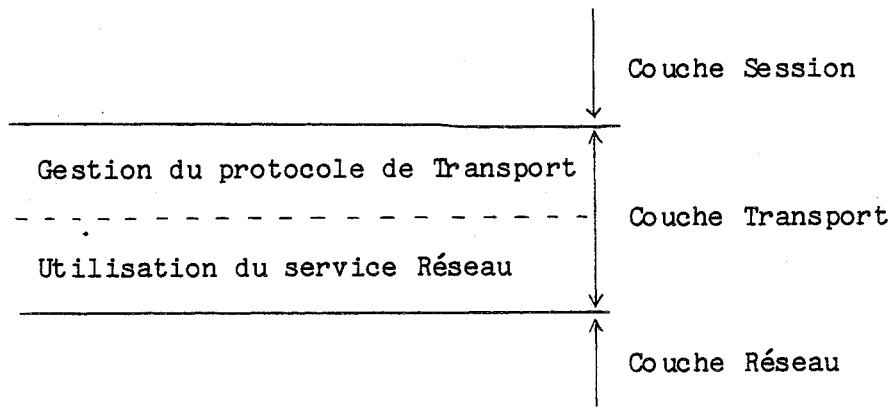


Figure 2

D'autre part, pour chaque type de connexion (Transport ou Réseau), nous retrouvons les deux catégories de fonctions à gérer :

- établissement et libération de connexions
- émission et réception de données

La représentation de ces fonctions et des interactions existant entre elles nous conduit au schéma fonctionnel retenu pour la réalisation de la Station de Transport (figure 3).

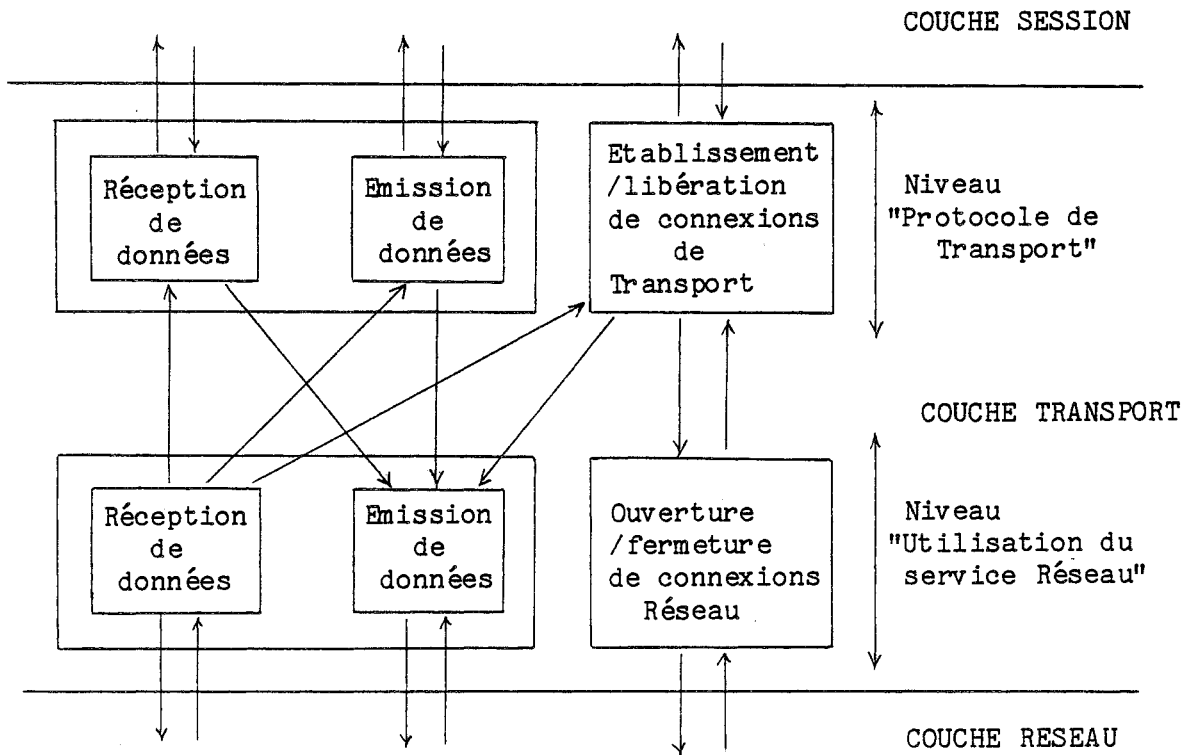


Figure 3 : Schéma fonctionnel d'une Station de Transport

A l'intérieur du niveau "Protocole de Transport" (resp. niveau "Utilisation du service Réseau") chaque connexion de Transport (resp. connexion Réseau) est représentée par un **contexte** appelé contexte de connexion de Transport (resp. contexte de connexion Réseau). C'est un ensemble d'informations relatives à la connexion concernée (identificateur, état de fonctionnement, etc...).

Le comportement de chaque module du schéma fonctionnel est décrit par un automate. Il est important de noter que cet automate est relatif à chaque contexte de connexion (Transport ou Réseau selon le niveau où se trouve le module correspondant).

Nous résumons dans ce qui suit les fonctions de chacun des modules du schéma.

a/ Module "Etablissement/libération de connexions de Transport"

Il gère d'une part l'établissement et la libération de connexions de Transport qui font partie des spécifications du protocole de Transport et d'autre part l'établissement et la libération de "**chemins**"; le "chemin" représente une liaison locale entre une connexion de Transport et une connexion Réseau. C'est un concept introduit par l'équipe de description formelle du projet RHIN pour son modèle PASCAL de la Station de Transport /D7/. L'établissement et la libération de chemins sont des opérations internes à la couche Transport permettant d'établir ou de rompre les liaisons entre les connexions de Transport et les connexions Réseau.

Le fonctionnement de ce module est décrit par un automate dans lequel les opérations sur les chemins s'enchaînent avec celles du protocole d'établissement et de libération de connexions de Transport.

b/ Module "Transfert de données sur une connexion de Transport"

Il est lui-même composé de 2 modules : Emission et Réception, qui prennent en charge le transfert de données sur une connexion de Transport lorsque celle-ci est établie.

Le module "Emission sur une connexion de Transport" gère deux flots de données en émission : un flot normal et un flot

express. Les données reçues à l'interface sont transformées en T-PDU (Transport Protocol Data Unit) de données normales ou express pour être envoyées ensuite au module "Emission sur une connexion Réseau" du niveau inférieur.

Quant au module "Réception sur une connexion de Transport", son travail consiste à fabriquer des unités de données de l'interface (IDU) à partir des T-PDU de données normales ou express reçues, et à les transmettre ensuite à la couche Session.

Les deux modules précédents gèrent au cours de l'émission et de la réception des données le contrôle de flux à 2 niveaux : le **crédit protocole** et le **crédit interface** dont la gestion nécessite des fonctions de calcul permettant de passer de l'un à l'autre.

c/ Module "Ouverture/fermeture de connexions Réseau"

Ce module réalise, sur la demande du module "Etablissement/libération de connexions de Transport", l'association "connexion de Transport - connexion Réseau" (appelée "chemin"). Elle peut être détruite soit sur l'initiative du module "Etablissement/libération de connexions de Transport", soit à la suite d'incidents signalés par la couche Réseau.

Ce module ne prend l'initiative de l'ouverture d'une connexion Réseau qu'à la suite d'une requête de demande de chemin et à condition qu'aucune connexion Réseau existante ne puisse satisfaire la demande. Il est possible qu'un multiplexage soit effectué au cours de cette phase.

En plus, ce module réagit aux indications venant du Réseau, concernant l'ouverture, la fermeture des connexions Réseau ou les incidents sur celles-ci.

Si une indication d'ouverture de connexion Réseau lui parvient, il accepte l'ouverture mais ne répercute pas l'événement à d'autres modules.

d/ Module "Transfert de données sur une connexion Réseau"

Ce module est composé de 2 modules : Emission et Réception.

Le module "Emission" est chargé d'envoyer des données à la couche Réseau en respectant son interface. D'autre part, il met en oeuvre une fonction de **concaténation** qui consiste à regrouper dans une même N-SDU (Network Service Data Unit) plusieurs T-PDU.

Le module "Réception" récupère les données arrivant du Réseau et en extrait des T-PDU (opération de **séparation** ).

L'émission et la réception sur une connexion Réseau sont par ailleurs régulées par le contrôle de flux à l'interface Réseau.

### 2.3 - Schéma d'exécution

Ce paragraphe présente la structure dynamique de la Station de Transport permettant la coordination de l'ensemble des modules présentés précédemment.

Le logiciel système que nous utilisons offre le concept de **tâches** et celui de **boîtes aux lettres** pour permettre la communication et la synchronisation entre tâches. Ces concepts sont illustrés par le schéma ci-dessous :

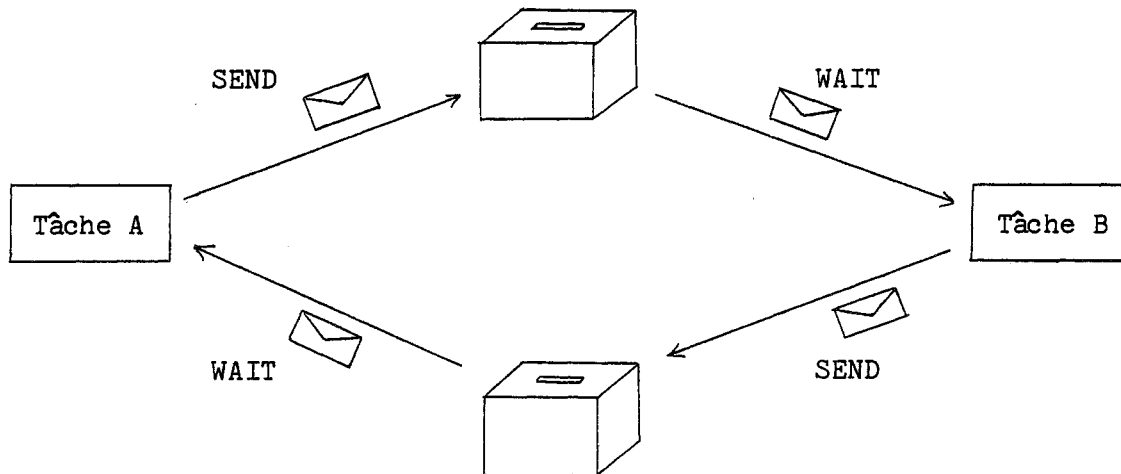


Figure 4

A chaque boîte aux lettres sont associées une file d'attente de tâches et une file d'attente de messages. D'une manière générale, une tâche qui cherche un message dans une boîte aux lettres (par la

primitive WAIT) est mise en attente s'il n'existe aucun message. Une tâche peut envoyer un message à une boîte aux lettres grâce à la primitive SEND. Si aucune tâche est en attente de messages sur cette boîte, le message sera mis en file. Dans le cas contraire, il sera attribué à la première tâche en attente.

Nous avons introduit, à cause de notre choix tout à fait arbitraire de réaliser la Station de Transport par une seule tâche, le mécanisme d' **attente multiple** . Il s'agit de permettre à une tâche d'être en attente sur plusieurs files de messages différentes. Ce mécanisme est réalisé dans le noyau système par des **boîtes aux lettres multiples** . A une boîte aux lettres multiple est associé un certain nombre de boîtes aux lettres ordinaires (celles qui sont décrites précédemment) ; une certaine priorité relative de traitement est attribuée à chacune d'entre elles.

Une tâche en attente sur une boîte aux lettres multiple (par la primitive MULTIPLE WAIT) sera réveillée quand il y aura eu un dépôt de message dans l'une des boîtes aux lettres ordinaires associées à cette boîte aux lettres multiple. Elle reçoit alors le message ainsi que le numéro de la boîte aux lettres ordinaire contenant ce message.

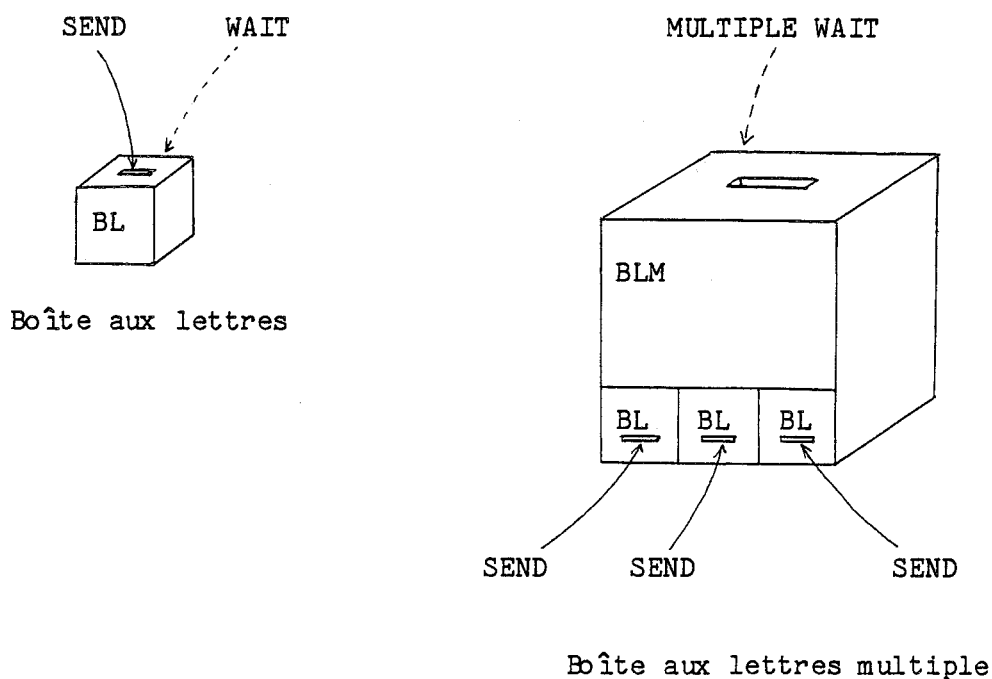


Figure 5

Utilisation de ces concepts pour la mise en oeuvre de la Station de Transport :

Les échanges de requêtes et d'indications entre la Station de Transport et les couches Réseau et Session et à l'intérieur de la Station de Transport utilisent le mécanisme de boîte aux lettres. Sept boîtes aux lettres sont réservées à cet effet et réparties de la façon suivante :

\* 4 boîtes aux lettres permettent la communication entre le niveau Session et la Station de Transport. Une boîte aux lettres (boîte n° 1 de la figure 6) est utilisée pour contenir les indications envoyées par la Station de Transport à la couche Session. Trois autres boîtes aux lettres permettent à la couche Session de déposer des requêtes pour la couche Transport selon leur type :

- requêtes concernant l'ouverture et la fermeture de connexions de Transport (boîte n° 2).
- émission de données normales et envoi de crédit d'interface (boîte n° 3).
- émission de données express et envoi de crédit d'interface (boîte n° 4).

L'utilisation de trois boîtes aux lettres (au lieu d'une) permet de séparer les différents types de requêtes provenant de la couche Session en trois files de messages auxquelles on peut donner des priorités de traitement différentes. Les données express par exemple doivent être traitées en priorité par rapport aux autres types de requêtes.

\* 2 boîtes aux lettres permettent la communication entre la Station de Transport et la couche Réseau. L'une (boîte n° 7) reçoit les requêtes envoyées par la couche Transport vers la couche Réseau, l'autre (boîte n° 6) les indications émises par la couche Réseau pour la couche Transport.

\* 1 boîte aux lettres (boîte n° 5) est réservée aux échanges à l'intérieur de la Station de Transport. Toutes les primitives de



communication entre les différents modules de la Station de Transport transitent par cette boîte aux lettres. Il est donc nécessaire de mettre en place un aiguillage des messages contenant ces primitives afin de transmettre chaque message au module responsable de son traitement.

La nécessité d'un aiguillage existe aussi pour les messages en provenance :

- de la couche Réseau : ils peuvent en effet être destinés à l'un quelconque des modules chargés de gérer les connexions Réseau.

- de la couche Session pour ce qui concerne les boîtes aux lettres de données normales ou de données express : il convient en effet de trier les requêtes d'émission de données qui sont destinées au module émission et les requêtes d'envoi de crédit qui concernent le module réception.

Pour des raisons historiques (simplicité apparente,...), nous avons choisi d'exécuter les modules de la Station de Transport dans une seule tâche. Mises à part les boîtes aux lettres 1 et 7 qui contiennent des messages destinés à la couche Session et à la couche Réseau, les autres boîtes (2,3,4,5,6) constituent les différentes files de messages en attente de traitement par la tâche Station de Transport. Une boîte aux lettres multiple est utilisée pour englober ces cinq dernières boîtes aux lettres.

La tâche Station de Transport attend l'arrivée des messages sur la boîte aux lettres multiple et réalise l'aiguillage. Pour traiter en premier les événements les plus urgents, une priorité est affectée à chacune des boîtes aux lettres 2,3,4,5,6 ; l'ordre décroissant des priorités est le suivant :

- boîte 4 (requêtes d'émission de données express)
- boîte 5 (requêtes et indications internes)
- boîte 6 (indications du Réseau)
- boîte 3 (requêtes d'émission de données normales)
- boîte 2 (requêtes d'établissement et de libération de connexions de Transport)

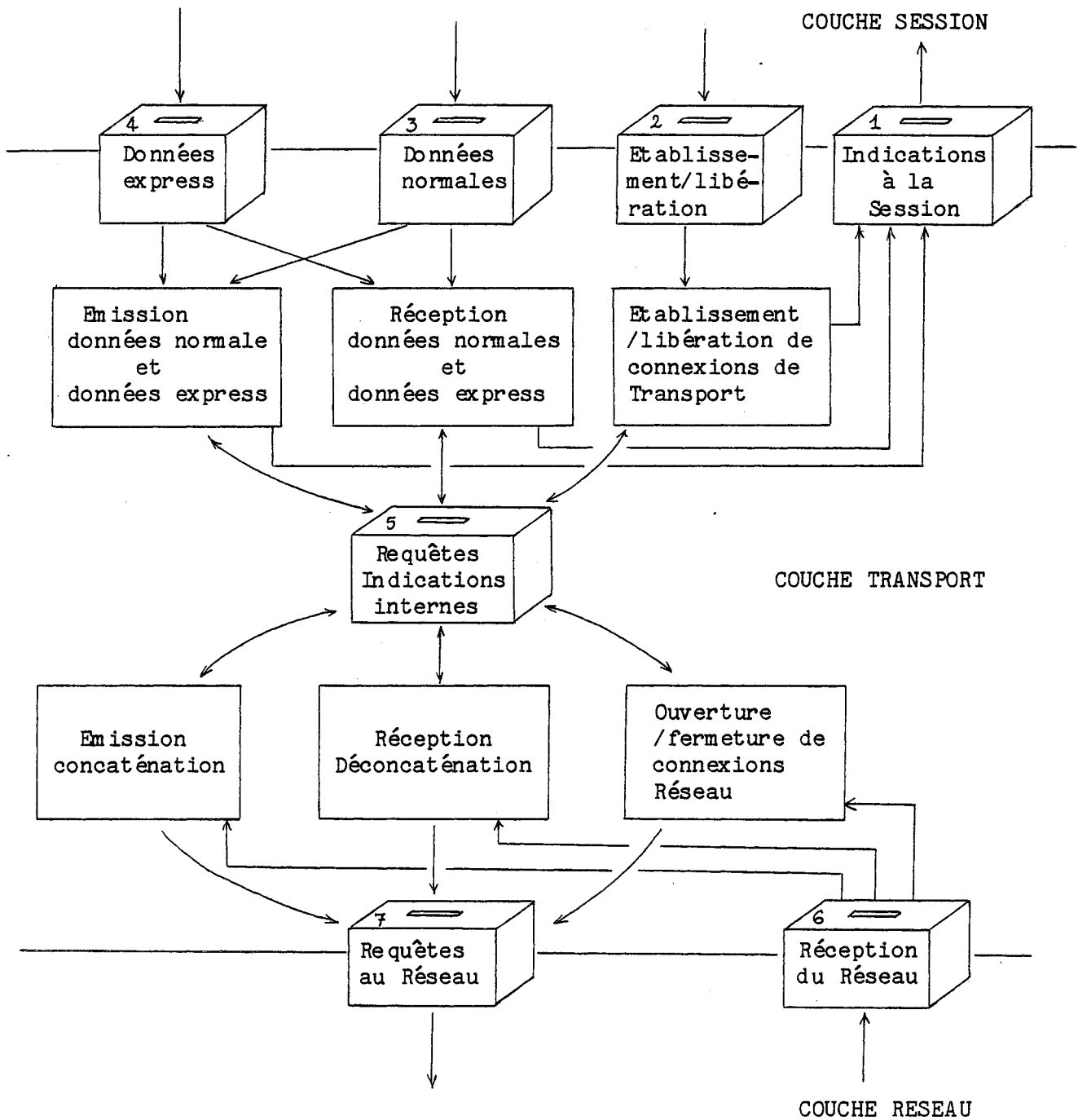


Figure 6 : Affectation des boîtes aux lettres  
(implantation d'une Station de Transport)

Les messages sont donc retirés des différentes boîtes aux lettres dans un ordre qui dépend de cette priorité. Après avoir été récupéré, un message est ventilé vers le module concerné par son traitement, chaque module étant exécuté sous le contrôle de la tâche Station de Transport. Au retour d'un module, la tâche de Transport se remet en attente de message sur la boîte aux lettres multiple.

L'algorithme général exécuté par la tâche de Transport est le suivant :

```
bal2,bal3,bal4,bal5,bal6 : boîtes aux lettres ;  
balm (bal2,bal3,bal4,bal5,bal6) : boîte aux lettres multiple ;
```

Répéter ;

```
message = MULTIPLE WAIT (balm) ;  
Cas message.type faire; /* aiguillage du message */  
  Emission-connexion-transport ;  
  Réception-connexion-transport ;  
  Etablissement-libération-connexion-transport ;  
  Emission-connexion-réseau ;  
  Réception-connexion-réseau ;  
  Ouverture-fermeture-connexion-réseau ;
```

Fincas;

Fin-répéter ;

Notons que la taille des boîtes aux lettres est infinie (le chaînage des messages est sans limite). Il n'y a donc pas de situation de saturation des boîtes aux lettres, mais il risque d'y avoir la pénurie de mémoire réservée aux messages.

#### 2.4 - Critiques

Faisons tout d'abord quelques constatations tirées de la mise en oeuvre de la Station de Transport :

- La première approche guidant la décomposition est la mise en évidence des niveaux de traitements successifs à l'intérieur d'une couche (la couche Transport est décomposée ici en deux niveaux).

- Par contre la décomposition en modules à l'intérieur d'un niveau de la Station de Transport présentée précédemment n'exprime pas nettement, du point de vue de la description fonctionnelle, tous les aspects du fonctionnement interne de ce sous-niveau.

C'est pourquoi, nous évoquons ci-dessous quelques aspects particulièrement importants qu'il faudrait expliciter dans les schémas fonctionnel et dynamique :

- Le choix de la classe de protocole, l'allocation d'un nouveau contexte de Transport,... sont implicitement effectués dans le module "Etablissement/libération de connexions de Transport". La décomposition fonctionnelle de la Station de Transport serait plus fine si l'on isolait ces fonctions dans un module à part jouant le rôle d' **administration locale** des connexions de Transport.

- Les opérations d'établissement et de rupture de liaisons entre les connexions de Transport et les connexions Réseau (demande et libération de chemins) interviennent dans l'automate associé à chaque contexte de connexion de Transport du module "Etablissement/libération de connexions de Transport". Il est préférable d'écarter ces opérations internes à la couche Transport de ce module afin que celui-ci n'ait qu'à gérer l'automate défini par les spécifications standard du protocole de Transport. Les opérations d'établissement et de rupture de liaisons entre les connexions de Transport et les connexions Réseau feront alors partie des fonctions d'administration locale.

- Les configurations de correspondance des connexions ne sont pas explicitées par les schémas. Si plusieurs classes du protocole étaient implantées, la couche assurerait tantôt une correspondance simple (classe 0 : une connexion de Transport associée à une connexion de Réseau), tantôt un multiplexage (classes 1,2,3,4 : plusieurs connexions de Transport associées à une connexion Réseau). Aucun mécanisme n'est prévu pour distinguer ces deux configurations lors de l'établissement de la liaison entre une connexion de Transport et une connexion Réseau (ce qui correspond ici à une opération de demande de chemin).

- Les connexions existant dans une même couche à un moment donné sont indépendantes par définition. C'est pourquoi on souhaite avoir dans le schéma d'exécution une expression du parallélisme du fonctionnement des connexions, indépendamment de la mise en oeuvre sur un système donné (qui peut ne pas permettre la réalisation de ce parallélisme). On constate qu'un schéma de processus communicants traduit mieux cette situation. Remarquons par ailleurs que le multiplexage des connexions pourrait être exprimé par un schéma "producteurs multiple - consommateur unique" dans le sens "connexions de Transport → connexion Réseau".

### 3 - REALISATION D'UNE ENTITE SESSION

Cette réalisation est conforme aux spécifications du protocole Session RHIN.

#### 3.1 - Fonctions mises en oeuvre

Par rapport au protocole de Transport qui s'occupe essentiellement de la fonction de transfert de données, le protocole de Session prend en charge un travail de toute autre nature : l'arbitrage de dialogue des entités de la couche Présentation.

Nous avons mis en oeuvre le sous-ensemble "Transfert de fichiers" du protocole de Session RHIN, qui comporte les fonctions suivantes :

- établissement de connexions de Session
- libération de connexions de Session
- échange de données normales
- échange de données express
- gestion du dialogue
- synchronisation de la connexion de Session

Notons qu'il n'y a pas de multiplexage de connexions dans la couche Session (par définition de l'ISO).

### 3.2 - Schéma fonctionnel

A l'intérieur de la couche Session deux niveaux sont aisément mis en évidence. Le premier concerne la gestion des connexions de Session conformément au Protocole de Session RHIN et à l'interface définie pour les entités de Présentation. Le second concerne l'utilisation des connexions de Transport conformément à l'interface fournie par le service de Transport de manière à supporter les connexions de Session.

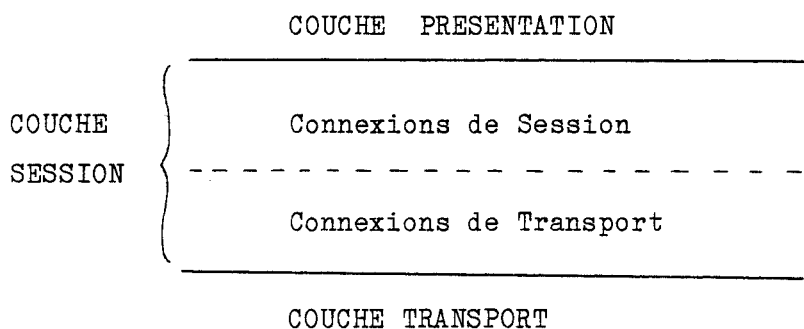


Figure 7 - Deux niveaux de la couche Session

En ce qui concerne la gestion des connexions de Session, nous introduisons les deux modules suivants :

a/ Module "Gestion d'environnement des connexion de Session" :

Grâce aux enseignements tirés de la réalisation de la Station de Transport, nous mettons en évidence ce module qui regroupe les fonctions d'administration locale des connexions de Session. Il prend en compte des demandes d'établissement de connexion de Session : ces demandes émanent soit de l'interface avec la couche Présentation (requêtes locales) soit d'une connexion de Transport avec la réception d'une PDU d'établissement distant d'une connexion de Session (une S-PDU-CN : Session Protocol Data Unit - Connect).

Le module assure la création de l'environnement nécessaire à une connexion de Session : création d'un contexte, allocation d'une connexion de Transport, démarrage de l'automate gérant la connexion. De même, en fin de Session, ce module est chargé de la libération de la connexion de Transport et de la destruction du contexte. Les incidents survenant en cours de Session sur une connexion de Transport sont récupérés par ce module et éventuellement répercutés sur la connexion de Session associée.

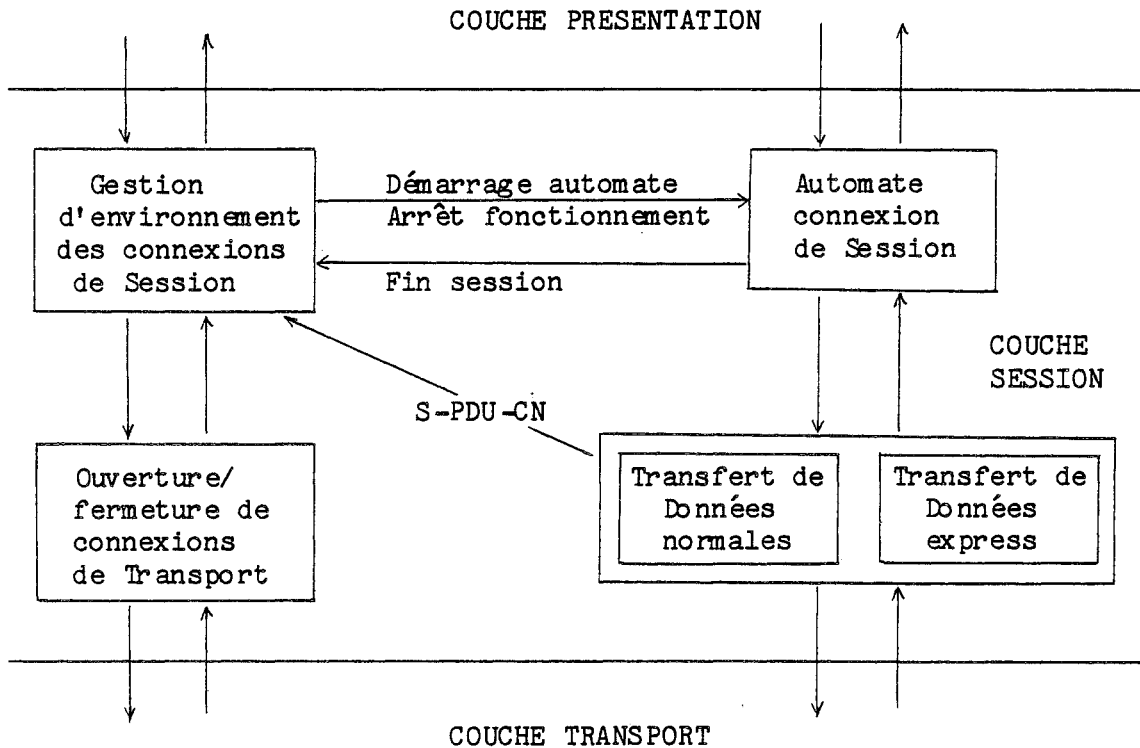


Figure 8 : Schéma fonctionnel d'une entité Session

b/ Module "Protocole de Session" :

Il déroule l'automate gérant chaque connexion de Session. Le démarrage de l'automate est provoqué par le module décrit précédemment ; ensuite, l'automate évolue au gré des primitives émises par

l'entité de Présentation relativement à cette connexion, ou des éléments de Protocole de Session reçus sur la connexion de Transport associée. Conceptuellement, on peut considérer qu'il existe un exemplaire de ce module par connexion de Session.

La figure 8 donne un premier aperçu des interactions de ces deux modules entre eux et avec les entités des niveaux adjacents.

La gestion des connexions de Transport est, elle aussi, réalisée par deux modules :

a/ Module "Ouverture/fermeture de connexions de Transport" :

Il assure l'ouverture et la fermeture des connexions de Transport selon l'interface fournie par le service de Transport. Il s'agit d'une part, de traiter les demandes d'allocation et de libération provenant du module "Gestion d'environnement des connexions de Session" et de générer en retour la réponse appropriée (connexion de Transport allouée, refusée ou libérée), et d'autre part, de traiter des événements provoqués soit par une entité Session distante (connexion de Transport ouverte) soit par le service de Transport à la suite d'incidents (connexion de Transport rompue).

b/ Module "Transfert de données sur des connexions de Transport"

Il est lui-même composé de deux modules :

b.1 - Transfert de données normales.

Il s'agit d'émettre sur la connexion de Transport associée les S-PDU générées lors des actions exécutées par l'automate d'une connexion de Session, et de transmettre à ce même automate les données reçues sur cette connexion de Transport.

Il est important de noter que l'association "connexion de Session - connexion de Transport" n'est pas effective tant que



l'automate gérant la connexion de Session n'a pas démarré. Il s'ensuit que tout S-PDU reçu sur une connexion de Transport non encore associée à une connexion de Session est transmis au module "Gestion d'environnement des connexions de Session : en particulier un S-PDU-CN permet la création d'un nouveau contexte de Session et le démarrage de l'automate correspondant. L'émission et la réception de données sur une connexion de Transport sont soumises à un contrôle de flux à l'interface avec le service de Transport.

#### b.2 - Transfert de données express.

Afin de supporter le Protocole de Session RHIN, l'ouverture d'une connexion de Transport sera toujours demandée avec l'option données express ; à l'instar du flot de données normales, le flot de données express est soumis à un contrôle de flux.

#### 3.3 - Schéma d'exécution

L'implantation d'une entité Session est analogue à celle de la Station de Transport : les modules sont exécutés par une seule tâche, appelée tâche Session ; ils communiquent entre eux et avec l'extérieur grâce à des boîtes aux lettres. Dans le schéma dynamique d'une entité Session (figure 9), une boîte aux lettres est rattachée à un niveau ou à un module de ce niveau. Nous avons introduit 8 boîtes aux lettres, réparties de la manière suivante :

- La boîte aux lettres 1, associée au module "Gestion d'environnement des connexions de Session", reçoit des indications du niveau d'utilisation des connexions de Transport, des requêtes d'établissement d'une connexion de Session de la part de l'utilisateur, et des événements provenant du module "Protocole Session".
- La boîte aux lettres 2, associée au module "Protocole Session", reçoit des événements provenant de l'utilisateur (requêtes d'utilisation et de terminaison de la connexion de Session), du module "Gestion d'environnement des connexions de Session", et du module "Transfert de données sur des connexions de Transport".

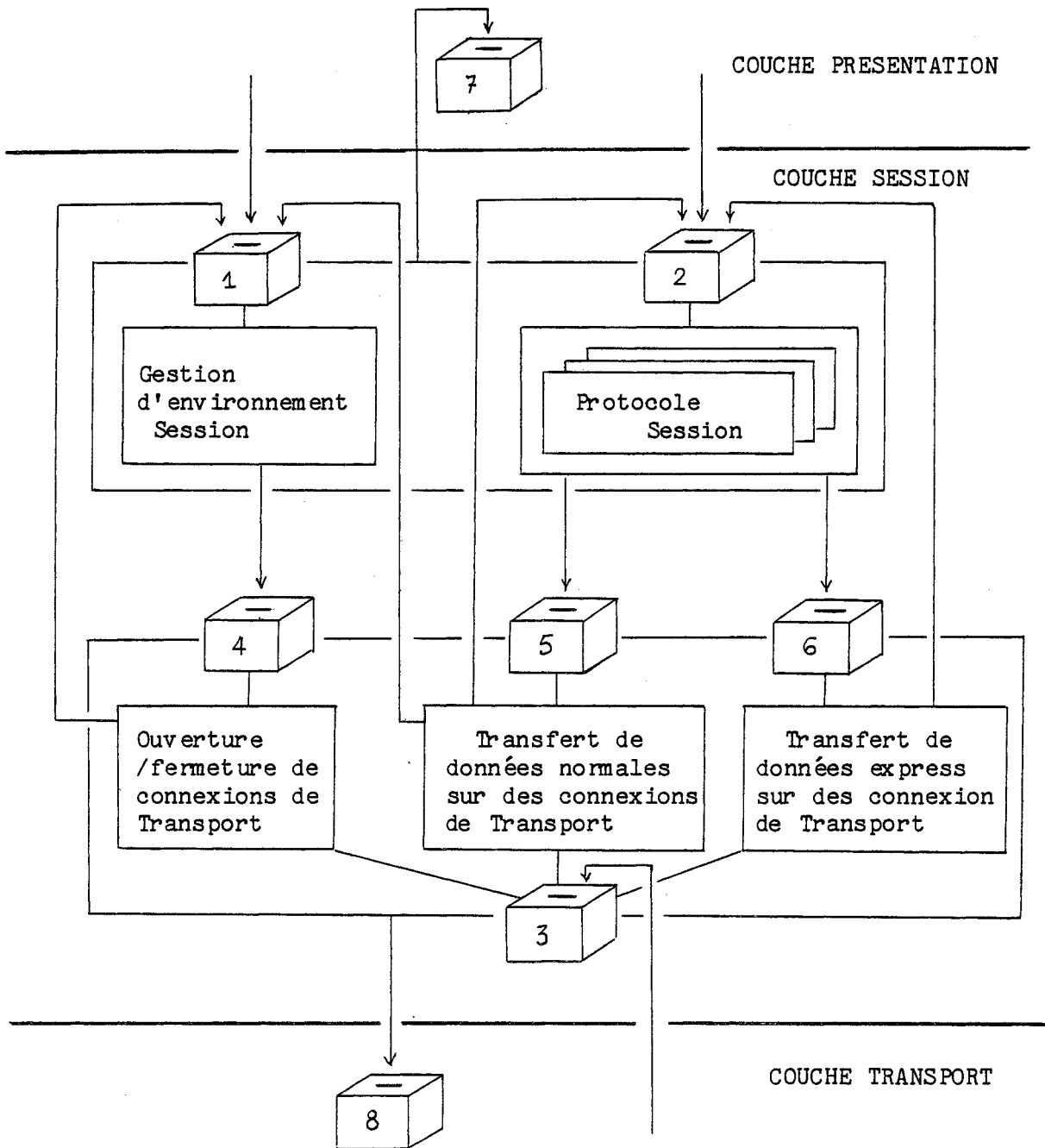


Figure 9 : Affectation des boîtes aux lettres (implantation d'une entité Session)

Les flèches correspondent à des événements (messages) envoyés à des boîtes aux lettres.

- La boîte aux lettres 3 concerne le niveau d'utilisation des connexions de Transport. Elle contient des messages venant de la couche Transport et destinés aux trois modules responsables de la gestion des connexions de Transport : "Ouverture/fermeture", "Transfert de données normales", "Tranfert de données express".
- La boîte aux lettres 4, réservée au module "Ouverture/fermeture de connexions de Transport", reçoit des événements issus du module "Gestion d'environnement des connexions de Session".
- Les boîtes aux lettres 5 et 6, réservées respectivement aux modules "Transfert de données normales" et "Transfert de données express" (sur des connexions de Transport), reçoivent des événements venant du module "Protocole Session".
- Les boîtes aux lettres 7 et 8 contiennent des messages envoyés par l'entité Session à la couche Présentation et à la couche Transport.

La tâche Session est à l'état dormant s'il n'y a aucun événement dans les six premières boîtes aux lettres et est réveillée par l'arrivée d'un événement dans l'une quelconque d'entre elles. Pour cela, on introduit une boîte aux lettres multiple qui englobe les boîtes aux lettres 1 à 6, et sur laquelle la tâche Session se met en attente.

Pour traiter en premier les événements les plus urgents, nous attribuons des priorités aux différents événements. Voici l'ordre décroissant des priorités :

- Les requêtes internes concernant le transfert de données express (boîte 6).
- Les indications de la couche Transport (boîte 3).
- Les requêtes internes concernant le transfert de données normales (boîte 5).
- Les requêtes internes concernant l'ouverture et la fermeture de connexions de Transport (boîte 4).
- Les requêtes concernant le protocole de Session (boîte 2).

- Les requêtes et indications destinées au module "Gestion d'environnement des connexions de Session" (boîte 1).

L'algorithme général de la tâche Session est le suivant :

bal1,bal2,bal3,bal4,bal5,bal6,bal7 : boîtes aux lettres ;  
balm (bal1,bal2,bal3,bal4,bal5,bal6,bal7) : boîte aux lettres  
multiple ;

Répéter ;

message = MULTIPLE WAIT (balm) ;

Cas message.type faire ; /\* aiguillage du message \*/

Gestion-environnement-session ;

Protocole-session ;

Ouverture-fermeture-connexion-transport ;

Transfert-données-express-connexion-transport ;

Transfert-données-normales-connexion-transport ;

Fincas;

Fin-répéter ;

Il est important de noter que, pour augmenter le parallélisme à l'exécution, on peut mettre en oeuvre plusieurs tâches dont chacune correspond à un module (ou un exemplaire de celui-ci).

### 3.4 - Critiques

Malgré un certain nombre d'améliorations apportées au modèle de réalisation d'une entité Session par rapport au modèle de la Station de Transport, il existe encore des points importants qu'il aurait fallu expliciter dans les schémas fonctionnel et dynamique :

- L'automate d'ouverture et fermeture des connexions de Transport fonctionne relativement à chaque connexion de Transport. Il en est de même pour le transfert de données. Il est nécessaire, sur le plan de la description, d'explicitier cette indépendance des connexions de Transport. Conceptuellement, le schéma dynamique devrait montrer qu'il

y a autant d'exemplaires des modules "Ouverture/fermeture" et "Transfert de données" (relatifs à une connexions de Transport) que de connexions de Transport existantes.

- Le modèle de référence d'OSI préconise que les durées de vie d'une connexion de Session et de la connexion de Transport associée peuvent être différentes. Deux situations peuvent se produire /B1/ :

a/ une même connexion de Transport prend en charge successivement plusieurs connexions de Session,

b/ plusieurs connexions de Transport prennent successivement en charge une même connexion de Session (cas de la reconfiguration).

Pour prendre en compte de telles situations, il faut introduire des fonctions d'administration locale des connexions de Transport au même titre que celles administrant les connexions de Session. Elles prendront en charge le contrôle de la durée de vie des connexions de Transport au service des connexions de Session.

\* \* \*

La réalisation des couches Transport et Session montre que malgré l'existence d'un protocole bien défini, il reste encore beaucoup de paramètres que le concepteur doit lui-même fixer lors de la mise en oeuvre. Par conséquent, un modèle de couche doit être suffisamment modulaire et présenter des frontières nettes entre les réalisations des différentes fonctions pour faciliter les modifications ultérieures éventuelles. Ces dernières sont dues à l'adaptation de la couche à un environnement Réseau donné, ou sont engendrées par des changements dans les nouvelles versions de normes.

De ce qui précède et de l'expérience que nous avons pu en tirer, nous allons dégager un modèle général de fonctionnement d'une couche en nous appuyant sur les idées suivantes :

a/ La décomposition en sous-couches est une opération préliminaire qui permet d'éclairer les différents niveaux de traitement d'une couche. Ce sera l'objet du chapitre IV.

b/ Il paraît logique de mettre en évidence, dans le schéma dynamique, le fonctionnement en parallèle et l'indépendance des connexions (n) et des connexion (n-1). Chacune de ces connexions doit être gérée par une tâche indépendante. Le chapitre V propose un modèle dynamique d'une couche de communication en utilisant le concept d'"activité" pour la représentation des tâches.

c/ Il faudra également mettre en évidence dans le modèle les fonctions d'administration des connexions (n) et des connexions (n-1) à l'intérieur d'une entité (n).



## CHAPITRE IV

### DECOMPOSITION EN SOUS-NIVEAUX FONCTIONNELS D'UNE COUCHE (n)

#### 1 - PRELIMINAIRE

#### 2 - STRUCTURATION D'UNE COUCHE (n) EN SOUS-COUCHES

2.1 - Services (n) et (n-1) de type sans connexion

2.2 - Service (n) orienté connexion, service (n-1) de type  
sans connexion

2.3 - Service (n) de type sans connexion, service (n-1)  
orienté connexion

2.4 - Services (n) et (n-1) orientés connexion

#### 3 - CONCEPT DE NOEUD DE CORRESPONDANCE





## 1 - PRELIMINAIRE

La décomposition d'une couche de communication en sous-niveaux fonctionnels a pour objectif d'affiner la description et d'explicitier certains enchaînements de traitements internes à la couche.

Dans cet objectif, ce chapitre constitue une suite logique du chapitre II dans lequel nous avons présenté les différentes fonctions globales d'une couche. Nous nous proposons d'étudier ici une organisation de ces fonctions en sous-niveaux fonctionnels distincts.

Le modèle introduit devrait permettre en premier lieu d'accroître la clarté de la description fonctionnelle d'une couche et en second lieu de faciliter sa réalisation. On pourra se reporter à l'exemple donné au chapitre précédent : c'est une ébauche de cette approche.

Dans un souci de clarté et pour ne pas mélanger les problèmes, nous avons volontairement limité l'étude au cas d'une couche enveloppant un seul service.

## 2 - STRUCTURATION D'UNE COUCHE (n) EN SOUS-COUCHES

Nous avons vu que la structuration en couches d'une architecture de communication consiste d'abord à décomposer la tâche de communication en des fonctions distinctes, hiérarchisées et ensuite à les abstraire successivement pour construire des niveaux de services.

Une conséquence importante de cette architecture est que les messages échangés entre deux entités homologues de même niveau sont confiés au niveau sous-jacent qui les considère comme une suite de caractères sans se préoccuper de leur signification. Il en est de même pour le niveau suivant, et ainsi de suite; chaque niveau ne prend en charge qu'une partie des fonctions de la communication.

Cette technique de structuration pourrait être utilisée pour organiser la structure interne d'une couche. En effet, les transformations successives des données au sein d'une couche qui ont été décrites dans le chapitre II (IDU(n) -- SDU(n) -- PDU(n) -- SDU(n-1) -- IDU(n-1)) reflète cette tendance de traitement hiérarchisé. C'est pour cette raison que nous introduisons la notion de sous-couche dans l'étude de la structuration d'une couche.

Considérons une couche (n) constituée d'un certain nombre d'entités serveurs (n), les interactions avec les couches (n+1) et (n-1) ayant lieu en des point d'accès (n) et (n-1).

En premier lieu, les fonctions relatives aux interfaces de la couche peuvent être mises dans les deux sous-couches suivantes :

- Une sous-couche d'interface (n)/(n+1) : qui se situe à l'interface (n)/(n+1) et assure les fonctions de gestion des interactions (émission, réception des primitives), d'adaptation du format des SDU à celui des IDU (et inversement), et de contrôle de flux à l'interface.

- Une sous-couche d'interface (n)/(n-1) : qui assure les mêmes fonctions que la sous-couche précédente mais cette fois vis à vis de la couche (n-1).

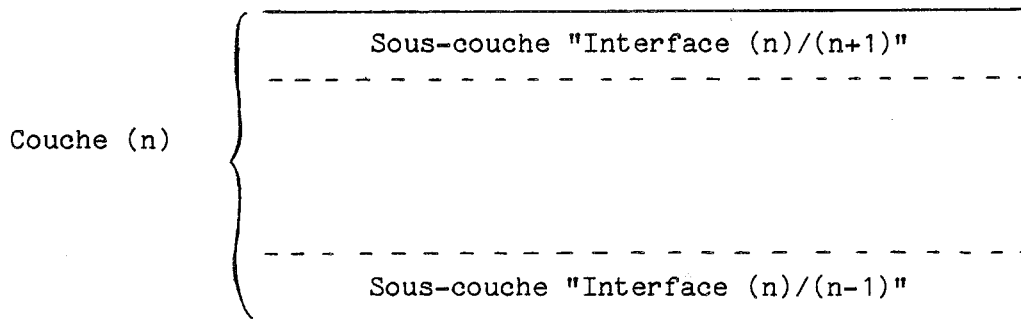


Figure 1 : Deux sous-couches d'interface

Dans le souci d'arriver à un modèle le plus général possible, nous allons étudier dans ce qui suit la structuration en sous-couches des autres fonctions de la couche. Nous passons en revue

les 4 cas de figure représentant les différents types de mise en correspondance des supports de communication entre deux services (n) et (n-1). On verra que le cas le plus complexe des quatre (services (n) et (n-1) tous les deux orientés connexion) est aussi le plus général.

#### 2.1 - Services (n) et (n-1) de type sans connexion

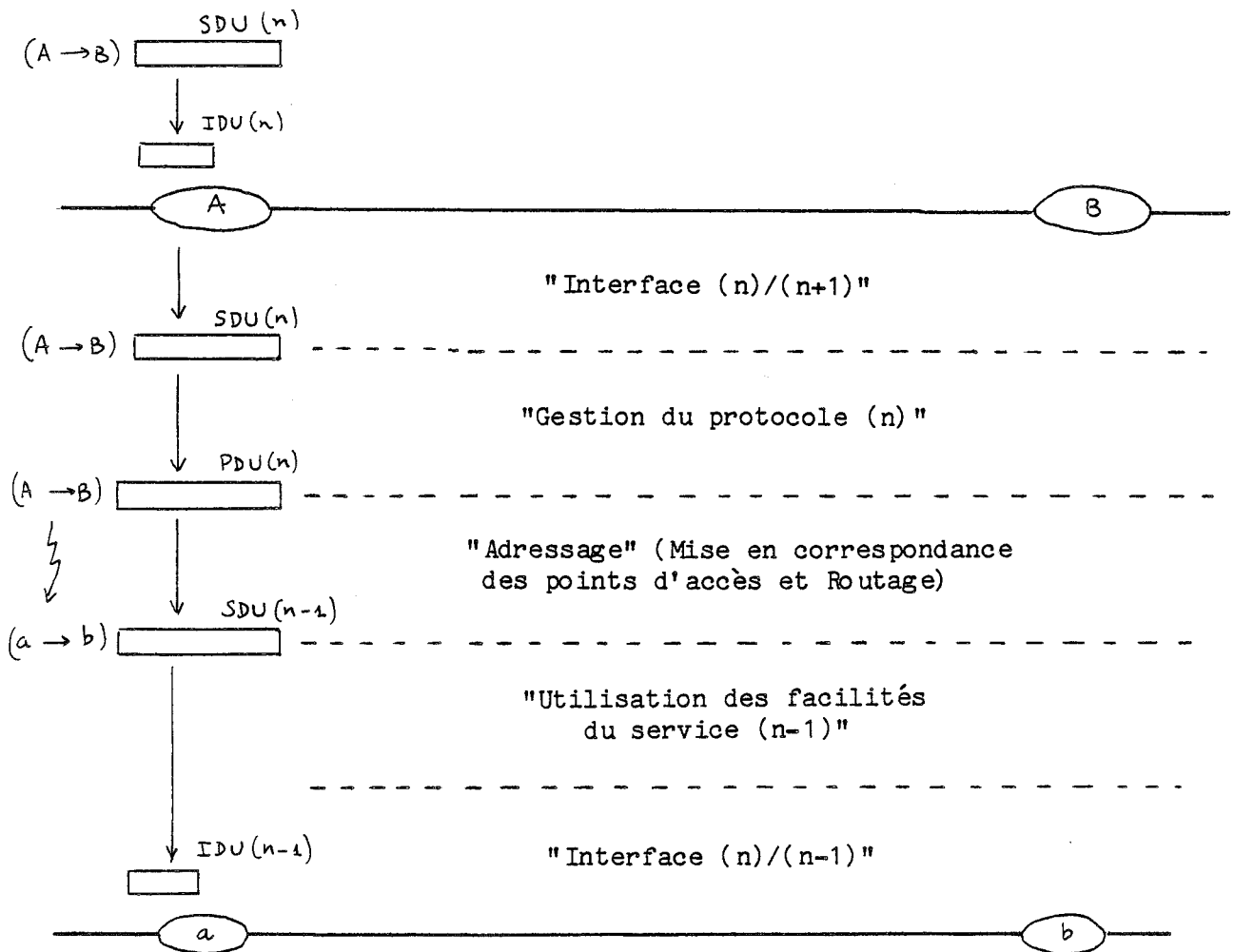
Mises à part les sous-couches d'interface, il apparaît trois niveaux de traitement importants au sein de la couche (n) :

- Une sous-couche de gestion du protocole (n) : caractérisée par un protocole (n) qui gère les fonctions de communications de la couche. La réception, l'émission et le traitement des PDU sont assurés par ce niveau. En ce qui concerne l'acheminement des PDU à travers le service (n-1), le travail est confié aux deux sous-couches suivantes :

- Une sous-couche d'adressage : qui code et décode des informations d'adressage (adresses (n) et (n-1)) accompagnant les PDU. C'est précisément à ce niveau qu'on effectue la correspondance des points d'accès et le routage. Les adresses (n-1) déterminées par ces fonctions indiquent le cheminement des informations à travers le service (n-1).

- Une sous-couche d'utilisation des facilités du service (n-1) : qui surveille l'utilisation des facilités du service (n-1) pour le compte de la couche (n) (se conformer à des règles de déroulement de ces facilités, donner des décisions sur les indications reçues du service (n-1),...). Par rapport à la sous-couche d'interface (n)/(n-1) qui gère les primitives isolées, cette sous-couche prend en compte l'enchaînement de celles-ci. Elle joue alors le rôle de gestionnaire des facilités du service inférieur (n-1).

Le schéma de la figure 2 montre un exemple de transfert de données avec la formation successive des différentes unités de données dans la couche.



Notation :

$(A \rightarrow B)$  : le message va de A à B

$(a \rightarrow b)$  : le message va de a à b

A,B sont des adresses (n); a,b sont des adresses (n-1)

Figure 2 : Décomposition en sous-couche - Services (n) et (n-1) de type sans connexion

## 2.2 - Service (n) orienté connexion, service (n-1) de type sans connexion

Dans ce cas, la couche (n) gère plusieurs flots de données indépendants appartenant à des connexions (n) existantes ; le service (n-1) n'offre pas de connexions. Cependant, pour optimiser le nombre de transferts de données à travers le service (n-1), les PDU issues de connexions (n) différentes et ayant même destination peuvent être concaténées dans une SDU (n-1) à l'émission et séparées ensuite à la réception.

Une nouvelle sous-couche assurant ces opérations de concaténation et de séparation est introduite. Elle est sous-jacente à la sous-couche d'adressage puisque c'est celle-ci qui détermine les chemins à travers le service (n-1) des différentes PDU (n). Cette sous-couche sera appelée "mise en correspondance des supports de communication" (supports de communication offerts par les services (n) et (n-1)).

Notons que pour chacune des connexions (n), le protocole (n) contrôle les opérations d'établissement, de libération de connexion, de transfert de données sur celle-ci, etc... à l'intérieur de la sous-couche de gestion du protocole (n). L'envoi des données du protocole au service (n-1) est assuré par les 4 sous-couches inférieures : adressage, mise en correspondance des supports de communication, utilisation des facilités du service (n-1) et interface (n)/(n-1).

Le schéma de la figure 3 donne un exemple de la formation successive des unités de données dans ce type de couche.

### Notation :

(A → B,1) : le message va de A à B sur la connexion 1

(A → B,2) : le message va de A à B sur la connexion 2

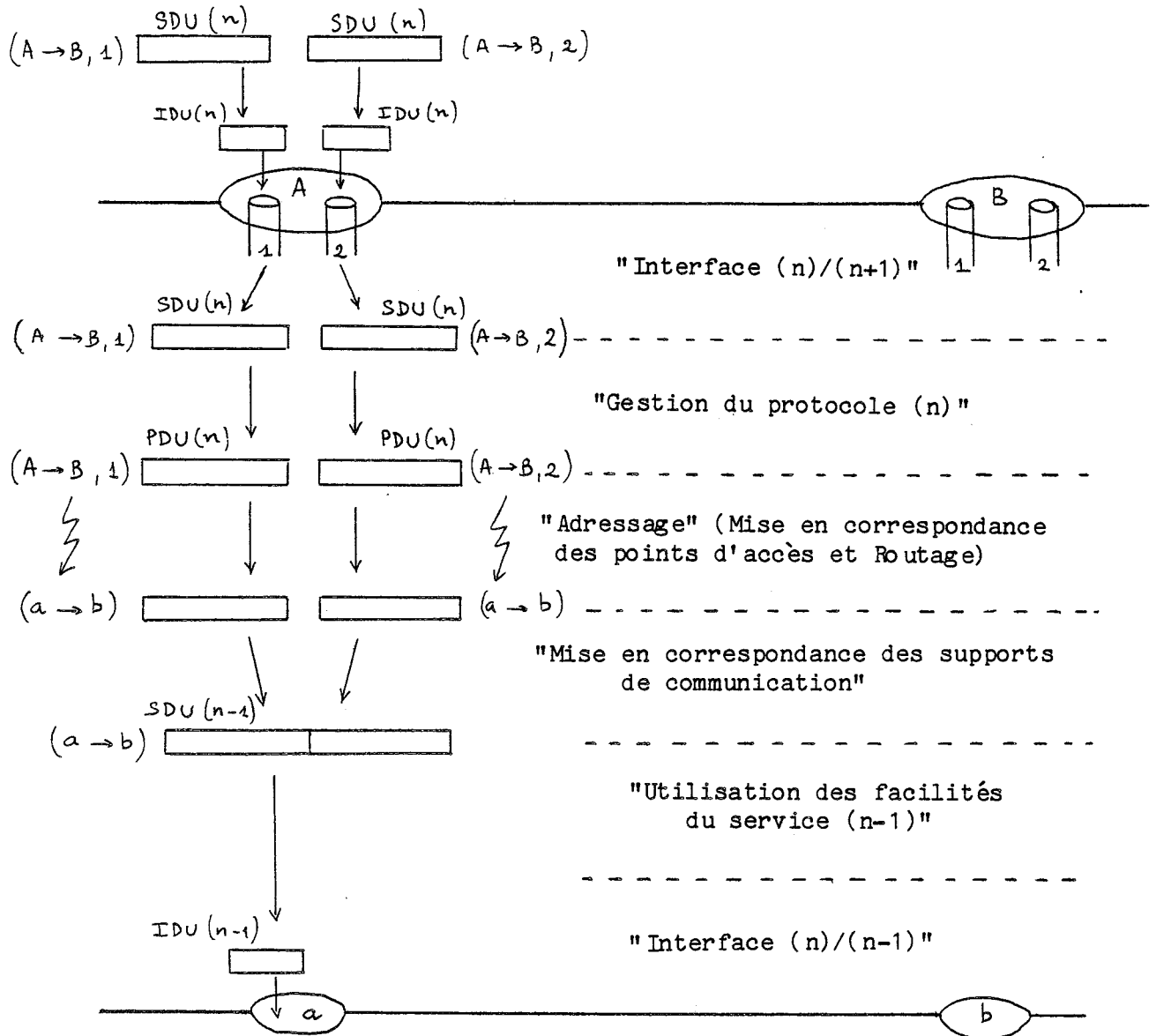


Figure 3 : Décomposition en sous-couches.

Service (n) orienté connexion,

Service (n-1) de type sans connexion

2.3 - Service (n) de type sans connexion, service (n-1) orienté connexion

La couche (n) utilise dans ce cas des connexions (n-1) pour l'acheminement des PDU (n). Pour augmenter la disponibilité du service

et la vitesse de transfert de données, on peut envisager une configuration d'éclatement : plusieurs connexions (n-1) prennent en charge l'acheminement des données entre deux entités (n).

Par rapport au cas précédent (2.2), la sous-couche de mise en correspondance des supports de communication gère l'éclatement et la sous-couche d'utilisation des facilités du service (n-1) doit prendre en charge le contrôle des connexions (n-1) existantes.

Le schéma de la figure 4 donne un exemple d'une configuration d'éclatement. Une PDU (n) est dupliquée pour former deux SDU(n-1) identiques partant séparément sur deux connexions (n-1) vers une même destination.

#### 2.4 - Services (n) et (n-1) orientés connexion

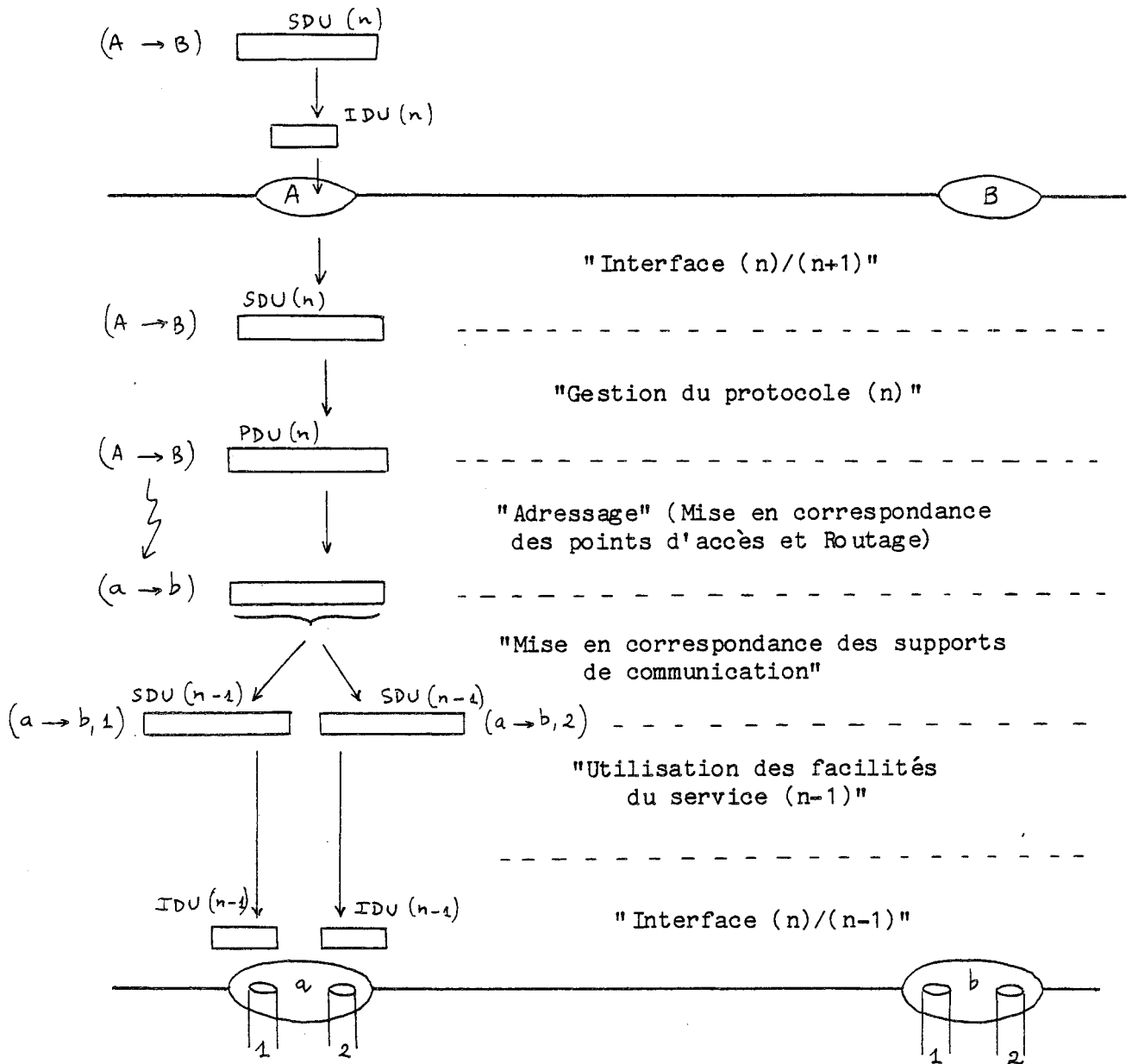
La couche (n) gère dans ce cas des connexions (n) et contrôle l'utilisation de connexions (n-1). Elle assure l'aiguillage des flots de données provenant des connexions (n) vers des connexions (n-1) et inversement.

La sous-couche de "mise en correspondance des supports de communication" peut être considérée comme une superposition des deux cas de figure précédents (2.2 et 2.3). On peut trouver différentes configurations : multiplexage, éclatement ou combinaison des deux.

Le schéma de la figure 5 donne un exemple de la formation des différents flots de données dans une configuration qui utilise à la fois le multiplexage et l'éclatement. Dans cet exemple, des fonctions de concaténation et de duplication sont réalisées dans la sous-couche de "mise en correspondance des supports de communication".

Les 3 premières configurations de couche (2.1,2.2,2.3) peuvent être considérées comme des cas particuliers du dernier modèle présenté. C'est pourquoi, la suite de l'étude portera essentiellement sur ce modèle.





Notation :

(a → b, 1) : le message va de a à b sur la connexion 1

(a → b, 2) : le message va de a à b sur la connexion 2

Figure 4 : Décomposition en sous-couches.

Service (n) de type sans connexion

Service (n-1) orienté connexion,

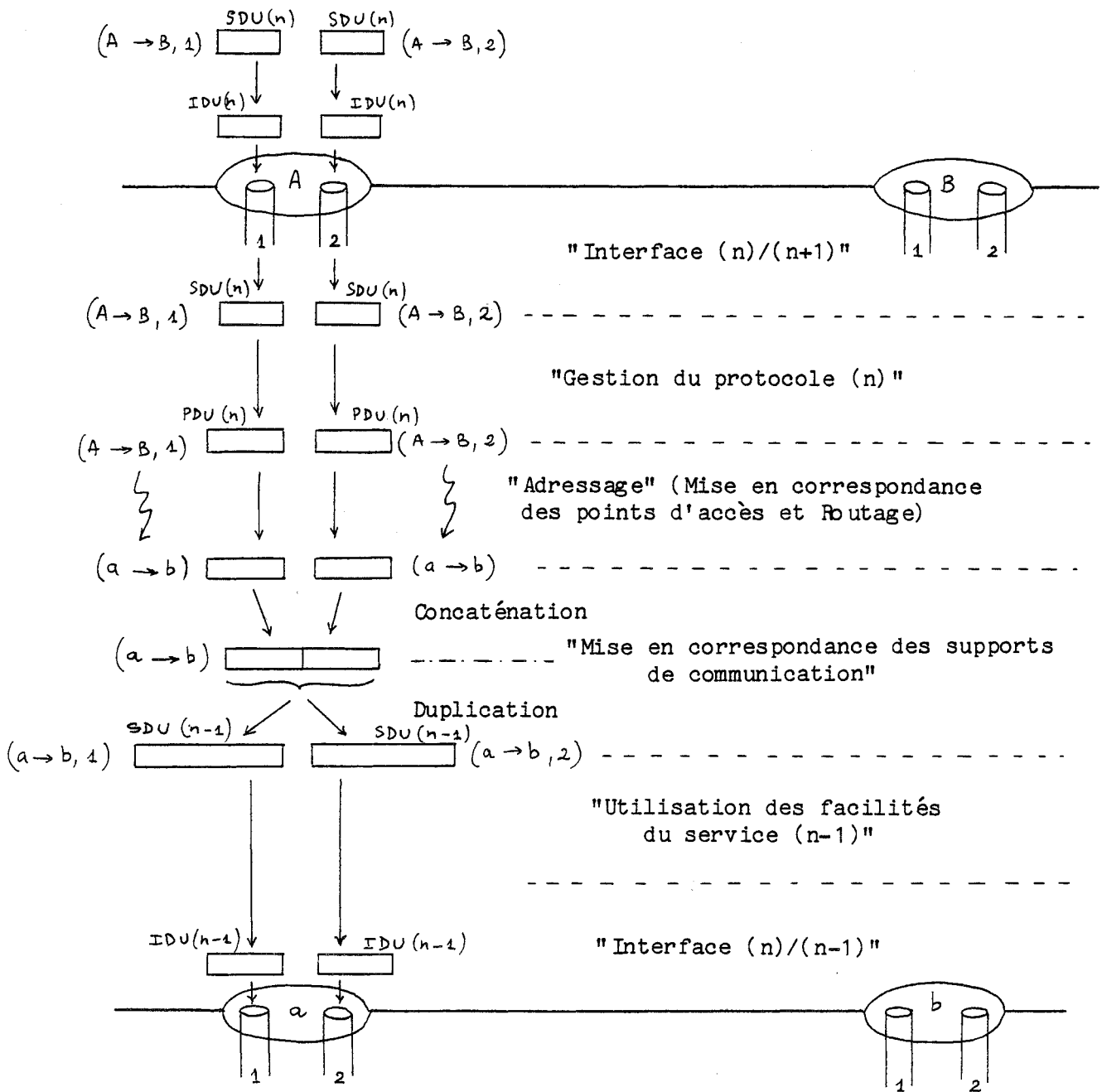


Figure 5 : Décomposition en sous-couches.

Services (n) et (n-1) orientés connexion,

### 3 - CONCEPT DE NOEUD DE CORRESPONDANCE

En toute généralité, une couche (n) doit pouvoir gérer les différents types de correspondance des connexions (figure 6).

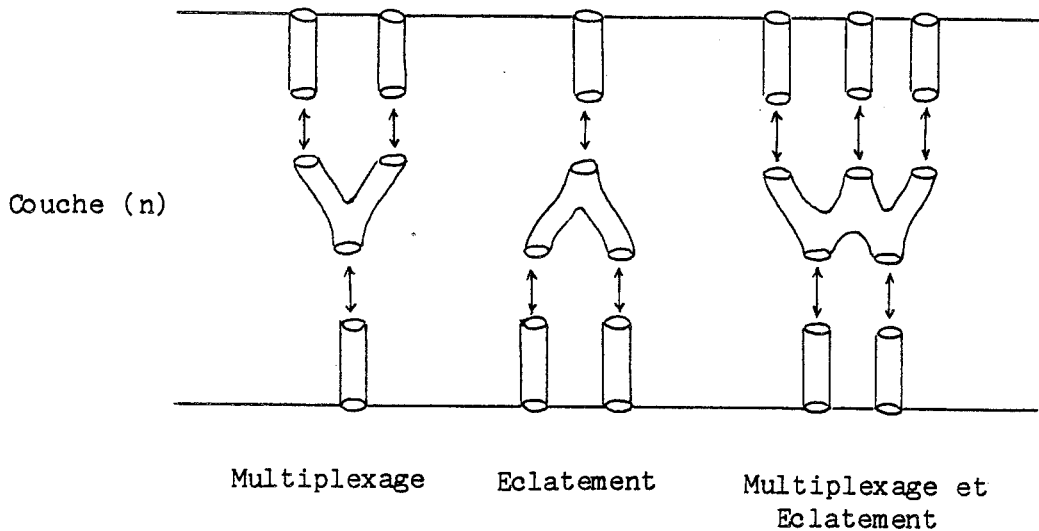


Figure 6 : Différents types de correspondance des connexions

Il faut dans ce cas établir des associations entre les connexion (n) et (n-1) et les concrétiser. Pour cela, nous introduisons le concept de noeud de correspondance dont voici les principales propriétés:

a/ Un noeud de correspondance est une association logique entre un certain nombre de connexions (n) et un certain nombre de connexions (n-1). Cette association peut représenter un multiplexage, un éclatement ou une combinaison des deux. Concrètement, le noeud de correspondance met en place une certaine **politique d'utilisation** du service (n-1) au profit des connexions (n).

b/ Des **caractéristiques** attribuées à un noeud lors de sa création (nombre maximal de connexions (n) multiplexées, nombre maximal de

connexions (n-1) dans une configuration d'éclatement, utilisation des fonctions spécifiques telles que la concaténation, la duplication ...) déterminent son fonctionnement dans le temps, c'est à dire la façon de transformer les flots de données (n) en ceux (n-1) et inversement.

c/ La création d'un noeud suit celle des connexions (n-1) et impose une politique d'utilisation à l'intérieur d'une couche (n). L'établissement d'un noeud (une configuration de correspondance) résulte d'un **accord** entre les entités (n). Il y a au moins deux façons de le mettre en oeuvre :

. Soit définir un protocole d'établissement de la correspondance pour obtenir cet accord par le dialogue. Ce protocole est alors géré par la sous-couche de mise en correspondance des connexions.

. Soit imposer une fois pour toutes des conventions statiques entre les entités (n) sur la manière d'utiliser une connexion (n-1) chaque fois qu'elle est créée ; un noeud de correspondance a alors des caractéristiques connues à l'avance. Cette solution simplifie la mise en oeuvre de la correspondance.

d/ Une connexion (n) créée possède un identificateur local ( le CEP-id : connection end point identifier). Se référer à cet identificateur est équivalent à désigner une liaison entre deux adresses (n). De même, un noeud de correspondance possède un **identificateur** local permettant aux entités supérieures de désigner une liaison entre deux adresses (n-1).

Le couple "identificateur d'une connexion (n) - identificateur d'un noeud de correspondance" représente la liaison entre une connexion (n) et une (ou plusieurs) connexion (n-1). Elle est effectuée lors de l'établissement d'une connexion (n) et permet d'éviter dans la phase de transfert de données les opérations d'adressage habituelles qui obligent à explorer des tables (mise en correspondance des points d'accès, routage).

\*

\*

\*

En résumé, dans ce chapitre nous avons mis en évidence un certain nombre de niveaux de traitement au sein d'une couche et introduit le concept de noeud de correspondance. Ce modèle reste incomplet et ne permet pas d'explicitier simplement les problèmes suivants :

- Comment exprimer l'indépendance et le parallélisme du fonctionnement des connexions (n), des noeuds de correspondance et des connexions (n-1) dans une couche (n) ?

- Comment relier les "objets" que nous venons de citer pour former des chemins de données à l'intérieur de la couche (n) ?

- Comment situer par rapport à d'autres fonctions, des fonctions d' "administration locales" qui contrôlent l'existence de ces "objets" et qui les individualisent en leur attribuant dynamiquement des caractéristiques à la création : les paramètres du protocole (n) relatifs à une connexion (n), les paramètres de correspondance relatifs à un noeud de correspondance, les paramètres de qualité relatifs à une connexion (n-1) ?

Ces problèmes relèvent de l'aspect dynamique du fonctionnement d'une couche que nous allons aborder dans le chapitre suivant.

## CHAPITRE V

### MODELE DYNAMIQUE D'UNE COUCHE (n)

1 - PRELIMINAIRE

2 - CONCEPT D'ACTIVITE

3 - MODELE DYNAMIQUE D'UNE ENTITE (n)

3.1 - Considérations initiales

3.2 - Description du modèle dynamique

3.3 - Fonctionnement du modèle

4 - QUELQUES CAS PARTICULIERS DU MODELE

4.1 - Service (n) orienté connexion, service (n-1) de type sans connexion

4.2 - Service (n) de type sans connexion, service (n-1) orienté connexion

5 - RESUME DES CARACTERISTIQUES DU MODELE



## 1 - PRELIMINAIRE

Les fonctions d'une couche d'une architecture de communication sont distribuées dans de différents sous-niveaux de traitement grâce à la structuration en sous-couches (cf chap IV). Par ailleurs, on a remarqué que ces fonctions peuvent être exécutées dynamiquement par des tâches (cf chap. III) et qu'il existe des tâches spécifiques d'administration internes à l'entité (n).

On se propose d'étudier dans ce chapitre un modèle dynamique du fonctionnement d'une couche qui mette en évidence les différents aspects venant d'être cités grâce à :

- l'utilisation du concept d' "activité" comme outil de représentation d'une certaine tâche de traitement dans une entité (n),

- l'introduction des activités d' "administration locale" à l'intérieur d'une entité (n). On trouve une représentation semblable dans un modèle introduit par T.F. PIATKOWSKI /E1/.

Par la suite, on pourra s'inspirer directement de ce modèle dans le cadre de la réalisation d'une couche.

Dans ce qui suit, nous rappelons d'abord quelques aspects essentiels du concept d'activité. Ensuite, nous introduisons le modèle dynamique d'une couche et étudions quelques scénarios de fonctionnement. L'application du modèle à des cas particuliers est présentée dans le paragraphe 4. Enfin, les caractéristiques du modèle sont résumées à la fin du chapitre.

## 2 - CONCEPT D'ACTIVITE

Le concept d'activité introduit dans /E2/ recouvre la notion classique de processus /E3/ et permet de décrire l'aspect dynamique des systèmes informatiques.



Une activité est la représentation du déroulement dans le temps des fonctions d'une certaine entité fonctionnelle (\*). Chaque activité a son propre **environnement d'exécution** (appelé aussi "espace d'exécution") et sa propre échelle de temps d'exécution.

Concrètement, une activité peut être considérée comme une représentation du déroulement d'une procédure ou d'autres concepts informatiques désignant un ensemble de fonctions exécutables.

Grâce à ce concept, on peut représenter plusieurs exemplaires d'une même entité fonctionnelle par des activités. Celles-ci fonctionnent alors de façon indépendante (figure 1).

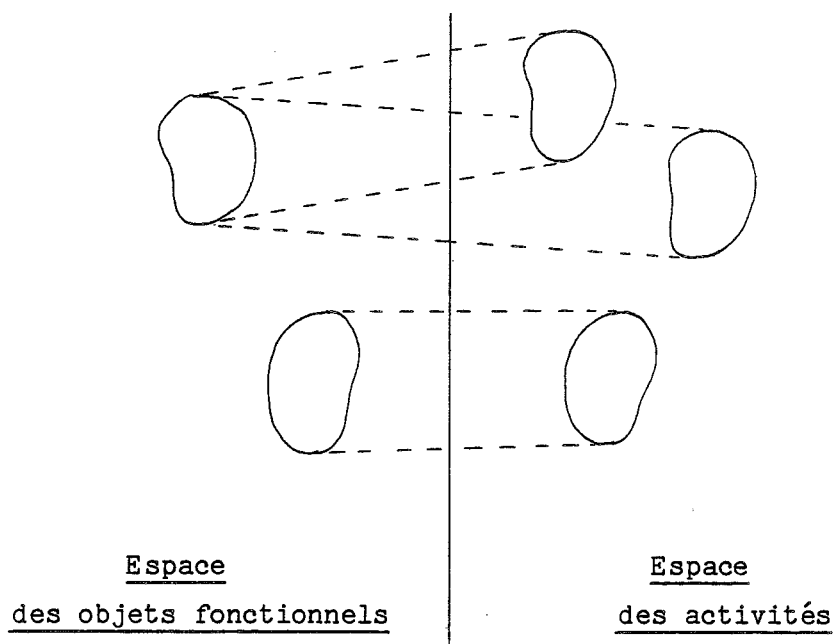


Figure 1 : Activités

---

\* Dans /E2/ une activité est associée à un "objet" ou un "réseaux d'objets". Dans notre contexte, les entités fonctionnelles peuvent être vues comme des objets (au sens de /E2/).

Exemple : Une activité "protocole (n)" représente le déroulement d'un protocole relatif à une connexion (n). On peut avoir autant d'activités "protocole (n)" que de connexions (n) qui existent à un moment donné. L'entité fonctionnelle "protocole (n)" est le "modèle" de ces activités. Chaque activité a son propre environnement d'exécution et sa propre échelle de temps d'exécution.

Les activités peuvent être créées et détruites dynamiquement. Lorsqu'elle existe, une activité peut être **active** ou **inactive** (l'activité se déroule ou est bloquée). Une activité a donc des états différents au cours du temps : inexistante, existante, active, inactive. Une activité peut agir sur une autre pour modifier son état à l'aide des opérations suivantes :

- Opérations d'existence : création, destruction d'une activité.

- Opération de synchronisation : suspension (rendre inactive une activité) , activation (rendre active une activité).

Les activités ont généralement besoin d'échanger des informations entre elles pour collaborer à un travail commun. Pour cela, il leur faut un moyen pour recueillir les messages, que l'on appellera un **objet de communication** . Il faut également définir des opérations que l'on peut faire sur cet objet. Par exemple, les deux opérations habituelles sur un objet de communication "boîte aux lettres" sont : "déposer un message" et "retirer un message".

Nous avons présenté ci-dessus les aspects essentiels du concept d'activité. Pour les besoins de description du fonctionnement dynamique d'une couche, nous introduisons une notion supplémentaire : celle d' **environnement de communication** d'une activité. Ce sont toutes les activités qui communiquent avec l'activité considérée. L'environnement de communication d'une activité peut changer au cours du temps grâce à la création et la destruction d' "objets de communication" permettant la liaison ou la rupture de communication entre activités.

### 3 - MODELE DYNAMIQUE D'UNE COUCHE (n)

A l'aide du concept d'activité, nous pouvons faire apparaître dans le modèle dynamique d'une couche les aspects suivants :

- l'indépendance et le parallélisme du fonctionnement des connexions (n) : nous avons vu que le protocole est en particulier spécifique à une connexion (n);

- l'indépendance et le parallélisme du fonctionnement des noeuds de correspondance : nous avons vu que plusieurs configurations de correspondance peuvent exister en même temps;

- l'indépendance et le parallélisme de l'accès à des connexions (n-1) : les connexions (n-1) dont dispose la couche (n) peuvent être de qualités différentes;

- la fonction d'administration locale qui supervise l'ensemble des activités existant au sein d'une entité (n) et assurant le fonctionnement des connexions (n), des noeuds de correspondance, et de l'accès aux connexions (n-1).

#### 3.1 - Considérations initiales

Dans les différentes sous-couches exposées dans le chapitre précédent, nous allons introduire les activités suivantes :

- "connexion (n)"
- "noeud de correspondance"
- "connexion (n-1)"
- "administration locale de l'entité (n)"

. Sous-couches d'interface (n)/(n+1) et de gestion du protocole(n)

Les fonctions suivantes :

- Contrôle de flux à l'interface
- Formation des IDU (n), des SDU (n) et des PDU (n)
- Protocole (n)

sont individualisées par connexion (n). C'est pourquoi, nous associons à chacune des connexions (n) existant dans l'entité (n) une activité qui déroule les fonctions précédentes et qui sera appelée par la suite l' **activité "connexion (n)"** .

Notons qu'on peut tirer parti du parallélisme possible entre deux fonctions "entrée de message" et "sortie de message" /E4/ pour décomposer plus finement l'activité "connexion (n)". Nous restons, pour cette étude, au niveau global d'une connexion.

. Sous-couche d'adressage.

La détermination des adresses (n-1) par les fonctions de mise en correspondance des points d'accès et de routage est généralement faite une seule fois lors de la création d'une connexion (n). Par la suite, le chemin de données est identifié par une liaison locale "connexion (n) - noeud de correspondance" (voir IV.3). Ces opérations d'adressage font donc partie de la préparation de l'environnement de communication pour une activité "connexion (n)". Elles seront réalisées par une activité spécifique que nous appellerons **activité d'administration locale de l'entité (n)** .

. Sous-couche de mise en correspondance des supports de communication.

Un noeud de correspondance représente une configuration de raccordement des connexions (n) aux connexions (n-1). Il peut en exister plusieurs dans une couche. C'est pourquoi, à chaque noeud existant nous associons une activité dite **activité "noeud de correspondance"** .

. Sous-couches d'utilisation des facilités du service (n-1) et d'interface (n)/(n-1).

Les fonctions suivantes de ces deux sous-couches :

- Contrôle d'utilisation des facilités du service (n-1)
- Contrôle de flux à l'interface
- Formation des SDU (n-1), IDU (n-1)

sont relatives à chacune des connexions (n-1) dont dispose l'entité (n). Par conséquent, on introduira une activité par connexion (n-1) exécutant les fonctions précédentes. Nous l'appellerons l' **activité "connexion (n-1)"** .

Outre les opérations d'adressage, l'activité d'administration locale est chargée de créer, détruire et synchroniser les activités rattachées aux "objets" ("connexion (n)", etc..) de l'entité (n). Nous détaillons le rôle de chaque activité dans ce qui suit.

3.2 - Description du modèle dynamique

A partir des considérations précédentes, nous proposons ici un modèle dynamique d'une entité (n) à trois niveaux d'activités (figure 2) :

- le niveau 1 : "connexions (n)"
- le niveau 2 : "noeuds de correspondance"
- le niveau 3 : "connexions (n-1)"

Pour faciliter la description, l'activité d'administration de l'entité (n) est décomposée en 3 activités dont chacune administre un niveau.

Par rapport à la décomposition en sous-couches qui a un caractère statique, cette décomposition repose sur l'organisation des fonctions de la couche dans des "objets" dynamiques. C'est pourquoi, plusieurs sous-couches peuvent se trouver dans un niveau d'activités.

Notation :

- A,B,C : activités d'administration locale
- a1,a2,a3,a4,a5 : activités "connexion (n)"
- b1,b2 : activités "noeud de correspondance"
- c1,c2,c3 : activités "connexion (n-1)"

- ↔ : relation de communication
- : relation de création, destruction et synchronisation

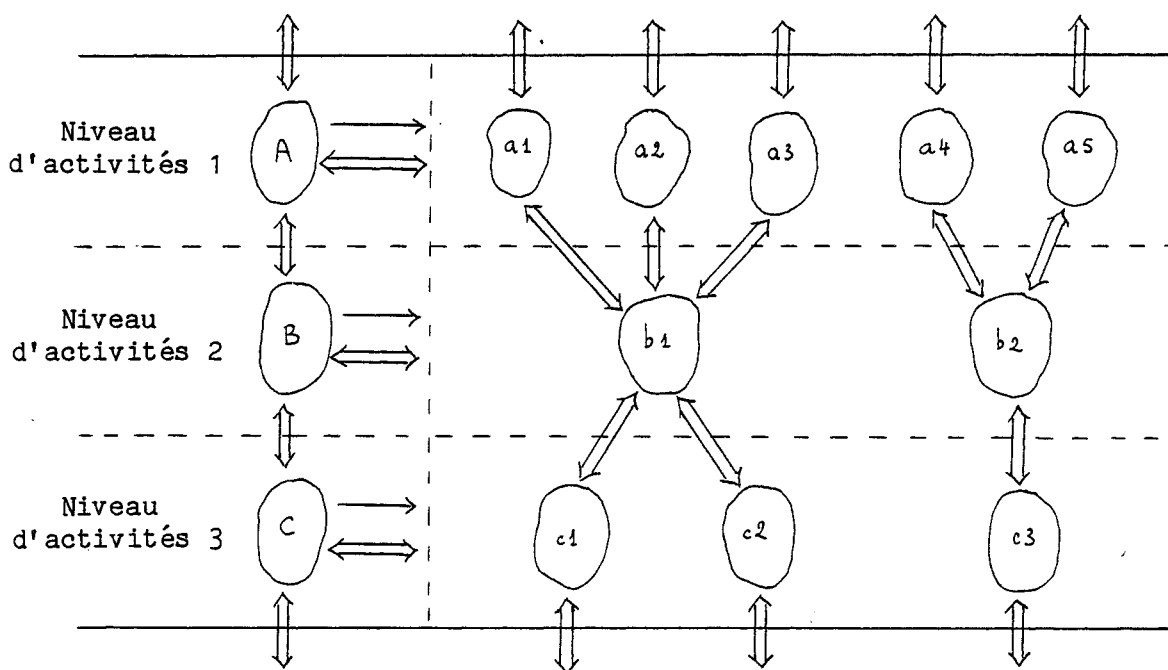


Figure 2 : Modèle dynamique d'une entité (n)

Les flèches horizontales représentent les interactions d'une activité d'administration locale avec toutes les autres activités de son niveau.

### 3.2.1 - Activité "connexion (n)" (a1,a2,a3,...)

Il en existe une par connexion (n) ; elle assure d'une part le déroulement du protocole (n) (établissement de la connexion, transfert de données, libération de la connexion,...), d'autre part la construction des unités de données (IDU (n), SDU (n), PDU (n)) des flots d'informations correspondants. Elle a pour partenaire de dialogue, une activité "connexion (n)" distante qui gère l'autre extrémité de la connexion.

Toute activité "connexion (n)" est créée par l'activité "administration des connexions (n)" (entité A de la figure 2). Après sa création, elle interagit directement avec l'entité supérieure (n+1) pour dérouler les facilités de service que demande celle-ci.

La fin d'une communication entraîne la destruction par l'activité A l'activité "connexion (n)" correspondante.

Toute activité "connexion (n)" doit avoir un environnement de communication comportant les activités suivantes (figure 3) :

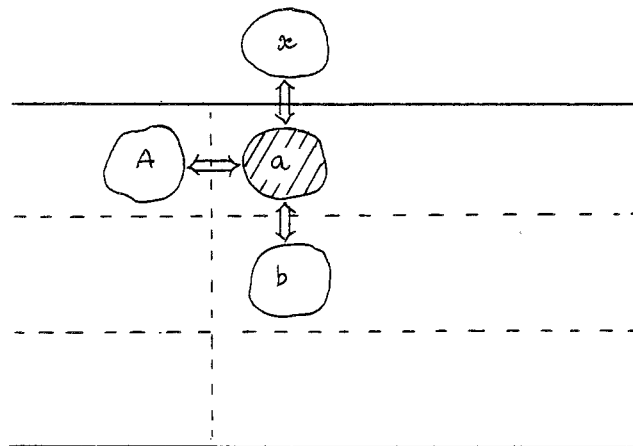


Figure 3 : Environnement de communication  
d'une activité "connexion (n)"

- une activité (n+1) de la couche supérieure à laquelle elle envoie des données reçues sur la connexion (n),
- l'activité "administration des connexions (n)" à laquelle elle envoie des événements de fin de fonctionnement ou d'anomalies,

- une activité "noeud de correspondance" à laquelle elle envoie des PDU (n) produites par le protocole.

### 3.2.2 - Activité "administration des connexions (n)" (entité A de la figure 2)

Elle contrôle l'existence des activités "connexion (n)" (par les opérations de création, destruction), les synchronise avec les autres activités des niveaux d'activités 2 et 3 (par des opérations d'activation, suspension).

Parmi les requêtes provenant de la couche supérieure, elle surveille l'arrivée de celles demandant l'établissement d'une connexion (n). Lorsqu'elle en reçoit une, après en avoir contrôlé la validité elle procède, si la charge en cours le permet (par exemple le nombre des connexions (n) existantes n'atteint pas encore la valeur limite), à la création d'une activité "connexion (n)" : d'une part elle effectue le choix des paramètres de fonctionnement de l'activité à créer (classe, sous-ensemble de protocole,...), d'autre part elle interagit avec le niveau d'activités 2 pour obtenir un noeud de correspondance permettant l'accès à une ou plusieurs connexions (n-1). Pour cela, elle fournit à l'activité "administration des noeuds de correspondance" (entité B de la figure 2) des adresses (n) pour que celle-ci puisse effectuer les opérations d'adressage, et rechercher un noeud de correspondance approprié.

La création d'une activité "connexion (n)" ne sera effective que lorsqu'un noeud de correspondance aura été fourni.

### 3.2.3 - Activité "noeud de correspondance" (b1,b2,...)

Cette activité créée par l'activité "administration des noeuds de correspondance" rend service aux activités "connexion (n)" qui lui sont rattachées.

Elle assure éventuellement l'établissement d'une configuration de correspondance en accord avec une activité "noeud de correspondance" distante : négociation des paramètres tels que le degré de multiplexage, d'éclatement, l'utilisation des fonctions de concaténation, de séparation, etc...



Dans la phase de transfert de données, l'activité "noeud de correspondance" assure la transformation des flots de données (n) en flots de données (n-1) et inversement (PDU (n) ↔ SDU (n)), en leur appliquant éventuellement des fonctions négociées précédemment (concaténation, séparation,...).

Une activité "noeud de correspondance" donnée communique avec un certain nombre d'activités constituant son environnement de communication (figure 4) :

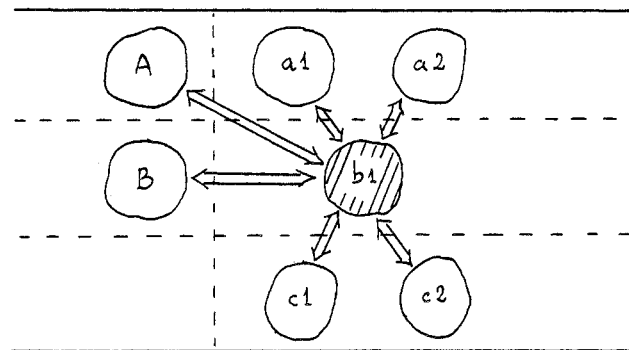


Figure 4 : Environnement de communication d'une activité "noeud de correspondance"

- une ou plusieurs activités "connexion (n)" auxquelles elle envoie des PDU (n) qui leur sont destinées,

- l'activité "administration des noeuds de correspondance" à laquelle elle envoie des événements de fin de fonctionnement ou d'anomalie,

- une ou plusieurs activités "connexion (n-1)" auxquelles elle envoie des SDU (n-1) qu'elle fabrique à partir des PDU (n).

- l'activité "administration des connexions (n)" à laquelle elle transmet des PDU d'établissement distant d'une connexion (n) qu'elle reçoit du niveau d'activités inférieur. Ici, on autorise une activité "noeud de correspondance", après sa création, à communiquer directement avec l'activité "administration des connexions (n)". Un autre trajet possible de ces PDU, qui apparaît plus strict sur le plan de

contrôles hiérarchisés, est de passer par l'activité "administration des noeuds de correspondance" avant d'atteindre l'activité "administration des connexions (n)" (voir figure 5). On constate que ce trajet est plus long et que l'activité "administration des noeuds de correspondance" ne fait à priori aucun traitement spécifique sur ces PDU (celles-ci ne sont significatives que pour le Protocole (n)). C'est pourquoi, nous gardons, pour la suite de notre étude, le schéma de communication de la figure 4.

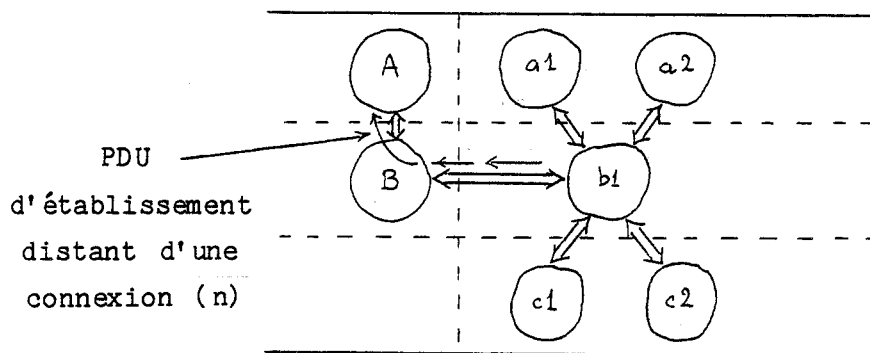


Figure 5

3.2.4 - Activité "administration des noeuds de correspondance"  
(entité B de la figure 2)

Cette activité contrôle l'ensemble des activités "noeud de correspondance" du niveau d'activités 2 qui gèrent le raccordement des connexions (n) aux connexions (n-1).

Elle assure en premier lieu les fonctions d'adressage de la couche (la mise en correspondance des points d'accès et le routage) pour traduire les adresses (n) fournies par le niveau d'activités 1 en des adresses (n-1) qui servent de paramètres dans les opérations suivantes :

- Rechercher un noeud de correspondance existant qui permet l'accès à une (ou plusieurs dans le cas d'éclatement) connexion (n-1)

associée aux adresses (n-1) et sur lequel on peut ajouter une nouvelle connexion (n) pour permettre la création d'une activité "connexion (n)" dans le niveau d'activités 1,

- Créer, si aucun noeud ne satisfait les conditions précédentes, un nouveau noeud de correspondance (donc une nouvelle activité de ce type) associé à ces adresses (n-1). En fonction du degré d'éclatement, il faut d'abord établir une ou plusieurs connexions (n-1). Pour cela, l'activité "administration des noeuds de correspondance" s'adresse au niveau d'activités 3 en fournissant comme paramètres les adresses (n-1).

Outre la création et la destruction des activités "noeud de correspondance", l'activité B est également chargée de les synchroniser avec d'autres activités des niveaux 1 et 3 (on trouvera un exemple de synchronisation d'activités dans §3.3.4)

### 3.2.5 - Activité "connexion (n-1)" (c1,c2,...)

Il en existe une par connexion (n-1) ; elle est créée par l'activité "administration des connexions (n-1)" (entité C de la figure 2). Initialement elle surveille l'établissement d'une connexion (n-1) en interagissant avec le service (n-1) et quand la connexion (n-1) aura été effectivement établie, elle y assurera le transfert de données en respectant les règles d'utilisation du service (n-1), notamment en ce qui concerne l'enchaînement des primitives et le contrôle de flux à l'interface (n-1). La construction des IDU (n-1) et des SDU (n-1) est également assurée par cette activité.

La figure 6 montre l'environnement de communication d'une activité "connexion (n-1)", l'activité "y" étant une activité de la couche (n-1). On autorise l'activité "connexion (n-1)", après sa création, à communiquer directement avec l'activité "administration des noeuds de correspondance". On verra un exemple de communication entre ces deux activités dans le paragraphe 3.3.

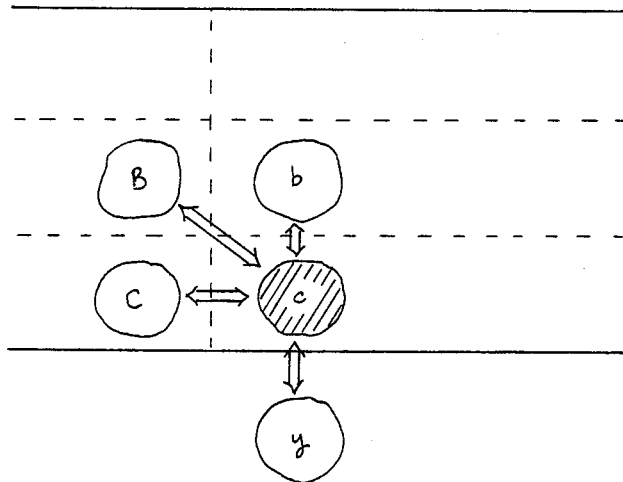


Figure 6 : Environnement de communication d'une activité "connexion (n-1)"

3.2.6 - Activité "administration des connexions (n-1)" (entité C de la figure 2)

Elle est responsable de l'existence des activités "connexion (n-1)" (c1,c2,c3,...). Elle applique éventuellement sur celles-ci des opérations de synchronisation (suspension, activation) afin de coordonner leur déroulement avec celui des activités des niveaux 1 et 2.

Parmi les indications du service (n-1), seules celles concernant l'établissement distant d'une connexion (n-1) ou affectant l'ensemble des connexions (n-1) (par exemple l'indication "service (n-1) en mauvais état") sont traitées par cette activité. Le traitement de ces événements entraînera des changements dans l'ensemble des activités "connexion (n-1)". Par exemple :

- une décision de reconfiguration d'une connexion (n-1) donne lieu au remplacement d'une activité "connexion (n-1) par une autre ;
- une indication d'établissement distant d'une connexion (n-1) a pour effet d'amener l'activité C à créer une nouvelle activité "connexion (n-1)" ;
- etc...

### 3.3 - Fonctionnement du modèle

Nous allons illustrer le fonctionnement du modèle en décrivant l'enchaînement des interactions entre les activités dans les situations suivantes :

- l'établissement d'une connexion (n)
- le transfert de données sur une connexion (n)
- la libération d'une connexion (n)
- la reconfiguration des connexions (n-1)

#### 3.3.1 - Etablissement d'une connexion (n)

L'établissement d'une connexion (n) entraîne la formation d'une chaîne d'activités sur les trois niveaux de la couche dans les deux systèmes en communication. Soient X et Y ces deux systèmes, nous allons montrer comment sont créées les différentes activités selon l'ordre chronologique des événements.

Supposons qu'aucune connexion (n) n'existe initialement entre les systèmes X et Y et que sur le système X, une entité (n+1) envoie une requête d'établissement d'une connexion (n) à l'entité (n).

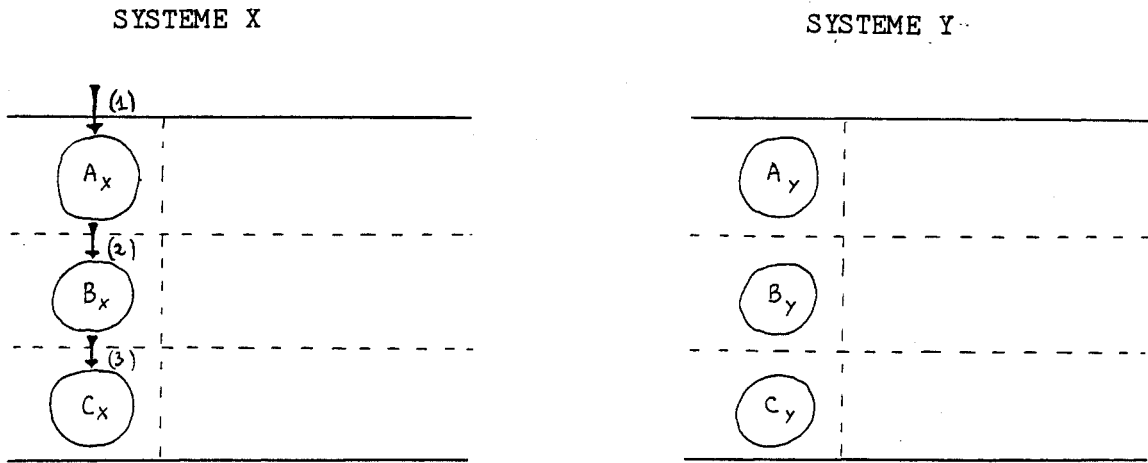
La suite de la description sera illustrée par les figures 7.1 à 7.9.

#### Notation :

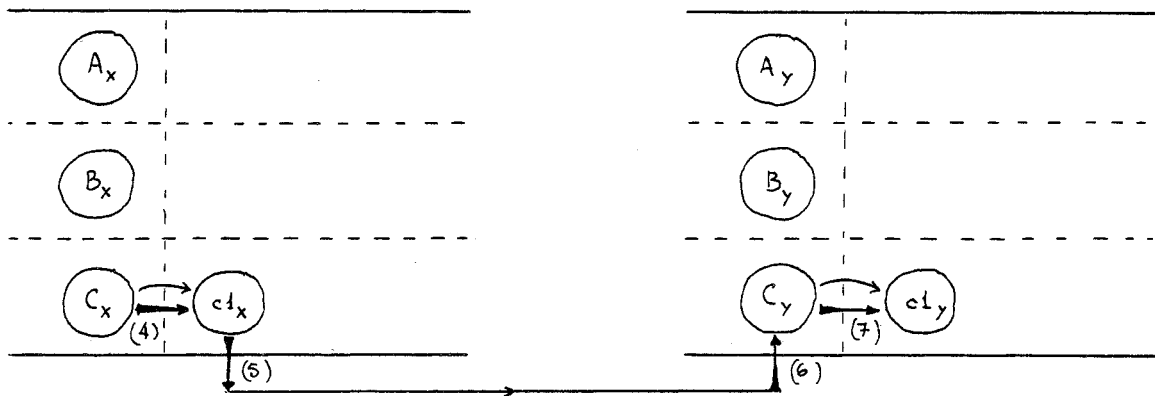
→ : création d'une activité

➡ : envoi d'un événement et des données

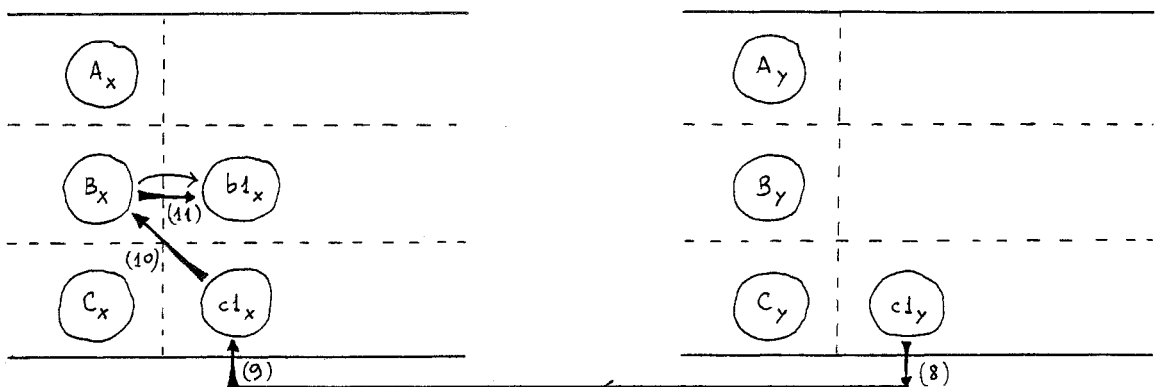
(n) : numéro d'ordre de l'événement dans le temps



**Fig 7.1 :** Demande d'établissement d'une connexion (n) sur le système X



**Fig 7.2 :** Création des activités "connexion (n)"



**Fig 7.3 :** Création d'une activité "noeud de correspondance" après l'établissement d'une connexion (n-1)

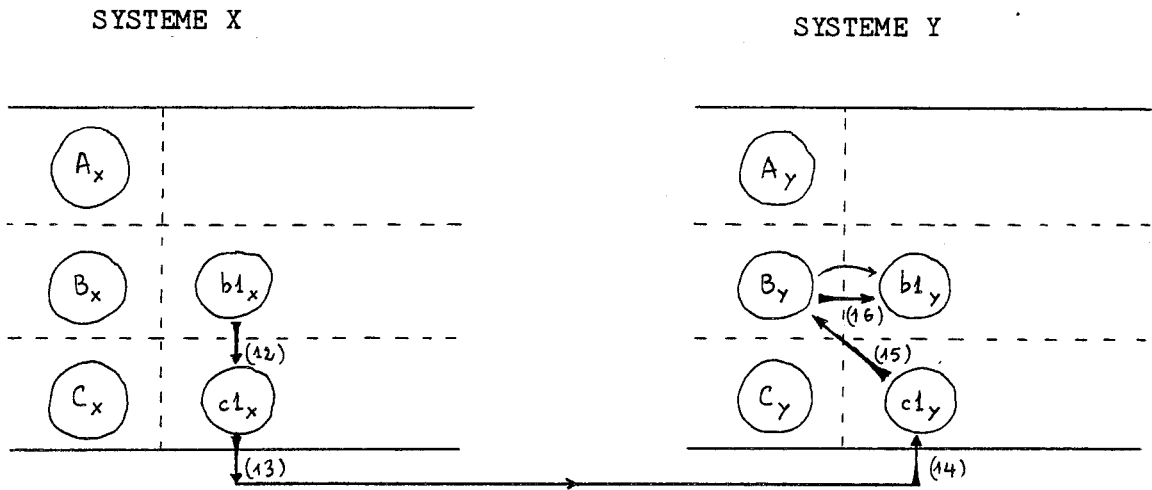


Fig 7.4 : Demande d'une configuration de correspondance

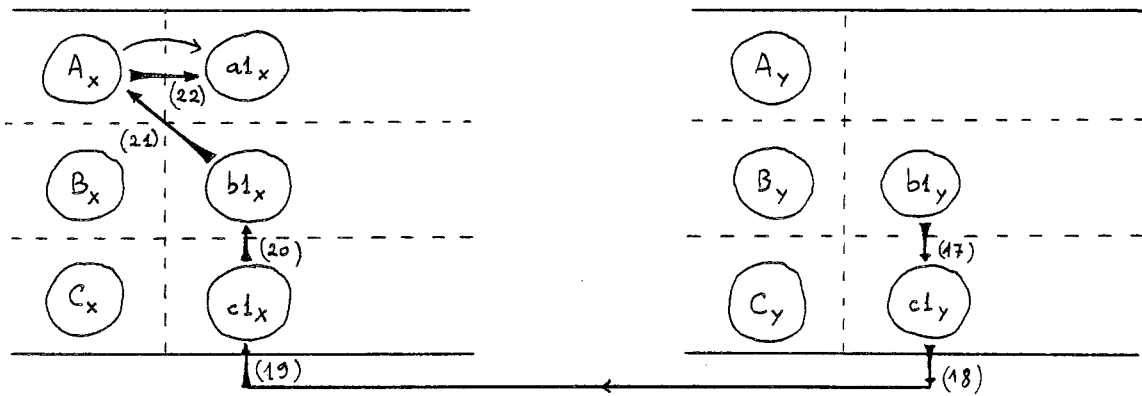


Fig 7.5 : Création d'une activité "connexion (n)" après l'établissement d'un noeud de correspondance

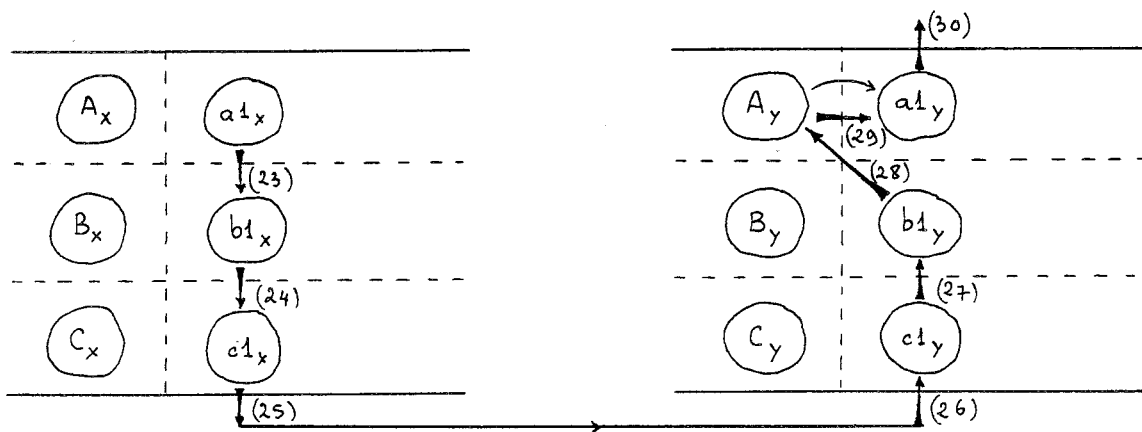
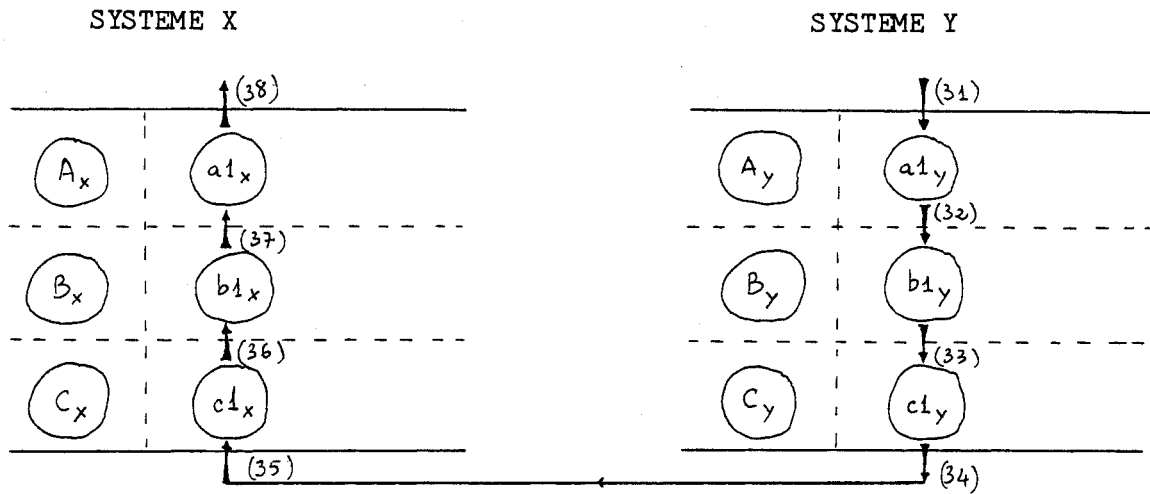
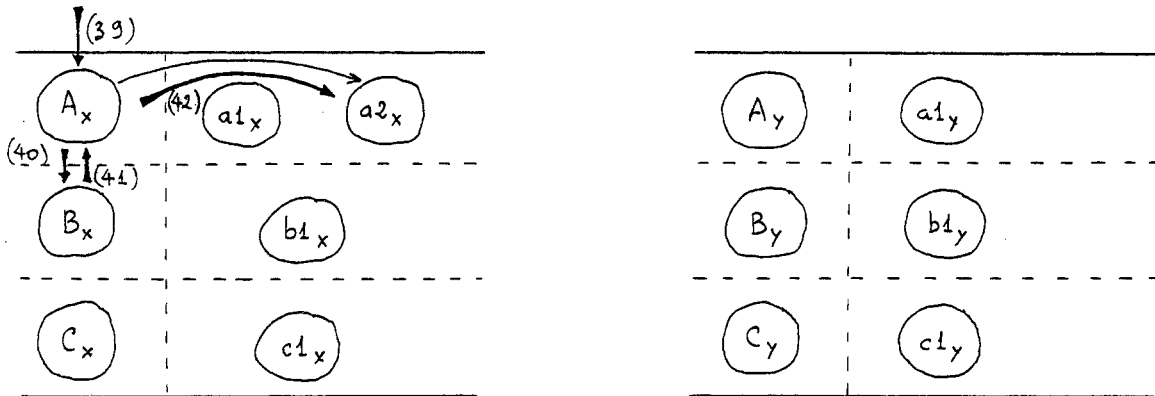


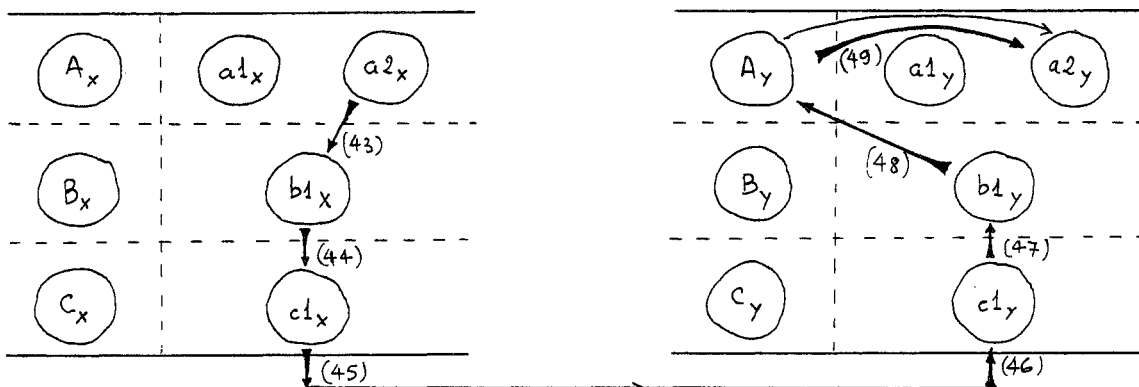
Fig 7.6 : Etablissement d'une connexion (n)



**Fig 7.7** : Réponse à la demande d'établissement d'une connexion (n)



**Fig 7.8** : Demande d'établissement d'une 2<sup>e</sup> connexion (n) sur le système X. Création d'une 2<sup>e</sup> activité "connexion (n)"



**Fig 7.9** : Création d'une 2<sup>e</sup> activité "connexion (n)" sur le système Y pour l'établissement de la 2<sup>e</sup> connexion (n)



L'enchaînement décrit par les figures 7.1 à 7.9 correspond aux différentes phases suivantes :

a/ Traitement de la requête d'établissement d'une connexion (n) sur le système X (figure 7.1)

L'activité "administration des connexions (n)" (entité A de X) traite la requête et procède à la création d'une première activité "connexion (n)" (on suppose que toutes les conditions locales sont favorables à cette création). Elle demande à l'activité "administration des noeuds de correspondance" (entité B de X) de lui fournir un noeud permettant le raccordement à une connexion (n-1) (on suppose que le noeud demandé est capable de supporter un multiplexage de p connexions (n) ( $p > 1$ ) et que son degré d'éclatement est égal à 1). L'activité "administration des noeuds de correspondance" doit alors obtenir une connexion (n-1) du niveau inférieur avant de créer le noeud demandé.

Les opérations d'adressage effectuées par l'activité B de X à partir des adresses (n) fournissent des adresses (n-1) qui seront ensuite transmises à l'activité "administration des connexions (n-1)" (entité C de X) pour permettre à celle-ci d'établir une connexion (n-1).

b/ Création sur les systèmes X et Y des activités "connexion (n-1)" qui surveillent l'établissement d'une connexion (n-1)  
(figure 7.2)

A la suite de l'étape a, l'activité C de X crée une activité "connexion (n-1)" notée c1 qui demande au service (n-1) l'établissement d'une connexion (n-1) en lui fournissant les adresses (n-1).

L'activité C du système Y sera avertie d'une demande d'établissement d'une connexion (n-1) par une indication issue du service (n-1). Elle créera alors une activité "connexion (n-1)" appelée c1 (sur le système Y) qui sera chargée de répondre à cette demande.

c/ Création des activités "noeud de correspondance" (figure 7.3, 7.4)

Si la réponse donnée par l'activité c1 de Y est positive, la confirmation reçue par c1 de X est répercutée à l'activité "administration des noeuds de correspondance" B de X. Celle-ci crée alors une activité "noeud de correspondance" notée b1 associée à l'activité c1 de X.

L'activité b1 de X commence par négocier une configuration de correspondance avec le système distant Y. Les informations de négociation envoyées par b1 de X arrivent à l'activité c1 de Y et sont reportées à l'activité "administration des noeuds de correspondance" B de Y. En fonction des valeurs des paramètres de la proposition, B de Y crée une activité "noeud de correspondance" notée b1 associée à l'activité c1 de Y et chargée de répondre à cette proposition.

d/ Création des activités "connexion (n)" (figure 7.5, 7.6, 7.7)

Supposons que l'activité b1 de Y réponde positivement à la configuration de correspondance proposée par b1 de X ; un noeud de correspondance est alors établi. L'activité "administration des connexions (n)" A de X, informée de l'existence de ce noeud, peut alors créer une activité "connexion (n)" a1 complétant la chaîne d'activités (b1,c1) de X. Cette activité chargée de gérer le protocole (n) commence par envoyer une PDU d'établissement de la connexion (n) au système Y.

Cette PDU reçue par l'activité b1 de Y est transmise à l'activité "administration des connexions (n)" A de Y qui, en fonction des caractéristiques de la connexion (n) demandée, crée une activité "connexion (n)" a1 sur le système Y. Celle-ci devient le partenaire de dialogue de l'activité a1 de X pour dérouler leur protocole (n). La communication entre a1 de X et a1 de Y est possible grâce aux deux chaînes d'activités établies : (a1,b1,c1)/X et (a1,b1,c1)/Y.

Pour résumer, l'établissement de cette connexion (n) dans la couche (n) a nécessité des créations successives de :

- un couple d'activités homologues "connexion (n-1)"

- un couple d'activités homologues "noeud de correspondance"
- un couple d'activités homologues "connexion (n)"

qui forment deux chaînes d'activités homologues gérant le passage des données de la connexion (n) à travers la couche (n).

e/ Etablissement d'une nouvelle connexion (n) utilisant la même connexion (n-1) (figures 7.8, 7.9)

Supposons que sur le système X une entité (n+1) envoie une requête d'établissement d'une connexion (n) entre les mêmes points d'accès (n) que la connexion (n) qui vient d'être établie.

Selon le schéma décrit précédemment, l'activité "administration des connexions (n)" A de X demande à l'activité B de X de lui fournir un noeud de correspondance. Etant donné que le noeud déjà établi dessert le même destinataire et peut supporter plus d'une connexion (n), son identificateur est tout de suite fourni à l'activité A de X. Ainsi, une nouvelle activité "connexion (n)" appelée a2 est immédiatement créée par l'activité A de X. Cette activité envoie une PDU d'établissement d'une connexion (n). Cette PDU empruntera le chemin formé par les chaînes d'activités (b1,c1)/X et (b1,c1)/Y.

Après avoir été récupérée par l'activité b1 de Y, la PDU est transmise à l'activité "administration des connexions (n)" A de Y. En fonction des paramètres contenus dans la PDU, A de Y crée une activité "connexion (n)" appelée a2 qui va assurer la gestion du protocole (n) en coopération avec l'activité distante a2 de X.

Nous constatons que dans ce cas l'établissement de la connexion (n) ne nécessite que la création d'activités "connexion (n)" car les supports d'accès au service (n-1) sont déjà disponibles dans les niveaux 2 et 3.

### 3.3.2 - Transfert de données

Dans la phase de transfert de données, les chaînes d'activités créées : (a1,b1,c1)/X, (a2,b1,c1)/X, (a1,b1,c1)/Y, (a2,b1,c1)/Y assurent le fonctionnement des deux connexions (n) de façon autonome (figure 8) sans aucune intervention des activités d'administration

locale, sauf dans des cas d'anomalie.

Les données à transférer traversent successivement les activités d'une chaîne et y subissent des opérations telles que le groupage, le dégroupage, la concaténation, la séparation, etc...

Nous remarquons que l'association des activités en une chaîne permettent d'éviter des opérations d'adressage à chaque transfert de données. Une modification des tables d'adressage n'est nécessaire que lors d'une opération de reconfiguration de la couche et est effectuée par les activités d'administration locale sans que les activités appartenant à des chaînes de transfert de données en soient informées (voir un exemple de reconfiguration en 3.3.4).

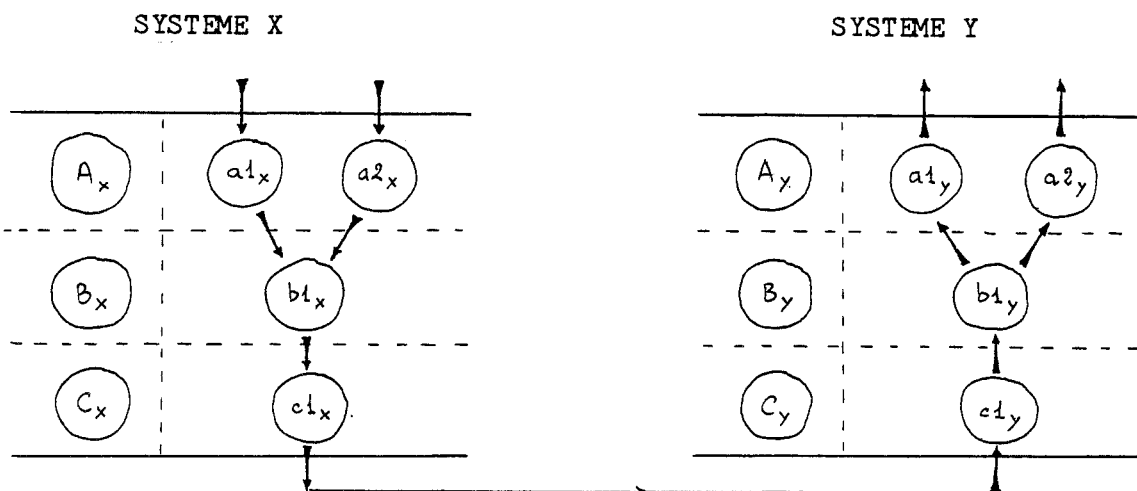


Figure 8 : Transfert de données

### 3.3.3 - Libération d'une connexion (n)

En général, lorsque la communication est terminée, les activités d'administration détruisent les activités de la chaîne de transfert de données concernée. L'ordre de destruction est le suivant:

- les activités "connexion (n)";
- les activités "noeud de correspondance". Celles-ci ne sont libérées que si elles ne sont plus utilisées par d'autres connexions (n). L'activité d'administration des noeuds de correspondance peut décider,

en cas d'utilisation fréquente d'une liaison (n-1), de garder en vie une activité "noeud de correspondance", même si toutes les connexions (n) multiplexées sont déjà libérées;

- les activités "connexion (n-1)", avec les mêmes remarques que précédemment.

### 3.3.4 - Reconfiguration des connexions (n-1)

Il peut arriver que le mauvais fonctionnement sur une connexion (n-1) ou une modification de routage à travers le service (n-1) entraîne un remplacement d'une connexion (n-1) par une autre. Pour cela, les activités d'administration prennent en charge la reconfiguration. Elles suspendent le fonctionnement des activités se trouvant sur les niveaux 1 et 2 et procède à un remplacement des activités du niveau 3. Des activités suspendues ne seront réactivées que lorsque la reconfiguration aura été faite.

Il est important de noter que la reconfiguration d'une connexion (n-1) s'effectue de manière invisible pour les activités des niveaux 1 et 2 de la chaîne concernée.

## 4 - QUELQUES CAS PARTICULIERS DU MODELE

Le modèle présenté vise le cas le plus complexe : celui d'une couche (n) fournissant un service (n) à base de connexion et utilisant un service (n-1) lui-aussi à base de connexion, les configurations de correspondance étant multiples. Nous allons montrer la simplification du modèle dans d'autres types de configurations.

### 4.1 - Service (n) orienté connexion, service (n-1) de type sans connexion

Dans ce cas, on trouve une activité du niveau 3, non plus par connexion (n-1), mais par point d'accès (n-1). Chacune de ces

activités gère l'accès au service (n-1) selon les règles d'utilisation imposées au point d'accès (n-1) auquel elle est associée. Elles peuvent exister en permanence puisque ces points d'accès sont généralement figés. Remarquons qu'en général il y a une seule activité de ce type car le nombre de points d'accès est souvent limité à un.

Une activité "noeud de correspondance" assure éventuellement la concaténation des PDU provenant d'un certain nombre de connexions (n) et ayant le même point d'accès (n-1) destinataire (figure 9).

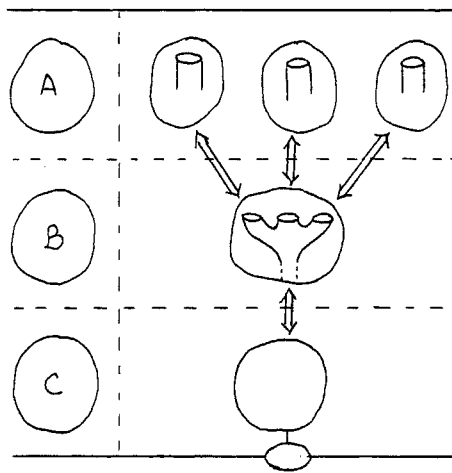


Figure 9

4.2 - Service (n) de type sans connexion, service (n-1) orienté connexion

A l'image du cas précédent, le premier niveau ne comporte plus des activités qui représentent les connexions (n), mais des activités qui desservent les points d'accès (n) existants. Des configurations d'éclatement peuvent exister dans la couche (figure 10)

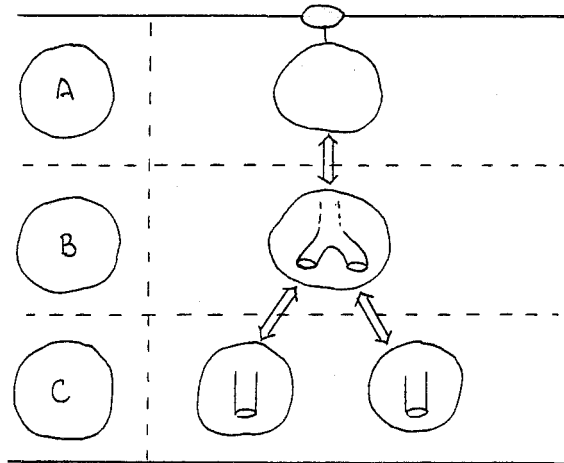


Figure 10

5 - RESUME DES CARACTERISTIQUES DU MODELE

Le modèle dynamique d'une entité (n) tel qu'il a été introduit possède deux qualités : **généralité** et **modularité** .

a/ La généralité réside dans les deux aspects suivants :

\* la facilité d'adaptation du modèle à des structures de couche différentes;

\* l'indépendance du modèle vis à vis des outils d'implantation. Ceci est dû à la généralité du concept d'activité. En effet, les activités ont été introduites comme des unités d'exécution de fonctions sans aucune hypothèse sur la façon dont elle sont réalisées. De même, les concepts tels que les opérations sur les activités et la communication entre les activités, qui leur sont associés, sont tout à fait généraux et ne présupposent rien quant à la réalisation.

b/ On obtient une structure modulaire d'une couche grâce à 2 façons d'organiser ses fonctions :

- Une organisation verticale qui consiste à diviser la couche en des niveaux de traitement successifs.

- Une organisation horizontale qui sépare les fonctions d'administration locale de celles gérant la communication et l'acheminement des données entre deux entités (n).

Le modèle ainsi obtenu a les propriétés suivantes :

- Le niveau d'activités 1 concerne l'implantation du protocole. Un changement du protocole se limite à la modification de ce niveau.

- Le niveau d'activités 2 permet une indépendance complète entre les connexions (n) et les connexions (n-1).

- La conception du niveau d'activités 3 dépend de l'interface d'utilisation de la couche inférieure. Chaque fois que cette interface change, les modifications se limitent à ce niveau.

- Mises à part les activités d'administration locale, chacune des autres activités d'une chaîne sur ces 3 niveaux effectue un traitement particulier sur un certain flot de données sans se préoccuper de son environnement de communication. Les interactions entre ces activités deviennent naturelles car ce ne sont que des **échanges d'unités de données** définies dans le cadre du modèle de référence d'OSI.

- L'attribution des paramètres de fonctionnement (classe, sous-ensemble de protocole,...) aux activités précédentes, l'activation et la suspension de celles-ci, l'établissement et la rupture des liens qui existent entre elles, ou toute autre opération mettant en jeu leur environnement de communication (la reconfiguration par exemple) sont des fonctions réalisées par les activités d'administration locale. Ces fonctions sont définies en grande partie par le concepteur



au moment de la réalisation d'une couche. Il est par conséquent important de les séparer des autres fonctions de la couche.

En outre, par la représentation en activités, on obtient une vision du fonctionnement en parallèle de différents éléments de la couche.

\*

\*

\*

## CONCLUSION

Nous nous sommes proposés d'étudier, après avoir présenté deux réalisations réelles de couches, un modèle de fonctionnement d'une couche de communication. Par cette démarche, nous avons voulu d'une part décrire notre étude telle qu'elle s'est réellement déroulée, d'autre part insister sur le fait que le modèle résulte essentiellement d'une expérience de réalisation.

Il est vrai que les réalisations présentées ne sont pas des solutions optimales : elles comportent, pour des raisons historiques, des choix arbitraires tant sur le plan de la structuration que sur le plan de l'implantation. Cependant, c'est précisément grâce à l'expérience acquise lors de ces réalisations que nous avons pu élaborer le modèle de couche présenté dans les chapitres IV et V. Ce modèle permet d'explicitier nettement les aspects importants suivants :

- Du point de vue fonctionnel, on voit apparaître dans la couche une hiérarchie de traitement verticale (par la mise en évidence des sous-couches) et une hiérarchie de traitement horizontale (par l'introduction des fonctions d'administrations locale).

- Du point de vue dynamique, l'indépendance des connexions ou des noeuds de correspondance s'exprime à l'aide des activités parallèles qui leur sont associées.

Le modèle fournit alors un guide pour le développement d'une couche particulière. En guise de conclusion, nous allons donner quelques idées qui peuvent orienter les prolongements possibles de cette étude :

- Le modèle donne un premier niveau de décomposition d'une couche qui facilitera par la suite une description plus détaillée de cette dernière.

Des langages de description formelle de protocoles intégrant les concepts d'OSI, tels que PDIL, LC/1 ou le langage FDT de l'ISO, ont été introduits récemment /F1,F2,F3/. On constate que ces langages permettent de décrire les activités et leur comportement (par des automates d'états finis ou des réseaux de Pétri) ainsi que la communication entre activités. Notre modèle pourra alors servir d'ossature pour une description de couche utilisant ces langages.

- Si l'on utilise le modèle introduit comme support d'implantation d'une couche, il faut réaliser les concepts d'activité et d'objet de communication à l'aide des outils système fournis par le système d'exploitation de la machine cible (processus, boîte aux lettres,...).

On constate que, pour notre type de problème (qui a trait à des protocoles de communication), les automates d'états finis ou les réseaux de Pétri sont habituellement utilisés pour décrire le comportement des activités : un événement qui arrive à une activité déclenche une action indivisible, modifie l'état de celle-ci et entraîne un envoi des événements de sortie vers d'autres activités. Ceci suggère l'implantation du modèle sur des systèmes du type "d'exécution atomique des transactions" ou de "traitement cadencé par événements". On peut trouver un exemple de ce type de système dans /F4/.

- On constate également que chaque activité du modèle peut être vue comme un "objet" (par exemple "une connexion (n)" avec les opérations "ouverture", "fermeture", etc...). De ce fait, on peut envisager une spécification en termes d'objets et de types abstraits /F5/ pour mieux formaliser la modularité du modèle.

Enfin, allier les deux notions "d'orientation d'objets" (aspect de modularité) et de "traitement cadencé par événements" (aspect de parallèle-réparti) nous paraît une idée bien adaptée à la formalisation de notre type de problème /F6/. Elle donne sans doute une orientation possible pour prolonger cette étude.

## BIBLIOGRAPHIE

/A1/ Richard DESJARDINS

ISO Open Systems Interconnection Standardization - Status Report  
SIGCOMM' 83 Symposium, Austin, Texas.  
Computer Communication Review, Vol 13, N° 2, March 1983.

/A2/ INFOREP.BNI

La normalisation des réseaux informatiques.  
Document d'INFOREP (Association générale des utilisateurs de  
l'informatique répartie), et BNI (Bureau d'orientation de la  
normalisation en informatique).  
1982.

/A3/ J. P. ANSART

RHIN ou les Réseaux Informatiques Hétérogènes Normalisés.  
ENJEUX N° 28, Septembre 1982.

/B1/ AFNOR Z70-001

Système de traitement de l'information. Modèle de référence de  
base pour l'interconnexion des systèmes ouverts.  
Juin 1982.

/B2/ A. LYMAN CHAPIN

Connectionless Data Transmission.  
Computer Communication Review, Vol 12, N° 2, April 1982.

/B3/ H. ZIMMERMANN

Réseaux informatiques.  
Techniques de l'Ingénieur, H3500, Mars 1980.

/B4/ H. J. BURKHARDT, S. SCHINDLER

Structuring Principles of the Communication Architecture of Open  
Systems - A Systematic Approach.  
Computer Networks, Vol 5, N° 5, 1981.

/B5/ V. G. CERF, P. T. KIRSTEIN

Issues in Packet Network Interconnection.

Proceedings of IEEE, Vol 66, N° 11, November 1978.

/B6/ V. G. CERF, R. E. KAHN

A Protocol for Packet Network Interconnection.

IEEE Transactions on Communication, Vol Com 22, N° 5, May 1974.

/B7/ D. WALDEN

A System for Interprocess Communication in a Resource Sharing  
Computer Network.

Communication of the ACM, Vol 15, April 1972.

/B8/ ISO/TC 97/SC 16/WG 1

Data Processing - Open Systems Interconnection - Service  
Conventions.

N 897, January 1982.

/B9/ AFNOR CF/TC 97/SC 16/GT 1

Concepts généraux d'adressage d'OSI (document de travail).

Groupe ad hoc sur l'adressage OSI, Octobre 1982.

/B10/ TRANSPAC

Spécifications techniques d'utilisation du réseau Transpac  
(STUR).

Document du réseau Transpac.

/B11/ M. GIEN, H. ZIMMERMANN

Design Principles for Network Interconnection.

Six Data Communications Symposium (IEEE/ACM), Pacific Grove,  
California, November 1979.

/B12/ L. POUZIN

Interconnection of Packet Switching Network.

Proceeding Subconference on Computer Networks, HICSS-7, Honolulu,  
Hawaï, Janvier 1974.

/B13/ CCITT X.121

Recommendation X.121 "International numbering plan for public data network".

Genève, 1978.

/B14/ CORNAFION

Systèmes informatiques répartis.

Edition Dunod, 1981.

/B15/ V. Di CICCIO, C. A. SUNSHINE, J. A. FIELD, E. G. MANNING

Alternatives for Interconnection of Public Packet Switching Data Networks.

Six Data Communications Symposium, Pacific Grove, California, November 1979.

/B16/ CCITT X.75

Recommendation X.75 "Terminal and transit call control procedures and data transfer system on international circuits between packet switched data networks".

Genève, 1978.

/B17/ CCITT X.25

Recommendation X.25 "Interface between data terminal equipment (DTE) and circuit terminating equipment (DCE) for terminals operating in packet mode on public packet networks".

Genève, 1977.

/C1/ H. ZIMMERMANN

OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection.

IEEE Transactions on Communications, Vol Com 28, N° 4, April 1980.

/C2/ ECMA-72

Transport Protocol - Standard.

January 1981.

/C3/ ECMA-75

Session Protocol - Final Draft.

December 1981.

/D1/ M. CART, Ba Thu NGUYEN

Implantation des cinq couches du modèle de l'ISO dans un micro-ordinateur - Aspect matériel - Logiciel de base du frontal - Intégration du logiciel X25.

Rapport interne, Convention RHIN N° 81/279, Février 1982.

/D2/ M. CART, Ba Thu NGUYEN

Implantation des cinq couches du modèle de l'ISO dans un micro-ordinateur - Mise en oeuvre de la couche Transport.

Rapport interne, Convention RHIN N° 81/279, Avril 1982.

/D3/ M. CART, Ba Thu NGUYEN

Implantation des cinq couches du modèle de l'ISO dans un micro-ordinateur - Service et architecture du frontal.

Rapport interne, Convention RHIN N° 81/279, Août 1982.

/D4/ M. CART, Ba Thu NGUYEN

Implantation des cinq couches du modèle de l'ISO dans un micro-ordinateur - Mise en oeuvre de la couche Session.

Rapport interne, Convention RHIN N° 81/279, Janvier 1983.

/D5/ Projet RHIN

Service et Protocole de Transport.

Préstandard RHIN, Version N° 1, 1981.

/D6/ Projet RHIN

Service et Protocole de Session.

Protocole expérimental RHIN, Version N° 1, 1982.

/D7/ M. LE FEVRE, O. RAFIQ, J. P. ANSART, Y. FISCHER

Modèle Pascal du protocole de Transport RHIN.

Technical Report, FDT 7502, Juin 1981.

/E1/ T. F. PIATKOWSKI

The ISO-ANSI Open Systems Reference Model - A Proposal for a System Approach.

Computer Networks, Vol 4, N° 4, 1980.

/E2/ S. GUIBOUD RIBAUD

Architecture des systèmes - Une vue synthétique : logicielle et matérielle.

Saint Etienne, Ecole des Mines, Juillet 1978.

/E3/ CROCUS

Systèmes d'exploitation des ordinateurs.

Edition Dunod, 1975.

/E4/ X. ROUSSET DE PINA, I. VATTON

Interface de la Station de Transport Danube.

Note interne, IMAG, 1979.

/F1/ J. P. ANSART, V. CHARI, M. NEYER, O. RAFIQ, D. SIMON

Description, Simulation and Implementation of Communication Protocols using PDIL.

SIGCOMM' 83 Symposium, Austin, Texas.

Computer Communication Review, Vol 13, N° 2, March 1983.

/F2/ J. M. AYACHE, J. P. COURTIAT

LC/1, Un langage pour la description, l'analyse et l'implémentation des protocoles.

Journées de Présentation du Projet RHIN, Agence de l'Informatique, Avril 1983.

/F3/ ISO/TC 97/SC 16/WG 1

- Architectural Specification Concepts.

- Extended Finite State Machine Specification.

- Temporal Ordering Specification Language.

Draft tutorial documents, N 34, N 35, N 36, December 1982.



/F4/ J. S. BANINO, A. CARISTAN, M. GUILLEMONT, G. MORISSET,  
H. ZIMMERMANN  
CHORUS : An Architecture for Distributed Systems.  
Rapport de recherche INRIA N° 42, Novembre 1980.

/F5/ B. G. CLAYBROOK  
A Specification Method for Specifying Data and Procedural  
Abstractions.  
IEEE Transactions on Software Engineering, Vol SE. 8, N° 5,  
September 1982.

/F6/ M. SHAPIRO  
Un cadre pour la conception de logiciel réparti, sûr et  
modulaire.  
BIGRE N° 30, Juin 1982.

\* \*  
\*

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,  
VU les rapports de présentation de M. M. HABIB et Melle M. CART

M. NGUYEN Ba Thu

est autorisé à présenter une thèse en soutenance pour l'obtention  
du diplôme de DOCTEUR-INGENIEUR, spécialité Informatique

Fait à Saint-Etienne, le 28 novembre 1983

Le Directeur de l'EMSE,







UN MODELE DE FONCTIONNEMENT D'UNE COUCHE D'UNE ARCHITECTURE  
DE COMMUNICATION

NGUYEN BA THU

N° d'ordre 36 II

Mots-clés : Réseaux

Interconnexion des systèmes ouverts

Architecture de communication

Structuration en couches

Sous-couches

Modèle dynamique.

RESUME

Dans le domaine des réseaux informatiques, la technique de structuration en couches est habituellement utilisée pour décomposer des architectures de communication. Dans ce travail, nous nous proposons d'étudier un modèle de fonctionnement d'une couche qui facilite la description et l'implantation de cette dernière. Le modèle introduit fournit une organisation hiérarchisée et modulaire des fonctionnalités de la couche.

Après avoir rappelé les notions fondamentales liées à l'architecture d'interconnexion de systèmes, nous décrivons deux réalisations réelles de couches. Ensuite, nous présentons le modèle selon deux axes complémentaires : la décomposition fonctionnelle en sous-couches et l'étude dynamique de la couche.