DOSIS : un serveur OSI pour l'ouverture des systèmes distribués au monde extérieur

Samer Haj Houssain

▶ **To cite this version:**

Samer Haj Houssain. DOSIS : un serveur OSI pour l'ouverture des systèmes distribués au monde extérieur. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1988. Français. NNT : . tel-00328632

HAL Id: tel-00328632
https://theses.hal.science/tel-00328632

Submitted on 10 Oct 2008

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

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

THESE

présentée par

Samer HAJ HOUSSAIN

pour obtenir le titre de

DOCTEUR
de l'INSTITUT NATIONAL POLYTECHNIQUE
DE GRENOBLE
(arrêté ministériel du 5 juillet 1984)

Spécialité : INFORMATIQUE

DOSIS : un serveur OSI pour l'ouverture des systèmes distribués au monde extérieur

Soutenue le 28 janvier 1988 devant la Commission d'Examen composée de :

MM.	Jacques MOSSIERE	Président
	Roland BALTER	Examinateurs
	Guy JUANOLE	
	Guy MAZARE	
	Guy PUJOLLE	
Mme	Dominique SERET	

Thèse préparée au sein du Laboratoire de Génie Informatique

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : Georges LESPINARD

Année 1988

Professeurs des Universités

BARIBAUD Michel	ENSERG
BARRAUD Alain	ENSIEG
BAUDELET Bernard	ENSPG
BEAUFILS Jean-Pierre	ENSEEG
BLIMAN Samuel	ENSERG
BLOCH Daniel	ENSPG
BOIS Philippe	ENSHMG
BONNETAIN Lucien	ENSEEG
BOUVARD Maurice	ENSHMG
BRISSONNEAU Pierre	ENSIEG
BRUNET Yves	IUFA
CAILLERIE Denis	ENSHMG
CAVAIGNAC Jean-François	ENSPG
CHARTIER Germain	ENSPG
CHENEVIER Pierre	ENSERG
CHERADAME Hervé	UFR PGP
CHOVET Alain	ENSERG
COHEN Joseph	ENSERG
COUMES André	ENSERG
DARVE Félix	ENSHMG
DELLA-DORA Jean	ENSIMAG
DEPORTES Jacques	ENSPG
DOLMAZON Jean-Marc	ENSERG
DURAND Francis	ENSEEG
DURAND Jean-Louis	ENSIEG
FOGGIA Albert	ENSIEG
FONLUPTJean	ENSIMAG
FOULARD Claude	ENSIEG
GANDINI Alessandro	UFR PGP
GAUBERT Claude	ENSPG
GENTIL Pierre	ENSERG
GREVEN Hélène	IUFA
GUERIN Bernard	ENSERG
GUYOT Pierre	ENSEEG
IVANES Marcel	ENSIEG
JAUSSAUD Pierre	ENSIEG
JOUBERT Jean-Claude	ENSPG
JOURDAIN Geneviève	ENSIEG
LACOUME Jean-Louis	ENSIEG
LESIEUR Marcel	ENSHMG
LESPINARD Georges	ENSHMG
LONGEQUEUE Jean-Pierre	ENSPG
LOUCHET François	ENSIEG
MASSE Philippe	ENSIEG
MASSELOT Christian	ENSIEG
MAZARE Guy	ENSIMAG
MOREAU René	ENSHMG
MORET Roger	ENSIEG
MOSSIERE Jacques	ENSIMAG
OBLED Charles	ENSHMG
OZIL Patrick	ENSEEG
PARIAUD Jean-Charles	ENSEEG
PERRET René	ENSIEG
PERRET Robert	ENSIEG
PIAU Jean-Michel	ENSHMG
POUPOT Christian	ENSERG
RAMEAU Jean-Jacques	ENSEEG
RENAUD Maurice	UFR PGP
ROBERT André	UFR PGP
ROBERT François	ENSIMAG
SABONNADIERE Jean-Claude	ENSIEG
SAUCIER Gabrielle	ENSIMAG
SCHLENKER Claire	ENSPG
SCHLENKER Michel	ENSPG
SILVY Jacques	UFR PGP
SIRIEYS Pierre	ENSHMG
SOHM Jean-Claude	ENSEEG
SOLER Jean-Louis	ENSIMAG
SOUQUET Jean-Louis	ENSEEG
TROMPETTE Philippe	ENSHMG
VEILLON Gérard	ENSIMAG
ZADWORNY François	ENSERG

Professeur Université des Sciences Sociales (Grenoble II)

BOLLIET Louis

Personnes ayant obtenu le diplôme d'HABILITATION A DIRIGER DES RECHERCHES

BECKER Monique
BINDER Zdenek
CHASSERY Jean-Marc
CHOLLET Jean-Pierre
COEY John
COLINET Catherine
COMMAULT Christian
CORNUEJOLS Gérard
COULOMB Jean- Louis
DALARD Francis
DANES Florin
DEROO Daniel
DIARD Jean-Paul
DION Jean-Michel
DUGARD Luc
DURAND Madeleine
DURAND Robert
GALERIE Alain
GAUTHIER Jean-Paul
GENTIL Sylviane
GHIBAUDO Gérard
HAMAR Sylvaine
HAMAR Roger
LADET Pierre
LATOMBE Claudine
LE GORREC Bernard
MADAR Roland
MULLER Jean
NGUYEN TRONG Bernadette
PASTUREL Alain
PLA Fernand
ROUGER Jean
TCHUENTE Maurice
VINCENT Henri

Chercheurs du C.N.R.S
Directeurs de recherche 1ère Classe

CARRE René
FRUCHART Robert
HOPFINGER Emile
JORRAND Philippe
LANDAU Ioan
VACHAUD Georges
VERJUS Jean-Pierre

Directeurs de recherche 2ème Classe

ALEMANY Antoine
ALLIBERT Colette
ALLIBERT Michel
ANSARA Ibrahim
ARMAND Michel
BERNARD Claude
BINDER Gilbert
BONNET Roland
BORNARD Guy
CAILLET Marcel
CALMET Jacques
COURTOIS Bernard
DAVID René
DRIOLE Jean
ESCUDIER Pierre
EUSTATHOPOULOS Nicolas
GUELIN Pierre
JOUD Jean-Charles
KLEITZ Michel
KOFMAN Walter
KAMARINOS Georges
LEJEUNE Gérard
LE PROVOST Christian
MADAR Roland
MERMET Jean
MICHEL Jean-Marie
MUNIER Jacques
PIAU Monique
SENATEUR Jean-Pierre
SIFAKIS Joseph
SIMON Jean-Paul
SUERY Michel
TEODOSIU Christian
VAUCLIN Michel
WACK Bernard

Personnalités agrées à titre permanent à diriger des travaux de recherche (décision du conseil scientifique)

E.N.S.E.E.G

CHATILLON Christian
HAMMOU Abdelkader
MARTIN GARIN Régina
SARRAZIN Pierre
SIMON Jean-Paul

E.N.S.E.R.G

BOREL Joseph

E.N.S.I.E.G

DESCHIZEAUX Pierre
GLANGEAUD François
PERARD Jacques
REINISCH Raymond

E.N.S.H.G

ROWE Alain

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

COURTIN Jacques

E.F.P.

CHARUEL Robert

C.E.N.G

CADET Jean
COEURE Philippe
DELHAYE Jean-Marc
DUPUY Michel
JOUVE Hubert
NICOLAU Yvan
NIFENECKER Hervé
PERROUD Paul
PEUZIN Jean-Claude
TAIB Maurice
VINCENDON Marc

Laboratoires extérieurs

C.N.E.T

DEVINE Rodericq
GERBER Roland
MERCKEL Gérard
PAULEAU Yves

Je remercie Mr Jacques Mossière, Professeur à l'INPG et Directeur du Laboratoire de Génie Informatique, de m'avoir fait l'honneur de présider le jury de cette thèse.

Je remercie,
Mr Michel Diaz, Directeur de recherche au CNRS ;
Mr Guy Pujolle, Professeur à l'université de Paris VI, qui ont bien voulu être rapporteur de ce travail.

Je remercie vivement Mme Dominique Seret, Maître de conférences à l'ENST de Paris, pour avoir accepté de juger ce travail. Ses remarques m'ont permis d'améliorer largement le contenu de ce document.

J'exprime ma sincère gratitude à Mr Guy Juanole, Maître de conférences à l'INSA de Toulouse, pour avoir accepté de juger ce travail et pour les précieux échanges que nous avons eus tout au long de la rédaction de cette thèse. L'intérêt qu'il a manifesté pour mon travail et les conseils qu'il m'a donnés, m'ont beaucoup aidé dans l'élaboration de ce document.

Je tiens à exprimer ma profonde reconnaissance à Mr Guy Mazaré, Professeur à l'INPG qui a bien voulu diriger cette thèse. J'ai été très sensible au soutien qu'il m'a apporté et à la confiance qu'il m'a accordée.

J'adresse mes sincères remerciements à Mr Roland Balter, responsable du Centre de Recherche Bull à Grenoble, qui m'a accueilli au sein de son équipe. Je tiens à lui témoigner toute ma reconnaissance et ma gratitude pour le soutien permanent qu'il m'a manifesté.

Je tiens à remercier,

MM. Jacques Bernadat et Gérard Vandôme, ingénieurs au Centre de recherches Bull, qui m'ont beaucoup appris. Je leur suis profondement reconnaissant.

Mr Xavier Rousset de Pina, Maître de conférences à l'université de Grenoble, pour les critiques qu'il m'a adressées lors de la rédaction. Son amitié m'a été une source d'encouragement indispensable.

Mlle Danièle Silvestre, en préparation d'un mémoire d'ingénieur CNAM, pour l'aide qu'elle m'a apportée en travaillant sur la réalisation d'une maquette de DOSIS.

Mes collègues et amis du Laboratoire de Génie Informatique et du Centre de Recherche Bull, pour leur sympathie.

Ce travail a été réalisé grâce au soutien du Centre de Recherche Bull.

RESUME

Nous nous sommes intéressés dans cette thèse au problème d'ouverture des systèmes distribués au monde extérieur. Cette ouverture est assurée au moyen d'un service de communication externe. Ce service permet aux applications s'exécutant dans le système distribué local de communiquer à plusieurs niveaux de protocoles avec d'autres applications s'exécutant sur des systèmes distants distribués ou centralisés.

L'examen des caractéristiques de la communication externe permet de constater que les services et les protocoles de communication OSI sont adaptés à cette communication. Le modèle OSI peut donc être considéré comme un modèle pour la communication entre les systèmes distribués.

Assurer la communication entre les systèmes distribués implique l'interconnexion de leurs systèmes de communication respectifs qui doivent être supposés totalement hétérogènes. L'analyse des techniques d'interconnexion basées sur l'encapsulation ou sur la conversion montre leurs limites pour assurer une solution satisfaisante. Nous proposons donc une nouvelle approche pour assurer la communication entre les systèmes distribués. Cette approche est basée sur un serveur OSI distribué (DOSIS). Ce serveur permet l'utilisation des systèmes de communication spécifiques à l'intérieur de chaque système distribué, tout en donnant à l'extérieur une vision de ce dernier qui est celle d'un système OSI unique.

L'architecture et le fonctionnement de DOSIS ont été définis pour permettre l'accès et le partage d'une instance unique des couches OSI dans le système distribué ainsi que la possibilité de distribuer ces couches sur plusieurs sites. Chaque utilisateur a l'impression de supporter l'ensemble des couches OSI sur son site local.

L'intérêt fondamental de l'approche DOSIS réside dans la séparation entre les protocoles de communication interne et les protocoles de communication externe dans un système distribué. Ainsi cette approche est applicable quel que soit le niveau de communication externe sans imposer de contraintes particulières sur le service de communication interne.

MOTS CLES

systèmes distribués, ouverture, communication, serveur de communication, modèle OSI, interconnexion de réseaux, passerelle.

SUMMARY

This thesis deals with the openness problem of distributed systems. The openess is provided by means of an external communication service. This service allows applications running on the local distributed system to communicate at several protocol levels with other applications running on either distributed or centralised remote systems.

Identifying the characteristics of the external communication shows that OSI communication services and protocols are adapted to such a communication. Thus, the OSI model is considered as a model for communication between distributed systems.

Providing communication between distributed systems implies the interconnection of their respective communication systems which should be supposed totaly heterogenous. The analysis of the interconnection techniques based either on encapsulation or conversion shows their limits in providing a satisfactory solution to our problem. Therefore we propose a new approach for communication between distributed systems based on a Distributed OSI Server (DOSIS). This approach allows the use of specific communication system inside the distributed system; while such a system appears from the outside as being single OSI system.

The function and architecture of DOSIS are designed in order to provide applications with access to a single and shared instance of the OSI layers. The facility of distributing OSI layers into several sites is also considered. Thus each user on the distributed system has the impression of having the full set of OSI layers on his own site.

The main feature of DOSIS consists of the separation between the internal and external communication protocols. Thus this approach can be applied to any external communication level without imposing specific requirements on the internal communication service.

KEY WORDS

distributed systems, openness, communication, communication server,
OSI model, networks interconnection, gateway.

TABLE DES MATIERES

Introduction 1

Chapitre 1

Les systèmes distribués sur réseaux locaux et l'ouverture au monde extérieur 5

1. Introduction 5

2. Les Systèmes Distribués sur Réseaux Locaux (SD-RL) 7
   - 2.1. De l'interconnexion à l'intégration 7
     - 2.1.1. Les SD-RL basés sur l'interconnexion 8
     - 2.1.2. Les SD-RL basés sur l'intégration 9

3. L'ouverture des SD-RL au monde extérieur 13
   - 3.1. Caractéristiques de la communication externe 15
   - 3.2. La communication externe et le modèle OSI 16

4. La communication interne dans un SD-RL et le modèle OSI 18
   - 4.1. Les protocoles OSI de transfert de données et les SD-RL 20
   - 4.2. Les protocoles OSI du niveau application et les SD-RL 23

5. Conclusion 24

Chapitre 2

Interconnexion de réseaux 27

1. Introduction 27

2. Classification des différents cas d'interconnexion de réseaux 30

2.1. Cas d'interconnexion des réseaux partiellement hétérogènes 31
2.2. Cas d'interconnexion des réseaux totalement hétérogènes 34

3. Eléments de base de l'interconnexion 35
3.1. Les passerelles 35
3.2. Le niveau d'interconnexion 40
3.3. Hétérogénéité des protocoles de communication 44

4. Quelques problèmes liés à l'interconnexion 46
4.1. Adressage et routage 46
4.1.1. Le mécanisme de routage dans un environnement d'interconnexion 47
4.1.2. Différents schémas d'adressage et l'interconnexion de réseaux 48
4.1.2.1. Adressage plat 48
4.1.2.2. Adressage hiérarchique 49
4.1.2.3. Adressage par représentant 51
4.2. Segmentation et réassemblage 52
4.3. Contrôle de flux et contrôle d'erreurs 55

5. Les techniques d'interconnexion 57
5.1. Critères d'évaluation 57
5.2. La technique d'encapsulation 60
5.2.1. L'interconnexion par encapsulation 60
5.2.2. L'application de la technique d'encapsulation 63
5.2.3. Le protocole d'encapsulation 64
5.2.4. Evaluation de la technique d'encapsulation 65
5.3. La technique de conversion 68
5.3.1. Le principe de la réalisation de conversion 71
5.3.2. Le processus de conversion 73
5.3.3. Evaluation de la technique de conversion 74

6. Exemples de passerelles 76
6.1. La passerelle ROSE 76
6.1.1. Le projet ROSE 76
6.1.2. Principe d'interconnexion 77

6.2. La passerelle TCP/IP - Transport ISO 80
6.2.1. Principe d'interconnexion 81
6.2.2. Comparaison entre les services du transport classe 4 ISO et du TCP 82
6.2.3. Définition d'un sous-ensemble commun de services 84

7. Conclusion 85

Chapitre 3

DOSIS: un serveur OSI distribué pour la communication avec le monde OSI 87

1. Introduction 87

2. Quelles solutions pour fournir les services OSI dans un SR-RL 88

3. DOSIS : une nouvelle approche pour l'ouverture des SD-RL au monde OSI 90
3.1. Définition et caractéristiques de DOSIS 90
3.2. Le principe du fonctionnement de DOSIS 93
3.2.1. Le modèle de service 93
3.2.2. La mise en œuvre du modèle client-serveur 93

4. L'architecture et le fonctionnement de DOSIS 95
4.1. Spécification du serveur OSI centralisé 96
4.1.1. Eléments de l'architecture du serveur OSI centralisé 96
4.1.2. Coopération entre les différentes entités 102
4.1.3. Protocole Client-Serveur 105
4.1.4. Quelques remarques sur le fonctionnement du serveur 112
4.2. Spécification du serveur OSI distribué 116
4.2.1. Eléments de l'architecture de DOSIS distribué 116
4.2.2. Un scénario d'utilisation du serveur OSI distribué 121

5. Intérêts de l'approche DOSIS 123
6. Réalisations 126
6.1. Le serveur transport-ROSE 127
6.1.1. Les interfaces transport-ROSE et TCP/IP 127
6.1.2. Le principe de l'implémentation et les choix retenus 129
6.2. Le serveur distribué session-transport 132
6.2.1. Les choix retenus pour l'implémentation du serveur session-transport 133
6.2.2. Le logiciel Courier 134
6.2.3. Réalisation à l'aide de Courier du serveur session-transport 135

Conclusion 139

Annexe

Les modèles d'interconnexion : ISO, DARPA, et CCITT 143

1. Introduction 143

2. Le modèle d'interconnexion OSI 144
2.1. Structure de la couche réseau 145
2.2. Les approches OSI pour l'interconnexion 148
2.2.1. L'approche "relais" 148
2.2.2. L'approche "internet" 151
2.2.3. Application des approches internet et relais 152
2.3. Scénarios d'interconnexion 154
2.3.1. Scénario basé sur l'approche orientée connexion 155
2.3.2. Scénario basé sur l'approche orientée sans connexion 155
2.4. Adressage et routage 158

3. Le modèle d'interconnexion DARPA 162
3.1. L'architecture DARPA 162
3.2. Le principe d'interconnexion 164
3.3. Le protocole IP 165
3.3.1. Format des datagrammes IP 165
3.3.2. Eléments du protocole IP 168
3.4. L'utilisation du protocole IP dans une configuration d'interconnexion de réseaux locaux à travers un réseau X25 174
3.4.1. Principe d'interconnexion 174
3.4.2. La mise en oeuvre de IP sur X25 176
3.4.3. Les fonctionnalités de l'interface IP/X25 177

4. Le modèle d'interconnexion CCITT 178
4.1. Principe d'interconnexion 179
4.2. L'adressage CCITT 183
4.3. Caractéristiques du protocole d'interconnexion X75 184

5. Conclusion 185

Bibliographie

Introduction

L'évolution dans le domaine de la communication à laquelle nous avons assisté dans la dernière décennie a donné une nouvel élan aux travaux de recherche dans le domaine des systèmes d'exploitation. Il s'agit en effet de tirer profit des nouveaux moyens de communication pour construire des systèmes informatiques distribués basés sur la coopération et le partage des ressources matérielles et logicielles d'un ensemble de machines situées dans un espace géographique limité. Cette coopération et ce partage, appelés aussi *intégration*, ne peuvent pas se faire sans communication, le rôle d'un système d'exploitation étant de rendre la coopération et le partage le plus efficace et le plus transparent possible à l'utilisateur, concepteur des applications.

L'environnement dans lequel se situe notre travail est celui d'un *Système Distribué sur Réseau Local (SD-RL)*. Un tel système est constitué physiquement d'un ensemble d'ordinateurs (stations) connectés par un réseau local. Un ensemble de logiciels de base est mis en œuvre pour assurer l'accès et le partage des ressources disponibles sur ces stations. Les applications supportées par le système sont typiquement des applications de génie logiciel et de bureautique.

Comme pour les systèmes centralisés, un SD-RL ne doit pas rester isolé, il a besoin d'échanger des informations avec d'autres systèmes. L'utilisation de ces systèmes distribués a donc de nouveau posé le problème de la coopération et de la communication, cette fois non pas à l'intérieur de ces systèmes mais entre les systèmes distribués eux-mêmes. Nous appelons ce problème *ouverture des systèmes distribués au monde extérieur*.

L'objectif de notre travail est donc de concevoir et de mettre en œuvre une solution qui assure l'ouverture des SD-RL au monde extérieur. Cette ouverture est assurée en fournissant aux applications d'un SD-RL un service de communication, appelé *service de communication externe*. Ce service met en relation une application s'exécutant dans le système distribué local avec une autre application qui se trouve à l'extérieur. Il permet l'échange d'informations à plusieurs niveaux de services et de protocoles entre deux SD-RL (ou entre un SD-RL et un système centralisé).

Assurer le service de communication externe au niveau du système distribué implique l'interconnexion du système de communication sous-jacent avec les systèmes de communication des SD-RL et des systèmes centralisés distants. L'étude des techniques d'interconnexion généralement utilisées montrera leurs limites pour assurer une solution satisfaisante au problème d'ouverture des SD-RL. Nous proposons donc une nouvelle approche d'interconnexion basée sur l'utilisation d'un serveur OSI distribué qui permet d'offrir le service de communication externe et rend les SD-RL ouverts au monde extérieur.

Cette thèse est organisée en trois chapitres. Le premier définit l'environnement SD-RL dans lequel est posé le problème d'ouverture. Il dégage les propriétés du système de communication dans cet environnement et les caractéristiques du service de communication externe. Nous discutons, dans ce chapitre, de la relation entre le modèle OSI et les systèmes de communication des SD-RL. Nous dégageons ensuite les raisons pour lesquelles les services définis dans le cadre du modèle OSI ne sont pas toujours utilisés dans le cadre des systèmes distribués sur réseaux locaux. Ceci ne doit pas être analysé comme un refus, donc un échec, de l'architecture OSI qui est orientée plus vers *l'interconnexion* des systèmes, alors que celle des SD-RL est orientée vers *l'intégration* des systèmes.

Le second chapitre est consacré à l'étude du problème d'interconnexion de réseaux. Une classification des cas d'interconnexions est présentée dans ce chapitre. Les éléments essentiels liés à l'interconnexion sont dégagés. Nous décrivons ensuite les principales approches proposées pour l'interconnexion des réseaux (l'application de ces principes dans les trois architectures de communication ISO, DARPA et CCITT est décrite dans l'annexe). Cette étude montrera les limites de ces techniques et la nécessité de rechercher une autre approche pour assurer l'ouverture des SD-RL.

Dans le troisième chapitre, nous proposons notre approche pour traiter le problème de la communication externe dans les SD-RL. Avec cette approche, le modèle OSI prend toute son importance en tant que norme pour la communication. Il assure l'ouverture des systèmes distribués vers l'extérieur en permettant aux utilisateurs de ces systèmes de communiquer avec des systèmes distants qu'ils soient distribués ou centralisés. L'approche retenue débouchera sur une solution appelée *DOSIS (Distributed OSI Server)*. Ce serveur distribué joue le rôle d'un frontal de communication OSI pour le compte des utilisateurs et des stations de travail connectées au système distribué. Ceci sans imposer de contraintes particulières sur les protocoles de communication interne qui a lieu entre les stations du même système distribué. Les fonctionnalités et l'architecture de DOSIS sont décrites dans ce chapitre. Nous montrons ensuite les caractéristiques et l'intérêt de DOSIS et en particulier ses avantages par rapport aux solutions d'interconnexion de réseaux citées dans le chapitre précédent. Les idées développées ont été validées par une réalisation dont les éléments essentiels seront présentés à la fin de ce chapitre.

Le problème posé dans cette thèse et le travail réalisé s'appuient sur l'expérience acquise au cours de notre participation dans deux projets ESPRIT : Le projet ROSE (*Research Open System for Europe*) qui est concerné par la mise en œuvre de l'architecture OSI, et le projet CSA (*Communication System Architecture*) qui a pour but de définir une architecture de communication pour des systèmes distribués dans un environnement de bureautique.

Chapitre 1

Les systèmes distribués sur réseaux locaux et l'ouverture au monde extérieur

1. Introduction

Nous avons assisté au cours de la dernière décennie à deux phénomènes importants. D'une part, la généralisation de l'utilisation des réseaux locaux et d'autre part, l'apparition de ce qu'on appelle les "stations de travail" qui ont fait et continuent à faire l'objet d'une formidable évolution.

Depuis l'apparition des premièrs réseaux locaux au milieu des années 70, la technologie dans la matière n'a pas cessé d'évoluer. Cette évolution aidée par un certain effort de normalisation a permis de passer du stade d'utilisation limitée à quelques centres expérimentaux au stade de grande diffusion. Celle-ci a touché d'abord les centres de recherche et les laboratoires universitaires où nous avons vu l'installation d'un grand nombre de réseaux locaux. Ces réseaux relient les moyens de calcul et de développement autrefois dispersés et isolés pour les mettre en commun à la disposition des utilisateurs. La grande diffusion des réseaux locaux est en train de toucher également les entreprises dans deux domaines différents : le domaine des applications industrielles de type commande et surveillance de processus industriels et le domaine des applications bureautiques qui est en pleine effervescence.

Parallèlement au progrès technologique en matière de réseaux locaux et à la généralisation de l'utilisation de ces réseaux, nous avons assisté ces dernières années à la naissance de ce qu'on appelle maintenant les "stations de travail". Ce sont des ordinateurs qui possèdent une grande puissance de calcul et un large espace de mémorisation mais qui sont caractérisés par un faible coût et des volumes réduits. Ces deux caractéristiques permettent leur utilisation comme poste de travail individuel ou partagé par un nombre réduit d'utilisateurs. Ces stations de travail présentent généralement une interface homme-machine sophistiquée (éditeur multi-fenêtres, souris, écran à point, etc...) et surtout la possibilité de se connecter à un réseau local.

L'apparition de ces stations de travail a permis de franchir un grand pas vers la décentralisation de l'informatique à l'intérieur même de domaines géographiques limités. Cette décentralisation ne pouvait pas être mise en œuvre sans assurer les moyens permettant à ces stations de travail d'accéder des ressources partagées car onéreuses (imprimantes à Laser, serveurs d'archivage, etc...). Elle implique aussi la mise à la disposition des utilisateurs des différentes stations de travail d'un service de communication qui assure les mêmes fonctions d'échange d'informations existantes dans les systèmes centralisés. Un réseau local relie donc les stations de travail entre elles et des serveurs spécialisés. Les deux fonctions d'accès à des ressources communes et d'échange d'informations sont alors assurées par un ensemble de protocoles supportés dans chaque station connectée au réseau. Ces protocoles offrent des services tels que l'accès terminal à distance, le transfert de fichier, le courrier électronique, etc...

La généralisation des réseaux locaux et la disponibilité de stations de travail puissantes ont contribué au développement de travaux de recherche sur ce qu'on peut appeler *"Système Distribué sur Réseau Local (SD-RL)"*. Ces travaux visent à passer du schéma de type stations de travail et serveurs avec communication explicite pour l'accès aux ressources distantes et pour l'échange d'informations à un schéma où la distribution est cachée autant que possible.

L'objet de ce chapitre est d'identifier le besoin d'*ouvrir* les SD-RL au monde extérieur. Il s'agit d'offrir aux différentes applications qui se déroulent sur un SD-RL la possibilité de communiquer avec d'autres applications situées dans d'autres systèmes distribués ou centralisés. Cette ouverture sera assurée au

moyen d'un service de communication externe. L'architecture de communication adaptée pour fournir ce service ainsi que les protocoles utilisés seront définis.

Le chapitre sera découpé en trois parties. Dans la première partie, nous définissons ce qu'est un système distribué sur réseau local, en décrivant sa structure générale et les services qu'il fournit. Dans la deuxième partie nous discutons le problème d'ouverture d'un tel système sur le monde extérieur. Ce besoin implique l'existence d'un service de communication appelé *service de communication externe* dont les fonctions et les caractéristiques seront décrites. La troisième partie discute la relation entre la communication interne, la communication externe dans un SD-RL et le modèle OSI. Elle conclut sur le rôle que peut jouer le modèle OSI dans l'environnement d'un système distribué sur réseau local.

2. Les systèmes distribués sur réseaux locaux (SD-RL)

Un système distribué sur réseau local est constitué typiquement d'un ensemble de stations de travail et de serveurs connectés par un réseau local. Un ensemble de logiciels est mis en œuvre pour fournir aux applications des services généraux qui tirent avantage de cette connexion en leur permettant d'accéder à un ensemble de ressources mises en commun. Le système constitue ainsi un environnement de travail général pour le développement de logiciels et la bureautique. Le terme SD-RL n'inclut pas généralement les systèmes distribués "spécialisés" comme les systèmes de gestion de bases de données réparties ou les systèmes distribués dédiés à des applications spécifiques (commande de processus, systèmes bancaires...etc). A ce stade de généralité cette définition peut englober une multitude de systèmes. L'objet des paragraphes suivants est de préciser le type de SD-RL qui constitue le contexte de notre travail.

2.1 De l'interconnexion à l'intégration

Le critère principal qui permet de distinguer les différents SD-RL est la *transparence de la distribution*. La transparence d'une ressource ou d'un objet dans un système distribué se traduit par les deux caractéristiques suivantes :

- l'application qui manipule cet objet n'a pas à connaître sa localisation physique (la machine sur laquelle il se trouve).

- la sémantique et la syntaxe des opérations d'accès ou de manipulation de cet objet sont indépendantes de sa localisation. En d'autres termes, ces opérations sont homogènes que l'objet soit local ou distant.

Selon les systèmes, cette transparence peut caractériser un sous-ensemble plus ou moins important des ressources gérées. En fonction du critère de transparence, on peut obtenir différents types de SD-RL qui vont des SD-RL basés sur *l'interconnexion* à ceux basés sur *l'intégration* en passant par des systèmes intermédiaires. Les termes "interconnexion" et "intégration" sont utilisés ici en opposition de l'un par rapport à l'autre pour décrire la vision que les applications ont du SD-RL. L'interconnexion signifie que les applications ont une vision d'un ensemble de *systèmes autonomes* mais capables d'échanger des informations. L'intégration indique que la visibilité offerte à l'application est celle d'un unique système. Nous résumons dans la suite les principales caractéristiques de chaque type de SD-RL.

2.1.1. Les SD-RL basés sur l'interconnexion

Les SD-RL basés sur l'interconnexion constituent le cas extrême où la distribution est totalement visible aux applications. Comme dans tous les SD-RL, des protocoles de communication sont mis en œuvre pour assurer l'échange d'informations entre les différentes stations du système. Cependant cet échange reste visible aux applications quelque soit le niveau d'abstraction. Les caractéristiques suivantes permettent de distinguer ce type de systèmes [Tanenbaum 85] :

- chaque station possède son propre système d'exploitation. Elle fonctionne avec beaucoup d'autonomie et d'une manière indépendante des autres stations ;

- l'utilisateur travaille habituellement sur une station. L'accès à d'autres processeurs se fait explicitement au moyen d'un protocole d'accès terminal à distance (remote login) ;

- l'accès à un fichier distant se fait en transférant ce fichier sur la station

locale explicitement au moyen d'un protocole de transfert de fichiers ;

- le service de communication dans ce cas est un service supplémentaire fourni par les différents systèmes d'exploitation des stations connectées au réseau. La mise hors fonction de ce service ou du réseau n'affecte que les applications qui ont besoin d'échanger des informations entre plusieurs stations ou qui demandent l'accès à des ressources distantes.

Cette catégorie de SD-RL inclut les réseaux locaux supportant une architecture de communication en couches. On trouve au niveau de la couche supérieure des applications de type transfert de fichiers, accès terminal à distance, messagerie, etc... Un exemple typique d'un SD-RL basé sur l'interconnexion est un réseau Ethernet reliant des stations qui supportent l'architecture de communication DARPA (TCP/IP, UDP/IP, TELNET, FTP, etc...). Généralement les stations du SD-RL ainsi que leurs systèmes d'exploitation sont hétérogènes. Cette hétérogénéité reste visible aux utilisateurs et aux applications. Ainsi l'accès à une station distante implique la connaissance de son système d'exploitation et de son environnement. Cette hétérogénéité peut se manifester aussi au niveau des structures de fichiers si le protocole de transfert de fichier utilisé n'assure pas la conversion nécessaire.

2.1.2. Les SD-RL basés sur l'intégration

Un pas important dans l'orientation vers les SD-RL basés sur l'intégration consiste à supporter sur le réseau local un système de gestion de fichiers réparti. Le fichier est une des ressources principales dans un système d'exploitation. Il s'agit de rendre la distribution des fichiers transparente. Ceci commence par donner à l'utilisateur (l'application) les moyens d'accéder aux fichiers distants de façon homogène à leurs accès locaux (en utilisant les mêmes primitives). Il n'y a donc plus de transfert explicite de fichier. L'évolution de ce type de SD-RL vers plus d'intégration aboutit à des systèmes où il existe un espace unique de désignation des fichiers. La localisation des fichiers est alors complètement cachée à l'utilisateur. Le système gère le stockage des fichiers en fonction du taux d'occupation des disques des différentes stations. Cependant chaque station supporte son propre système d'exploitation indépendant et autonome. Seul l'accès aux fichiers implique le fonctionnement du service de communication entre les différents systèmes d'exploitation, mais cela reste transparent à l'application. La mise en œuvre de la gestion répartie des fichiers

suppose que les systèmes de gestion de fichiers soient homogènes sur les différentes stations. Le cas échéant une harmonisation entre les différents systèmes de gestion de fichiers est nécessaire. Cela passe généralement par la définition d'un système de gestion de fichiers virtuel auquel on fait correspondre les systèmes locaux existants.

Plusieurs SD-RL basés sur une gestion de fichiers répartie ont été développés. On peut noter parmi d'autres : Newcastle Connexion [Brownbridge 82], LOCUS [Walker 84], NFS-Sun [Lyon 84].

Une évolution supplémentaire permet de généraliser l'intégration en assurant la transparence d'accès à toutes les ressources (et pas seulement aux fichiers). Ceci donne lieu alors aux systèmes distribués basés sur l'intégration.

Un système distribué basé sur l'intégration (appelé aussi un système distribué intégré) repose sur le principe d'intégrer l'ensemble des ressources des différentes stations dans une "machine virtuelle". L'ensemble de ces ressources est géré par un seul et unique système d'exploitation dont les composants sont répartis sur l'ensemble des stations. Ce système offre aux utilisateurs et aux applications la visibilité d'une machine virtuelle partagée. Il ne s'agit plus ici d'interconnecter des systèmes existants mais de concevoir un système d'exploitation qui prend en compte dès le départ les problèmes de la distribution. On obtient ainsi un système qui a les caractéristiques suivantes :

- la distribution est entièrement transparente aux utilisateurs. Les ressources et les objets sont accessibles de manière indépendante de leur localisation ;

- le système offre un espace de désignation global des objets gérés ;

- la communication entre les différentes stations est invisible au niveau des applications. Le service de communication dans les SD-RL intégrés devient un service de base indispensable au fonctionnement du système.

Dans un tel système, les ressources physiques telles que les espaces de stockage secondaire et plus particulièrement les processeurs sont alloués automatiquement par le système d'exploitation (en fonction de critères de performance et de disponibilité). Ces opérations font appel d'une manière

intense au système de communication qui devient un élément essentiel du système d'exploitation réparti. Les stations perdent beaucoup de leur autonomie par rapport au cas des systèmes basés sur l'interconnexion.

De manière schématique, on peut considérer un SD-RL intégré comme étant composé de quatre niveaux hiérarchiques (Fig 1.1).

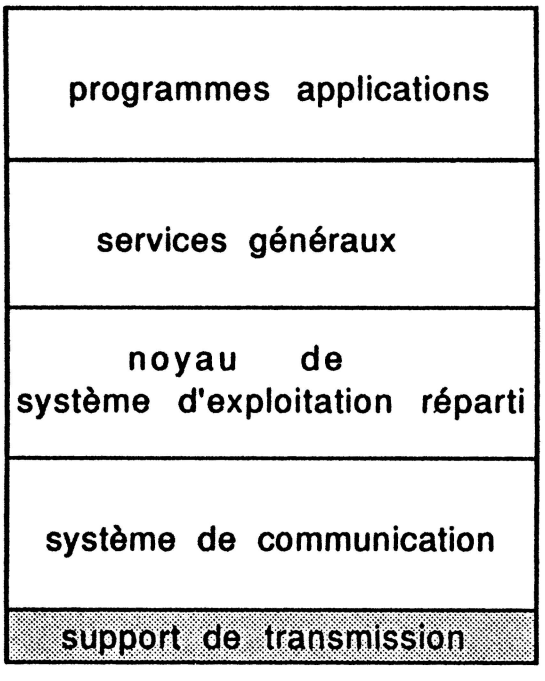


Fig 1.1 Structure hiérarchique générale d'un système distribué sur réseau local

Au niveau le plus bas, on trouve le système de communication qui inclut le support physique (le réseau) et les protocoles de communication. Ces protocoles vont jusqu'au niveau transport avec souvent un niveau de communication supplémentaire qui offre un service de communication inter-processus d'un niveau d'abstraction élevé telle que la communication par message ou l'appel de procédure à distance. Au niveau supérieur, on trouve le noyau du système d'exploitation réparti qui gère les objets et les ressources de base du système ainsi que leur allocation. Au dessus de ce noyau, un ensemble de services de bases sont offerts aux applications tel qu'un serveur de fichiers,

un serveur d'impression, un serveur graphique avec gestion de fenêtres, serveur de courrier électronique, etc... Au niveau le plus élevé, on trouve les applications des utilisateurs. Cette structure montre le rôle primordial du système de communication qui est à la base de l'ensemble de services offerts par le système.

La majeure partie des travaux de recherche en matière de systèmes d'exploitation est orientée depuis quelques années vers les SD-RL intégrés. On peut citer à titre d'exemple : Accent [Rashid 81], Mach [Jones 86], Chorus [Zimmerman 81] et V-system [Cheriton 83], ainsi que des SD-RL intégrés orientés vers l'approche objet comme : Argus [Liskov 84], Eden [Almes 85], Emrald [Black 86] et Guide [Balter 87].

L'environnement dans lequel se place notre travail est un système distribué sur réseau local. Le problème posé et la solution proposée sont valables quel que soit le type de SD-RL (basé sur l'interconnexion ou sur l'intégration). Cependant, l'approche que nous retenons trouve tout son intérêt dans un environnement de SD-RL intégré.

Il est à noter qu'en général le terme "système distribué" est utilisé dans la littérature pour désigner les systèmes orientés vers l'intégration. Dans la suite de ce document l'abréviation "SD-RL" sera utilisée pour désigner de tels systèmes.

3. L'ouverture des SD-RL au monde extérieur

Un système distribué doit fournir aux applications les mêmes types de services que ceux offerts habituellement par les systèmes centralisés à temps partagé. Une des propriétés identifiée comme essentielle dans les systèmes centralisés à temps partagé est *l'ouverture*. L'ouverture consiste à fournir aux applications un service de communication leur permettant d'échanger des informations avec d'autres systèmes. Au niveau d'abstraction des applications, un SD-RL est équivalent à un système centralisé. Par conséquent, les applications qui s'exécutent dans l'environnement d'un SD-RL, ont également besoin de communiquer avec le monde extérieur. La propriété d'ouverture est donc nécessaire dans un environnement SD-RL. Cette propriété est assurée par un service de communication qui est appelé *service de communication externe* par opposition au *service de communication interne* qui a lieu entre les différentes stations du SD-RL (Fig 1.2).

Le problème d'ouverture des systèmes n'est pas en lui-même un problème nouveau. Il s'était déjà posé en ce qui concerne les systèmes centralisés, il y a un dizaine d'années. La réponse à ce problème s'est traduite par la définition de l'architecture de communication OSI *(Open System Interconnection)* [ISO/IS 7498] définie par l'ISO (International Standard Organization). Ce qui nous intéresse ici est d'extrapoler cette notion d'ouverture pour l'appliquer globalement à un système distribué sur réseau local et d'étudier le rapport entre la communication interne qui est à la base des SD-RL et la communication externe nécessaire pour ouvrir ces SD-RL au monde extérieur.

Notre objectif est donc de concevoir et de réaliser une solution qui assure l'ouverture des SD-RL au monde extérieur au moyen d'un service de communication externe. Pour cela nous allons commencer par présenter les caractéristiques de la communication externe. Ceci permettra de définir la nature des services qui doivent être offerts et les protocoles adaptés qui peuvent être utilisés.

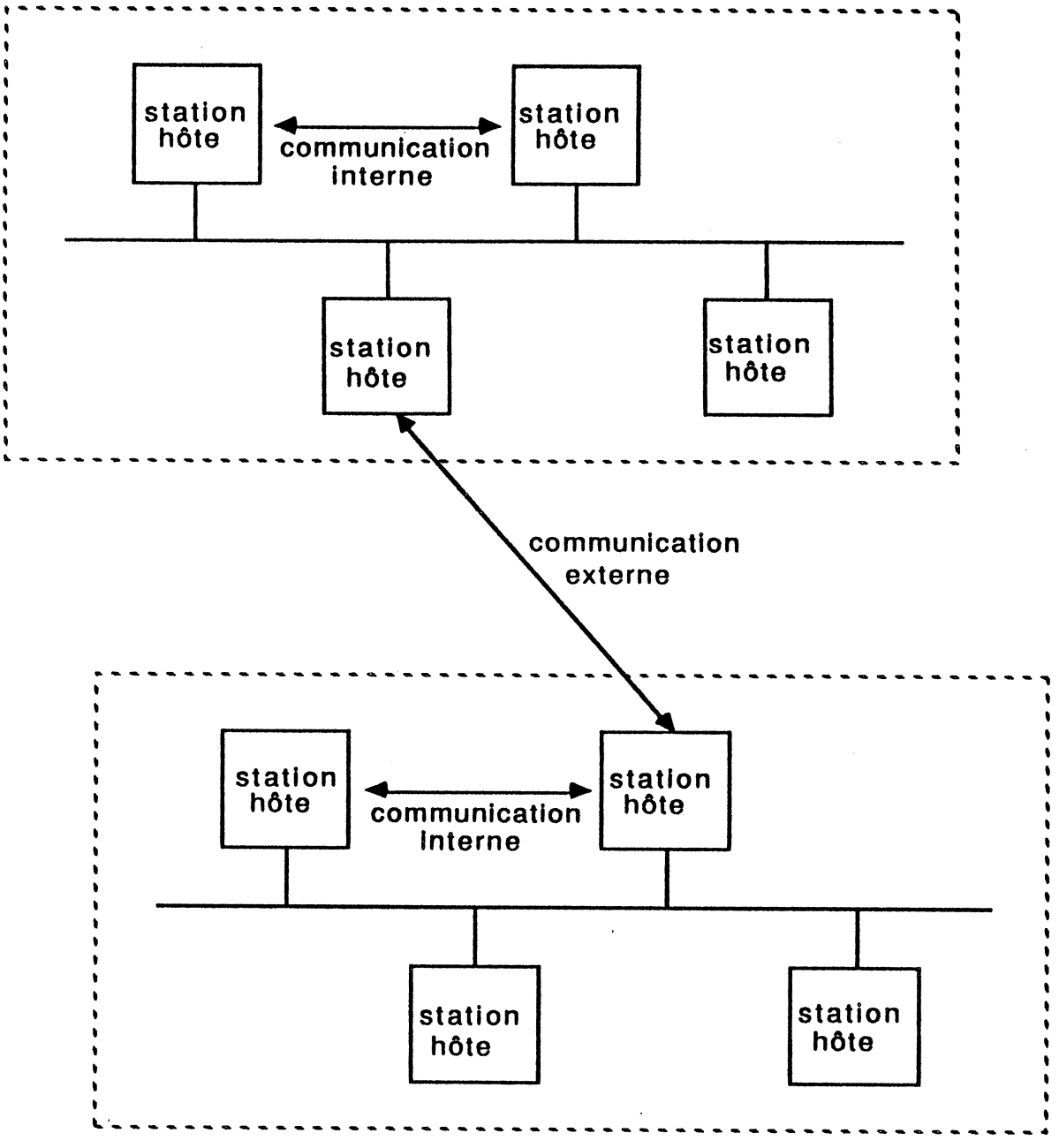


Fig 1.2 La communication interne et la communication externe dans un système distribué sur réseau local

3.1. Caractéristiques de la communication externe

Le choix entre le principe de l'interconnexion et celui de l'intégration se pose de nouveau en ce qui concerne la communication entre les SD-RL. Il s'agit de préciser à quel niveau d'abstraction l'échange d'informations entre les SD-RL doit être assuré. A priori, deux approches existent :

- la première consiste à assurer cet échange de manière implicite au niveau du noyau du système d'exploitation réparti. Ceci signifie l'intégration de l'ensemble des SD-RL dans une nouvelle machine virtuelle qui cache la distribution et la communication inter-SD-RL aux applications. Cette intégration nous semble peu intéressante pour les raisons suivantes :

  - les noyaux des systèmes d'exploitation distribués qui gèrent les différents SD-RL sont a priori hétérogènes car spécifiques à l'environnement et aux applications supportées par chaque SD-RL. Or l'intégration nécessite l'utilisation d'un noyau commun.

  - les différents SD-RL devant communiquer ne se trouvent pas généralement dans un espace géographique limité. Or l'intégration ne peut se faire que dans un environnement local, car un système intégré fait appel de manière intense à la communication. Ceci implique un débit d'information élevé et une bonne disponibilité des moyens de communication. Ces conditions ne sont généralement pas vérifiées dans le cas de la communication à grande distance.

- la seconde approche consiste à assurer la communication entre les SD-RL au niveau de leurs systèmes de communication. Ceci signifie une interconnexion entre ces SD-RL (et non pas une intégration). La communication entre les SD-RL sera donc explicite et visible au niveau des applications. Le service de communication externe ne sera donc pas un service de base mais un service supplémentaire. L'utilisation du service de communication externe est moins fréquente que celle du service de communication interne. Cette utilisation est faite par les applications seulement et non pour le fonctionnement du SD-RL lui-même. Par conséquent, les valeurs des paramètres de délai et de fiabilité ne sont plus critiques et la communication externe peut se faire à grande distance.

Pour les deux raisons que nous avons citées ci-dessus ainsi que pour d'autres liées à la complexité de la gestion, nous écartons l'approche d'intégration des SD-RL pour nous intéresser à la seconde approche basée sur l'interconnexion des SD-RL. Celle-ci permet à chaque SD-RL de conserver son autonomie et fournit aux applications la possibilité de communiquer avec des applications distantes d'une manière explicite.

3.2. La communication externe et le modèle OSI

Nous avons identifié dans le paragraphe précédent les propriétés de la communication externe pour un SD-RL. Il reste donc à définir les types de services que l'on souhaite offrir pour cette communication externe et les protocoles mis en œuvre pour réaliser ces services.

Le problème d'ouverture fait depuis quelques années l'objet de travaux importants par les instances de normalisation. Ils ont donné lieu à la définition du modèle de communication OSI comme solution à ce problème. Il est vrai que presque toute les implémentations de l'architecture OSI ont été réalisées sur des systèmes centralisés. Cependant, d'après le modèle de référence [ISO/IS 7498], la définition du terme "système" relatif aux systèmes OSI (un système OSI est un système ouvert supportant l'architecture OSI et communiquant au moyen des protocoles OSI) est une définition qui présente dans le cadre de notre environnement distribué, un grand intérêt. Cette définition est la suivante: "Un système, selon le concept du modèle OSI, est constitué d'un ou plusieurs ordinateurs, des logiciels associés, des périphériques, des terminaux, des opérateurs humains, des processeurs physiques, des moyens de transfert d'informations, etc... Cet ensemble forme un tout autonome capable de traiter et de transferer des informations".

L'intérêt que présente l'application de cette définition dans notre environnement apparaît dans la possibilité de considérer un SD-RL dans sa totalité comme étant conceptuellement un système OSI unique. Par conséquent, cette définition permet de retenir le modèle et les protocoles OSI pour la communication externe qui traverse les frontières du SD-RL. Ainsi, un SD-RL se comporte comme un système OSI capable de communiquer avec d'autres systèmes au moyen des protocoles OSI. Ceci indépendamment du modèle et des protocoles de communication internes au SD-RL. En effet, la définition citée ci-dessus ne fait aucune hypothèse sur le fonctionnement interne du système

OSI. Le choix de supporter l'architecture et les protocoles OSI pour la communication externe dans les SD-RL nous paraît très intéressant à plusieurs égards :

- le modèle OSI a été conçu comme une norme pour la communication, le but étant d'apporter une solution unique pour l'interconnexion des systèmes. Ceci permet de réduire le nombre des problèmes soulevés par l'interconnexion d'un système à N systèmes différents à un seul et unique problème. Cet avantage reste présent quelle que soit la nature des systèmes à interconnecter et en particulier pour assurer la communication entre différents SD-RL.

- la communication OSI est une communication orientée vers l'interconnexion. Les services OSI, quels que soient leurs niveaux, n'ont pas pour objectif de cacher la distribution à l'utilisateur de ces services. Ainsi les applications OSI présentent une interface dans laquelle la communication avec les systèmes distants est explicite. D'autre part, nous avons vu que la communication externe pour un SD-RL doit être aussi basée sur le principe de l'interconnexion. Par conséquent, le modèle OSI est, de ce point de vue, parfaitement adapté à la communication externe pour les SD-RL.

- le modèle et les protocoles OSI ont été conçus pour être utilisables dans des environnements hétérogènes. C'est le cas de notre environnement, où chaque SD-RL est a priori géré par un système d'exploitation réparti spécifique.

- dans un environnement de systèmes distribués qui supportent des applications générales de bureautique et de génie logiciel, le besoin en terme de communication externe consiste à disposer des services habituels de communication de haut niveau comme le courrier électronique, l'accès aux fichiers à distance, l'exécution à distance, etc... Il est aussi intéressant de disposer d'un service de communication de plus bas niveau (communication par message), d'une part pour pouvoir développer des applications plus spécifiques, et d'autre part pour pouvoir communiquer avec des systèmes qui ne supportent pas les services de communication du niveau application. Les services OSI qui ont été définis suffisamment variés et généraux répondent donc aux besoins de communication entre les systèmes distribués

Pour ces raisons nous pensons que l'architecture et les protocoles OSI sont adaptés à la communication entre les SD-RL. Ouvrir un SD-RL revient donc à offrir à ses utilisateurs, quelle que soit leur localisation, l'ensemble des services OSI de niveau application en plus d'un service de communication par message tel que le service du niveau transport. Le problème est alors de trouver une solution générale pour offrir ces services qui soit indépendante du service de communication interne au SD-RL. Ceci signifie qu'aucune hypothèse ne doit être faite quant à l'architecture et les protocoles qui sont utilisés pour la communication interne. Les systèmes de communication des différents SD-RL doivent être considérés comme a priori hétérogènes. Nous expliquons dans le paragraphe suivant les raisons de cette hypothèse.

4. La communication interne dans un SD-RL et le modèle OSI

L'interconnexion et la communication entre les SD-RL auraient été des problèmes relativement simples à résoudre si les systèmes de communication internes aux SD-RL étaient basés sur le modèle OSI. Prenons le cas de deux SD-RL gérés par des systèmes d'exploitation différents. Supposons que les deux SD-RL supportent l'architecture et les protocoles de communication OSI et que ces protocoles sont utilisés pour la communication interne dans chaque SD-RL. Chaque station implémente donc les couches OSI. L'interconnexion entre les SD-RL consiste à assurer la communication entre les stations d'un SD-RL avec les stations de l'autre au moyen des services OSI. Le problème d'interconnexion se réduit alors à l'utilisation d'une passerelle simplifiée (voir chapitre 2 et annexe pour l'interconnexion de réseaux homogènes). Une station quelconque d'un SD-RL peut communiquer avec une autre station appartenant à l'autre SD-RL au moyen des protocoles OSI supportés par chacune. Les mêmes couches de communication servent alors aussi bien pour la communication interne que pour la communication externe. La seule différence entre les deux cas est que, pour la communication interne, ces services OSI sont utilisés par le noyau du système d'exploitation réparti, alors que pour la communication externe ils sont accessibles directement par les programmes utilisateurs.

En examinant les exemples des SD-RL intégrés cités précédemmment nous constatons que le modèle OSI n'est pas souvent retenu comme système de communication interne dans les SD-RL. Par conséquent, il est plus réaliste de

supposer priori que les systèmes de communication des SD-RL sont hétérogènes. La recherche d'une solution pour l'interconnexion des SD-RL doit donc prendre en compte cette contrainte.

Depuis l'apparition du modèle OSI, le débat "pour ou contre le modèle OSI" n'a pas cessé d'exister. Il ne s'agit pas ici de prendre position dans ce débat, cependant, il nous semble nécessaire d'analyser brièvement les raisons pour lesquelles les concepteurs des SD-RL intégrés ont souvent tendance à s'orienter vers des systèmes de communication différents de celui de l'OSI. Parmi les raisons avancées, il existe des raisons concernant le modèle OSI en général, mais surtout des raisons liées à l'utilisation du modèle et des protocoles OSI dans un environnement de système distribué intégré. C'est sur ces dernières que nous mettons l'accent dans notre discussion.

Le rapport entre le modèle OSI et les systèmes distribués intégrés peut être discuté en distinguant trois niveaux de communication : le niveau des protocoles d'accès au support, le niveau des protocoles de transfert de données et le niveau des protocoles d'applications.

La normalisation OSI des couches basses dans les réseaux locaux a été définie suite aux travaux réalisés par IEEE dans le cadre du projet 802. Ces travaux ont largement pris en compte des protocoles d'accès au support existants et expérimentés. Pour répondre aux nouveaux besoins dans ce domaine, de nouvelles normes sont en cours d'étude. Les protocoles d'accès au support normalisés de type CSMA/CD avec une topologie en bus, ou de type jeton avec une topologie en bus ou en anneau, répondent aux besoins des différentes applications envisagées actuellement. En ce qui concerne l'environnement qui nous intéresse, c'est-à-dire un système d'exploitation réparti supportant des applications générales de bureautique et de développement de logiciel, il n'existe pas de contraintes particulières vis-à-vis des protocoles d'accès au support qui peuvent être utilisés. Par conséquent, la mise en œuvre des couches basses de l'OSI pour les réseaux locaux ne pose pas de problèmes particuliers dans un environnement de SD-RL.

Le problème de l'utilisation du modèle OSI dans les SD-RL apparaît au niveau des protocoles de transfert de données et surtout au niveau de la couche application. C'est ce que nous examinons dans les paragraphes suivants.

4.1. Les protocoles OSI de transfert de données et les SD-RL

Nous utilisons le terme "protocoles de transfert de données" pour désigner les protocoles qui correspondent aux niveaux réseau, transport et session. L'ISO ne propose pas, concernant ces niveaux et les niveaux supérieurs, des protocoles particuliers pour le cas des réseaux locaux. Il est important de noter l'existence de variantes des protocoles et des options possibles qui peuvent être implémentées dans chacune de ces couches. La couche réseau ainsi que la couche transport supportent chacune les deux modes de communication : avec connexion et sans connexion. Cinq classes de protocoles sont proposées au niveau transport. Le choix de la classe ou des classes à implémenter dépend des caractéristiques du réseau. Trois profils regroupant chacun un ensemble d'options sont proposés au niveau session (à noter que la couche session est en cours de réorganisation, des modifications importantes de sa structure peuvent être apportées).

Il existe deux types de raisons pour lesquelles l'utilisation des couches OSI de transfert de données dans les SD-RL n'est pas courante : des raisons historiques et des raisons techniques.

1) les raisons historiques

Les travaux de normalisation dans le domaine de la communication ont commencé à la fin des années 70, les premiers résultats sont apparus au début des années 80. Depuis, la définition des protocoles OSI a beaucoup avancé. Cependant les travaux de normalisation ne sont pas terminés, des protocoles sont encore en cours de spécification , en particulier les protocoles des couches hautes du modèle OSI. Cependant, d'autres architectures et protocoles de communication sont actuellement totalement opérationels. On peut noter par exemple les protocoles DARPA qui sont largement diffusés dans les milieux scientifiques, en particulier depuis leur intégration avec les différentes versions du système UNIX™. D'autres architectures de communication spécifiques-vendeurs comme DSA, SNA ou DECNET étaient également opérationnelles avant la spécification de l'architecture OSI. Ceci explique en partie le fait que le développement des systèmes distribués intégrés se fasse essentiellement sur la base des systèmes de communication existants, en attendant une plus large diffusion de l'architecture OSI.

2) Les raisons techniques

Nous avons vu que la communication dans les SD-RL est un service de base. Le noyau de système d'exploitation distribué qui gère le SD-RL fait constamment appel à ce service. Dans ces conditions, les *performances* de la communication interne constituent un élément essentiel des performances globales du système. Ces performances sont assurées généralement par le choix d'une architecture et d'un ensemble de protocoles qui ont les deux caractéristiques suivantes [Svobodova 87] :

- l'architecture choisie doit minimiser le nombre des couches. Un nombre important de couches peut pénaliser fortement les performances. Le cas le plus typique concerne l'ouverture et la fermeture des connexions (on suppose l'utilisation des protocoles en mode connexion). L'ouverture d'une connexion à un niveau donné, qui constitue une opération lourde consommant beaucoup de ressources et de temps, provoque généralement le déclenchement du processus d'ouverture sur tous les niveaux inférieurs [Seret 85].

  Cette minimisation du nombre de couches est faite en fonction des caractéristiques du réseau utilisé et en fonction des services demandés par le niveau supérieur c'est-à-dire par le noyau de système d'exploitation réparti. Elle est réalisée en écartant les fonctionnalités qui ne sont pas nécessaires dans l'environnement spécifique en question et en intégrant le maximum de fonctionnalités dans une même couche. Un exemple important de cette intégration est le protocole d'appel de procédure à distance (RPC). Les protocoles RPC sont largement utilisés dans les systèmes distribués où ils représentent généralement le niveau le plus élevé de communication sur lequel est bâti le noyau de système d'exploitation réparti. Le protocole RPC intègre des fonctionnalités qui peuvent appartenir à plusieurs niveaux dans le modèle OSI comme la session, la présentation et l'application.

- la seconde caractéristique des protocoles utilisés dans les SD-RL est la simplicité. Cette simplicité est également obtenue en prenant en compte les caractéristiques du réseau et les services demandés par le niveau supérieur. Cette simplicité peut être obtenue par exemple grâce à

l'utilisation des protocoles en mode sans connexion, chaque fois que cela est possible.

Les réserves qui sont annoncées habituellement vis-à-vis de l'utilisation du modèle OSI dans les SD-RL sont basées sur le fait que l'architecture et les protocoles OSI ne présentent pas les caractéristiques citées ci-dessus. Ceci ne doit pas être considéré comme un point faible du modèle OSI dont l'objectif est d'offrir des services de communication généraux adaptés à différents types de réseaux et à une large gamme d'applications. Au contraire, la communication interne, qui est une fonction de base dans les SD-RL, exige des contraintes de performance. Ces contraintes imposent des optimisations qui ne peuvent être obtenues que par l'utilisation d'architectures et des protocoles spécifiques.

Dans certains cas l'utilisation des protocoles spécifiques n'est pas seulement faite dans le souci d'améliorer les performances, mais elle est due surtout à la la quantité limitée des *ressources* disponibles dans les stations du SD-RL et au faible débit du réseau qui les connecte. Il n'est donc pas possible dans ces cas de supporter une architecture générale telle que l'OSI qui nécessite des ressources importantes pour sa mise en œuvre. Un exemple typique de ce cas est le réseau AppleTalk[Apple 85] qui relie des micro-ordinateurs Macintosh.

Un choix adéquat des classes et des options proposées par l'OSI peut, dans certains cas, constituer un ensemble de protocoles adaptés à un environnement de SD-RL. Cependant, ceci limite la possibilité de communiquer avec d'autres systèmes OSI qui se trouvent en dehors du SD-RL. L'intérêt principal de l'utilisation du modèle OSI se trouve ainsi diminué.

4.2. Les protocoles OSI du niveau application et les SD-RL

Les réserves les plus importantes émises vis-à-vis de l'utilisation du modèle OSI dans les systèmes distribués intégrés apparaissent au plan fonctionnel. Les fonctionnalités et les services offerts par les protocoles OSI, en particulier au niveau application, ne sont pas adaptés aux SD-RL.

D'abord, certaines fonctionnalités qui sont nécessaires dans les SD-RL ne sont pas prises en compte ou ne sont pas encore intégrées dans le modèle OSI comme par exemple la diffusion. La communication par appel de procédure à

distance représente un exemple intéressant d'un service très utilisé dans les SD-RL mais qui n'est pas encore intégré dans l'architecture OSI (des travaux sur ce sujet ont commencé il y a peu de temps au sein de l'ECMA et l'ISO [ECMA 87]).

La base du problème tient à la différence de principe entre les SD-RL et le modèle OSI. Comme nous l'avons vu, un "vrai" SD-RL est basé sur le principe d'intégration des ressources dans une seule machine virtuelle gérée par un seul système d'exploitation réparti. Cette machine présente une interface aux applications dans laquelle la distribution est invisible. Un espace de désignation global des ressources permet leurs accès indépendamment de leur localisation. A l'opposé, le modèle OSI est basé sur l'interconnexion. Chaque système OSI est un système indépendant et autonome géré a priori par un système d'exploitation spécifique. Les services qui sont offerts au niveau application permettent, en dépit de l'hétérogénéité, d'accéder des ressources distantes mais ne permettent pas de rendre la distribution transparente.

Cette différence dans le principe de base apparaît, par exemple, au niveau des protocoles d'accès aux fichiers à distance. L'accès à un fichier distant, selon le protocole FTAM [ISO/DP 8571] défini par l'ISO, nécessite l'ouverture d'une connexion avec une entité FTAM distante. L'utilisateur de ce service doit connaître la localisation du fichier à accéder et l'adresse de l'entité FTAM distante (sauf s'il accède à un service de nommage). L'espace de désignation des fichiers locaux et celui des fichiers distants sont disjoints. Par ailleurs, le service de fichier virtuel (VFS) défini dans le cadre du FTAM permet de donner une vision virtuelle de la structure des fichiers distants, de leurs attributs et de la sémantique des opérations applicables sur ces fichiers. Cependant, ces paramètres ne correspondent pas a priori à ceux du système de gestion de fichiers local. L'utilisateur est donc conscient de la distribution, il doit connaître à la fois le système de gestion de fichiers local pour l'accès aux fichiers locaux, et celui de l'OSI pour l'accès aux fichiers distants.

5. Conclusion

Nous concluons ce chapitre en récapitulant les idées essentielles discutées et en définissant la démarche que nous adaptons pour la recherche de la solution.

- un SD-RL est un ensemble de stations reliées par un réseau local et géré par un système d'exploitation réparti.

- il nous semble nécessaire de ne pas isoler les stations d'un SD-RL du monde extérieur. Notre but est donc de rendre les SD-RL ouverts.

- l'ouverture d'un SD-RL consiste à offrir à une application s'exécutant sur une station quelconque (ou un ensemble de stations) un service de communication externe. Ce service permet à cette application de communiquer avec d'autres applications s'exécutant sur un système centralisé distant ou sur une station quelconque appartenant à un autre SD-RL.

- le service de communication externe doit être basé sur le modèle OSI, par conséquent les protocoles de communication externe seront les protocoles OSI. Le SD-RL est vu comme étant un seul système OSI.

- pour assurer le maximum d'ouverture, les applications doivent avoir un accès à l'ensemble des services OSI du niveau application, mais aussi un accès direct aux services de communication de niveau plus bas (comme le transport ou la session).

- la solution que nous cherchons pour assurer l'ouverture des SD-RL doit tenir compte de la contrainte d'hétérogénéité des systèmes de communication des différents SD-RL.

La figure 1.3 schématise le problème que nous traitons. Il s'agit d'assurer la communication entre deux SD-RL. Chaque SD-RL, supporte un système de communication spécifique. Dans chaque SD-RL deux types de services de communication existent : un service de communication interne, accédé par le noyau du système d'exploitation réparti qui gère le SD-RL et un service de communication externe, orienté OSI et accessible par les applications pour la

communication avec l'autre SD-RL. Le problème de la communication entre les SD-RL revient donc, dans ce contexte, à un problème d'interconnexion de systèmes de communication hétérogènes au moyen des protocoles OSI.

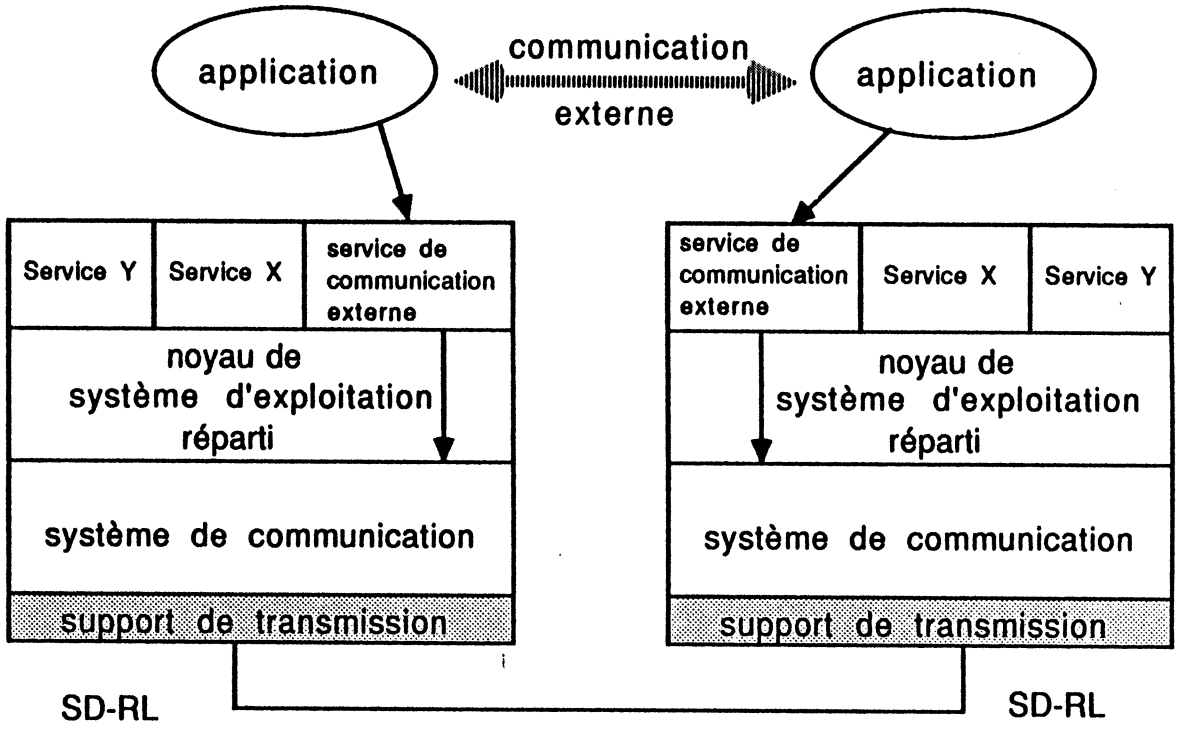


Fig 1.3 L'interconnexion des systèmes de communication pour assurer la communication inter-SD-RL

Le chapitre suivant sera donc consacré à une étude des techniques d'interconnexion de réseaux.

Chapitre 2

Interconnexion de réseaux

1. Introduction

Nous avons défini dans le chapitre précédent la notion d'*ouverture* vers le monde extérieur pour un SD-RL. Cette ouverture est assurée au moyen d'un service de communication externe dont les propriétés ont été identifiées.

Assurer un service de communication externe pour un SD-RL nécessite l'*interconnexion* du système de communication sous-jacent avec d'autres systèmes de communication. Cette interconnexion ne doit faire aucune hypothèse quant à l'architecture des systèmes de communication à interconnecter qui doivent être considérés hétérogènes. Cependant, elle doit permettre d'offrir plusieurs services de communication externe, tels que la messagerie, le transfert et l'accès des fichiers à distance, la communication par message ...etc.

L'objet de ce chapitre est d'étudier le problème d'*interconnexion de réseaux* et d'analyser les principales techniques utilisées pour résoudre ce problème. Cette analyse cernera les limites de ces techniques et définira leurs domaines d'application. Il nous permettra ensuite de juger si elles peuvent constituer une solution pour fournir le service de communication externe ; c'est-à-dire, assurer la communication entre les SD-RL à travers l'interconnexion de leurs systèmes de communication hétérogènes.

L'interconnexion de réseaux peut être définie comme étant la technique qui permet l'échange de données entre deux ou plusieurs réseaux. Elle fournit donc aux utilisateurs d'un réseau A les moyens de transmettre et de recevoir des données vers et en provenance des utilisateurs d'un réseau B. Ce problème a existé dès l'apparition des réseaux de communication ; il s'agissait au départ, d'interconnecter des réseaux à grande distance WAN (Wide Area Network) et, ensuite, d'interconnecter des réseaux locaux LAN (Local Area Network) entre eux ou avec des réseaux WAN.

Bien que la littérature sur ce sujet soit fournie, il est toujours difficile d'avoir une vue globale des problèmes et des techniques liés à l'interconnexion des réseaux. Cela est dû à la diversité des cas d'interconnexion qui peuvent se présenter : chaque cas est distingué, d'une part, par l'ensemble des protocoles qui sont mis en jeu dans l'interconnexion et, d'autre part, par le niveau d'hétérogénéité (défini plus loin) qui existe entre les réseaux à interconnecter. Cette diversité explique l'absence d'une méthodologie générale dans la recherche de solutions qui sont souvent de type "ad-hoc". Nous essaierons néanmoins, dans la suite de ce chapitre, de constituer une vue synthétique du problème d'interconnexion afin de vérifier si les techniques associées répondent au besoin du service de communication externe dans un SD-RL.

Notre étude des problèmes d'interconnexion est décomposée en cinq parties. Dans la première partie nous définissons un critère de classification des différents cas d'interconnexion. Ce critère permettra de distinguer deux classes principales. La deuxième partie introduit les éléments de base architecturaux et conceptuels liés à l'interconnexion. La troisième partie présente certains problèmes qui se posent dans un environnement d'interconnexion et les mécanismes qui permettent de les résoudre. Les principales approches et les techniques associées pour la mise en œuvre de l'interconnexion ainsi que leurs champs d'application sont exposés dans la quatrième partie. La dernière partie est consacrée à l'étude de deux exemples d'interconnexion.

Il convient auparavant de définir la terminologie que nous utiliserons au cours de ce chapitre :

Réseau global : c'est le réseau constitué de l'interconnexion de plusieurs réseaux.

Sous-réseau : un composant (réseau) d'un réseau global. Le terme *réseau* sera cependant utilisé en l'absence.d'ambiguïté possible.

Communication intra-réseau :

la communication entre des systèmes appartenant au même sous-réseau est appelée*communication intra-réseau.*

Communication inter-réseaux :

la communication entre deux systèmes appartenant à deux sous-réseaux différents est appelée *communication inter-réseaux.*

La terminologie OSI sera employée pour désigner les couches, les structures de données qui sont échangées au niveau du protocole. Cette terminologie sera parfois appliquée par extrapolation sur les autres architectures de communication :

(N)-PDU : *Protocol Data Unit*, ce sont les unités de données échangées au niveau du protocole de couche N.

(N)-SDU : *Service Data Unit*, ce sont les unités de données échangées au niveau de l'interface N/N+1.

(N)-SAP : *Service Access Point*, c'est le point d'accès au service de la couche N. C'est encore l'identificateur des entités communicantes utilisatrices des services de la couche N (les entités N+1).

2. Classification des différents cas d'interconnexion de réseaux

Le problème essentiel qui apparaît lors de l'interconnexion des réseaux est le problème d'*hétérogénéité* de leurs systèmes de communication. Cette hétérogénéité peut se manifester sur le plan architectural ainsi que sur le plan des protocoles et des services utilisés. Elle peut apparaître à plusieurs niveaux (couches) de communication. La nature des problèmes et les solutions qu'on peut apporter dépendent fortement du *niveau d'hétérogénéité*. Ce niveau indique la couche de protocole à partir duquel les deux réseaux interconnectés utilisent des protocoles identiques. Mais il n'indique pas, pour les protocoles qui ne sont pas identiques à un niveau donné, leur degré d'hétérogénéité. En d'autres termes, c'est le niveau d'hétérogénéité vertical et non pas horizontal entre deux architectures de communication. Ainsi, deux réseaux peuvent être hétérogènes aux niveaux les plus bas, c'est-à-dire au niveau de leur support de transmission et de leurs protocoles d'accès au support. L'hétérogénéité peut atteindre des niveaux supérieurs. Le cas extrême apparaît quand les réseaux à interconnecter deviennent incompatibles à tous les niveaux de protocole. Ceci signifie que chacun des réseaux utilise une architecture et des protocoles de communication qui lui sont spécifiques.

Notre classification des cas d'interconnexion est essentiellement basée sur le critère de *niveau d'hétérogénéité* entre les réseaux à interconnecter. Ce critère nous semble important parce qu'il oriente le choix entre les différentes techniques d'interconnexion qu'on peut utiliser dans chacun des cas.

Nous distinguons deux niveaux d'hétérogénéité : *hétérogénéité partielle*, et *hétérogénéité totale*. Deux principaux cas d'interconnexion sont par conséquent définis :

1) cas d'interconnexion des réseaux partiellement hétérogènes,
2) cas d'interconnexion des réseaux totalement hétérogènes.

2.1. Cas d'interconnexion des réseaux partiellement hétérogènes

Nous appelons réseaux *partiellement hétérogènes* des réseaux qui supportent des protocoles de communication appartenant à la *même famille de protocoles* et respectant la *même architecture de communication*. On peut citer comme exemple les familles ISO, CCITT, DARPA, DSA, SNA.

La propriété essentielle de cette classe de problème d'interconnexion est la suivante :

Il existe un niveau n de protocole à partir duquel les réseaux à interconnecter utilisent les mêmes protocoles de communication (les protocoles sont identiques pour les niveaux n, n+1, n+2...). L'hétérogénéité qui existe dans les niveaux inférieurs à n est donc complètement invisible à l'utilisateur de ce niveau n (Fig 2.1). Un protocole de bout en bout est donc possible à partir du niveau n.

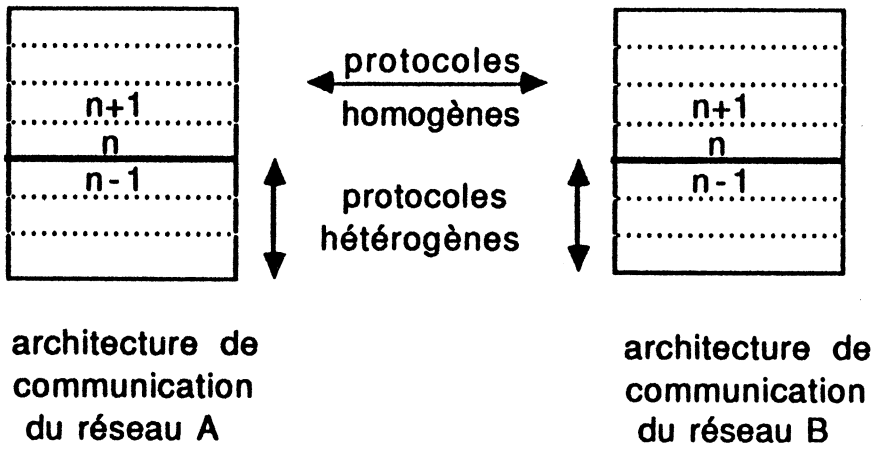


Fig 2.1 Réseaux partiellement hétérogènes

Bien que les réseaux à interconnecter appartiennent à la même famille, ils peuvent néanmoins présenter une certaine hétérogénéité. Cette dernière est due à des caractéristiques fonctionnelles différentes pour chaque réseau (étendue géographique, technique d'accès au support, etc...). Trois cas d'interconnexion de réseaux partiellement hétérogènes sont distingués :

1) Interconnexion LAN-WAN

Les problèmes du partage de la voie de communication et du routage dépendent souvent de la topologie du réseau utilisé. Certaines topologies sont adaptées à un environnement local alors que d'autres sont adaptées à des réseaux à grande distance. Le partage du support ainsi que le routage sont assurés généralement par les couches basses de l'architecture de communication (les trois premiers niveaux selon le modèle OSI). L'interconnexion LAN- WAN fait donc apparaître une hétérogénéité au niveau de ces couches. Dans le modèle OSI, ces fonctionnalités sont fournies dans les trois premières couches. Ces couches supportent les protocoles X25 niveaux physique, trame et paquet, alors que dans le cas des réseaux locaux, elles supportent des protocoles qui assurent les fonctionnalités des niveaux MAC (accès au support) et LLC (contrôle de liaison de données).

2) Interconnexion LAN-LAN

Dans le cas d'interconnexion de réseaux locaux, la nature de l'hétérogénéité est essentiellement liée aux différentes topologies utilisées et surtout aux protocoles d'accès au support (MAC) associés. La prise en compte de plusieurs protocoles d'accès au support dans la même architecture de communication est due, à la fois, à des arguments techniques (protocole déterministe versus non déterministe), et à des arguments politiques et commerciaux. Ainsi le modèle OSI définit plusieurs protocoles MAC comme le 8802.3 CSMA/CD, le 8802.4 token/ring, et le 8802.5 token/bus, d'où le problème d'interconnexion de ces différents réseaux.

3) Interconnexion WAN-WAN

Deux formes d'hétérogénéité peuvent exister lors de l'interconnexion des réseaux à grande distance. La première apparaît dans une architecture de communication permettant plusieurs protocoles d'accès au réseau. Dans cette architecture, chaque réseau, appelé généralement sous-réseau, utilise ses propres protocoles de bas niveau, c'est-à-dire les protocoles d'accès et de communication à travers le réseau, qui peuvent être de type circuit virtuel ou de type datagramme. Il s'agit donc d'interconnecter ces

sous-réseaux pour offrir aux niveaux supérieurs de protocole la visibilité d'un seul réseau. L'exemple le plus typique de cette configuration d'interconnexion est celui qui existe dans l'architecture de communication DARPA.

Une deuxième forme d'hétérogénéité apparaît dans le cas où les deux réseaux à interconnecter utilisent les mêmes protocoles d'accès au réseau, mais sont différents par leurs fonctionnalités internes telles que leurs protocoles internes, leur politique de routage, de contrôle de congestion, leur fonction d'administration de réseau etc... Pour des raisons évidentes de sécurité et de fiabilité, il n'est pas souhaitable d'intégrer les deux réseaux en un seul. Ils sont généralement interconnectés en gardant l'autonomie de chacun. Un exemple de ce cas d'interconnexion est celui du protocole X75 défini par le CCITT qui permet l'interconnexion des réseaux X25.

Comme nous l'avons indiqué, la classe des problèmes d'interconnexion des réseaux partiellement hétérogènes concerne des réseaux qui respectent le même modèle et les mêmes protocoles de communication. Les problèmes d'hétérogénéité sont connus au moment de la définition de cette architecture qui peut donc prendre en compte ces problèmes et intégrer les solutions qui permettent de couvrir les cas d'interconnexion. Nous examinons dans l'annexe les approches d'interconnexion qui sont proposées dans trois familles principales de protocoles de communication qui sont : ISO, CCITT et DARPA.

L'hétérogénéité qui apparaît dans la même famille de protocoles est liée aux types de réseaux utilisés. Cette hétérogénéité se manifeste généralement au niveau des couches basses de communication (jusqu'au niveau réseau). Le niveau transport étant lui indépendant du type de réseau (on ne considère pas les différentes classes de transport dans le modèle OSI comme des protocoles hétérogènes). La communication à ce niveau est de bout en bout. C'est donc généralement la couche transport qui représente le niveau n à partir duquel les protocoles sont identiques à l'intérieur de la même architecture de communication.

Un cas particulier d'interconnexion de réseaux partiellement hétérogènes consiste à interconnecter deux ou plusieurs réseaux qui supportent des

protocoles identiques à travers un réseau intermédaire qui supporte des protocoles différents. C'est par exemple le cas lorsque deux ou plusieurs réseaux locaux similaires sont interconnectés à travers un réseau public de type X25.

2.2. Cas d'interconnexion des réseaux totalement hétérogènes

Nous appelons *réseaux totalement hétérogènes* les réseaux qui supportent des familles différentes de protocoles (ISO et DARPA par exemple). Chaque réseau supporte sa propre architecture de communication. L'hétérogénéité apparaît au niveau du nombre de couches, de leurs fonctionnalités, et au niveau des protocoles qui sont mis en œuvre et qui sont a priori hétérogènes pour l'ensemble des couches. Par conséquent, aucune communication directe ne peut avoir lieu entre les deux réseaux quel que soit le niveau de cette communication.

Contrairement à l'interconnexion des réseaux partiellement hétérogènes, les solutions pour l'interconnexion des réseaux totalement hétérogènes ne sont pas intégrées au préalable dans les différentes architectures de communication. Il s'agit donc d'interconnecter des systèmes de communication conçus séparément. Les solutions qui sont proposées pour ce type d'interconnexion sont spécifiques à chaque contexte. Elles dépendent, d'une part, des deux systèmes de communication à interconnecter et, d'autre part, de leur niveau d'interconnexion. Ceci reflète la difficulté qu'il y a à définir une méthodologie générale qui permette la mise en œuvre de solutions systématiques pour ce cas d'interconnexion.

3. Eléments de base de l'interconnexion

Nous discutons dans cette section trois éléments de base du problème d'interconnexion de réseaux. Le premier élément est la nécessité d'introduire un système intermédiaire entre les réseaux interconnectés. L'architecture générique d'un tel système est décrite. Le deuxième élément est la notion de niveau d'interconnexion. L'importance de ce paramètre ainsi que les contraintes et les besoins qui déterminent sa valeur sont présentés. Le troisième élément est l'hétérogénéité des services et des protocoles ainsi que ses conséquences sur le problème d'interconnexion.

3.1. Les passerelles

L'interconnexion de deux ou plusieurs réseaux (qu'ils soient faiblement ou totalement hétérogènes) nécessite l'introduction d'un système qui a des fonctionnalités particulières, ce système est appelé généralement *passerelle (gateway)*. La définition de la passerelle qui nous paraît la plus adéquate, car la plus générale, est celle qui est donnée dans [Cerf 78] : une passerelle est un système matériel et logiciel attaché (connecté) à deux ou plusieurs réseaux permettant aux données de passer d'un réseau vers un autre. A noter les remarques suivantes:

- la passerelle peut être soit un système dédié à la réalisation de l'interconnexion auquel cas elle ne supporte pas d'applications utilisateurs, soit un système hôte (existant) dans lequel on intègre les fonctionnalités de la passerelle.

- les fonctionnalités nécessaires à la réalisation de l'interconnexion ne sont pas forcément toutes concentrées dans la passerelle, certaines peuvent exister dans les systèmes hôtes.

- la passerelle entre deux réseaux permet la communication entre les utilisateurs de ces deux réseaux à un ou plusieurs niveaux, mais elle ne doit pas être visible à ces utilisateurs.

- la passerelle doit être vue comme une entité logique, qui peut être parfois séparée physiquement en deux parties (Fig 2.2). Chaque partie, appelée *demi-passerelle (gateway-half)* est implémentée sur une machine attachée

à un réseau. Les deux demi-passerelles communiquent et assurent la fonction d'interconnexion. Cette communication utilise soit une voie simple (liaison directe) soit une voie complexe (réseau). Un cas typique de l'utilisation des demi-passerelles correspond à l'interconnexion de deux réseaux homogènes (supportant des protocoles identiques) à travers un troisième réseau qui supportent des protocoles différents.

- la passerelle est le point de passage de toutes les communications inter-réseaux. Elle peut constituer un goulot d'étranglement si les ressources dont elle dispose ne sont pas suffisantes.

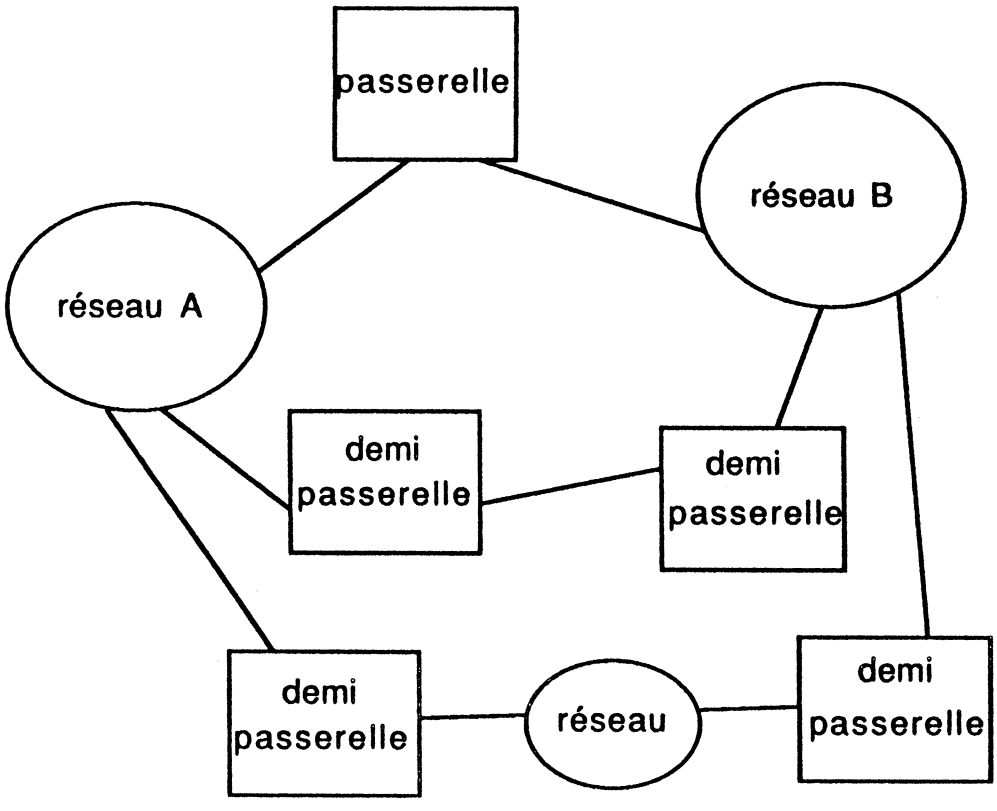


Fig 2.2 Différents types de passerelles

Afin de décrire l'architecture de la passerelle en terme de module, nous considérons l'interconnexion entre deux réeseaux R_a et R_b. Le réseau R_a supporte l'architecture de communication L_a, R_b supporte l'architecture L_b.

Nous ne faisons pour le moment aucune hypothèse sur le degré d'hétérogénéité entre les deux architectures concernées (elles peuvent appartenir à la même famille). La passerelle doit être alors composée de trois modules (Fig 2.3) :

- les deux modules *communicateurs* $L_a(n)$, $L_b(m)$: chacun de ces modules contient l'ensemble ou un sous-ensemble des couches de communication de l'architecture L_a (respectivement L_b), où n (respectivement m) est la couche supérieure supportée par la passerelle. Ce module permet à la passerelle de communiquer avec le réseau R_a au niveau de la couche n et avec le réseau R_b au niveau de la couche m.

- le module *adaptateur* : ce module réalise les fonctions nécessaires à l'interconnexion. Le type d'interaction de l'adaptateur avec les deux modules précédents ainsi que les services fournis dépendent des protocoles utilisés, du niveau d'hétérogénéité et surtout de l'approche retenue pour l'interconnexion. Ainsi pour l'approche *encapsulation* examinée plus loin, le module adaptateur n'est qu'une couche de communication.

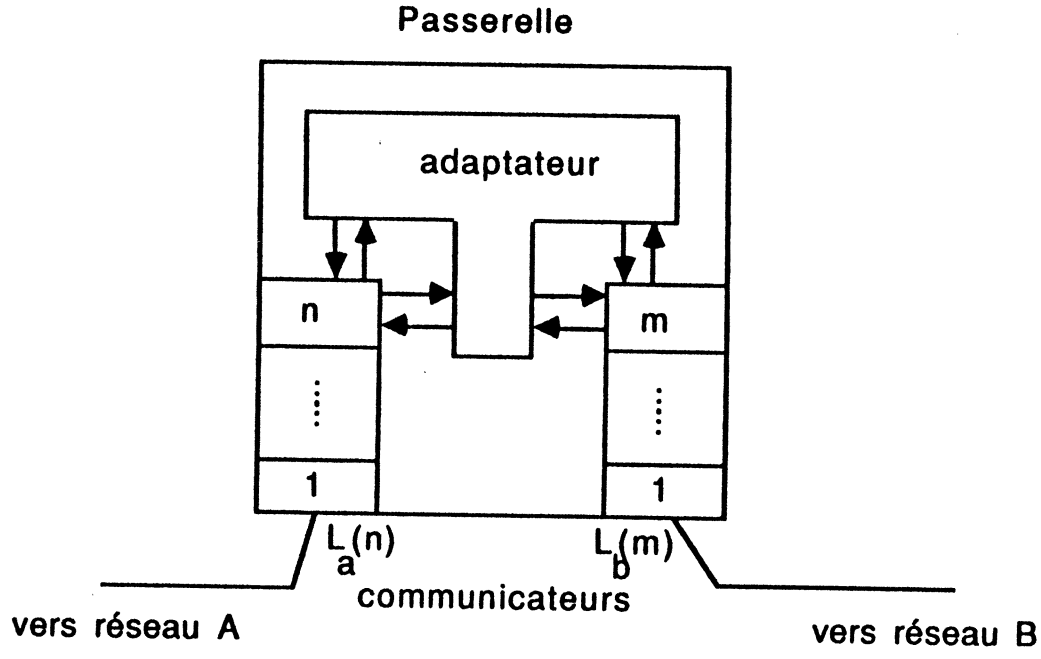


Fig 2.3 Architecture générique d'une passerelle

L'architecture des passerelles décrite ci-dessus permet, grâce au principe de la structuration en couches, de faire abstraction de toute hétérogénéité qui peut exister au niveau inférieur à [n,m]. Le problème se ramène ainsi à l'interconnexion des protocoles [n,m].

La spécification d'une passerelle consiste donc à décrire d'une part les couches supportées par les modules communicateurs, et d'autre part les fonctions de l'adaptateur et son interaction avec les modules communicateurs. Dans la suite de ce chapitre, le module communicateur sera spécifié par la couche supérieure qu'il supporte ; l'existence des couches inférieures est toujours implicite.

La configuration minimale d'une passerelle consiste à supporter seulement les deux couches physiques $L_a(1)$ et $L_b(1)$ et un adaptateur simple. Les deux couches physiques assurent l'attachement physique de la passerelle aux réseaux à interconnecter. La fonction de l'adaptateur est de récupérer les signaux qui arrivent d'un côté pour les amplifier et les réémettre de l'autre côté. Cette configuration minimale correspond en réalité à une forme primitive de passerelle appelée généralement *répéteur*. Le répéteur est utilisé pour interconnecter des réseaux locaux homogènes supportant les mêmes protocoles d'accès au support MAC. L'interconnexion est imposée, dans ce cas, à cause des contraintes physiques telles que la longueur de support ou le nombre maximum de stations.

Nous avons vu qu'une passerelle permet d'interconnecter deux ou plusieurs réseaux. Une généralisation de ce schéma d'interconnexion constitue ce qu'on appelle un *réseau global* composé de plusieurs réseaux interconnectés selon une topologie maillée par des passerelles. C'est une extension du schéma bien connu d'un réseau à commutation de paquets. Dans ce schéma général d'interconnexion (Fig 2.4), les passerelles jouent le rôle des commutateurs inter-réseaux aiguillant les données d'un réseau vers un autre, et réalisant ainsi une fonction de *routage*.

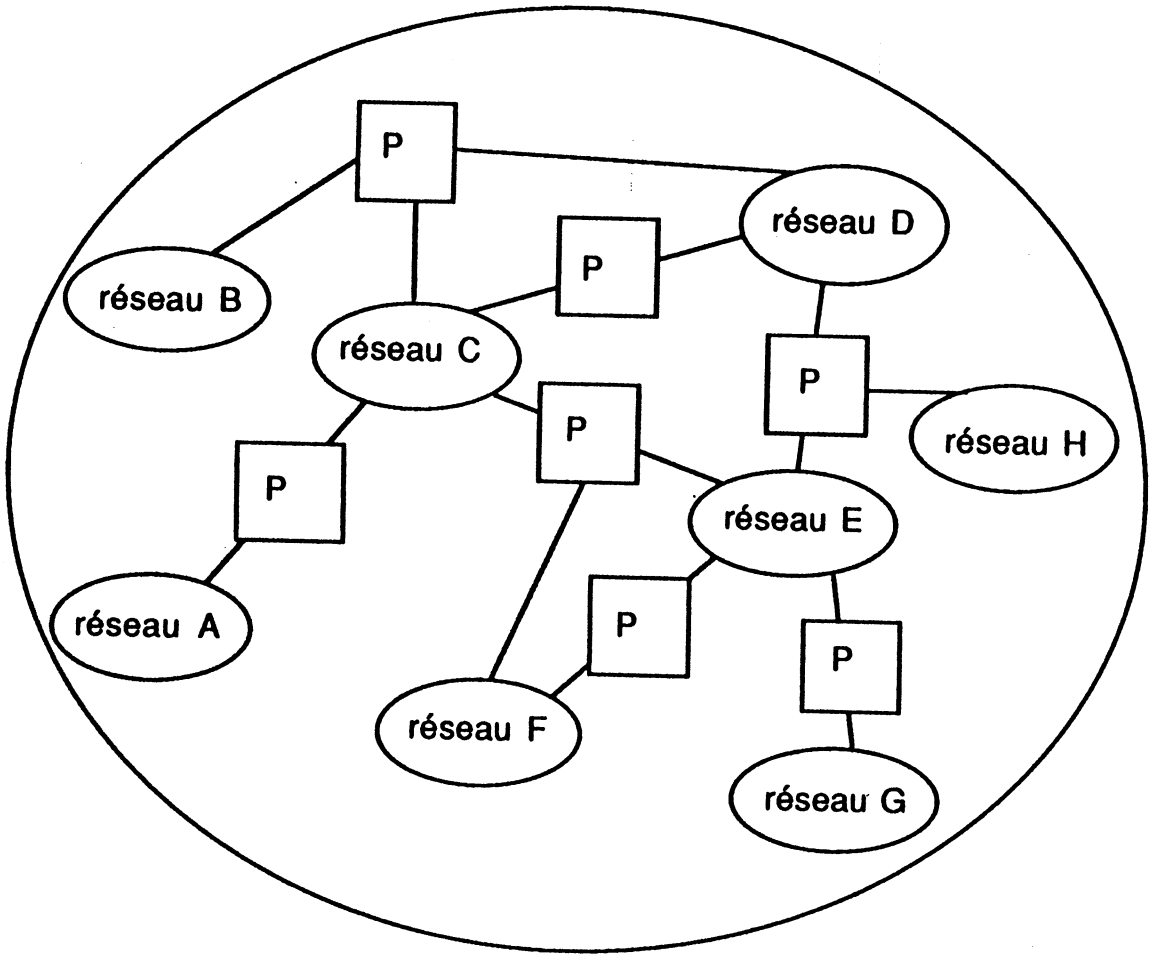


Fig 2.4 Structure d'un réseau global

3.2. Le niveau d'interconnexion

Le *niveau d'interconnexion* est un élément important du problème d'interconnexion car il détermine un niveau n de protocole au-dessus duquel (les niveaux n+1, n+2, ...) les deux réseaux interconnectés apparaissent conceptuellement comme un seul réseau. Supposons une interconnexion au niveau n entre deux réseaux. Ceci signifie que le service de communication offert par le niveau n permet aux entités communicantes au niveau n+1 qui se trouvent sur différents réseaux de communiquer entre elles (effectuer une communication inter-réseaux).

En ce qui concerne la passerelle, le niveau d'interconnexion correspond aux couches supérieures (relatives à chacun des réseaux interconnectés) qui y sont implémentées, soit dans les communicateurs soit dans l'adaptateur. Le niveau d'interconnexion est donc le niveau le plus élevé de la passerelle [Juanole 87]. Il est à noter que si l'architecture de communication n'est pas la même dans les deux réseaux, la numérotation des couches fonctionnellement équivalentes (si une telle correspondance est possible) sera différente. Par conséquent, le niveau d'interconnexion n'a pas la même valeur si on regarde d'un côté ou de l'autre de la passerelle.

La figure 2.5 permet de clarifier la notion de niveau d'interconnexion. La passerelle interconnecte les réseaux R_a et R_b au niveau des couches [n,m]. Elle intervient dans la communication entre les deux réseaux jusqu'à ce niveau. Elle permet donc de cacher l'hétérogénéité existant aux niveaux inférieurs. Pour qu'une communication inter-réseaux ait lieu aux niveaux supérieurs, il faut que les protocoles correspondants soient identiques. Les communications sont de bout en bout à ces niveaux (la passerelle est transparente). Ceci signifie que les structures de données PDU (Protocol Data Unit) et SDU (Service Data Unit) des niveaux [n,m] et des niveaux inférieurs sont manipulées et éventuellement transformées (typiquement pour les couches n et m) par la passerelle. Par contre, les structures des niveaux supérieurs traversent la passerelle d'une manière transparente. La communication au niveau [n,m] (le niveau d'interconnexion) est de *proche en proche.*

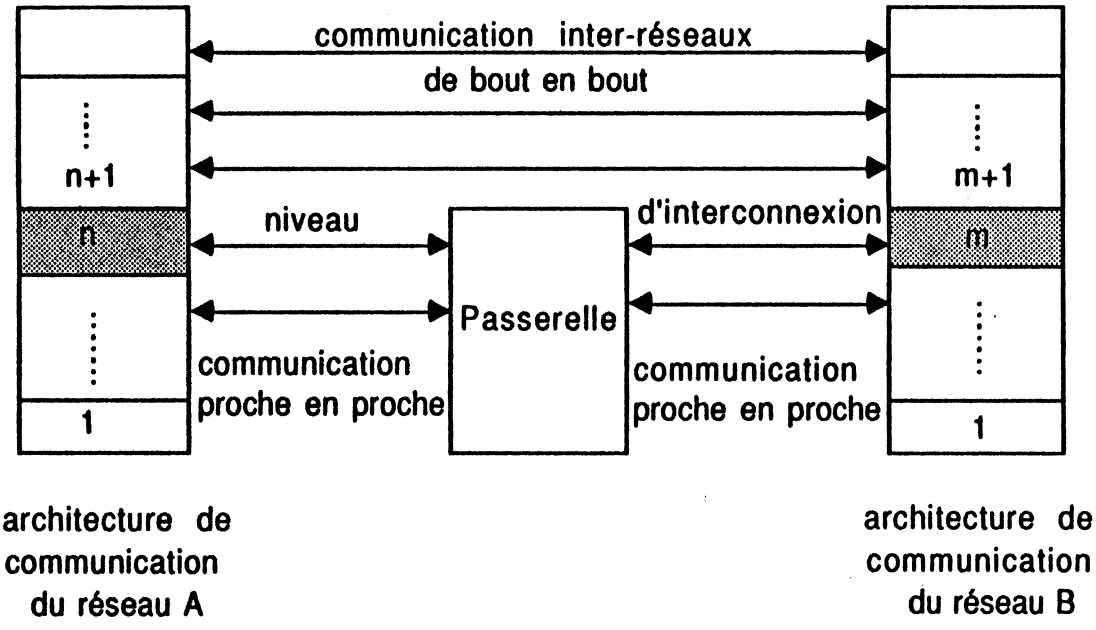


Fig 2.5 La notion de niveau d'interconnexion

Le choix d'un niveau d'interconnexion pour une configuration donnée est lié au niveau d'hétérogénéité. Si on considère le cas où les deux réseaux R_a et R_b sont hétérogènes jusqu'au niveau [n,m], l'interconnexion doit se faire *au moins* à ce niveau. La structure interne de la passerelle est alors constituée des deux modules communicateurs implémentant l'un les couches [1...n], l'autre les couches [1...m], et du module adaptateur. En l'absence de contraintes particulières et pour des raisons de performance et d'efficacité, l'interconnexion doit se faire au niveau le plus bas possible. Cela minimise le nombre de couches implémentées dans la passerelle. Par conséquent, le nombre total de couches traversées et le nombre d'opérations de codage et de décodage que subissent les messages échangés entre les deux réseaux sera réduit. Ceci a pour effet d'améliorer le délai d'acheminement des données entre les deux réseaux interconnectés.

Interconnexion des réseaux locaux au niveau du protocole d'accès au support :

Nous avons décrit dans un paragraphe précédent la passerelle de type *répéteur* permettant une interconnexion au niveau physique. Afin d'illustrer la notion de niveau d'interconnexion et d'introduire quelques problèmes liés à l'interconnexion, nous présentons dans ce paragraphe le fonctionnement d'une passerelle de type *pont (bridge)*. Un *pont* est une passerelle qui assure l'interconnexion de deux (ou plusieurs) réseaux locaux au niveau de leurs protocoles d'accès au support MAC. Il s'agit donc d'interconnecter deux réseaux R_a et R_b qui utilisent deux protocoles d'accès au support différents MAC_a et MAC_b, le protocole au niveau supérieur (niveau LLC) étant supposé le même sur les deux réseaux. Les deux modules communicateurs du pont supportent les couches MAC_a et MAC_b. La fonction essentielle du module adaptateur est une fonction de *filtrage*. Cette opération est basée sur l'adresse source et l'adresse destinataire contenues dans les trames circulant sur les réseaux. Les actions suivantes résument le fonctionnement du module adaptateur dans le pont :

- il doit récupérer, par l'intermédiaire du protocole MAC_a (respectivement MAC_b), les trames qui circulent sur le réseau R_a (respectivement R_b).

- il doit identifier et stocker celles qui sont émises par le réseau R_a (respectivement R_b) et qui sont adressées au réseau R_b (respectivement R_a). C'est une action de filtrage.

- il réémet vers leur réseau destinataire respectif les trames filtrées, en utilisant les protocoles MAC correspondant à chaque réseau.

L'intérêt de l'interconnexion au niveau MAC réside dans le fait que le pont n'introduit aucune modification sur les PDU de la sous-couche LLC qui le traversent. Ainsi au niveau LLC, les deux réseaux apparaissent virtuellement comme étant un seul réseau étendu. Le pont est donc complètement transparent aux stations connectées aux réseaux.

Le filtrage effectué par l'adaptateur nécessite un mécanisme permettant d'identifier les trames qui doivent traverser le pont pour passer d'un réseau à un autre. Cela peut se fait de la manière suivante :

le pont contient une table qui maintient des associations entre les adresses des différentes entités MAC (les différentes stations) et les réseaux auxquels elles sont attachées. Cela permet l'utilisation d'un espace d'adressage plat, mais nécessite une mise à jour de cette table à l'occasion de toute modification d'adresse ou de l'introduction d'une nouvelle station. La création et la mise à jour de la table peut s'effectuer soit par l'administrateur réseau, soit automatiquement par le pont qui travaille alors en mode espion en observant le trafic de chaque réseau. Il suffit alors pour remplir cette table d'y enregistrer l'adresse source de chaque trame circulante et le réseau sur lequel elle a été observée [Hawe 84].

L'implémentation du pont dans un contexte donné d'interconnexion peut poser certains problèmes techniques si les caractéristiques des deux réseaux telles que le débit, le délai d'acheminement des trames, le taux d'erreurs, etc..., sont très différentes. En particulier, un problème de congestion peut apparaître si le délai moyen nécessaire pour l'envoi d'une trame sur l'un des deux réseaux est sensiblement inférieur au délai moyen de réception sur l'autre réseau. Nous pouvons modéliser le comportement de pont de ce point de vue par deux files d'attente, l'une en réception sur un réseau et l'autre pour l'émission sur l'autre réseau. Il faut donc assurer que la longueur moyenne de chacune des files d'attente n'excède pas le nombre de tampons (buffers) disponibles.

Une autre application de l'interconnexion par un pont est le cas d'interconnexion de deux réseaux locaux homogènes. Cette solution s'avère, dans certains cas, plus intéressante que leur intégration dans un seul réseau (ou l'utilisation des répéteurs),en particulier grâce à la possibilité de contrôler et éventuellement de filtrer les données échangées entre les réseaux.

3.3. Hétérogénéité des protocoles de communication

Le problème d'hétérogénéité des architectures et des protocoles de communication des réseaux est l'obstacle essentiel à franchir pour assurer leur interconnexion. Par la présence et l'architecture des passerelles qui interconnectent les réseaux, ce problème est réduit au problème d'hétérogénéité entre les protocoles placés au niveau d'interconnexion. L'hétérogénéité entre deux protocoles de communication présente plusieurs aspects :

1) hétérogénéité sémantique

C'est la différence qui se manifeste au niveau des services qui sont offerts par l'un et l'autre des protocoles interconnectés dont voici des exemples :

- une interconnexion entre un protocole offrant un service en mode connexion et un autre qui offre un service en mode sans connexion.

- une interconnexion entre un protocole qui supporte l'envoi de données express, et un autre qui supporte seulement l'envoi de données normales.

- une interconnexion entre un protocole qui fournit un service de transmission en mode multipoint et diffusion et un protocole qui ne fournit que le service de communication point-à-point.

Cette hétérogénéité, qui est liée à l'existence d'un éventail de services possibles, apparaît d'une manière plus importante encore au niveau des couches hautes de communication où les protocoles ont des fonctions de traitement de données.

2) hétérogénéité syntaxique

Il s'agit ici de protocoles qui offrent des services équivalents du point de vue sémantique, mais l'hétérogénéité apparaît, d'une part, au niveau de l'interface de ces services (les primitives d'accès aux services et leurs paramètres) et d'autre part, au niveau du format des PDU qu'utilise chaque protocole.

3) hétérogénéité quantitative

Il s'agit ici de services équivalents existant dans les protocoles interconnectés, mais qui nécessitent une certaine adaptation ou réglage. A titre d'exemple on peut noter :

- la taille maximale des données véhiculées,
- les paramètres de contrôle de flux.

Le problème de la différence des tailles maximales des PDU-données et les mécanismes de segmentation/réassemblage ainsi que le réglage des paramètres de contrôle de flux dans un environnement d'interconnexion seront traités dans la section suivante.

4. Quelques problèmes liés à l'interconnexion

L'interconnexion de réseaux fait apparaître certains problèmes. Ces problèmes ne sont pas nouveaux, ils existent dans chaque environement de communication. Il s'agit en réalité d'examiner comment les mécanismes associés peuvent être étendus à un environement d'interconnexion où deux éléments nouveaux interviennent : l'introduction des passerelles et l'hétérogénéité des protocoles.

4.1. Adressage et routage

Bien que nous nous intéressions ici au problème d'adressage lié à l'interconnexion des réseaux, il convient d'éclaircir les concepts de base de l'adressage à l'aide d'une définition d'ordre général comme celle donnée dans [Shoch 78]. Shoch définit trois termes *nom, adresse,* et *route*: "un nom désigne une ressource, une adresse désigne sa localisation et une route indique comment la retrouver".

Dans un environnement de réseaux interconnectés, le *nom* est un symbole convivial permettant de désigner un utilisateur, un programme d'application, ou un serveur avec lequel un autre utilisateur souhaite communiquer. L'*adresse* est une structure de données respectant un format précis et permettant de déterminer la localisation d'une entité communicante sur le réseau global [Danthine 82]. En dehors du contexte d'interconnexion, le schéma d'adressage dans un réseau (quels que soient son architecture et le niveau de communication considéré) doit permettre de localiser une entité communicante sur ce réseau.

Une relation existe entre le nom et l'adresse ; en effet, l'adresse est obtenue par manipulation du nom. Cette manipulation fait généralement appel à *un serveur de nom* qui contient des informations de correspondance entre les noms et les adresses. Ce problème entre dans un cadre plus général que celui de l'interconnexion des réseaux. C'est pourquoi nous le laissons de côté pour nous intéresser aux problèmes d'adressage et de routage qui sont propres à l'environnement des réseaux interconnectés.

Une *route* est le chemin (ensemble de passerelles traversées) que doivent emprunter les données pour atteindre leur destination finale déterminée par la

source. Si les données traversent une seule passerelle, la route est appelée *route directe*. Si plusieurs passerelles sont traversées la route est alors appelée *route indirecte*. La route peut être définitivement spécifiée par la source, c'est le *routage source*, ou alors spécifiée progressivement par les différentes passerelles traversées c'est le *routage de proche en proche*.

4.1.1. Le mécanisme de routage dans un environement d'interconnexion

La décision de routage consiste à déterminer, à partir de certaines informations correspondant généralement à l'adresse destinataire, le prochain système vers lequel il faut rediriger les données reçues.

Nous distinguons deux centres de décision concernant le routage. Le premier est la station hôte source. Le protocole de communication implémenté sur cette dernière analyse l'adresse de destination. Cette analyse permet d'identifier si les PDU sont destinés à une station hôte du sous-réseau local ou à une station d'un sous-réseau distant, auquel cas il faut acheminer les PDU vers la passerelle, et si elle n'est pas unique, choisir la passerelle appropriée. Le deuxième centre de décision est constitué par les passerelles. La décision de routage effectuée consiste alors à déterminer la prochaine passerelle vers laquelle il faut acheminer les PDU si le sous-réseau destinataire n'est pas directement connecté à la passerelle locale.

La décision de routage présente deux aspects différents :

- le premier consiste à identifier, à partir de certaines informations comme l'adresse destinataire, la prochaine passerelle vers laquelle il faut rediriger les données sans qu'il y ait d'alternatives possibles (plusieurs chemins possibles). La décision est appelée *déterministe*. Un cas particulier, mais très courant dans une topologie d'interconnexion simple, est le cas où la prochaine passerelle est unique. La décision de routage est alors simple. Il consiste à déterminer s'il faut envoyer les données vers cette passerelle.

- le deuxième aspect concerne le cas où une ou plusieurs alternatives sont possibles. C'est le cas d'une topologie d'interconnexion complexe où les passerelles jouent le rôle de commutateurs inter-réseaux dans un réseau global constitué d'un ensemble de réseaux interconnectés. Il s'agit alors de faire un choix parmi les chemins possibles. La décision est alors plus

complexe et fait appel à des algorithmes de routage semblables à ceux utilisés dans un réseau à commutation de données.

Une relation existe entre le schéma d'adressage et la fonction de routage. Certains schémas d'adressage peuvent permettre d'exploiter efficacement la sémantique associée aux adresses pour en tirer des informations concernant le routage.

4.1.2. Différents schémas d'adressage et l'interconnexion de réseaux

Un schéma d'adressage dans un environnement d'interconnexion doit permettre la constitution d'un espace d'adressage pour le réseau global, il doit donc fournir le moyen de désigner une entité communicante sur le réseau global. Cette désignation globale n'est possible qu'au niveau d'interconnexion qui offre le service de communication inter-réseaux et la visibilité du réseau global. Nous considérons, comme étant un service fourni, la possibilité de désigner les entités communicantes sur un sous-réseau. Il s'agit donc d'examiner la possibilité d'étendre ce service sur le réseau global.

Deux aspects du problème d'adressage peuvent être distingués dans un environnement d'interconnexion. Le premier est lié au fait que les schémas d'adressage des différents sous-réseaux interconnectés sont hétérogènes, soit au niveau de la sémantique que portent les adresses, soit au niveau de leurs formats. Le deuxième aspect est lié à la nécessité de créer un espace de désignation global même si les structures d'adresses sur les différents sous-réseaux sont uniformes.

Trois schémas d'adressage pour l'interconnexion des réseaux sont identifiés :

1) adressage plat (appelé aussi "adressage absolu"),
2) adressage hiérarchique,
3) adressage par représentant.

4.1.2.1. Adressage plat

Ce schéma d'adressage correspond à des adresses non structurées. Aucune sémantique n'est associée au contenu de l'adresse. Un exemple de ce schéma est l'adressage de machines connectées à un réseau Ethernet. Il est à noter que

l'adresse ne contient pas d'information concernant le routage et, plus particulièrement, concernant la localisation du destinataire. Cela rend ce schéma d'adressage inutilisable directement dans un environnement d'interconnexion. Il est donc nécessaire d'accéder à *une table de correspondance* implémentée dans chaque station hôte et dans chaque passerelle. Cette table contient les relations d'association entre les adresses des destinataires et la passerelle à travers laquelle les PDU doivent être envoyés. Le nombre d'entrées dans cette table correspond au nombre de stations dans le réseau global. Le problème qui se pose alors est la mise à jour de ces tables lors des modifications (attachement et détachement des machines). Cette mise à jour est réalisée généralement par l'administrateur du réseau. Concernant la gestion de ces tables, on peut noter le cas particulier où la communication au niveau de l'interconnexion se fait naturellement en mode diffusion (c'est le cas d'interconnexion au niveau MAC de type CSMA/CD). Dans ce cas, la gestion des tables de correspondance est simplifiée : d'une part il n'est plus nécessaire d'implémenter la table dans les stations hôtes, d'autre part le remplissage et la mise à jour de cette dernière peut se faire automatiquement en mode *espion*. Un exemple typique de ce fonctionnement est le pont d'interconnexion au niveau MAC décrit précédemment.

Un inconvénient d'une stratégie d'adressage plat est l'obligation d'assurer l'unicité des adresses sur le réseau global. Par conséquent, une seule autorité administrative doit s'occuper de l'attribution des adresses à l'ensemble des stations hôtes connectées aux différents sous-réseaux.

4.1.2.2. Adressage hiérarchique

L'idée consiste à partitionner l'espace d'adressage en sous-espaces, ces derniers pouvant être aussi partitionnés, et ainsi de suite d'une manière récursive. Chaque partition est désignée par un identificateur. Ceci donne une organisation arborescente de l'espace d'adressage. Pour désigner globalement un élément appartenant à un sous-espace, il faut déterminer l'identificateur de l'espace contenant l'identificateur du sous-espace et enfin, l'identificateur local de l'élément. En général, le partitionement peut se faire selon plusieurs arguments. Mais, afin d'exploiter un tel schéma d'adressage dans un environnement de réseaux interconnectés, le partitionnement doit refléter la structure hiérarchique d'interconnexion. Ainsi, on associe un espace d'adressage au niveau du réseau global, un sous-espace au niveau de chaque

sous-réseau et, enfin, un sous-(sous-espace) à chaque station hôte. Ceci permet de désigner une entité communicante parmi plusieurs sur cette station. La structure de l'adresse aura donc le format suivant :

<adresse globale>=<id-sous-réseau> <id-station hôte> <id-entité>

La sémantique contenue dans l'adresse indique donc la localisation d'une entité communicante. L'identificateur de sous-réseau désigne le sous-réseau destinataire. L'identificateur de la station hôte permet de désigner une station hôte attachée à ce sous-réseau. Cet identificateur ne correspond pas nécessairement à l'adresse de la station destinatrice (au sens adresse physique) selon le schéma d'adressage propre au sous-réseau, mais il doit permettre de retrouver ce dernier. L'adresse utilisateur désigne une entité communicante dans la station destinatrice (c'est l'équivalent de la notion de point d'accès au service dans le modèle OSI).

L'adressage hiérarchique présente beaucoup d'intérêt, ce qui explique son utilisation courante dans les environnements d'interconnexion. L'adressage X121 pour les réseaux publics, et l'adressage *internet* dans l'architecture DARPA ainsi que les structures des adresses NSAP OSI sont trois exemples importants d'adressage hiérarchique. Cette stratégie d'adressage présente les avantages suivants :

- l'allocation des adresses et leur gestion sur chaque sous-réseau se fait d'une manière indépendante des autres sous-réseaux, car chaque sous-réseau possède son propre sous-espace d'adressage.

- l'introduction de nouveaux réseaux est relativement simple du point de vue de l'adressage. Il suffit d'associer un identificateur à ce sous-réseau. L'adresse globale d'une station hôte est alors obtenue en concaténant l'adresse locale de la station et l'identificateur de sous-réseau.

- une partie des informations nécessaires pour effectuer le routage est contenue directement dans l'adresse, en particulier dans le champ "identificateur du sous-réseau destinataire". Pour déterminer la passerelle appropriée, la station hôte source doit consulter une table de routage. Comparée à la stratégie d'adressage plat, la table de routage contient un nombre réduit d'entrées, qui est le nombre de sous-réseaux dans le réseau

global. Chaque entrée établit une association entre un identificateur de sous-réseau et l'adresse de la passerelle qui permet d'y accéder. L'adresse de la passerelle appartient à l'espace d'adressage du sous-réseau local.

4.1.2.3. Adressage par représentant

Les deux schémas d'adressage présentés ci-dessus peuvent être utilisés pour l'adressage à travers le réseau global dans un environnement d'interconnexion, à condition qu'au niveau de l'interconnexion le même schéma et le même format d'adressage soient supportés par l'ensemble des sous-réseaux. Cela n'est vrai a priori que si les différents sous-réseaux supportent la même famille de protocoles, ce qui est le cas de l'interconnexion des réseaux partiellement hétérogènes. Les précédents schémas d'adressage ne peuvent donc pas être appliqués si les différents sous-réseaux supportent des structures hétérogènes d'adresses, ce qui est le cas généralement des réseaux totalement hétérogènes. Il s'agit donc de définir une méthode permettant l'adressage inter-réseaux sachant que chaque sous-réseau continue à utiliser son propre, et éventuellement différent, schéma d'adressage. Cela peut être réalisé par la technique d'adressage par *représentant (proxy)*.

Le principe de l'adressage par représentant est simple : pour chaque entité communicante distante (résidant sur un sous-réseau différent) accessible par une passerelle, une adresse selon le schéma local (le schéma d'adressage dans le sous-réseau local) est associée dans la passerelle. Chaque passerelle contient donc des représentants, en termes d'éléments appartenant à l'espace d'adressage local, de l'ensemble des entités communicantes accessibles. Elle contient en plus une table qui réalise la correspondance entre les adresses des représentants selon le schéma local et leurs adresses selon le schéma d'adressage du réseau distant. Cette table est mise à jour à chaque modification ou introduction de nouvelles entités communicantes.

Le schéma d'adressage par représentant devient lourd à gérer quand le nombre de sous-réseaux devient important et quand leur structure d'interconnexion devient complexe. Toutefois, si les structures d'adresse utilisées dans les sous-réseaux ne permettent pas une partition hiérarchique, le schéma par représentant reste la seule solution générale possible au problème d'interconnexion dans un environnement totalement hétérogène.

4.2. Segmentation et réassemblage

Les caractéristiques des différents sous-réseaux interconnectés pour former un réseau global imposent l'utilisation d'unités de données de protocole (PDU) dont la taille peut être différente pour chaque sous-réseau (ce qui nous intéresse ici est la différence de taille des PDU de données qui concernent les protocoles au niveau de l'interconnexion). Le choix de la taille maximale de ces PDU pour un sous-réseau peut être influencé par plusieurs paramètres :

- la taille des buffers de mémorisation ;
- le taux d'erreurs qui peut exiger une petite taille de PDU pour augmenter la probabilité d'une transmission correcte des PDU ;
- le nombre de bits alloués au champ longueur de données qui détermine la taille maximale des PDU ;
- la performance de la communication ;
- la conformité à certains standards.

Cette différence de taille des PDU de données peut poser un problème lors de l'interconnexion. Il s'agit d'acheminer des données utilisateurs (SDU selon la terminologie ISO) générées dans une station hôte à travers une suite de sous-réseaux qui permettent de véhiculer des PDU dont la taille maximale est différente et généralement inférieure à la taille des données à acheminer. La solution de ce problème est basée sur l'extension de la technique de segmentation (ou fragmentation) et réassemblage, bien connue dans la communication intra-réseau, pour l'appliquer dans un environnement d'interconnexion à la communication inter-réseaux. La segmentation consiste à découper des données utilisateurs ou venant d'une couche supérieure (SDU) en segments (chaque segment est inclus dans un PDU) et à les réassembler dans la station destinatrice pour reformer le SDU initial. La décision de segmentation dans un environnement d'interconnexion doit être prise dans la passerelle qui se trouve en possession de données à router vers une autre passerelle à travers un sous-réseau qui ne supporte pas des PDU d'une telle taille. Trois politiques sont proposées pour répondre à ce problème [Shoch 79] [Callon 83] :

1) Choix d'une route appropriée

Cette politique consiste à choisir une route, par la source, à travers les

sous-réseaux de telle sorte que la taille des données véhiculées soit inférieure à la taille maximale des PDU que supporte chacun des sous-réseaux traversés. Cette stratégie présente l'avantage d'éviter la segmentation et le réassemblage mais a l'inconvénient de compliquer les algorithmes de routage et surtout de ne pas être applicable dans le cas où aucune route ne vérifie la propriété recherchée. Une telle politique ne peut donc être retenue que comme complément à l'une des deux politiques de segmentation présentées ci-dessous.

2) Segmentation intra-réseau

Cette politique consiste à réaliser une segmentation des données en fonction de la taille maximale des PDU dans chaque passerelle traversée. Cette taille est celle admise par le prochain sous-réseau qui véhiculera les segments vers la prochaine passerelle qui les réassemblera pour former les données (SDU) d'origine. La fonction de segmentation est généralement réalisée par le protocole de communication intra-réseau (spécifique au sous-réseau traversé), à condition qu'il supporte le service de segmentation ou alors par le niveau supérieur. Les avantages de la segmentation intra-réseau sont les suivants :

- le module adaptateur dans la passerelle est déchargé de la fonction de segmentation (dans le cas où le protocole de communication sur chaque réseau est capable de segmenter).

- dans l'approche d'interconnexion basée sur l'encapsulation, le module adaptateur est une couche de communication. Dans ce cas, la stratégie de segmentation intra-réseau évite de dupliquer l'en-tête associé à la couche d'encapsulation dans chaque segment.

- on obtient une bonne efficacité d'utilisation de la voie de communication, car la longueur des segments est toujours déterminée de proche en proche en fonction des sous-réseaux traversés.

Cependant cette stratégie présente plusieurs inconvénients :

- la segmentation et le réassemblage se font dans chaque passerelle. Cela peut avoir une conséquence néfaste au niveau de la performance. Cette stratégie aboutit à l'augmentation du délai moyen d'acheminement des données de bout en bout. Cela est dû, d'une part, à l'accroissement du délai

de traitement dans les passerelles à cause de la segmentation et, d'autre part, à l'introduction d'un temps mort qui représente le délai d'attente nécessaire à l'arrivée de tous les segments.

- dans certains cas, l'opération de réassemblage réalisée par une passerelle intermédiaire s'avère inutile. C'est le cas où la passerelle segmente les données avec une longueur de segment équivalente à celle des données qui viennent d'être réassemblées (deux réseaux successifs utilisent la même taille).

3) Segmentation inter-réseaux

Cette politique consiste à segmenter les données dans la station source ou dans une passerelle intermédiaire pour être réassembler dans la station destinatrice. Il n'y a donc pas de réassemblage dans les passerelles intermédiaires. Des segments peuvent être eux-même segmentés par une passerelle quelconque sur la route, si nécessaire. Cette politique a les avantages suivants :

- réduire le nombre d'opérations de segmentation et de réassemblage. La segmentation est réalisée seulement quand c'est nécessaire. Quant au réassemblage, il n'est effectué qu'une seule fois.

- permettre un routage indépendant pour chaque segment. Cela est vrai pour des protocoles de communication intra-réseau en mode sans connexion.

Deux inconvénients apparaissent dans cette stratégie de segmentation :

- il est nécessaire que les protocoles de segmentation et de réassemblage soient identiques sur l'ensemble des systèmes traversés afin de pouvoir réassembler les différents segments à l'arrivée.

- dans le cas de l'utilisation d'une couche d'encapsulation pour l'inter-connexion, la segmentation est effectuée par celle-ci. Cela signifie que l'en-tête associé à cette couche sera dupliqué dans chaque segment.

4.3. Contrôle de flux et contrôle d'erreurs

Le contrôle d'erreurs est basé généralement sur l'envoi d'acquittements de la part du destinataire vers la source. La fonction de contrôle de flux est basée sur le principe des *fenêtres,* qui doit assurer la régulation de flux de données entre la source et la destination. Ce qui nous intéresse ici est l'influence de l'interconnexion sur ces deux fonctions. Les aspects communs à ces deux fonctions seront considérés, car les acquittements sont généralement aussi utilisés pour véhiculer des paramètres de contrôle de flux tels que le *crédit.*

Nous avons vu que l'interconnexion entre deux réseaux met en jeu au niveau d'interconnexion une concaténation de deux communications intra-réseau pour composer la communication inter-réseaux. Compte tenu de cette concaténation, on peut mettre en œuvre, concernant la fonction de contrôle de flux et la fonction de contrôle d'erreurs, soit un contrôle de proche en proche soit un contrôle de bout en bout.

La figure 2.6 montre la signification de proche en proche et la signification de bout en bout pour la fonction de contrôle d'erreurs. Quand cette fonction est réalisée de proche en proche, la passerelle transmet l'acquittement vers la source dès qu'elle reçoit les données, alors que, dans le cas d'un fonctionnement de bout en bout, l'acquittement n'est envoyé qu'à la réception de l'acquittement par le destinataire.

Les conséquences de l'utilisation de l'un ou l'autre mode de fonctionnement apparaissent essentiellement au niveau de deux paramètres : le délai de transmission des données et la fiabilité de la communication. Un fonctionnement de bout en bout évite les pertes non signalées des PDU mais aboutit à un délai de transmission de données relativement long (à noter que ce fonctionnement implique la modification du délai de temporisation avant retransmission). Par contre, le fonctionnement de proche en proche permet de d'envoyer en cascade des PDU en diminuant ainsi le délai de transmission de données par rapport au fonctionnement de bout en bout. Mais dans ce cas, il existe une possibilité de perte des PDU non signalée à la source. Cette perte apparaît quand un PDU relayé par la passerelle se perd alors que celle-ci avait transmis son acquittement vers la source.

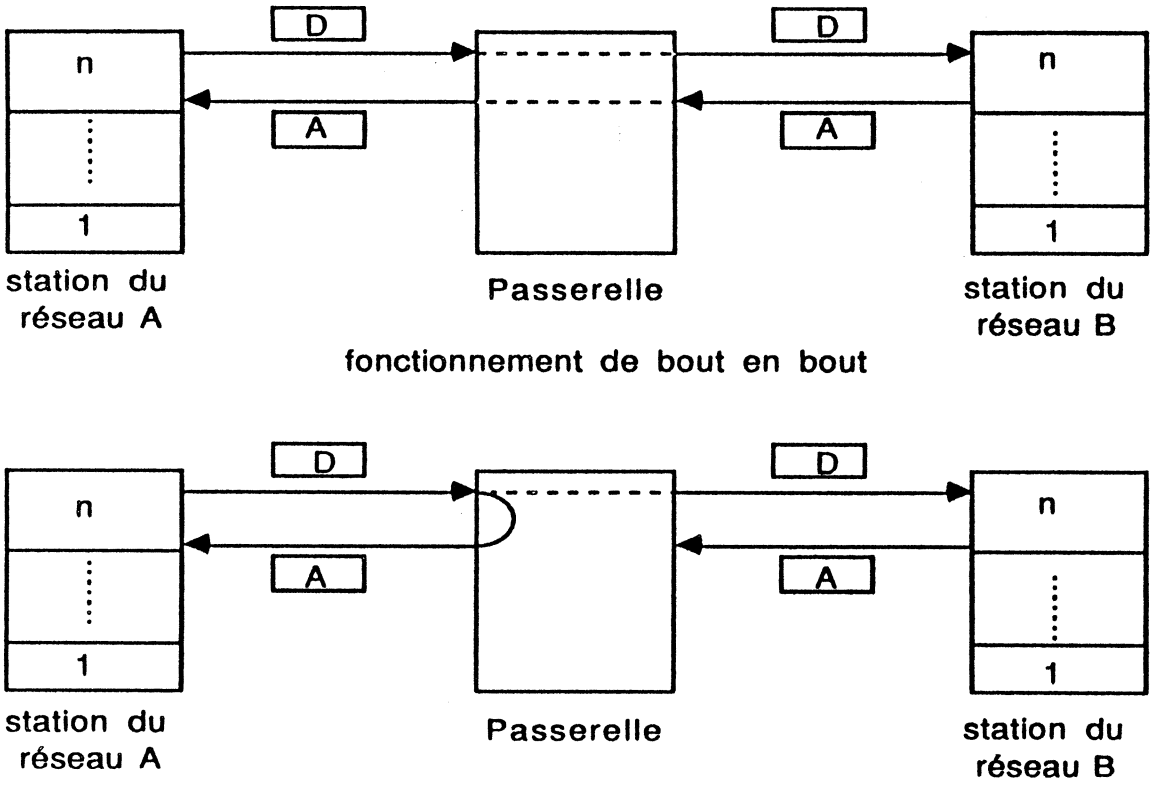


Fig 2.6 Contrôle d'erreurs au niveau d'interconnexion

Le fonctionnement de proche en proche implique une gestion par la passerelle afin d'adapter les paramètres de contrôle de flux de chacune des deux communications de proche en proche. Cette gestion doit assurer un délai de stockage minimal des PDU dans la passerelle. Pour cela il faut que la fenêtre d'émission (la possibilité d'émettre) du côté de chaque réseau soit supérieure à la fenêtre de réception (la possibilité de recevoir) du côté de l'autre réseau.

Nous avons examiné dans les paragraphes précédents l'ensemble des éléments à prendre en compte lors de l'interconnexion. Le choix entre les différentes solutions envisageables pour chaque problème ne peut pas se faire séparément des autres problèmes. L'ensemble de ces choix dépend essentiellement de l'approche architecturale et fonctionnelle retenue pour la réalisation de l'interconnexion qu'on appelle la technique d'interconnexion.

5. Les techniques d'interconnexion

Nous avons défini la passerelle comme étant un système intermédiaire entre deux ou plusieurs réseaux et qui permet leur interconnexion. Une architecture générique de la passerelle a été décrite dans le paragraphe 3.1 où nous avons défini sa structure, composée de deux modules communicateurs et d'un module adaptateur. Le nombre de couches de communication dans les modules communicateurs dépend du niveau d'interconnexion, alors que les fonctions du module adaptateur dépendent de la technique utilisée pour l'interconnexion.

L'objet de ce paragraphe est d'étudier les approches de base utilisées pour l'interconnexion et, pour chaque approche, de décrire les fonctions essentielles de l'adaptateur. Nous présenterons aussi les caractéristiques de chaque technique, son domaine d'application en fonction des cas d'interconnexion et les conditions nécessaires pour sa réalisation. Des exemples réels de chaque technique seront donnés dans les paragraphes suivants. Mais auparavant, il nous semble nécessaire de définir quelques paramètres ou critères qui permettent d'évaluer l'intérêt de chaque approche et de guider le choix vers un modèle d'interconnexion dans un contexte donné.

5.1. Critères d'évaluation

1) Localisation des fonctions d'interconnexion :

Les techniques d'interconnexion sont basées sur l'utilisation de la passerelle, mais, dans certains cas, il est nécessaire pour les stations hôtes de supporter une partie des fonctions nécessaires à la réalisation de l'interconnexion. Dans ce cas, les hôtes doivent implémenter des logiciels de communication supplémentaires qui coopèrent avec les logiciels qui s'exécutent dans la passerelle, afin d'assurer l'interconnexion et d'offrir le service de communication externe (communication inter-réseaux). Dans certains cas, une telle modification peut apparaître comme une contrainte qui limite l'intérêt de l'approche choisie pour l'interconnexion.

2) Visibilité et interface

Du point de vue de l'utilisateur de la communication inter-réseaux, la

passerelle doit rester un élément transparent. Cela se manifeste surtout au niveau de l'adressage qui doit permettre d'identifier directement l'entité communicante distante. Cela étant, la technique d'interconnexion choisie et le schéma d'adressage utilisé peuvent donner lieu à des différents degrés de visibilité :

- l'utilisateur doit connaître la topologie d'interconnexion,
- l'utilisateur est conscient de traverser les frontières de son réseau pour communiquer avec un réseau distant, sans pour autant avoir besoin de connaître la structure détaillée de l'interconnexion,
- l'utilisateur a la visibilité virtuelle d'un seul réseau ; une abstraction complète de l'interconnexion est réalisée.

En ce qui concerne l'interface et le service de communication, deux cas peuvent être distingués :

- le premier cas consiste à ne pas modifier l'interface de communication interne. Cela signifie que la technique d'interconnexion qui est mise en œuvre assure une interface et un service de communication externe (inter-réseaux) similaires à ceux de la communication interne (intra-réseau).

- dans le deuxième cas, les deux services de communication interne et externe sont indépendants et donc a priori différents. L'utilisateur doit donc manipuler deux interfaces de communication, selon que l'entité communicante distante est locale ou externe à son réseau.

3) Modularité et évolution

Ce paramètre indique si la technique d'interconnexion utilisée permet l'extension. Cette extension peut se faire dans deux directions :

- la première concerne la possibilité d'interconnecter, toujours selon la même technique, de nouveaux réseaux hétérogènes avec un minimum de modification du logiciel d'interconnexion, qu'il soit implementé dans la passerelle ou dans les stations hôtes.

- la deuxième concerne la possibilité d'appliquer la même technique

d'interconnexion, avec un minimum de modification, pour réaliser l'interconnexion à des niveaux différents.

En d'autres termes, le critère de modularité indique le volume de modifications à apporter à l'approche utilisée et à sa réalisation pour les généraliser à différents réseaux (hétérogénéité) ainsi qu'à différents niveaux d'interconnexion.

4) Domaine d'application

Deux paramètres permettent de cerner le domaine d'application d'une technique d'interconnexion :

- le niveau d'hétérogénéité des réseaux interconnectés : cela indique si la technique utilisée peut être appliquée dans le cas des réseaux faiblement hétérogènes, dans celui des réseaux totalement hétérogènes ou indifféremment dans les deux cas .

- le niveau d'interconnexion : il s'agit d'identifier si la technique en question est adaptée à une interconnexion au niveau des couches hautes ou des couches basses des protocoles.

5) Performances

Plusieurs paramètres peuvent être considérés comme, par exemple, le délai moyen d'acheminement des données au niveau d'interconnexion. En ce qui concerne la communication inter-réseaux, la performance diminue avec l'accroissement des encapsulations et des décapsulations (e.g, le nombre de couches traversées) que subissent les données échangées. Par conséquent, le nombre de couches implémentées dans la passerelle et la complexité du travail réalisé par le module adaptateur ont une influence importante sur les performances de la communication inter-réseaux.

Le critère de performance concerne également la communication interne (intra-réseau). En effet, certaines solutions d'interconnexion supposent d'implémenter dans les stations hôtes une part du logiciel d'interconnexion sous forme d'une couche de communication. Si cette couche est insérée dans l'ensemble des couches de communication interne, la communication externe

(inter-réseaux) introduira une modification des performances de la communication interne.

En plus de l'ensemble des paramètres définis ci-dessus, d'autres d'ordre plus général peuvent être pris en compte pour évaluer les solutions proposées, tels que la fiabilité de la communication inter-réseaux ainsi que le coût et la complexité de la mise en œuvre de ces solutions.

5.2. La technique d'encapsulation

D'une manière générale le mécanisme d'encapsulation consiste à modifier le service initial fourni par une couche logicielle au moyen d'une couche d'encapsulation. Cette dernière utilise le service de la couche encapsulée pour offrir le service modifié noté Sm [Gien 79]. Ceci permet d'harmoniser les différents services offerts par un ensemble d'entités au moyen des entités d'encapsulation qui présentent toutes la même interface harmonisée (Fig 2.7).

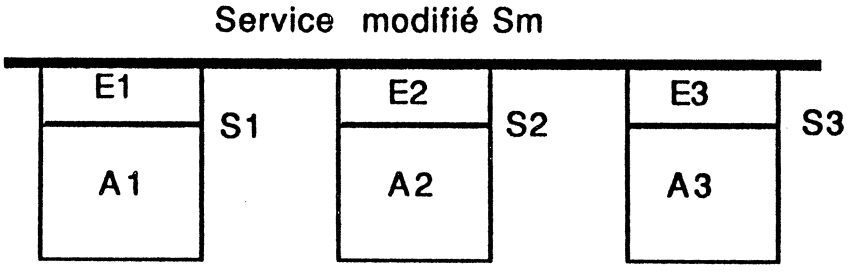


Fig 2.7 Le principe général de l'encapsulation

5.2.1. L'interconnexion par encapsulation

Examinons maintenant comment on peut mettre en œuvre cette technique d'encapsulation dans un environnement d'interconnexion. Nous considérons les deux réseaux R_a et R_b (Fig 2.8), où Ln_a est une couche de communication appartenant à l'architecture du système de communication du réseau R_a, et Ln_b est une couche de communication appartenant à l'architecture du système de communication du réseau R_b. Les deux protocoles de communication réalisés par Ln_a et Ln_b sont hétérogènes. Il n'est donc pas possible de

communiquer directement à ce niveau. L'approche d'encapsulation permet d'interconnecter les deux réseaux précédents. Il faut pour cela :

- implémenter une couche d'encapsulation En au-dessus des couches Ln_a et Ln_b sur l'ensemble des stations hôtes des deux réseaux. Cette couche offre un service de communication au moyen du protocole d'encapsulation PEn qui utilise les services de la couche Ln_a sur le réseau R_a, et les services de la couche Ln_b sur le réseau R_b. Ainsi les services et le protocole de communication au niveau En sont identiques sur l'ensemble des deux réseaux.

- introduire une passerelle entre les deux réseaux à interconnecter. Les deux modules communicateurs de la passerelle du côté du réseau R_a et du côté du réseau R_b contiennent respectivement les couches [1...Ln_a] et [1...Ln_b]. Le module adaptateur est constitué de la couche d'encapsulation En, qui interagit à la fois avec la couche Ln_a et avec la couche Ln_b.

Le schéma de la figure 2.8 montre que la couche En présente une interface de communication commune sur l'ensemble des deux réseaux et offre donc la visibilité d'un réseau global. La communication inter-réseaux au niveau En se fait de proche en proche. En revanche la communication au niveau supérieur (si des protocoles identiques sont utilisés à ce niveau) est de bout en bout. Les SDU de l'utilisateur (ou d'une entité de la couche supérieure à En) qui sont émis à partir d'une station hôte sur le réseau R_a sont encapsulés par les PDU de la couche En. Ces PDU traversent le réseau R_a au moyen de protocoles de communication de ce réseau [1...Ln_a] pour arriver vers la couche En dans la passerelle. Cette couche les route en les réémettant au moyen des protocoles du réseau R_b [1...Ln_b] vers la couche En sur la station hôte destinatrice qui les décapsule et délivre les SDU à l'utilisateur destinataire.

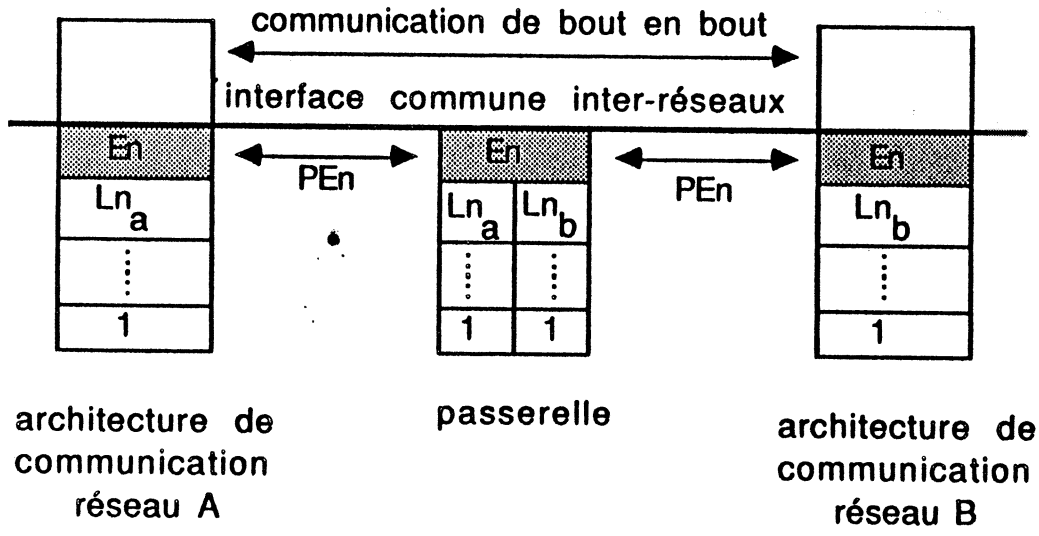


Fig 2.8 Le principe d'interconnexion par encapsulation

Dans un schéma d'interconnexion général, la couche encapsulation concatène dans les passerelles une suite de communications intra-réseau pour former une communication inter-réseaux. Elle doit pour cela réaliser les fonctions suivantes :

- fonction de segmentation et de réassemblage pour adapter la taille des données, qui sont échangées à travers le réseau global, à la taille des données qui peuvent être véhiculées par les sous-réseaux traversés. La politique de segmentation adaptée à l'approche d'encapsulation est celle de la segmentation inter-réseaux. La couche encapsulation étant implémentée dans les stations hôtes, elle peut donc intégrer une fonction de segmentation où les données sont segmentées dans la station source pour être réassemblées dans la station destinatrice.

- la couche En réalise une fonction de routage qui peut être plus ou moins sophistiquée selon la topologie de l'interconnexion. Cette fonction de routage est une partie intégrante de la couche En ; elle opère sur des informations (généralement l'adresse destinataire finale) contenues dans l'en-tête des unités de protocole En.

5.2.2. L'application de la technique d'encapsulation

Nous examinons dans ce paragraphe la possibilité d'appliquer la technique d'encapsulation dans les deux cas d'interconnexion définis précédemment.

1) cas des réseaux partiellement hétérogènes :

Nous avons vu que l'hétérogénéité est surtout présente dans ce cas, au niveau des couches basses (jusqu'à la couche réseau), afin de prendre en compte les différentes technologies réseau. Les couches supérieures, à partir du transport, sont donc identiques. La technique d'encapsulation est parfaitement adaptée à ce cas d'interconnexion. Il suffit d'introduire la couche d'encapsulation au dessous du niveau à partir duquel les protocoles deviennent homogènes (le transport). Les différences qui existent au niveau des couches inférieures seront cachées par la couche d'encapsulation qui présentera une interface commune d'accès au réseau global. Pour l'utilisateur du niveau transport, la couche d'encapsulation est invisible. Le protocole de transport sera de bout en bout par rapport à la communication inter-réseaux, faisant abstraction des différents réseaux et de la passerelle qui les interconnecte. Ce principe d'encapsulation (souvent appelé *internet model*) correspond à une interconnexion au niveau réseau. L'architecture DARPA intègre ce modèle d'interconnexion pour des réseaux supportant la famille de protocoles DARPA.

2) cas des réseaux totalement hétérogènes :

L'application du principe d'encapsulation, dans le cas où l'hétérogénéité apparaît à tous les niveaux, semble ne pas avoir beaucoup de sens ou en tous cas être artificielle. En effet, introduire la couche encapsulation à un niveau quelconque permet de cacher les différences des niveaux inférieurs ; mais aucune communication ne peut avoir lieu aux niveaux supérieurs car il sont eux-mêmes hétérogènes. La seule communication possible est lorsque l'utilisateur accède directement à la couche d'encapsulation. Cette façon d'utiliser le principe d'encapsulation nous semble peu intéressante pour les raisons suivantes :

- une nouvelle interface et un nouveau service (ceux de la couche d'encapsulation) sont imposés à l'utilisateur, pour lequel la couche

d'encapsulation est visible.

- il faudrait introduire une couche d'encapsulation pour chaque couche accédée par l'utilisateur ce qui augmenterait considérablement le nombre de couches.

- l'hétérogénéité étant existante à tous les niveaux, le besoin d'interconnexion implique une communication inter-réseaux au moins au niveau application. L'utilisation de la technique d'encapsulation dans ce cas consiste à introduire une couche qui fait converger les services offerts au niveau application de chaque réseau vers un service identique. En plus de l'inconvénient dû à la modification de service au niveau application, la communication à ce niveau perd son caractère de bout en bout.

La solution possible dans ce cas (en dehors de l'utilisation de la technique de conversion au niveau application) consiste à remplacer les protocoles d'application hétérogènes par des protocoles identiques sur les deux réseaux à interconnecter. Une couche d'encapsulation peut être ainsi introduite au niveau inférieur à celui de l'application sur les stations hôtes pour cacher l'hétérogénéité. Ceci permet de réaliser au moyen d'une passerelle appropriée une interconnexion au niveau application.

5.2.3. Le protocole d'encapsulation

Nous avons vu que l'utilisation typique de la technique d'encapsulation est celle qui permet d'offrir, à une couche transport commune, une interface d'accès commune à des réseaux hétérogènes. Un des points à déterminer est le type de service offert par la couche d'encapsulation : *mode connexion ou mode sans connexion*. Le choix entre les deux modes de communication en général est un problème difficile qui nécessite un large développement. Il s'agit ici d'un cas particulier qui consiste à déterminer le mode de communication adapté pour le protocole d'encapsulation, c'est-à-dire dans un environnement d'interconnexion.

Un protocole de communication en mode connexion offre généralement des avantages certains : la possibilité de réaliser un contrôle de flux, de préallouer les ressources nécessaires, de maintenir le séquencement des PDU et d'autres

avantages encore. En résumé, il offre une communication fiable. Cela ne signifie pas qu'il garantit la livraison des données sans erreur mais il assure au moins que les erreurs de communication seront signalées à l'utilisateur. Les inconvénients d'un tel protocole sont, d'une part, le coût de développement (en termes de complexité) et le coût de fonctionnement (en termes de ressources consommées) et, d'autre part, la performance de la communication qui présente une augmentation du délai d'acheminement des données. L'avantage d'un protocole en mode sans connexion réside dans sa simplicité et ses performances, en revanche sa fiabilité est moins grande. Le choix d'un protocole d'encapsulation pour une interconnexion au niveau réseau est guidé par les deux remarques suivantes :

- toutes les communications inter-réseaux traversent par la passerelle, ce qui constitue un goulot d'étranglement si les ressources nécessaires à la gestion de ces communications sont plus importantes que les ressources disponibles.

- la couche d'encapsulation est implémentée sur les machines hôtes, mais elle intervient aussi dans les communications intra-réseau.

Ces deux points montrent l'influence du protocole d'encapsulation sur les performances de la communication inter-réseaux mais aussi sur celles de la communication intra-réseau et sur la quantité de ressources nécessaires dans les passerelles. Afin d'améliorer les performances et de minimiser les ressources utilisées, un protocole en mode sans connexion est toujours retenu pour une interconnexion par encapsulation au niveau réseau. La fiabilité de la communication est alors assurée par le niveau supérieur, constitué généralement par une couche transport.

5.2.4. Evaluation de la technique d'encapsulation

Les principales caractéristiques de la technique d'encapsulation peuvent être déterminées selon les critères définis précédemment :

a) La mise en œuvre de la technique d'encapsulation nécessite d'implémenter les fonctions nécessaires à l'interconnexion (la couche d'encapsulation), à la fois dans la passerelle et dans toutes les stations hôtes. Cela influence les performances de la communication inter-réseaux, mais introduit aussi une

certaine *pollution* de la communication intra-réseau. En effet, la couche encapsulation intervient dans la communication intra-réseau alors qu'elle est "inutile" par rapport à cette communication.

Afin d'éviter la pollution, une variante de l'approche d'encapsulation peut être envisagée. Il s'agit d'utiliser la première couche commune à l'ensemble des réseaux interconnectés comme couche d'encapsulation. Le module adaptateur est alors constitué de la couche en question, à condition d'y implémenter la fonction de routage nécessaire à l'interconnexion. Cela présente l'avantage d'éviter de rajouter une nouvelle couche dans les stations hôtes. Cependant, puisque la communication au niveau de la couche d'encapsulation ne peut être que de proche en proche, cette solution devient peu satisfaisante quand les services de la couche qui joue le rôle d'encapsulateur sont normalement de bout en bout. C'est le cas lorsqu'on réalise une encapsulation avec la couche transport par exemple.

Un autre cas d'interconnexion par encapsulation, sans introduire de nouvelle couche, consiste à interconnecter deux réseaux locaux utilisant des protocoles MAC hétérogènes et une couche LLC identique (Fig 2.9).

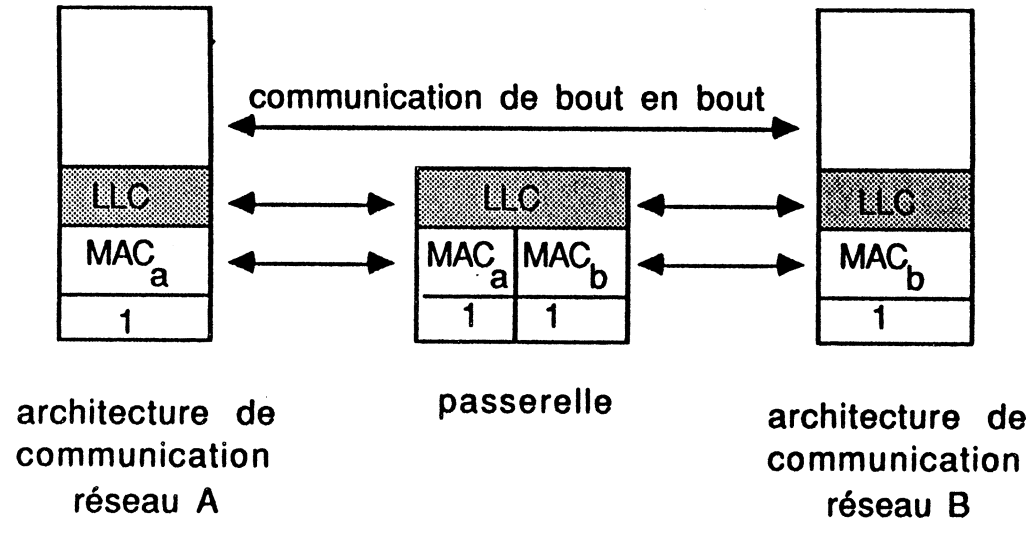


Fig 2.9 Interconnexion des réseaux locaux par encapsulation au niveau LLC

Dans ce cas, la couche LLC représente la couche d'encapsulation. Il est à noter que dans un tel cas, la solution de *pont* présente l'intérêt de minimiser le nombre de couches dans la passerelle et de réaliser une communication de bout en bout au niveau supérieur à celui de LLC.

b) Nous avons vu que l'application la plus appropriée (la plus courante aussi) de cette technique consiste à interconnecter des réseaux hétérogènes jusqu'au niveau réseau et qui supportent une couche transport identique (des réseaux partiellement hétérogènes supportant la même famille de protocoles). La couche d'encapsulation offre alors à la couche transport une seule interface d'accès au réseau global. Dans ce cas, la couche d'encapsulation ainsi que la passerelle sont complètement transparentes à tout utilisateur accédant au service transport ou aux services des couches supérieures. La visibilité d'un tel utilisateur est celle d'un unique réseau global. La structure d'adressage est uniforme au niveau de la couche d'encapsulation.

c) La définition du protocole d'une couche d'encapsulation "internet" dans une architecture de communication se fait de manière à ce qu'il soit réalisable par une grande variété de services de communication et d'accès réseau. L'interconnexion d'un nouveau réseau, qui supporte la même famille de protocoles, est relativement simple. Pour cela il faut implémenter dans la passerelle le module communicateur correspondant au réseau en question, et interfacer la couche encapsulation avec ce module. En d'autres termes, seule la partie interface inférieure de la couche encapsulation est dépendante dans sa réalisation des réseaux interconnectés.

L'intérêt de l'approche d'encapsulation réside dans le fait qu'elle résoud globalement les problèmes d'hétérogénéité entre les services de communication des différents sous-réseaux (hétérogénéité sémantique). Ceci est réalisé grâce à l'introduction d'une couche identique qui offre des services communs. Ainsi les services, qui n'étaient pas compatibles au niveau du protocole d'accès à chaque sous-réseau, ne sont plus accessibles par l'utilisateur. Mais cela suppose, par contre, que le service offert par la couche d'encapsulation soit réalisable par les couches de communication de chaque sous-réseau. Cela est vérifié généralement par le choix d'un protocole d'encapsulation simple laissant le problème de la fiabilité à la charge du niveau supérieur (le transport généralement).

La simplicité de la mise en œuvre de l'approche d'encapsulation, et surtout sa possibilité d'intégrer de nouveaux réseaux, expliquent l'intérêt de cette approche. Il reste que son domaine d'application est limité au cas des réseaux partiellement hétérogènes qui implémentent des protocoles appartenant à une même famille dont l'architecture de communication prévoit, a priori, une couche d'encapsulation pour réaliser l'interconnexion.

Cette limitation est importante car elle signifie que des architectures de communication totalement hétérogènes ne peuvent pas s'interconnecter à l'aide de cette approche. Nous allons voir comment l'approche conversion permet de répondre, sous certaines conditions, à ce type de problème d'interconnexion.

5.3. La technique de conversion

Nous considérons deux réseaux R_a et R_b, dans lesquels n_a est une couche de communication appartenant à l'architecture du système de communication du réseau R_a, et n_b est une couche de communication appartenant à l'architecture du système de communication du réseau R_b. Les deux protocoles de communication réalisés par n_a et n_b sont a priori hétérogènes. Aucune hypothèse n'est faite en ce qui concerne les couches inférieures qui peuvent être soit hétérogènes, soit identiques. On souhaite réaliser une interconnexion au niveau de n_a et n_b. Il faut, pour cela, introduire une passerelle entre les deux réseaux à interconnecter. Les deux modules communicateurs de la passerelle du côté du réseau R_a et du côté du réseau R_b contiennent respectivement les couches $[1...n_a]$ et $[1...n_b]$. Le module adaptateur réalise une fonction de *conversion* (appelée aussi *translation*) entre le niveau n_a et le niveau n_b. Une communication inter-réseaux au niveau $[n_a, n_b]$ met en jeu les protocoles correspondants à ces deux couches et la fonction de conversion entre eux. Elle est donc décomposée en deux sous-communications. Chacune fournit les services réalisées par le protocole correspondant. Cependant, l'utilisateur du service de la couche n_a sur le réseau R_a (respectivement n_b sur le réseau R_b) a l'impression d'effectuer une communication inter-réseaux au moyen des services et du protocole de la couche n_a (respectivement n_b) avec l'utilisateur de la couche n_b (respectivement n_a).

La fonction de conversion entre deux protocoles consiste à exprimer la sémantique des services offerts par l'un au moyen des services offerts par l'autre. Il s'agit donc de cacher l'hétérogénéité entre les deux protocoles en réalisant une conversion sémantique et syntaxique. Ce principe peut être mis en œuvre suivant deux méthodes, chacune correspond à un niveau d'intéraction différent entre le module de conversion et les couches à convertir. Les deux méthodes sont les suivantes :

1) Conversion au niveau interface

Le module de conversion se comporte comme un utilisateur des services des deux couches n_a et n_b (Fig 2.10). Il intervient alors au niveau de l'interface de chacune de ces couches. Sa fonction est d'utiliser les services fournis par l'un des deux protocoles à travers son interface pour assurer la continuité des services fournis par l'autre protocole. Ceci est réalisé par une correspondace entre les primitives des deux protocoles :

(n_a)-Primitive <=> (n_b)-Primitive

2) Conversion au niveau protocole

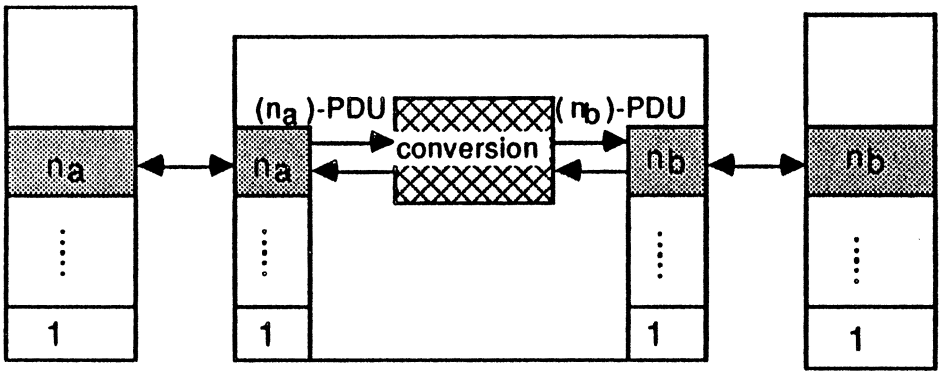
Dans ce cas, le module de conversion intervient au niveau protocole (Fig 2.10) en réalisant une correspondance entre les unités de protocole des deux couches :

(n_a)-PDU <=> (n_b)-PDU

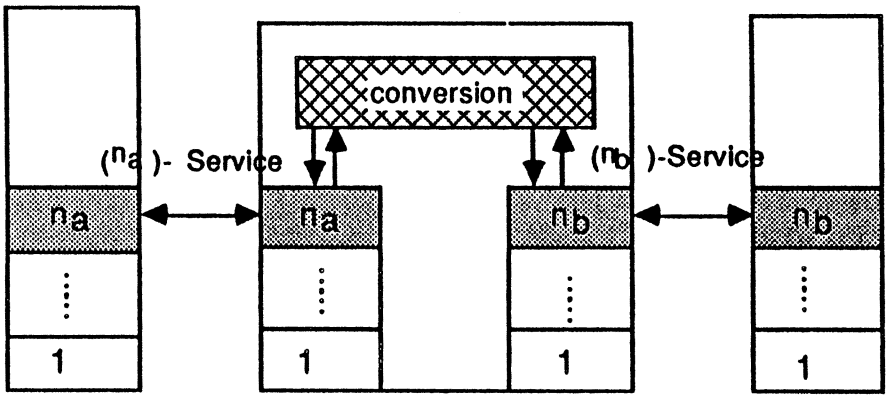
L'effort dans la conversion sémantique des services offerts par les couches à interconnecter est le même dans les deux méthodes. La différence se trouve au niveau de la conversion syntaxique qui concerne les primitives de l'interface, dans la première méthode, et les unités de protocole, dans la deuxième. Le choix entre les deux méthodes est essentiellement lié à des critères de facilité d'implémentation.

L'avantage de la conversion au niveau interface est le fait que le programme de conversion joue le rôle d'un programme utilisateur par rapport aux couches à convertir, aucune modification n'est donc nécessaire dans le code des protocoles. Alors que pour la conversion au niveau protocole, le code du

programme de conversion doit être intégré dans le code des protocoles à convertir. Cependant il existe une propriété intéressante de la conversion au niveau protocole : la possibilité de réaliser certaines fonctions de communication de bout en bout, ce qui n'est pas toujours possible dans le cas de la conversion au niveau interface. A titre d'exemple, prenons la fonction de contrôle d'erreurs par acquittement : l'envoi des acquittements est une fonction au niveau protocole, invisible donc qui ne peut pas être gérée au niveau interface.



Passerelle
de conversion au niveau protocole



Passerelle
de conversion au niveau interface

Fig 2.10 Conversion au niveau interface et conversion au niveau protocole

5.3.1. Le principe de la réalisation de conversion

La conversion entre deux protocoles P_a et P_b est basée sur la possibilité d'exprimer la sémantique véhiculée par les PDU de l'un au moyen du format et de la syntaxe des PDU de l'autre [François 83]. La mise en œuvre de ce principe consiste à décrire le comportement du module de conversion sous forme d'une machine d'états finis. Les événements entrants du côté du protocole P_a provoquent, en fonction de l'état interne du module de conversion, l'exécution d'actions. Ces actions consistent à générer du coté du protocole P_b des événements sortants équivalents aux événements entrants (Fig 2.11).

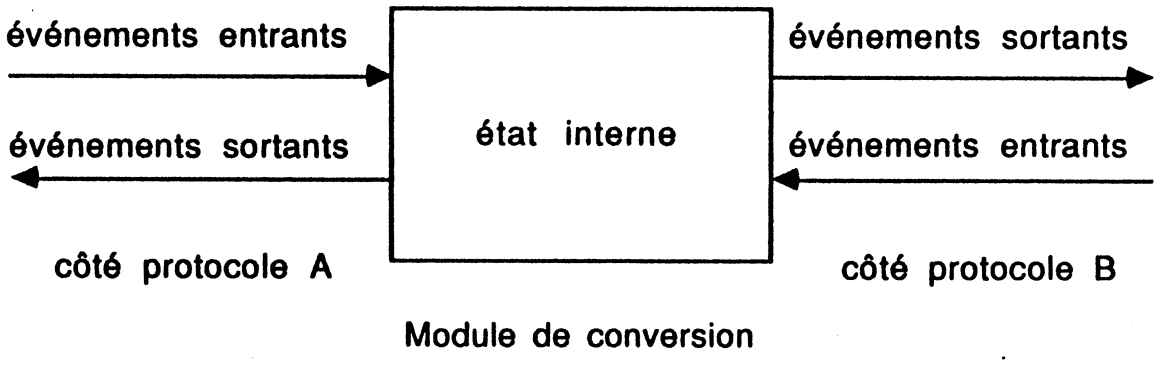


Fig 2.11 Le principe de modélisation de la fonction de conversion à l'aide d'une machine d'états

Lorsqu'il existe une correspondance directe entre les services offerts par chaque protocole, le module de conversion est appelé "relais". Son fonctionnement peut être schématisé de la manière suivante (Fig 2.12) : l'arrivée d'une indication de service du coté du protocole du réseau A *S(A)-Indication* génère la demande de service équivalente du côté du réseau B *S(B)-Request*. L'arrivée d'une confirmation de service du côté du réseau B *S(B)-Confirmation* génère la réponse de service équivalente du côté du protocole du réseau A *S(A)-Response*.

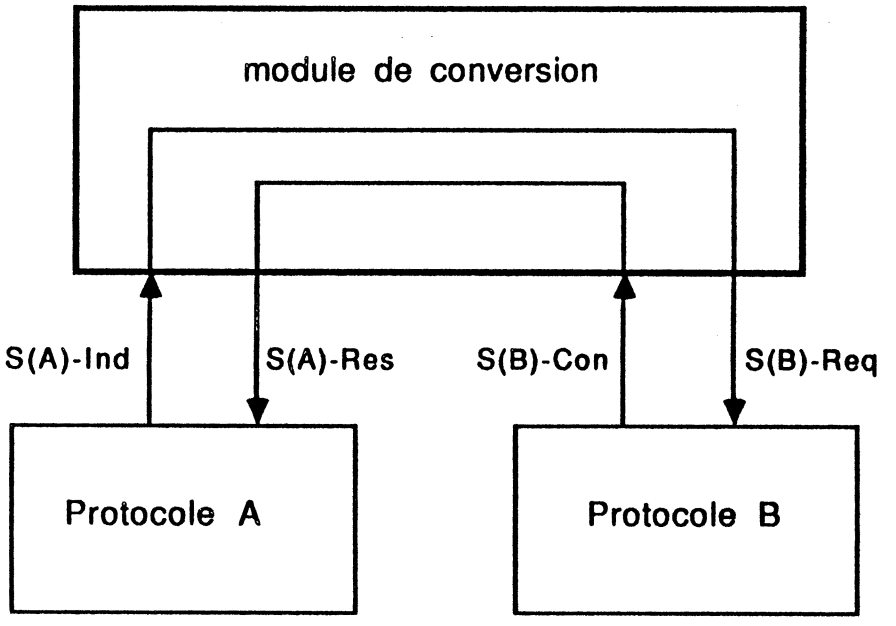


Fig 2.12 Schéma simplifié de la conversion

En réalité, le processus de conversion est plus complexe, car il n'existe pas une correspondance un-pour-un entre les services des protocoles à convertir. L'arrivée d'un événement d'un côté de la passerelle peut génerer aucun ou plusieurs événements de l'autre coté. Prenons, à titre d'exemple, le cas d'une conversion entre un protocole en mode connexion et un protocole en mode sans connexion :

- l'arrivée d'une indication d'ouverture de connexion du coté du protocole en mode connexion ne génère aucun événement du coté du protocole sans connexion. En revanche une réponse positive d'ouverture de connexion doit être transmise par le module de conversion du côté du protocole en mode connexion.

- l'arrivée d'une indication de données, du côté du protocole en mode sans connexion provoque la génération, par le module de conversion, de plusieurs actions du côté du protocole en mode connexion. D'abord, il faut établir une connexion avec l'entité communicante destinataire (si une telle connexion n'existe pas encore), ensuite il faut générer l'événement requête de données sur cette connexion.

5.3.2. Le processus de conversion

La mise en œuvre du principe de conversion dépend dans sa complexité de l'hétérogénéité qui existe entre les protocoles à convertir. L'opération de conversion n'est pas toujours possible. Elle doit obéir au conditions suivantes [François 83][Zoline 85] :

1) la fonction de conversion entre deux protocoles hétérogènes est réalisable seulement si ces protocoles offrent des services dont les sémantiques sont comparables. Les protocoles sont alors *convertibles*.

2) des protocoles hétérogènes offrent des services comparables, s'il existe un sous-ensemble de services communs offerts par les deux protocoles. La fonction de conversion s'applique uniquement sur ce sous-ensemble de fonctionnalités communes. Les services qui ne sont pas convertibles ne sont pas utilisables pour la communication inter-réseaux.

3) afin d'avoir un sous-ensemble de services communs suffisament large, il est nécessaire dans certains cas d'enrichir l'un des deux protocoles par certaines fonctionnalités offertes par l'autre protocole.

Les conditions citées ci-dessus montrent que le principe de la conversion n'est pas toujours réalisable. L'hétérogénéité sémantique entre les protocoles à convertir détermine la compléxité de la réalison de la conversion. Le choix des niveaux de protocoles à convertir pour l'interconnexion de deux réseaux est un choix important. Il doit viser l'obtention d'un sous-ensemble commun de services le plus large possible. Cependant ce choix n'est pas libre, car il est dicté essentiellement par le niveau de communication inter-réseaux que doit assurer l'interconnexion. On peut s'attendre généralement à ce que la complexité de l'interconnexion augmente avec l'accroissement du niveau des protocoles à convertir [Green 86]. Les fonctions réalisées par les niveaux supérieurs de protocole effectuent un traitement sur les données, elles sont donc de niveau sémantique plus élevé que les protocoles de bas niveau.

La démarche qui aboutit au développement de la solution de conversion entre deux protocoles hétérogènes se décompose en trois étapes :

1) comparaison entre les services de chaque protocole à travers les primitives et les paramètres de l'interface de chacun des protocoles. Cette comparaison doit permettre la définition d'un sous-ensemble commun de services.

2) pour le sous-ensemble des services communs, il faut définir la correspondance entre les services offerts par chaque protocole. Il faut également déterminer les restrictions à appliquer sur chaque protocole en fonction de l'ensemble de services qui ne sont pas convertibles.

3) la troisième étape consiste à décrire la machine d'états finis qui réalise la conversion.

Nous examinerons plus loin l'application de cette démarche à la conversion entre le protocole de transport ISO et le protocole TCP de DARPA.

5.3.3. Evaluation de la technique de conversion

Les caractéristiques essentielles de la technique de conversion sont les suivantes :

1) les fonctions qui réalisent l'interconnexion sont regroupées dans la passerelle. Aucune modification du système de communication n'est introduite dans les stations hôtes. Il résulte de ceci deux avantages importants de la technique de conversion :

   - cette technique n'introduit aucune modification dans les performances de la communication intra-réseau (ce qui n'est pas le cas de la technique d'encapsulation).

   - l'interface de la communication au niveau de l'interconnexion n'est pas modifiée. Il y a donc une homogénéité entre le service de communication interne et le service de communication externe à ce niveau.

2) La réalisation de la conversion peut s'avérer complexe si les protocoles à convertir ne présentent pas de services suffisamment comparables. La solution de conversion dépend beaucoup dans sa réalisation des

protocoles convertis. Par conséquent, elle n'est pas extensible (à l'opposé de l'encapsulation). Tout le processus de conversion doit être répété à chaque interconnexion de nouveau réseau et à chaque niveau d'interconnexion. On peut remédier aux problèmes de la complexité et l'extensibilité par l'utilisation de la notion de *méta-protocole*. Ceci consiste à convertir les protocoles à interconnecter vers un protocole commun appelé "meta protocol" ou "super protocol" [Green 86].

3) La conversion est applicable aussi bien dans le cas d'interconnexion de réseaux partiellement hétérogènes que dans le cas d'interconnexion des réseaux totalement hétérogènes. Dans le premier cas, le niveau d'interconnexion doit être choisi au dernier niveau où l'hétérogénéité apparaît. Dans le cas d'interconnexion entre des systèmes de communication hétérogènes à tous les niveaux de protocoles, la solution basée sur la conversion constitue la seule possibilité pour réaliser l'interconnexion. La conversion doit être réalisée à chaque niveau de communication accessible par les utilisateurs et en particulier au niveau application.

6. Exemples de passerelles

Cette section est consacrée à la présentation d'exemples concrets de passerelles d'interconnexion. Nous avons choisi de décrire deux passerelles qui ont été réalisées dans des environnements d'interconnexion et selon des approches différentes :

- la passerelle ROSE
- la passerelle TCP/IP -Transport OSI

Nous avons choisi la passerelle ROSE comme exemple d'interconnexion de réseaux locaux basés sur l'architecture OSI à travers des réseaux publics X25. Quand à la passerelle TCP/IP-Transport OSI, elle représente un cas typique d'application de la technique de conversion entre deux protocoles de transport largement diffusés.

6.1. La passerelle ROSE

Cette passerelle permet l'interconnexion de réseaux locaux à travers des réseaux X25. L'architecture de communication est basée sur le modèle et les protocoles OSI. Deux approches d'interconnexion ont été mises en œuvre : l'une utilise le principe d'interconnexion "internet" tel qu'il est proposé dans le modèle OSI, l'autre est basée sur le principe de relais au niveau transport tel qu'il est proposé par l'ECMA. C'est cette dernière solution qui est présentée ici.

6.1.1. Le projet ROSE

Cette passerelle a été réalisée dans le cadre du projet ESPRIT ROSE (Research Open Systems for Europe). Le but de ce projet [ROSE 85] qui a démarré en 1984 est de construire une infra-structure de communication pour l'ensemble des projets ESPRIT. Cette infrastructure est constituée de plusieurs réseaux locaux interconnectés à travers des réseaux publics X25. L'architecture de communication utilisée dans ce projet est basée sur le modèle OSI et les recommandations de l'ECMA. Cette architecture est implémentée sur des machines supportant le système UNIX. En plus des applications OSI comme le transfert de fichier (FTAM) et la messagerie (MHS), le projet a intégré également dans l'architecture de communication développée les applications de

communication fournies par UNIX telles que UUCP, CU, Mail et News.

Deux environnements de communication sont donc considérés dans le projet ROSE :

1) L'environnement de réseau local dans lequel des machines communiquent à travers un réseau local de type CSMA/CD. En ce qui concerne les protocoles de niveau supérieur, la première approche retenue était basée sur les recommandations de l'ECMA [ECMA/14 82] (c'est cette approche que nous considérons dans la suite de cette présentation). Cette approche consiste à supporter un protocole transport classe 4 directement au-dessus du protocole Ethernet de niveau trame, la couche réseau étant vide. Le protocole de transport classe 4 assure ainsi le contrôle d'erreurs et les reprises. Au niveau supérieur, on trouve le protocole de session ou des applications qui utilisent directement le protocole de transport telles que UUCP ou le PAD. La session supporte à son tour les applications OSI telles que MHS et FTAM.

2) Le deuxième environnement considéré concerne la communication à travers un réseau public X25 (ou un ensemble de réseaux X25 interconnectés). Cette communication est réalisée au moyen des protocoles X25 de niveau physique, ligne et paquet et un protocole de transport classe 3. Le but de la communication à travers les réseaux X25 est essentiellement d'assurer l'interconnexion de réseaux locaux.

6.1.2. Principe d'interconnexion

Chaque réseau local possède une passerelle connectée au réseau X25 qui lui permet de communiquer avec d'autres réseaux locaux. Cette passerelle n'est qu'une station hôte particulière. Le principe d'interconnexion est basé sur l'utilisation d'un relais au niveau transport. La fonction du relais est réalisée entre le protocole de transport classe 4 (coté réseau Ethernet) et le protocole de transport classe 3 (coté réseau X25). Le réseau Ethernet est considéré comme étant de type C et le réseau X25 comme étant de type B au sens ISO. Ainsi une communication au niveau transport entre deux stations attachées à deux réseaux locaux différents est composée de la concaténation de trois connexions transport et elle met en jeu deux passerelles (Fig 2.13) :

- une connexion transport classe 4 entre la station hôte du réseau local A et la passerelle de ce réseau.

- une connexion transport classe 3 entre la passerelle du réseau local A et la passerelle du réseau local B à travers le réseau X25

- une connexion transport classe 4 entre la passerelle du réseau local B et la station hôte sur ce réseau.

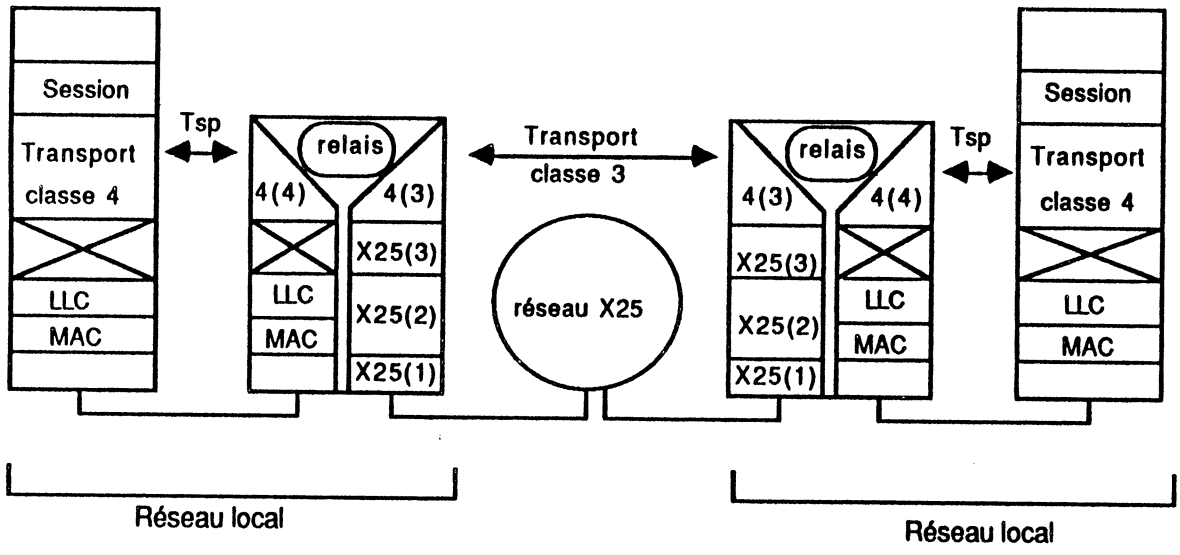


Fig 2.13 L'architecture de communication et le principe d'interconnexion dans ROSE

La communication globale au niveau transport n'a pas les caractéristiques d'une communication de bout en bout. Ceci, en plus de l'impossibilité de communiquer entre une station hôte ROSE et une station X25, constitue les deux inconvénients majeurs de cette stratégie d'interconnexion des réseaux locaux à travers les réseaux publics. Les avantages de cette solution sont, d'une part, la bonne performance de la communication à l'intérieur du réseau local due à l'absence de couche réseau et d'autre part, la simplicité de la mise en œuvre de la fonction de relais.

Le module relais est implémenté dans la passerelle ROSE comme une entité communicante utilisatrice des services du protocole transport classe 3 et transport classe 4. Son fonctionnement se décompose en trois phases :

- la première phase consiste à se mettre en attente de demande de connexion transport classe 3 (du côté réseau X25) ou classe 4 (du côté réseau local). L'arrivée d'une demande de connexion d'un côté de la passerelle déclenche la procédure d'ouverture de connexion de l'autre côté. Une fois les connexions transport ouvertes, le module passe à la deuxième phase de fonctionement.

- la deuxième phase consiste à entrer dans un cycle de balayage des deux connexions transport classe 4 et classe 3. Toute donnée reçue sur une connexion est réémise sur l'autre connexion. Ce cycle continue jusqu'à l'arrivée d'une demande de déconnexion sur une des deux connexions.

- le module relais entre dans la phase de déconnexion à l'arrivée d'une demande de déconnexion sur l'une des deux connexions. Ceci provoque la fermeture de l'autre connexion.

L'implémentation de cette fonction de relais est simple. Les interfaces offertes par le transport classe 4 et le transport classe 3 sont identiques.

Examinons maintenant le principe d'adressage au niveau transport dans l'environnement d'interconnexion ROSE. Le réseau global ROSE est composé d'un ensemble de réseaux locaux. Chaque réseau local constitue un *site*. Une station hôte est appelée *système*. L'adressage au niveau transport utilise la notion de *TSAP (transport service access point)* définie dans l'OSI. L'espace d'adressage global est partitionné hiérarchiquement. Ainsi la structure des adresses TSAP reflète la structure d'interconnexion constituée d'un ensemble de sites où chaque site est un ensemble de systèmes. La structure des adresses TSAP est la suivante :

<adresse TSAP> = <nom de site> <nom de système> <sélecteur>

le *nom de site* désigne un site ROSE, *le nom de système* identifie une station hôté sur ce site, le *sélecteur* est un identificateur qui désigne une entité communicante utilisatrice du transport sur la station hôte.

Afin d'assurer la fonction de routage ainsi que la fonction de correspondance entre les noms logiques utilisés dans les adresses et les adresses physiques, deux types de tables de correspondance sont définis :

- *la table des sites :* les entrées de cette table sont constituées de l'ensemble des noms des sites. Chaque entrée représente une association entre le nom d'un site et l'adresse X25 de sa passerelle. Il existe une seule table des sites au niveau du réseau global ROSE. Un exemplaire de cette table est implémenté dans chaque passerelle.

- *les tables des systèmes :* il existe une table des systèmes par site, elle est implémentée dans chacun des systèmes (station hôtes) de ce site. Cette table contient l'ensemble des noms des systèmes de ce site. A chaque système est associée son adresse Ethernet correspondante. Un indicateur permet d'identifier le système qui constitue la passerelle de ce site.

La structure des adresses transport, la table des sites et les tables des systèmes permettent d'effectuer facilement la fonction routage. Celle-ci est réalisée par la couche transport. Le champ site de l'adresse destinataire est analysé dans la station source. Si le site n'est pas le site local, le transport accède à la table des systèmes pour déterminer l'adresse Ethernet de la passerelle, à laquelle le TPDU est envoyé (c'est le TPDU de demande d'ouverture de connexion). La table des sites est accédée dans la passerelle pour déterminer, toujours à partir du nom du site destinataire, l'adresse X25 de la passerelle du site destinataire. La passerelle du site destinataire accède à la table des systèmes pour déterminer l'adresse Ethernet du système destinataire.

6.2. La passerelle TCP/IP - Transport ISO

La large diffusion des protocoles DARPA et le nombre important des applications supportées par ces protocoles ont accéléré le développement de solutions pour l'interconnexion entre les protocoles DARPA et les protocoles ISO. Nous présentons dans ce paragraphe une solution proposée dans [Groenbaek 86]. Cette solution assure l'interconnexion entre un réseau basé sur les protocoles DARPA (TCP/IP et UDP/IP) et un réseau basé sur les protocoles ISO.

6.2.1. Principe d'interconnexion

Le principe d'interconnexion retenu est basé sur la conversion au niveau transport. La conversion se fait entre le protocole transport DARPA et le protocole transport classe 4 de l'ISO. Le choix de ce niveau d'interconnexion est lié au fait qu'aux niveaux supérieurs, les deux réseaux supportent des protocoles identiques. Il est difficile en effet de réaliser une conversion au niveau application entre les applications ISO et les applications DARPA. Le choix de la technique de conversion permet de faire fonctionner les applications de transport déjà existantes sans introduire de nouveaux logiciels de communication sur les systèmes hôtes. La figure 2.14 schématise le modèle d'interconnexion utilisé.

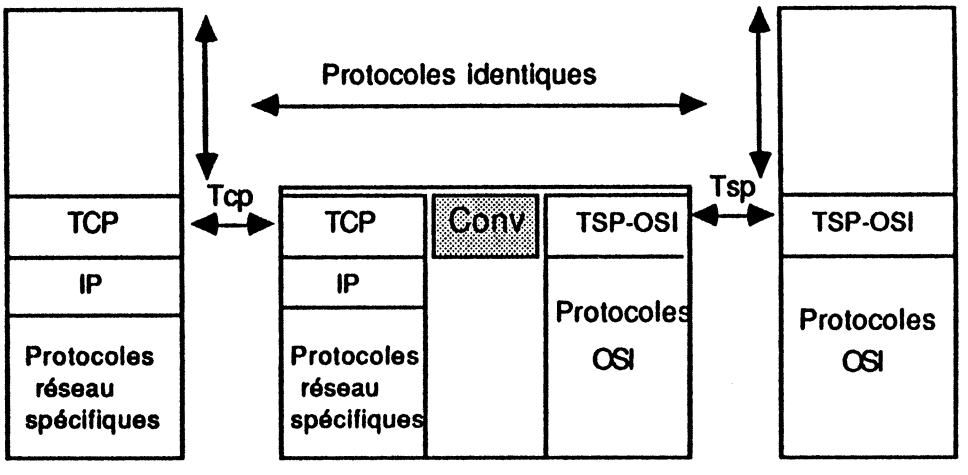


Fig 2.14 Interconnexion par conversion entre le transport OSI et le transport DARPA (TCP)

Ce modèle est basé sur la définition d'un ensemble commun de services assurés par les deux protocoles de transport et par l'introduction d'une passerelle de conversion. Il est important de noter que la conversion est effectuée au niveau protocole ce qui permet de conserver l'aspect de bout en bout des fonctions transport. Le développement de cette solution consiste d'abord à comparer les services offerts par les deux protocoles et à définir ensuite le sous-ensemble de services communs entre les deux protocoles (les services convertibles).

6.2.2. Comparaison entre les services du transport classe 4 ISO et du TCP

Nous donnons ici une brève comparaison [Gronbaek 86][ISO/DIS 8073] entre les services assurés par le transport OSI (appelé dans la suite TSP-OSI) et le transport TCP de DARPA. Le but de cette comparaison est de dégager les différences essentielles entre les deux protocoles.

1) phase d'ouverture de connexion transport

La fonction d'ouverture de connexion transport offre globalement les mêmes services dans les deux protocoles. Néanmoins certaines différences existent :

a) adressage

- TCP : l'adressage dans TCP est basé sur la notion de *socket*. Un socket est la concaténation de l'adresse internet (IP) qui identifie une station hôte dans un environnement d'interconnexion et l'adresse d'un *port* qui identifie une application utilisant TCP. Une connexion relie une paire de sockets.

- TSP-OSI : L'adressage dans TSP-ISO est basé sur la notion de Transport Service Access Point (TSAP). Une connexion relie deux TSAP. La structure et la sémantique des adresses TSAP ne sont pas définies. Une structure hiérarchique équivalente au socket peut donc être utilisée.

b) ouverture simultanée de connexion

Une demande d'ouverture simultanée sur la même paire de sockets dans le protocole TCP aboutit à l'établissement d'une seule connexion. Dans le cas de TSP-OSI, cette demande provoque l'établissement de deux connexions indépendantes liant la même paire de TSAP.

c) les données utilisateurs

Le TSP-OSI permet de supporter des données utilisateurs dans le TPDU de demande d'ouverture. Ce service n'est pas offert par TCP.

2) phase de transfert de données

a) données normales

le TSP-OSI permet de délimiter les données utilisateurs en unités appelées Transport Service Data Unit (TSDU). Cette notion n'existe pas dans TCP.

b) données express

- TSP-OSI : en plus des données normales le TSP-OSI offre le service de transfert de données express. Ces données sont l'objet d'un contrôle de flux indépendant. Elles doivent être prioritaires par rapport aux données normales. Cependant la mise en œuvre de cette priorité est spécifique à chaque implémentation.

- TCP : la notion de données express n'existe pas. Cependant l'utilisateur de TCP peut qualifier une suite de données comme étant urgente. Le protocole TCP véhicule cette information sous forme d'un indicateur positionné dans le segment qui transporte ces données. L'utilisateur qui reçoit ces données est informé de la valeur de cet indicateur. Du point de vue du protocole TCP ces données sont traitées comme les autres, elles ne bénéficient d'aucune priorité.

3) phase de fermeture de la connexion

Le service de fermeture de connexion offert par le TSP-OSI ne garantit pas que les données en attente de livraison ne soient pas perdues, en particulier quand la fermeture de la connexion est invoquée par le protocole de transport. Le transport TCP offre deux services de déconnexion, un service de déconnexion "propre" sans perte d'information et un service de déconnexion "brutal" équivalent à celui du TSP-OSI. Le service de déconnexion TCP ne permet pas de transporter des données utilisateurs comme c'est le cas pour le TSP-OSI.

6.2.3. Définition d'un sous-ensemble commun de services

La comparaison entre les deux protocoles montre que la majeure partie des services offerts par les deux protocoles sont communs, ils sont par conséquent convertibles directement. Cependant des restrictions doivent être imposées à l'utilisation de certains services :

- l'envoi des données utilisateurs au moment de l'ouverture ou de la fermeture des connexions transport ne doit pas être utilisé dans TSP-OSI. Ces données doivent être envoyées au début de la phase de transfert de données. Ceci peut être pénalisant, en particulier, quand cette possibilité est utilisée pour réaliser une extension d'adresse.

- le service des données express dans TSP-OSI doit être supprimé, ainsi que l'utilisation du flag "urgent" dans le protocole TCP. La conversion entre ces deux services n'est pas possible parce qu'ils ont des sémantiques différentes. Dans le cas de TSP-OSI, les données express peuvent être alors envoyées comme données normales car, de toutes façons la notion de priorité associée à ces données dépend de l'implémentation, elle ne fait pas partie de la spécification du protocole. En revanche le flag "urgent" associé à des données TCP est visible au niveau interface. Il est utilisé dans certaines applications (comme Telnet) pour véhiculer des conditions d'exception. Son interdiction affecte le fonctionnement de ces applications.

L'exemple de conversion présenté ici montre qu'en dépit de la similitude des services offerts par les protocoles TCP-OSI et TCP, des restrictions sur les services initiaux sont imposées. Un des intérêts de la technique de conversion qui consiste à conserver les interfaces initiales de chaque protocole se trouve ainsi compromis.

7. Conclusion

Nous avons présenté dans ce chapitre l'ensemble des concepts et des problèmes liés à l'interconnexion des réseaux. Nous avons dégagé les principales approches utilisées pour réaliser l'interconnexion. Les caractéristiques de chaque approche ainsi que les contraintes liées à son application dans les différents contextes d'interconnexion ont été discutés. Notre objectif était d'avoir la vue la plus globale et la plus synthétique. Ceci afin de déterminer si les techniques d'interconnexion peuvent servir directement comme base pour assurer la communication entre les SD-RL.

Avant de donner une première réponse à cette interrogation, il convient de récapituler les trois solutions générales permettant d'assurer l'interconnexion des réseaux :

1) Ponts et répéteurs

C'est la solution la plus efficace quand il s'agit d'interconnecter des réseaux locaux homogènes, ou hétérogènes uniquement au niveau de leurs protocoles d'accès au support.

2) La couche internet

C'est la solution la plus générale quand l'hétérogénéité ne dépasse pas le niveau réseau. Chaque réseau supporte son protocole d'accès réseau spécifique. La couche internet supporte un protocole en mode sans connexion (datagramme) et présente une interface homogène au protocole transport qui doit être identique sur l'ensemble des réseaux interconnectés.

3) La conversion

Quand l'hétérogénéité affecte des niveaux supérieurs au niveau réseau, la conversion est alors la solution envisagée. Si les architectures de communication des réseaux à interconnecter sont complètement hétérogènes une conversion au niveau application sera nécessaire. Comme nous l'avons vu, la mise en œuvre de la conversion peut être complexe. Elle peut même devenir impossible à réaliser si les protocoles à convertir n'offrent pas un sous-ensemble commun de services. Cela est particulièrement vrai pour les

protocoles du niveau application où la sémantique des services peut être fortement spécifique à chaque architecture de communication.

Il est donc clair qu'à partir du moment où nous faisons l'hypothèse de l'hétérogénéité totale des architectures de communication des SD-RL, la seule technique d'interconnexion envisageable pour assurer leur ouverture serait la conversion. Cette conversion doit être réalisée au niveau de la communication inter-réseaux souhaité par l'utilisateur (c'est généralement le niveau application). La complexité et l'impossibilité éventuelle de la mise en œuvre de cette solution nous incite à chercher une autre approche pour rendre les SD-RL ouverts sur le monde extérieur. Cette approche doit être caractérisée par son applicabilité systématique. Cela signifie que sa propriété essentielle devra être son indépendance (autant que possible) en terme d'applicabilité vis-à-vis :

- du niveau (au sens hiérarchique) du service de communication externe qu'elle doit offrir ;

- des protocoles de communication supportés dans chaque SD-RL.

C'est dans cette direction que s'oriente notre recherche pour une nouvelle approche permettant l'ouverture des SD-RL au monde extérieur. Ceci est l'objet du prochain chapitre.

Chapitre 3

DOSIS : un serveur OSI distribué pour la communication avec le monde OSI

1. Introduction

Nous avons identifié dans le premier chapitre la nécessité pour un système distribué d'avoir une ouverture vers le monde extérieur. Cette ouverture est assurée au moyen d'un service de communication externe. Nous avons vu également que l'architecture et les protocoles OSI sont adaptés à cette communication externe. Ils peuvent donc être utilisés comme un moyen pour la communication entre les systèmes distribués. Ceci doit se faire sans imposer leur utilisation à l'intérieur de chacun de ces systèmes. Nous faisons donc l'hypothèse de l'hétérogénéité totale des systèmes de communication des différents SD-RL.

L'objectif de ce chapitre est de rechercher une solution permettant d'offrir aux applications qui se déroulent sur un SD-RL les moyens d'accéder à un ensemble de services OSI. Ces services parmettent aux applications de communiquer avec d'autres applications sur d'autres SD-RL ou systèmes centralisés. Nous discutons d'abord les raisons pour lesquelles les solutions basées sur les techniques d'interconnexion étudiées dans le deuxième chapitre ne sont pas satisfaisantes. Nous présentons ensuite les éléments de base d'une nouvelle approche que nous proposons pour ce problème. Cette approche est appelée DOSIS (Distributed OSI Server) ; l'architecture et les fonctionnalités de ses différents composants sont décrits. L'intérêt de cette approche ainsi que ses avantages par rapport aux solutions d'interconnexion étudiées sont discutées. Nous terminons ce chapitre par la description des implémentations réalisées de DOSIS.

2. Quelles solutions pour fournir les services OSI dans un SD-RL ?

Rappelons donc qu'il s'agit d'un environnement de réseau local reliant un ensemble de systèmes au moyen d'un service de communication interne réalisé par des protocoles qui ne correspondent pas à ceux de l'OSI. Nous cherchons une solution qui offre aux applications qui s'exécutent dans un tel environnement l'accès à des services OSI. Ces applications peuvent être centralisées (s'exécutant sur une machine du SD-RL) ou distribuées (elles mettent en jeu un ensemble de processus s'exécutant sur plusieurs machines et coopérant au moyen des services de communication interne du SD-RL). Deux approches peuvent être envisagées pour assurer l'accès aux services OSI :

1) L'approche "individualiste"

Il s'agit simplement d'implémenter les protocoles OSI dans chacune des stations du SD-RL. Chaque station est alors considérée comme étant un système OSI *ouvert*. Ce système est capable de communiquer avec tout système OSI à l'intérieur ou à l'extérieur du SD-RL. Chaque station doit supporter à la fois l'architecture de communication spécifique au SD-RL pour la communication interne et l'architecture OSI pour la communication externe ce qui constitue un inconvénient majeur. Du point de vue de la communication OSI, le réseau apparaît comme un réseau local supportant l'architecture OSI incluant la passerelle nécessaire pour l'interconnexion au réseau public et à d'autres réseaux locaux. Les deux architectures de communication doivent cohabiter sur le même réseau. Ceci implique l'utilisation d'un protocole unique d'accès au support de communication.

Cette solution présente peu d'intérêt car elle nécessite l'implémentation dans chaque station de deux architectures de communication distinctes, sachant que les ressources demandées pour la mise en œuvre d'un système de communication sont importantes. L'aspect positif de cette approche consiste à éviter toute dépendance entre les protocoles de communication interne et les protocoles de communication externe. Ceci signifie qu'en dehors du problème de la disponibilité des ressources, cette solution est applicable quels que soient l'architecture et les protocoles utilisés pour la communication interne.

2) L'approche de conversion

Cette approche est basée sur l'utilisation des techniques d'interconnexion de réseaux. L'étude que nous avons faite du problème d'interconnexion dans le deuxième chapitre montre que la seule technique d'interconnexion qui peut être envisagée dans ces conditions pour assurer la communication entre les SD-RL est celle de la conversion. Il s'agit dans ce cas d'une conversion entre les protocoles spécifiques à chaque SD-RL et les protocoles OSI équivalents. Cette conversion, réalisée dans une passerelle, concerne chaque niveau de protocole dont les services sont considérés comme étant des services de communication externe. Typiquement cette conversion doit être réalisée au niveau transport et au niveau application (ce sont les services typiques de communication externe entre les SD-RL tels que nous l'avons vu dans le premier chapitre).

Cette solution présente deux avantages :

- une interface unique et commune est offerte pour la communication interne et pour la communication externe.

- une seule instance des couches OSI est supportée par le SD-RL, elle est implémentée dans la passerelle de conversion.

Le problème majeur de cette solution est lié à sa faisabilité. Comme nous l'avons vu, la conversion suppose que la sémantique des services offerts par chacun des protocoles convertis soit semblable. Cette condition est difficile à vérifier pour les services de niveau application, en particulier dans un système distribué à haute intégration offrant la visibilité d'une unique machine virtuelle gérée par un système d'exploitation réparti. L'autre inconvénient majeur de cette solution réside dans le fait qu'elle ne peut être systématique. Le processus de conversion est spécifique à chaque niveau de service de communication et à chaque SD-RL.

Les limites de la solution de conversion nous ont incité à rechercher une solution plus systématique au problème d'ouverture et adaptée à l'environnement d'un système distribué. Cette solution est décrite dans la section suivante.

3. DOSIS : une nouvelle approche pour l'ouverture des SD-RL au monde OSI

L'approche que nous proposons est basée sur la mise en œuvre d'un serveur de communication appelé *DOSIS (Distributed OSI Server)*. L'objectif de ce serveur est d'offrir aux utilisateurs et aux applications s'exécutant dans un environnement de SD-RL l'accès aux services OSI de plusieurs niveaux. Ceci leurs permettra de communiquer avec d'autres applications qui s'exécutent sur un autre SD-RL supportant DOSIS ou tout autre système ouvert au sens OSI (supportant l'architecture et les protocoles OSI). Il est à noter que le terme "application" désigne un ensemble de processus qui constituent un programme utilisateur, on utilisera le terme "couche application" quand il s'agit du niveau application au sens OSI.

Dans la suite de cette section, nous présentons la définition de DOSIS ainsi que son principe de base. Nous décrivons ensuite l'architecture de DOSIS et les fonctions des différents modules qui le composent. Les avantages et les caractéristiques de ce serveur sont discutés. Afin d'illustrer le fonctionnement et l'intérêt de DOSIS, des exemples de scénarios de communication sont donnés. Nous terminons par la description d'une réalisation d'un prototype de DOSIS.

3.1. Définition et caractéristiques de DOSIS

DOSIS est composé d'un ensemble d'entités qui sont implémentées sur plusieurs stations d'un SD-RL. Ces entités coopèrent en échangeant des informations afin d'offrir aux applications qui s'exécutent dans l'environnement de ce SD-RL l'accès aux services OSI de plusieurs niveaux (transport, session, et applications par exemple). Les principales caractéristiques de DOSIS sont les suivantes :

1) DOSIS contient une seule instance des couches OSI. Ces couches constituent des ressources partagées dont les services sont accessibles par toute application s'exécutant sur le SD-RL. Afin de respecter le principe de la transparence essentielle dans les SD-RL intégrés, la localisation des couches OSI doit rester transparente aux applications. Ainsi l'accès aux services offerts par les couches OSI est indépendant de la localisation de ces dernières. Ceci a deux conséquences importantes en terme de visibilité :

- l'utilisateur d'un programme d'application n'est pas de conscient de la distribution. Il a l'impression de supporter les couches OSI sur sa station locale. Ceci signifie que DOSIS offre une machine virtuelle OSI à chaque utilisateur du SD-RL quelle que soit sa localisation.

- de l'extérieur, l'ensemble du SD-RL apparaît comme *un système OSI unique* (Fig 3.1). Ce point de vue est en parfait accord avec l'esprit du modèle OSI (cf. chapitre 1). Ce dernier est concerné par l'échange des informations entre systèmes ouverts et non pas par le fonctionnement interne des composants de chaque système [ISO/DIS 7498]. Il suffit donc de définir la frontière du système ouvert, ce qui correspond dans notre cas à la frontière du SD-RL. L'approche que nous retenons et qui considère le SD-RL comme étant un seul système a l'avantage de cacher complètement la structure interne du SD-RL aux utilisateurs extérieurs. Ceci permet en particulier de simplifier le problème d'adressage comme nous le verrons dans les prochains paragraphes. L'idée de donner à un réseau local la vision d'un seul système a été exprimée plusieurs fois dans la littérature [Braden 83], [Cheriton 83], [Burg 84], [ECMA/TR21 84] sans que son exploitation soit toujours réelle ou complète.

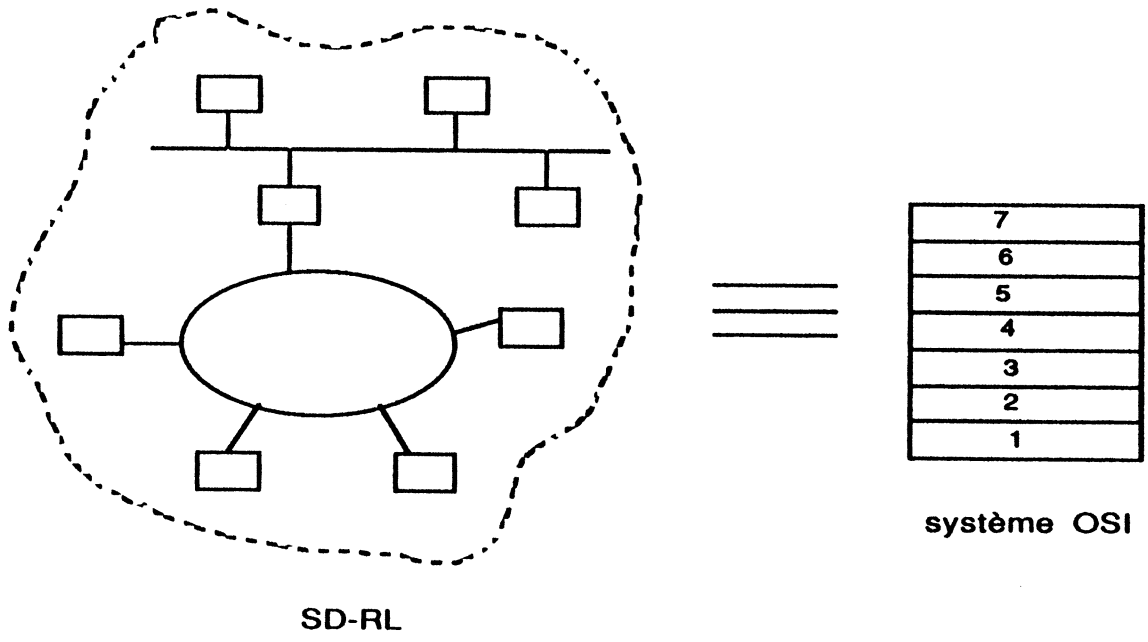


Fig 3.1 La vision externe d'un SD-RL supportant DOSIS

2) Afin de permettre la communication avec le plus grand nombre de systèmes, en particulier avec ceux qui implémentent partiellement les protocoles OSI (un sous-ensemble des couches OSI), DOSIS doit offrir aux applications une interface d'accès direct à plusieurs couches. L'architecture de DOSIS doit prendre en compte ce besoin. Le choix des couches devant offrir cette interface directe dépend des besoins spécifiques des utilisateurs de chaque SD-RL.

3) La mise en œuvre de DOSIS pour l'accès aux services OSI doit éviter de modifier (dans la mesure de possible) la syntaxe des primitives qui constituent l'interface originale des couches accessibles par les programmes utilisateurs. Ceci permettra le portage, dans l'environnement SD-RL, des programmes existants qui utilisaient les services offerts par les couches OSI dans un environnement centralisé.

4) Par définition, l'architecture de DOSIS est une architecture distribuée, car elle doit permettre l'accès à distance aux services offerts par les couches des protocoles OSI. Cependant, cette architecture peut tirer davantage profit des facilités de distribution dans un SD-RL et s'orienter encore plus vers la distribution. Ceci concerne particulièrement les couches de protocole. Ainsi on peut obtenir deux configurations de DOSIS :

   - la configuration "serveur OSI centralisé". Dans cette configuration, toutes les couches OSI se trouvent centralisées sur une seule machine physique (une station du SD-RL).

   - la configuration "serveur OSI distribué". Dans cette configuration, le fournisseur de service, qui est dans ce cas l'ensemble des couches OSI, sera distribué. Ceci signifie que toutes les couches OSI ne sont plus localisées sur la même machine, certaines résideront sur d'autres stations du SD-RL. L'intérêt de cette configuration est de distribuer la charge, en traitement et en stockage, sur plusieurs machines. Nous verrons par la suite comment cette distribution est réalisée.

3.2. Le principe du fonctionnement de DOSIS

3.2.1. Le modèle de service

Le fonctionnement de DOSIS est basé sur le modèle client-serveur. Le client est un programme d'application qui s'exécute sur une machine quelconque du SD-RL et qui utilise les services OSI d'une couche N pour communiquer avec des systèmes externes au SD-RL. Le serveur (DOSIS) est composé d'un ensemble d'entités coopérantes. L'entité principale est le fournisseur de service qui est constitué de l'ensemble des couches OSI. Ce fonctionnement client-serveur peut être schématisé de la façon suivante : lorsqu'un client a besoin de communiquer avec des systèmes OSI externes au SD-RL, il s'adresse au serveur en précisant le niveau de communication externe (une couche OSI) et le service demandé. Le fournisseur de service exécute le service demandé au moyen du protocole correspondant, il envoie ensuite la réponse au client. L'application de ce modèle, dans le but d'assurer la communication externe, fait apparaître un intérêt fondamental par rapport aux solutions d'interconnexion de réseaux exposées dans le chapitre précédent. Cet intérêt consiste à découpler le service de communication externe (inter-réseaux) du service de communication interne (intra-réseau). Le service de communication interne est utilisé comme moyen pour réaliser la communication externe (ceci est transparent à l'utilisateur du service de communication externe). Cependant, la condition contraignante qui consiste à disposer un sous-ensemble de services communs entre la communication externe et la communication interne, cette condition qui est nécessaire dans le cas de la conversion disparaît avec l'approche DOSIS. Nous examinerons avec plus de détails les avantages et les conséquences de cette approche.

3.2.2. La mise en œuvre du modèle client-serveur

Dans le modèle OSI, les interactions entre un fournisseur de service (une couche N) et l'entité utilisatrice des services de la couche N (soit la couche supérieure N+1, soit un processus utilisateur) se fait au moyen d'une interface. Cette interface est constituée concrètement d'un ensemble de primitives et de leurs paramètres. Dans un environnement centralisé, ces interactions sont directes, souvent sous forme d'appel de procédure. Il s'agit d'étendre ce modèle de service à l'environnement distribué d'un SD-RL où l'entité utilisatrice du service et l'entité qui fournit le service se trouvent a priori sur

des machines distinctes. Cette extension doit respecter autant que possible la contrainte de transparence. Ceci signifie que l'entité utilisatrice doit avoir la l'impression de supporter sur la machine locale l'entité qui fournit le service (la couche N), et de la même façon, cette dernière doit avoir l'impression d'interagir avec une entité utilisatrice du service résidant sur sa machine locale.

L'extension de ce modèle de service à l'environnement de SD-RL implique donc que les interactions entre l'entité utilisatrice des services de la couche N et la couche N elle-même ne soient plus directes. Ces interactions passent désormais à travers le système de communication interne du SD-RL. Le rôle de ce dernier est d'acheminer les demandes de services ainsi que les réponses entre le client (l'entité utilisatrice) et le fournisseur de service (la couche N). Afin d'assurer la transparence par rapport à l'introduction du système de communication entre le client et le fournisseur de service, deux entités logiques doivent être introduites (Fig 3.2). La première entité est introduite sur le site client, cette entité joue le rôle d'un représentant de l'entité fournisseur de service sur le site client. La transparence totale est assurée lorsque cette entité présente la même interface que celle de l'entité fournisseur de service (la couche N). La seconde entité est introduite sur le site de la couche N fournisseur de service. Cette entité représente le client sur ce site. Elle reproduit les appels aux services de la couche N faits par le client, elle reçoit les réponses de la part de la couche N. Ainsi, grâce à l'introduction de ces deux entités supplémentaires, la transparence est assurée. Ceci signifie qu'aucune modification n'est à apporter, ni sur le client ni sur le fournisseur de service, du fait qu'il s'exécutent sur deux machines distinctes du SD-RL. Les fonctions de ces deux entités supplémentaires seront décrites par la suite.

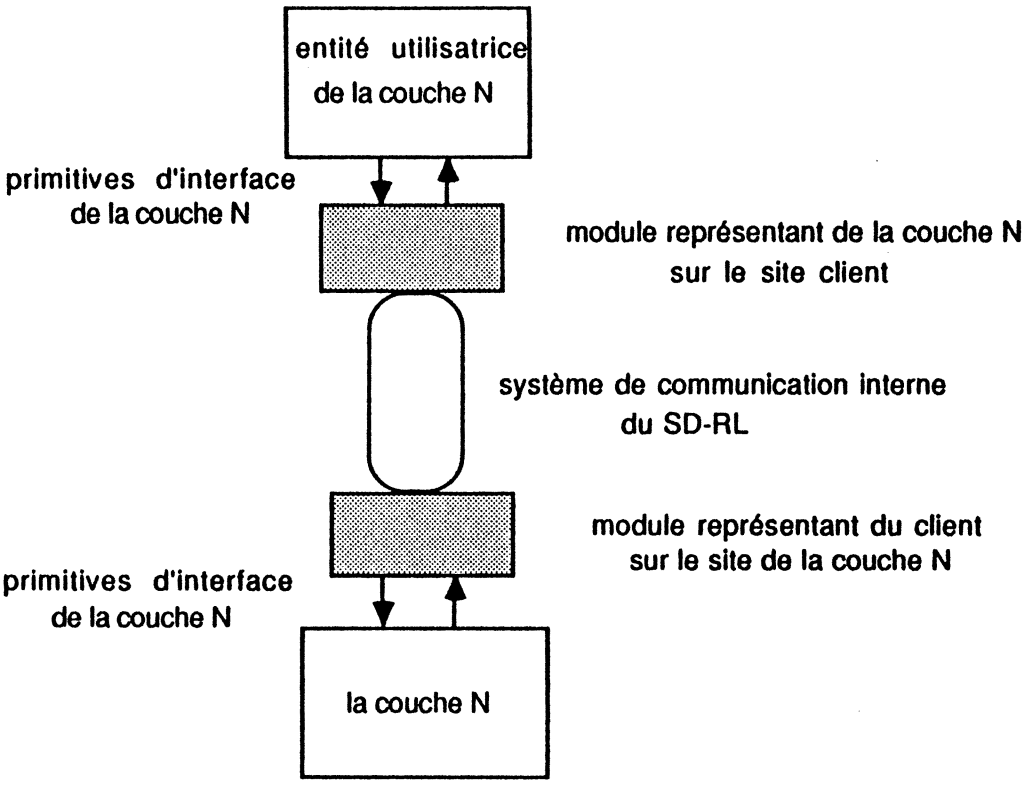


Fig 3.2 Le principe de fonctionnement de DOSIS

4. L'architecture et le fonctionnement de DOSIS

Comme nous l'avons indiqué, deux configurations de DOSIS sont proposées : le serveur centralisé et le serveur distribué. Les deux configurations sont basées sur la même approche et le même principe de base de fonctionnement. L'aspect de distribution existe également dans les deux configurations en ce qui concerne les interactions entre le programme utilisateur et la couche de communication OSI à laquelle il accède. C'est pourquoi le terme DOSIS est utilisé pour désigner l'approche en général. La configuration distribuée propose en plus la possibilité de distribuer les interactions entre les couches OSI en implémentant certaines couches sur des machines distinctes du SD-RL. La configuration de serveur centralisé peut être vue comme un cas particulier de la configuration distribuée. Nous préférons cependant, pour des raisons de clarté de l'exposé, respecter la démarche intellectuelle que nous avons suivie pour concevoir et développer DOSIS en commençant par la spécification du serveur centralisé suivie par celle du serveur distribué.

4.1. Spécification du serveur OSI centralisé

Nous rappelons d'abord que l'objectif du serveur OSI est de permettre à tout programme application s'exécutant dans l'environnement du SD-RL l'utilisation des services de communication OSI. Ces services sont assurés par l'ensemble des couches OSI et permettent aux applications du SD-RL de communiquer avec d'autres applications qui s'exécutent dans un environnement externe au SD-RL. Comme nous l'avons vu, l'idée principale du serveur est de partager une seule instance des couches OSI. Le serveur est constitué donc des sept couches OSI qui réalisent l'ensemble des protocoles OSI des différents niveaux. Ces couches OSI sont localisées sur une station du SD-RL appelée *le site serveur*. Nous appelons *site client* toute station du SD-RL capable de supporter un programme application faisant appel aux primitives de communication OSI. C'est le cas a priori de toutes les stations du SD-RL.

4.1.1. Eléments de l'architecture du serveur OSI centralisé

L'architecture du serveur OSI centralisé est composée de quatre types d'entités logiques (Fig 3.3). Chaque entité réalise un certain nombre de fonctions. Elle communique avec les autres entités afin de fournir les services du serveur OSI. Les quatre types d'entités sont les suivants :

- *fournisseur de services OSI*
- *agent client*
- *agent serveur(N)*
- *contrôleur*

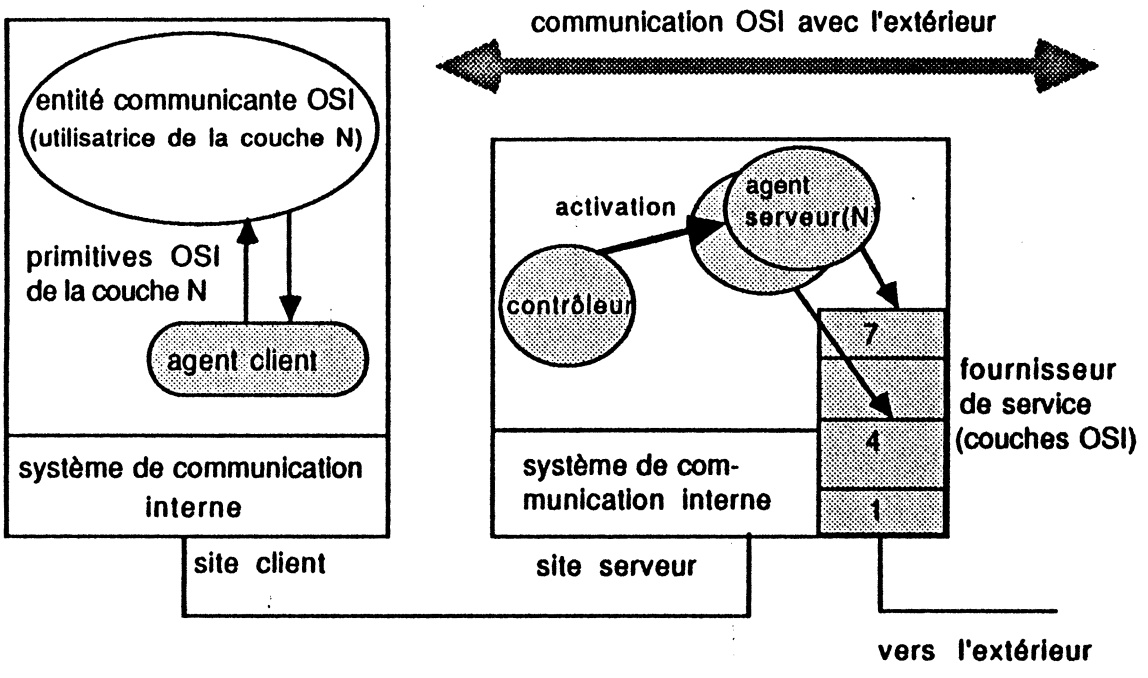


Fig 3.3 Eléments de l'architecture du serveur OSI centralisé

Nous examinons par la suite les fonctionnalités principales de chaque entité et ses interactions avec les autres entités.

1) Fournisseur de services OSI

Nous appelons l'ensemble des couches OSI : l'entité *fournisseur de services.* Le fournisseur de services est localisé sur le site serveur (ce qui caractérise le serveur OSI centralisé). Toute station du SD-RL peut être un site serveur à la seule condition de posséder une liaison physique avec le monde extérieur, c'est-à-dire avec d'autres systèmes, soit directement, soit à travers un réseau.

Pour répondre au besoin de communication externe que nous avons défini, le fournisseur de services OSI présente aux programmes utilisateurs des interfaces d'accès direct à plusieurs niveaux de communication, tels que le niveau transport et le niveau application. Ces interfaces sont constituées des primitives (et de leurs paramètres) qu'on appelle les primitives OSI.

2) Agent client

Une entité agent client réside sur chaque site client. Le rôle de cette entité est de représenter, sur les sites clients, les couches OSI dont les services sont exportés, c'est-à-dire accessibles à distance. Nous utiliserons le terme "agent client(N)" pour désigner un agent client qui est spécialisé pour l'accès à une couche N uniquement. Pour un agent client général ou lorsqu'il n'y a pas d'ambiguïté concernant la couche représentée, nous utiliserons le terme "agent client". C'est l'entité agent client qui permet de cacher la distribution des couches OSI aux programmes utilisateurs qui s'exécutent sur les sites clients. Une instance de l'entité agent client est associée à chaque programme utilisateur (client). L'interface que présente cette entité est constituée de l'ensemble des services des couches OSI accessibles à partir des sites clients. Ces services sont concrétisés par un ensemble de primitives dont la syntaxe et la sémantique doivent être identiques à celles qui sont présentées à l'origine par les couches OSI. L'interface d'accès est alors unique quelle que soit la nature de l'accès à ces services : local ou distant. Ceci permet, par conséquent, la portabilité des applications existantes utilisant les services OSI d'une façon locale vers des sites clients distants. Nous verrons plus loin qu'il sera parfois, pour des raisons liées à des implémentations spécifiques, difficile de respecter totalement cette contrainte de portabilité. L'interface offerte par l'agent client aux programmes utilisateurs sur les sites clients permet de leur offrir la vision de supporter une "machine virtuelle OSI". Ainsi, du point de vue des utilisateurs sur un site client quelconque, l'agent client est équivalent à l'ensemble des couches OSI.

Les fonctions de l'agent client sont les suivantes :

- l'agent client doit d'abord envoyer au site serveur une requête de demande d'ouverture d'une session d'accès aux services OSI de la couche N. Le terme "session" désigne ici une suite de demande de services OSI, il n'a pas de rapport avec celui utilisé dans la couche session OSI.

- à chaque appel utilisateur à une primitive OSI, l'agent client envoie vers le site serveur des informations (essentiellement constituées de l'identificateur de l'appel ainsi que les paramètres associés). Ceci permet de reconstituer cet appel sur le site serveur. La primitive est alors exécutée par le fournisseur de services. Ainsi une communication OSI

avec un système distant externe est générée.

- l'agent client est chargé ensuite de recevoir les paramètres de retour concernant la primitive exécutée et de les délivrer au programme utilisateur (client). Ce dernier a donc l'impression de supporter sur la station locale les couches OSI.

3) Agent serveur(N)

Une entité agent serveur(N) est une entité qui représente un client (programme utilisateur) sur le site serveur. N est la couche OSI de niveau N dont les services sont accédés à distance par ce client. Il existe autant d'entités agents serveurs que de couches potentiellement accessible à distance. Pour un serveur OSI qui offre l'accès aux services du niveau transport et du niveau application, il y aura deux entités agents serveurs : l'agent serveur(4) et l'agent serveur(7). A chaque client est associée une instance de chaque entité agent serveur(N). Ainsi, pour un client qui utilise le service transport, on lui associera une instance de l'agent serveur(4) ; pour un client qui utilise les services application, on lui associera un agent serveur(7) ; alors que pour un client qui utilise à la fois le service transport et le service application,on lui associera sur le site serveur un agent serveur(4) et un agent serveur(7).

Afin de minimiser l'utilisation des ressources sur le site serveur, la création des instances agents serveurs est dynamique. De manière simplifiée, on peut dire que les instances agents serveurs ont pratiquement la même durée de vie que les clients. Quand un client est actif (en cours d'exécution), ses agents serveurs sont actifs. Ils disparaissent quand le client se termine. Nous expliquons plus loin comment ces agents serveurs sont créés et détruits.

En ce qui concerne l'association entre les agents clients et les agents serveurs, les trois solutions suivantes sont valables sur le plan fonctionnel :

- un seul agent serveur, pour l'ensemble des couches accessibles, est associé à chaque client ;

- un agent serveur par couche est associé à tous les clients ;

- un agent serveur par couche est associé à chaque client.

Sur le plan de la réalisation, le choix d'avoir un agent serveur par couche et par client nous semble le plus approprié. En effet l'option qui consiste à avoir un seul agent serveur pour l'ensemble des couches accessibles a l'inconvénient de réserver beaucoup des ressources, alors que le client utilise généralement les services d'une seule couche à la fois. Nous avons aussi écarté la solution qui consiste à associer un agent serveur pour plusieurs clients pour la raison suivante : un agent serveur peut se terminer prématurément (en cas d'appel OSI erroné de la part d'un client par exemple), ceci affecte le fonctionnement des autres clients (ils voient leur agent serveur disparaître).

L'association entre un agent serveur(N) et un client consiste concrètement à créer une connexion logique entre l'agent client correspondant au client en question et l'agent serveur(N). Cette connexion permet de véhiculer, dans un sens, les primitives de la couche N appelées par le client vers l'agent serveur et, dans le sens inverse, les paramètres de retour qui résultent de l'exécution de ces primitives. Ainsi l'agent serveur fonctionne pour le compte d'un seul client. La durée de vie de cette connexion dépend de la durée de vie du client et de l'agent serveur. Nous discuterons des stratégies de création et de destruction des agents serveurs et de leurs connexions avec les agents clients par la suite.

Les fonctions de l'agent serveur(N) sont les suivantes :

- écouter les informations en provenance du client auquel l'agent serveur est associé.

- recevoir les informations concernant les appels aux primitives OSI de la couche N effectués par le client. Ces informations sont envoyées par l'agent client.

- reconstituer, à partir des informations envoyées par l'agent client, les appels aux primitives OSI, et les adresser au fournisseur de service (la couche OSI concernée) qui les exécute.

- transmettre les paramètres de retour qui résultent de l'exécution des primitives OSI vers l'agent client qui les délivre au programme utilisateur.

La communication entre l'agent client et l'agent serveur utilise un protocole simple de type client-serveur. Ce protocole consiste à envoyer des requêtes de l'agent client à l'agent serveur. Ce dernier demande leur exécution au fournisseur de service et retourne les réponses à l'agent client. Ce protocole est réalisé au moyen du service de communication interne qui doit être capable de transférer des données entre les sites de manière fiable.

La communication entre un agent client et un agent serveur a les caractéristiques suivantes :

- le protocole client-serveur est d'autant plus simple que le service de communication interne est fiable. Une hypothèse raisonnable, par rapport au service de communication interne dans un SD-RL, consiste à considérer la communication interne comme étant du niveau transport.

- la communication entre l'agent client et l'agent serveur est transparente aussi bien pour le programme utilisateur que pour les couches OSI.

- il n'existe aucune relation entre la sémantique des services accédés et le protocole client-serveur. Le même protocole permet indifféremment l'accès aux services OSI de différents niveaux. Ceci constitue une caractéristique fondamentale de l'approche du serveur OSI. Nous examinerons ce point lors de la présentation des avantages du serveur OSI.

- il est important de noter que les (N)-PDU relatifs aux communications OSI s'échangent exclusivement entre le site serveur et un système externe au SD-RL. L'échange entre les sites clients et le site serveur concerne uniquement les primitives d'interfaces OSI de la couche N.

4) Contrôleur

Une seule instance de l'entité contrôleur existe. Elle réside sur le site serveur. Le rôle du contrôleur est de gérer l'accès au serveur OSI et son fonctionnement. Le contrôleur est responsable des actions suivantes :

- création des instances agents serveurs sur le site serveur. A la demande des agents clients, le contrôleur crée les agents serveurs qui seront associés aux agents clients. Cette création n'est pas toujours possible comme nous le

verrons ci-dessous. Une fois l'agent serveur créé, la communication avec l'agent client associé a lieu sans l'intervention du contrôleur. L'opération de création est en réalité une activation d'une instance de l'entité agent serveur. Du point de vue de l'implémentation, cette opération consiste à créer un processus. Nous utiliserons indifféremment les termes "création" ou "activation" pour désigner cette opération.

- sur le site serveur, il existe au moins autant d'agents serveurs que de clients sur tous les sites clients. Ces agents serveurs nécessitent des ressources importantes afin de gérer les connexions avec les agents clients. Par conséquent, le contrôleur ne peut pas créer un nombre infini d'agents serveurs. Des critères doivent être définis pour allouer équitablement les agents serveurs aux clients, et aussi pour éviter l'écroulement des performances de la station du site serveur, ce qui a des conséquences directes sur les performances de la communication externe. Ces critères, pour être valables, doivent être définis en fonction de la spécificité des protocoles de communication interne et de l'implémentation dans chaque environnement de SD-RL. A titre indicatif, on peut limiter la création des agents serveurs selon les deux critères suivants : un nombre maximal d'agents serveurs par site client, et un nombre maximal total d'agents serveurs à ne pas dépasser.

- le contrôleur peut gérer également d'autres fonctions secondaires comme la gestion des droits d'accès aux services OSI ou la gestion d'un historique de ces accès.

Les quatre entités que nous avons décrites coopèrent en communiquant entre elles afin d'assurer les services fournis par le serveur OSI. Ceci est détaillé ci-dessous.

4.1.2. coopération entre les différentes entités

A partir de la description des fonctions des différentes entités du serveur OSI centralisé, nous présentons dans ce paragraphe les interactions entre ces différentes entités et le fonctionnement de l'ensemble. Celui-ci peut être résumé de la manière suivante (la figure 3.4 schématise cette coopération à un moment ou l'agent serveur est déjà créé) :

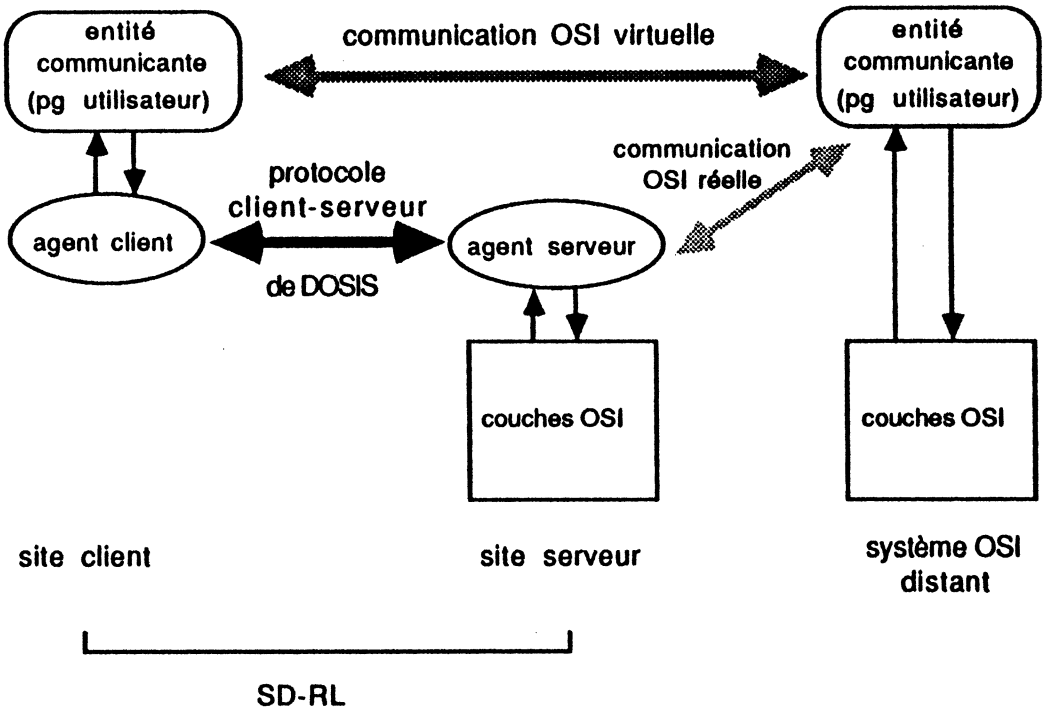


Fig 3.4 Coopération entre les entités de DOSIS

- à chaque client est associé un agent client sur le site client. Cet agent client intercepte les appels aux primitives OSI faits par le client.

- pour chaque client qui est utilisateur des services OSI de la couche N, un agent serveur(N) sur le site serveur est activé. Cette opération d'activation est réalisée par le contrôleur.

- une connexion logique est ensuite créée entre l'agent client associé au client et l'agent serveur. Cette connexion est réalisée au moyen des protocoles de communication interne. Un agent client peut être connecté à plusieurs agents serveurs (mais un par couche accessible). Par contre un agent serveur(N) travaille pour le compte d'un seul agent client.

- les primitives (avec leurs paramètres) interceptées par l'agent client sont codées et transmises à l'agent serveur. L'agent serveur demande l'exécution des primitives à la couche N, il récupère les paramètres de retour et les transmet à l'agent client. Cet échange entre l'agent client et l'agent serveur est réalisé moyennant un protocole simple de type client-serveur décrit plus loin.

- l'exécution des primitives OSI génère une communication OSI avec un programme communicant résidant sur le système externe adressé dans les primitives. La transparence de l'échange entre l'agent client et l'agent serveur par rapport au client (le programme utilisateur) et par rapport au serveur (la couche N) a la conséquence recherchée suivante : chacun des deux programmes utilisateurs résidant sur le site client et sur le site externe distant a l'impression de réaliser une communication au moyen des services et des protocoles OSI avec l'autre programme utilisateur (Fig 3.4). Cette communication virtuelle est en réalité composée de deux communications concaténées. La première communication est réalisée au moyen de service de communication interne selon un protocole spécifique Client-Serveur qui sera défini ultérieurement. Elle a pour but d'acheminer les primitives d'interface vers le site serveur où elles sont exécutées, ce qui déclenche la deuxième communication qui est une communication réalisée par les protocoles OSI avec un système externe distant supportant l'architecture OSI.

Un problème qui reste à résoudre est celui qui consiste à définir la stratégie de création (activation) et de destruction (déactivation) des agents serveurs. Pour des raisons de limitation des ressources disponibles, le nombre des agents serveurs actifs au même instant est limité. Il faut donc éviter qu'un agent serveur ne soit actif alors qu'il ne sert plus parce qu'il n'y a plus d'appels aux primitives OSI. La solution idéale consiste à activer un agent serveur(N) juste avant le premier appel aux primitives OSI de niveau N, et de désactiver cet agent serveur juste après le dernier appel à ces primitives. Il s'avère difficile d'arriver à cette solution idéale. Cependant plusieurs stratégies peuvent être envisagées :

1) la première stratégie consiste à exprimer l'activation et la désactivation des agents serveurs au niveau du programme utilisateur sous forme de deux appels supplémentaires. L'appel d'activation sera alors nécessaire pour pouvoir utiliser les services OSI. Par contre, rien ne peut obliger le programmeur à effectuer l'appel de désactivation ce qui peut donner lieu à des agents serveurs actifs alors que leurs clients ont disparu. Cette stratégie a aussi l'inconvénient de ne pas permettre la portabilité des applications déjà existantes qui n'utilisent pas ces appels supplémentaires.

2) la deuxième stratégie consiste à exprimer l'activation et la désactivation par l'agent client en fonction des appels d'ouverture et de fermeture de connexions OSI. Ceci signifie que les activations et les déactivations des agents serveurs se font au rythme des ouvertures et des fermetures des connexions. Ceci n'est valable que pour les protocoles OSI en mode connexion.

3) la troisième stratégie consiste à exprimer l'activation d'un agent serveur(N) par l'agent client au premier appel fait par le client aux primitives de la couche N. L'agent serveur reste actif indépendamment de la durée de la communication OSI. La désactivation est provoquée au moment de la terminaison du programme client.

C'est cette dernière stratégie que nous retenons par la suite car elle est satisfaisante fonctionnellement, indépendante du mode de communication (avec ou sans connexion) et réalisable facilement dans notre contexte d'implémentation.

4.1.3. Protocole Client-Serveur

Comme nous l'avons vu, le fonctionnement du serveur OSI est basé sur l'échange entre le site client et le site serveur. Cet échange a lieu entre l'entité agent client d'une part et les entités contrôleur et agent serveur d'autre part. L'objet de cet échange est de véhiculer sur le réseau des informations relatives aux services OSI appelés par les clients afin de les reproduire sur le site serveur. Nous définissons dans ce paragraphe les éléments essentiels du protocole utilisé pour gérer cet échange.

La figure 3.4 schématise la position de ce protocole dans l'architecture du serveur. La fonction de ce protocole est de reproduire fidèlement les interactions qui ont lieu entre le client et l'agent client, concernant les services de la couche N, au niveau de l'interface entre l'agent serveur et la couche N. Ceci donne l'impression au client de supporter localement les couches OSI (N, N-1, ...1), et de réaliser ainsi une communication OSI avec son correspondant qui est situé sur un système OSI externe.

On peut noter deux caractéristiques concernant ce protocole :

- reproduire les interactions qui ont lieu sur le site client sur le site serveur revient à exécuter des opérations à distance. Dans ce schéma d'exécution à distance, c'est l'agent client qui émet la demande d'exécution des primitives et c'est l'agent serveur qui les effectue et retourne les réponses. Il s'agit donc d'un protocole dissymétrique de type client-serveur.

- afin de rendre ce protocole transparent au client (ce qui est un objectif essentiel du serveur), l'échange entre le client et le serveur doit être synchrone. Ceci signifie que le client doit être bloqué en attendant la réponse du service demandé.

Ce protocole est donc proche d'un protocole d'appel de procédure à distance.

Nous présentons dans la suite une description informelle de ce protocole à travers les éléments suivants :

- les règles d'échange ;
- les structures de données utilisées appelés CS-PDU (Client-Server Protocol Data Unit). Nous utilisons le terme "PDU" quand il n'y a pas d'ambiguïté concernant le protocole auquel les PDU sont associées ;
- l'interface abstraite de ce protocole.

Le protocole client-serveur est décomposé en trois phases qui sont décrites ci-dessous. Cette description est indépendante d'une stratégie d'activation ou de désactivation particulière des agents serveurs.

La figure 3.5 schématise les échanges selon le protocole client-serveur.

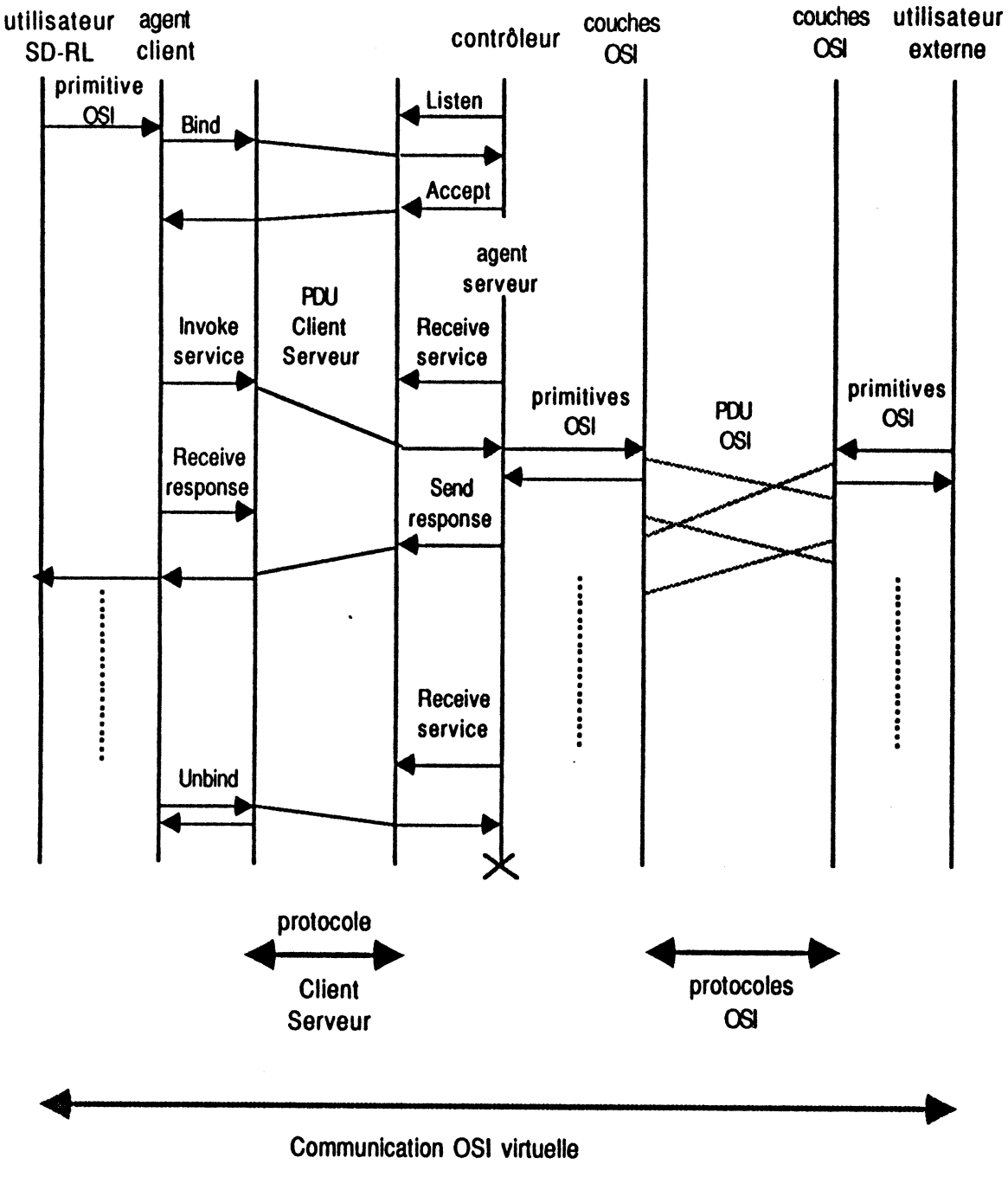


Fig 3.5 Echanges agent client - agent serveur
suivant le protocole client-serveur

1) Phase d'ouverture d'une session de service

L'objet de cette phase est d'ouvrir une session pour l'utilisation des services de communication de la couche N. Cette phase met en communication un agent client et le contrôleur. La demande d'ouverture de service est toujours initiée par l'agent client et adressée vers le contrôleur. L'exécution de cette demande d'ouverture consiste pour l'agent client à réaliser les actions suivantes :

- ouverture d'une connexion avec le contrôleur au moyen du service de communication interne. L'adresse du contrôleur selon le schéma d'adressage interne est une information supposée connue par les agents clients.

- envoi sur la connexion ouverte avec le contrôleur d'une PDU qui demande l'ouverture d'une session de service concernant les services de la couche(N).

- le contrôleur est normalement en état d'attente de demande d'ouverture de service. A la réception de cette demande, il vérifie le droit d'accès du client, la disponibilité des ressources nécessaires pour la réalisation de service et la possibilité de créer un agent serveur(N) pour le compte du client demandeur. Si une de ces conditions n'est pas vérifiée le contrôleur envoie vers l'agent client une PDU de compte-rendu négatif. La session de service ne peut donc pas avoir lieu. Dans le cas inverse le contrôleur crée un agent serveur(N) en lui passant l'identificateur de la connexion ouverte avec l'agent client. Il envoie ensuite vers l'agent client une PDU de compte-rendu positif. La session de service est alors ouverte. La communication entre l'agent client et le contrôleur se termine. Ce dernier retourne alors à l'état d'attente de demande d'ouverture de service.

En fonction de la stratégie d'activation que nous avons choisie, la demande d'ouverture est déclenchée au premier appel à une primitive de la couche N fait par le client.

Les primitives suivantes sont utilisées lors de la phase d'ouverture

BIND(N, ID_Client, ID_Session, DIAG)
/*
Cette primitive est appelée par l'agent client pour demander l'ouverture d'une session de service avec un agent serveur. Elle provoque l'ouverture d'une connexion avec le contrôleur et envoie sur cette connexion une PDU de demande d'ouverture vers le contrôleur. C'est un appel bloquant en attendant l'arrivée d'une PDU de compte-rendu. Une seule session est ouverte à un moment donné avec un agent serveur(N). Les paramètres suivant sont utilisés :

N : le numéro de la couche accédée (paramètre d'appel)
ID_Client : identificateur du client (paramètre d'appel), il permet au contrôleur de vérifier les droits d'accès du client.
ID_Session : identificateur interne de la session de service (paramètre de retour). Une bijection existe entre une session de service et une connexion au niveau du service de communication interne. Par conséquent la gestion des indentificateurs des sessions peut être simplifiée en faisant une correspondance directe avec les identificateurs des connexions.
DIAG : un compte-rendu positif ou négatif de l'ouverture de la session (paramètre de retour).
*/

LISTEN(N, ID_CLIENT)
/*
Cette primitive est appelée par le contrôleur. C'est un appel bloquant en attendant l'arrivée d'une PDU de demande d'ouverture d'une session de service. N et ID_CLIENT sont des paramètres de retour. A l'arrivée d'une telle demande, le contrôleur tente d'activer un agent serveur(N) pour le compte du client demandeur.
*/

Selon que l'activation a réussi ou a échoué, une des deux primitives suivantes est appelée :

ACCEPT()
/*
Cette primitive est appelée par le contrôleur lorsqu'il a réussi la création d'un agent serveur(N). Le contrôleur transmet à l'agent serveur l'identificateur de la connexion qui le relie à l'agent client. La connexion relie désormais l'agent client et

l'agent serveur. L'exécution de cette primitive provoque l'envoi d'une PDU de compte-rendu positif vers l'agent client.

*/

REFUSE ()

/*

Cette primitive est appelée par le contrôleur lorsqu'il a échoué dans sa tentative de création d'un agent serveur. Son exécution provoque l'envoie d'une PDU de compte-rendu négatif vers l'agent client.

*/

2) Phase d'échange de données

Une fois la session de service ouverte entre l'agent client et l'agent serveur, un échange de type requête-réponse peut avoir lieu. L'objet de cette échange est de véhiculer les primitives OSI appelées par le client et le résultat de leur exécution par la couche N. Comme dans la phase d'ouverture et de fermeture, c'est l'agent client qui initialise les requêtes et l'agent serveur qui répond. Dans cette phase, l'agent client envoie généralement une requête à chaque appel d'une primitive OSI fait par le client. Cette requête est transmise dans une PDU de demande de service. A chaque arrivée d'une PDU vers l'agent serveur (cette PDU transporte des informations concernant la primitive OSI appelée par le client), une primitive OSI est appelé par l'agent serveur ce qui génère un échange de PDU OSI avec le correspondant sur le système externe distant. eci est différent des phases d'ouverture et de fermeture dans lesquelles les requêtes sont consommées par l'agent serveur sans générer de communication vers l'extérieur. Comme nous l'avons vu, l'échange entre l'agent client et l'agent serveur est de type synchrone (l'agent client est bloqué en attendant la réponse de l'agent serveur). Il n'y a donc qu'une PDU qui circule à la fois dans un sens ou dans l'autre. Les primitives suivantes sont utilisées :

INVOKE-SERVICE(ID_SESSION, SB, RB)

/*

Cette primitive est appelée par l'agent client suite à un appel à une primitive OSI de la couche N par le client. Son exécution génère l'envoi d'une PDU de demande de service vers l'agent serveur, demandant l'exécution de la primitive OSI. L'appel est

bloquant en attendant la réception d'une PDU de réponse qui contient les paramètres de retour de la primitive OSI de la part de l'agent serveur. Les paramètres suivants sont utilisés :

ID_SESSION : C'est l'identificateur de la session de service (voir ci-dessus). Il permet d'aiguiller les primitives OSI vers l'agent serveur approprié.

SB : Service Bloc. C'est un paramètre d'appel qui est constitué d'une structure de données de taille variable. Elle contient l'identificateur de la primitive OSI appelée par le client et les paramètres d'appel de cette primitive. L'agent client est chargé de constituer le SB à partir de l'appel à la primitive OSI effectué par le client. Il doit remplacer toutes les variables de type référence par les valeurs référencées. Le SB est transmis dans la PDU de demande de service.

RB : Result Bloc. C'est un paramètre de retour. Il est constitué d'une structure de données de taille variable. Cette structure, qui est transmise par l'agent serveur dans le PDU de réponse, contient les valeurs des paramètres de retour de la primitive OSI appelée.

*/

RECEIVE_SERVICE (SB)

/*

C'est une primitive bloquante appelée par l'agent serveur. Elle permet la réception d'un SB envoyé par l'agent client dans une PDU de demande de service. A partir du contenu du SB, l'agent serveur reconstitue la primitive OSI appelée par le client. Il demande l'exécution de cette primitive à la couche concernée et transmet les paramètres de retour dans une PDU de réponse au moyen de la primitive SEND_RESPONSE. Un autre type de PDU peut arriver lorsque l'agent serveur est en état d'attente de réception de service, c'est la PDU de fermeture de session. L'agent serveur récupère alors un SB particulier indiquant la fermeture de la session de service par le client.

*/

SEND_RESPONSE (RB)

/*

C'est une primitive appelée par l'agent serveur. Il permet de transmettre les paramètres de retour d'un appel de primitive OSI vers l'agent client. Les informations concernant ces paramètres et le compte-rendu de l'exécution de la primitive sont codés dans la structure RB qui est transmise dans une PDU de réponse. L'agent serveur doit remplacer dans RB tout paramètre de type référence par la valeur référencée. A la

réception, l'agent client effectue la transformation inverse avant de délivrer ces paramètres au client.

*/

3) Phase de fermeture de session de service

La fermeture d'une session de service est initialisée par l'agent client. L'agent serveur qui est normalement dans un état d'attente de requête de l'agent client reçoit l'indication de fermeture de service. Il libère les ressources réservées pour cette session et se termine. La primitive utilisée est la suivante :

```
UNBIND (ID_SESSION)
/*
   Cette primitive est appelée par l'agent client pour terminer la session de
   service(ID_SESSION) avec l'agent serveur. L'appel est non bloquant (pas d'attente de
   confirmation de l'agent serveur). L'exécution de cette primitive provoque l'envoi d'un
   PDU de fermeture de session et ensuite la fermeture de la connexion qui relie
   l'agent client et l'agent serveur.
*/
```

4.1.4. Quelques remarques sur le fonctionnement du serveur

Nous commentons dans ce paragraphe certains aspects du fonctionnement du serveur et du protocole client-serveur.

a) Complexité du protocole client-serveur

Le protocole client-serveur présenté ci-dessus est assez simple, il ne fait aucun contrôle d'erreurs, car il suppose un service de communication interne fiable en mode connexion. En plus, il ne fait pas apparaître le traitement à faire en cas de rupture de moyen de communication. Dans un environnement réel, ces aspects doivent être considérés en fonction des caractéristiques du service de communication interne utilisé.

b) Primitives OSI bloquantes ou non bloquantes

Le choix d'une communication de type synchrone entre l'agent client et l'agent serveur est nécessaire pour rendre cette communication transparente au client. Ceci permet en particulier de conserver la nature bloquante ou non bloquante des primitives OSI lorqu'elles sont appelées par le client (à distance).

c) Appels OSI entrants et sortants

La communication globale entre le programme utilisateur sur un site client et son correspondant sur un site externe est composée de deux communications. La première a lieu entre l'agent client et l'agent serveur, elle a pour objectif d'échanger les (N)-SDU (Service Data Unit) entre le client utilisateur des services de la couche N et la couche N sur le site serveur. La deuxième communication a lieu entre les deux couches N homologues sur le site serveur et sur le site externe, dont l'objet est d'échanger les (N)-PDU. La relation dissymétrique entre l'agent client et l'agent serveur représente l'interface entre l'utilisateur de service de la couche N et la couche N. A cette interface, selon le modèle de service OSI, les flux d'information peuvent "monter" de la couche N vers l'utilisateur (Indication ou Confirmation) ou "descendre" de l'utilisateur vers la couche N (Requête ou Réponse). Dans la plupart des implémentations, ces flux d'information sont contrôlés par l'utilisateur de la couche N (c'est l'utilisateur qui fait l'appel à la couche N). Ceci se traduit, dans le cas du serveur OSI, par la relation dissymétrique client-serveur entre l'agent client et l'agent serveur. Cependant, ce modèle ne produit aucune dissymétrie entre l'utilisateur sur le SD-RL et son correspondant sur le site externe. L'un ou l'autre peut être indifféremment l'initiateur dans la communication OSI.

d) Adressage

L'adressage entre les entités communicantes OSI est rendu simple grâce aux trois éléments suivants qui caractérisent l'approche DOSIS :

- le service de communication externe est un service qui possède son interface propre visible à l'utilisateur qui est l'interface OSI. Par

conséquent, il n'est pas nécessaire de réaliser une correspondance entre le schéma d'adressage interne et celui de l'adressage externe qui est le schéma d'adressage OSI.

- la vision du SD-RL de l'extérieur est celle d'un système OSI unique. Par conséquent le seul site adressable de l'extérieur est le site serveur.

- selon le principe de DOSIS, à chaque client est associé un agent serveur qui le représente sur le site serveur. Les données qui sont reçues par l'agent serveur sont aiguillées grâce à cette association vers le client.

Pour adresser une entité communicante quelconque, il suffit donc d'adresser son agent serveur sur le site serveur en utilisant le schéma d'adressage OSI (on adresse le (N)-SAP auquel est lié l'agent serveur). L'aiguillage des données vers le client destinataire se fait au moyen de la connexion qui relie l'agent serveur à l'agent client attaché au client destinataire.

e) Correspondance appels client - appels agent serveur

Le fonctionnement idéal du serveur consiste à ce que l'agent serveur reproduise fidèlement les appels du client aux primitives OSI. Ainsi, un appel OSI du client doit générer l'exécution de la primitive de demande de service par l'agent client et ensuite la demande d'exécution de cette primitive par l'agent serveur. La réalisation de ce schéma simple peut poser des problèmes dans certains cas. Supposons une primitive OSI du type envoi de données (au niveau transport par exemple). La taille de ces données (SDU) n'est pas limitée dans les protocoles OSI. Ceci peut poser un problème à deux niveaux :

- le système de communication local peut imposer une limite sur la longueur des données transmises. Ceci peut être résolu en utilisant un mécanisme de segmentation et de réassemblage au niveau du protocole client-serveur.

- un problème plus délicat apparaît au niveau de l'agent serveur. Il est possible que les ressources mémoires ne soient pas suffisantes pour reconstruire l'appel OSI avec la même longueur de données. Dans ce cas deux solutions peuvent être proposées. La première consiste à

remplacer l'appel d'origine par une suite d'appels d'envoi de données au niveau de l'agent serveur. Cette infidélité dans la reconstruction des appels peut aboutir en cas d'erreurs à des incohérences entre la vision du client et la réalité concernant l'état de la communication. C'est pourquoi nous préférons la deuxième solution qui consiste simplement à imposer une limite sur la longueur des données (SDU) de l'utilisateur.

f) Dépendance entre le client et son agent serveur

L'agent serveur est le représentant d'un client sur le site serveur, le nombre des agents serveurs actifs à un instant donné n'est limité que par les ressources disponibles sur le site serveur. En plus d'une bonne stratégie d'activation et désactivation (cf 4.1.2), il est nécessaire de prendre en compte les cas d'exception suivants :

- terminaison par exception du client ou de l'agent client,
- panne du moyen de communication interne.

Dans ces cas, un agent serveur actif devient inutile, et une communication OSI reste établie alors que le client a disparu. Il faut donc que ces événements provoquent la libération des ressources allouées pour la communication OSI et la terminaison de l'agent serveur. Ceci peut-être réalisé de manière dépendante de l'environnement de l'implémentation et des caractéristiques du protocole de communication interne (certains événements d'exception peuvent être signalés sur la connexion qui relie l'agent client et l'agent serveur). La solution la plus générale reste l'utilisation d'un mécanisme basé sur la temporisation au niveau du protocole client-serveur.

4.2. Spécification du serveur OSI distribué

L'idée du serveur distribué est de tirer avantage de l'environnement de SD-RL pour distribuer la charge qui résulte de l'implémentation des couches OSI. En effet la mise en œuvre de ces couches nécessite beaucoup de ressources. Avec le serveur distribué, les couches OSI ne sont plus concentrées sur un site serveur mais distribuées sur plusieurs sites. La distribution maximale est obtenue en implémentant chaque couche sur un site différent. Pour des raisons de performance, il n'est pas intéressant de retenir ce cas limite. Une position intermédiaire plus raisonnable consisterait à distribuer les sept couches sur deux ou trois sites. La configuration de cette répartition des couches ne fait pas partie de la spécification du serveur distribué. C'est à la charge du réalisateur de DOSIS de choisir cette configuration en fonction de la charge et des ressources disponibles sur les différents sites.

4.2.1. Eléments de l'architecture de DOSIS distribué

Le principe de fonctionnement et l'architecture du serveur centralisé sont extrapolés pour être appliqués au serveur distribué. Ainsi les interactions entre une couche N+1 et la couche inférieure N ne sont plus directes si ces deux couches sont localisées sur deux sites différents. Ces interactions (les primitives d'interface et les paramètres) sont véhiculées par le système de communication interne du SD-RL. Ceci doit être transparent pour les couches N+1 et N, de telle sorte que la couche N+1 doit avoir l'impression de supporter la couche N sur son site, et de la même façon la couche N doit avoir l'impression de supporter la couche N+1 sur son site. Cette transparence est réalisée (comme pour le serveur centralisé) grâce à deux modules : l'agent client(N) qui est le représentant de la couche N sur le site de la couche N+1, et le module agent serveur(N) qui est le représentant de la couche N+1 sur le site de la couche N (Fig 3.6).

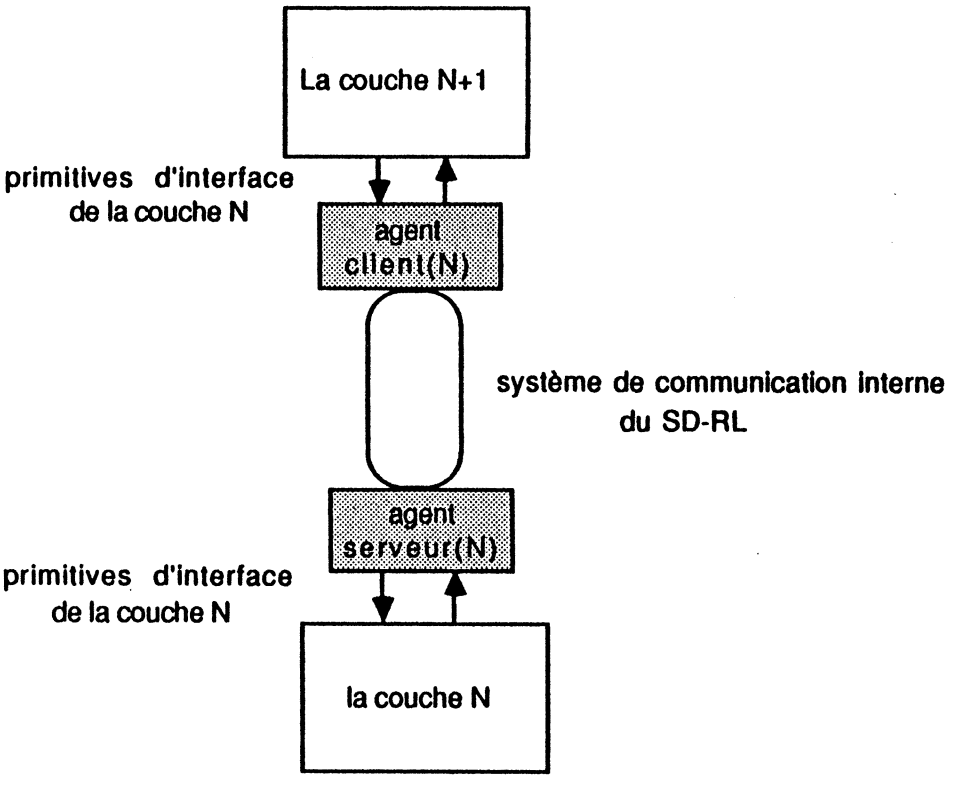


Fig 3.6 Le principe de fonctionnement du serveur OSI distribué

Dans cette configuration distribuée on retrouve les entités de base que nous avons vues pour la configuration centralisée :

- *agent client(N)*

  c'est un agent client qui travaille pour le compte de la couche N+1 ou un un programme utilisateur. Il intercepte les primitives de la couche N appelées par la couche N+1 pour les communiquer à l'agent serveur(N). Il se trouve sur le même site que celui de la couche N+1.

- *agent serveur(N)*

  c'est un agent serveur qui représente la couche N+1. Il communique avec l'agent client(N) pour échanger les primitives et les paramètres appelés par la couche N+1. Il reproduit ces primitives et demande leur exécution à

la couche N. Il se trouve sur le même site que celui de la couche N.

- *Fournisseur de service(N,M)*

  c'est une suite de couches OSI (N est la couche supérieure et M est la couche inférieure). La conséquence de l'approche distribuée est que le fournisseur de service n'est plus unique, car toutes les couches ne sont plus regroupées sur le même site. Il y a autant de fournisseurs de service que de partitions des couches OSI. Le cas particulier d'un fournisseur de service(7,1) représente le serveur centralisé.

- *Contrôleur*

  on associe à chaque fournisseur de service(N,M) un contrôleur qui a le rôle d'activer les agents serveurs, soit pour le compte d'une couche supérieure, soit pour le compte d'un utilisateur (si la couche est accessible directement par les utilisateurs).

Les fonctions de chacune des entités ainsi que celles du protocole client-serveur entre les agents clients et les agents serveurs ne changent pas par rapport au cas du serveur centralisé. Cependant, les éléments nouveaux suivants apparaîssent :

- un agent serveur(N) existe, soit parce que la couche N est accessible par les programmes utilisateurs, soit parce que c'est la couche supérieure dans un site qui contient un fournisseur de service(N,M).

- dans le cas du serveur centralisé, nous avions deux types de sites : le site client (la plupart des stations du SD-RL sont des sites clients) et le site serveur (un seul site serveur existe, c'est celui qui contient les couches OSI). Dans le cas du serveur distribué nous avons les trois types de sites suivants (Fig 3.7) :

  - *site client* : c'est un site qui peut accéder aux services du serveur OSI distribué (en principe, c'est le cas de toute station du SD-RL). Il contient les agents clients de toutes les couches accessibles directement par l'utilisateur.

- *site intermédiaire* : c'est un site qui contient un fournisseur de service(N,M) constitué d'un sous-ensemble des couches OSI. Il y a autant de sites intermédiaires que de partitions des couches OSI. Ce site doit également supporter un agent client(M-1) qui réalise la liaison avec la couche M-1 et un agent serveur(N) qui permet à la couche N+1 l'accès au service de la couche N.

- *site passerelle* : c'est un site qui contient le fournisseur de service dont la couche inférieure est la couche physique(1), c'est-à-dire un fournisseur de service(K,1). C'est le point de connexion entre l'environnement local et le monde extérieur. Ce site contient également un contrôleur.

- tous les agents serveurs ne sont plus localisés sur un site. Par conséquent chaque agent client(N) doit connaître la localisation de l'agent serveur(N).

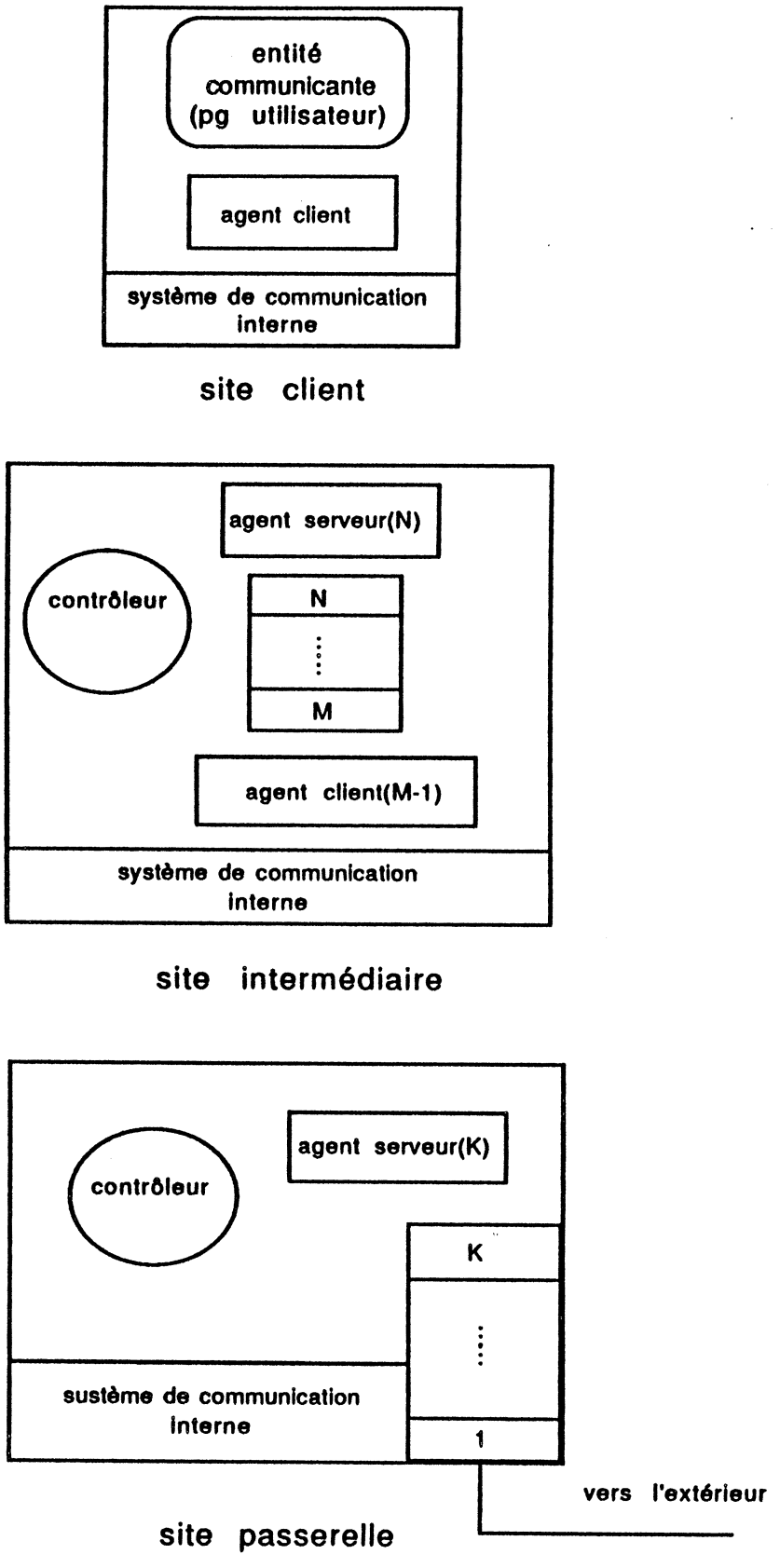


Fig 3.7 Eléments de l'architecture du serveur OSI distribué

4.2.2. Un scénario d'utilisation du serveur OSI distribué

Afin de clarifier le fonctionnement du serveur distribué, nous décrivons dans ce paragraphe un scénario possible de l'utilisation de ce serveur. Dans ce scénario, il s'agit de réaliser une communication OSI au niveau application entre un programme utilisateur situé sur un site faisant partie d'un système distribué basé sur un réseau local et un programme utilisateur situé sur un système distant centralisé supportant l'architecture OSI. Le réseau local supporte une architecture de communication spécifique, il est cependant connecté au réseau public X25. Le système distant est également connecté au réseau X25.

La configuration que nous retenons ici (choisie à titre d'exemple) est celle qui consiste à partitionner les couches OSI en deux groupes : d'une part la couche application et d'autre part les couches X25, transport, session, et présentation.

Par conséquent, nous aurons la structure DOSIS suivante (Fig 3.8) :

- le site client qui contient l'agent client application qui représente, pour le programme utilisateur communicant, une abstraction de la couche application OSI.

- un site intermédiaire qui contient le fournisseur de service(7) qui est la couche application, le contrôleur associé, l'agent serveur application et enfin, l'agent client présentation qui représente une abstraction de la couche présentation sur ce site.

- un site passerelle qui contient le fournisseur de service(6,1) (les couches X25, transport, session, et présentation), le contrôleur associé, et enfin, l'agent serveur présentation qui travaille pour le compte de la couche application.

Grâce à cette structure, le programme utilisateur, sur le site client, peut maintenant communiquer au niveau application avec son correspondant sur le système OSI distant. Cette communication OSI virtuelle est composée de trois communications intermédiaires :

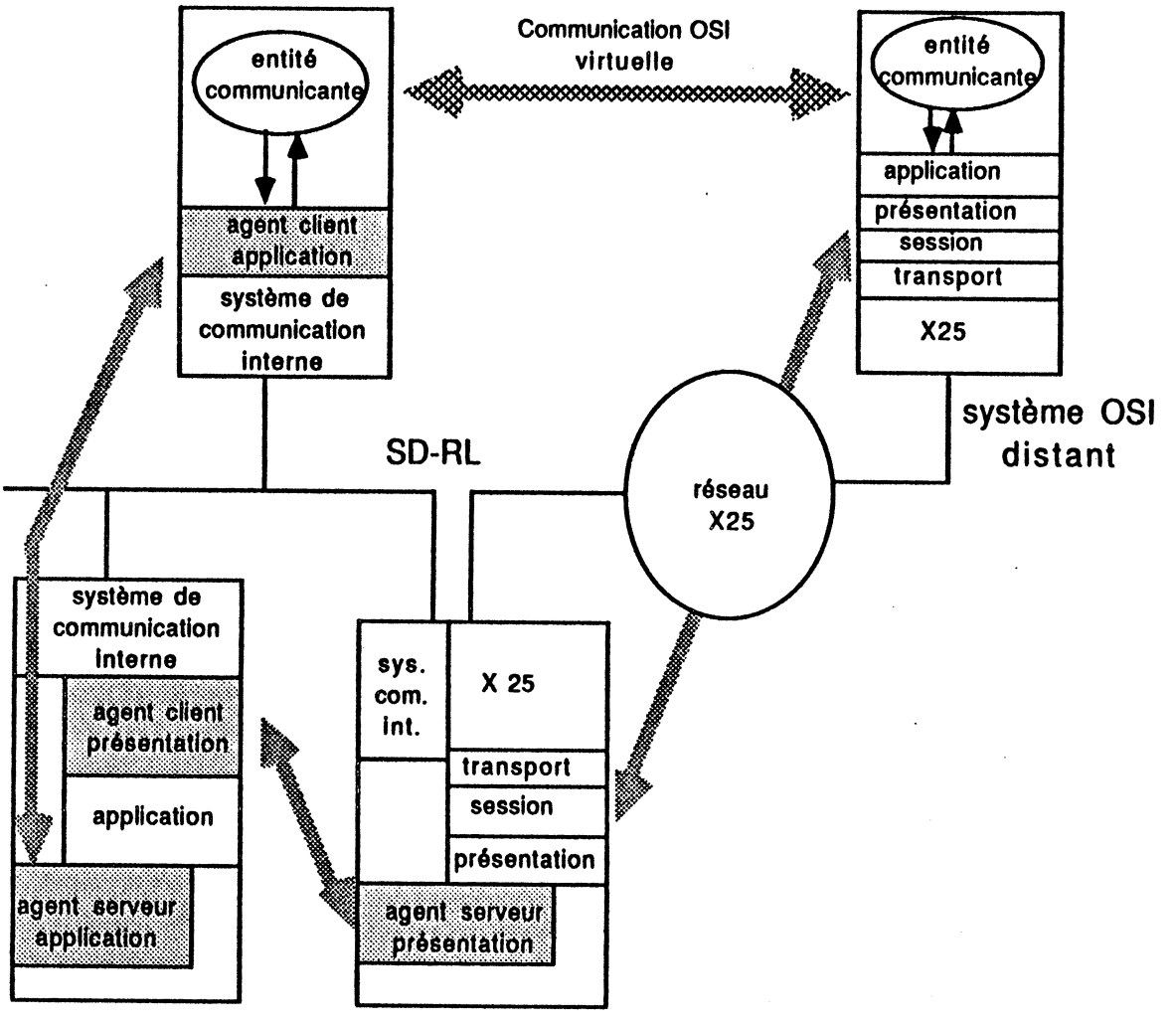


Fig 3.8 Scénario de communication utilisant le serveur OSI distribué

- la première communication a lieu entre l'agent client application sur le site client et l'agent serveur application sur le site intermédiaire. Cette communication se déroule selon le protocole client-serveur basé sur les protocoles de communication interne au SD-RL. Elle permet d'acheminer les primitives de la couche application OSI appelées par le client. L'exécution de ces primitives par la couche application génère la communication suivante.

- la deuxième communication a lieu entre l'agent client présentation sur le site intermédiaire et l'agent serveur présentation sur le site passerelle. Elle permet d'acheminer les primitives de la couche présentation appelées par la couche application (toujours selon le protocole client-serveur). L'exécution de ces primitives génère la communication suivante.

- la troisième communication est une communication OSI qui a lieu entre les deux couches présentation, celle sur le site passerelle, et celle sur le site OSI distant.

La combinaison de ces trois communications donne aux deux programmes communicants sur le site client et sur le système OSI distant la vision d'une communication OSI au niveau application.

5. Intérêts de l'approche DOSIS

L'approche DOSIS présente beaucoup d'intérêts comme solution pour l'ouverture des systèmes distribués au monde extérieur. Elle peut aussi être vue comme une méthode pour interconnecter des réseaux hétérogènes. A cet égard, DOSIS présente des avantages par rapport aux solutions d'interconnexion étudiées dans le deuxième chapitre. Les points suivants résument les intérêts et les avantages principaux de DOSIS :

- DOSIS est une solution qui est compatible sur le plan conceptuel avec les résultats des travaux de normalisation et notamment avec le modèle OSI. Elle utilise les services et les protocoles OSI pour assurer l'ouverture des systèmes informatiques les uns vers les autres, ce qui est l'objectif principal de l'architecture OSI.

- DOSIS est basé sur le principe de partage des ressources qui est un des aspects essentiels des SD-RL. D'une part, il réalise une économie de ressources en évitant l'implémentation des couches OSI sur toutes les machines du SD-RL, d'autre part il permet une meilleure distribution de la charge grâce à la version distribuée. A cause de leurs ressources limitées, certaines stations ne peuvent pas supporter l'architecture OSI, DOSIS leur permet cependant l'accès aux services OSI.

- DOSIS permet facilement d'intégrer toute modification, nouvelle fonction, ou nouveau protocole qui résulte des travaux en cours de l'ISO ou de l'évolution future du modèle OSI. Cette facilité d'intégration est due à deux caractéristiques importantes :

    - le principe de la mise en œuvre de DOSIS est indépendant des protocoles OSI proprement dits.

    - un seul exemplaire des couches OSI est implémenté dans un SD-RL.

- la vision du SD-RL offerte par DOSIS, qui est celle d'un système OSI unique, évite aux systèmes externes de connaître la structure interne du SD-RL. Ceci a pour conséquence de simplifier le mécanisme d'adressage. Cette vision est assurée sans imposer des contraintes sur le fonctionnement interne du SD-RL.

- DOSIS permet d'assurer la communication entre des réseaux totalement hétérogènes. L'approche utilisée est systématique à la fois dans sa conception et dans sa mise en œuvre, quel que soit le niveau de la communication externe et quel que soit le service de la communication interne. Aucune contrainte n'est imposée sur le service de communication interne pour la réalisation de DOSIS. Ce qui est fondamental dans cette approche, c'est la séparation entre le service de communication externe et le service de communication interne. Ceci est à l'opposé de la technique de conversion qui nécessite que les protocoles convertis soient proches fonctionnellement l'un de l'autre.

- la mise en œuvre de la solution DOSIS est simple car elle ne fait pas appel à de nouveaux mécanismes de communication ou de systèmes d'exploitation. Cette mise en œuvre peut être fortement réduite dans des systèmes distribués à haute intégration.

- DOSIS permet la portabilité des applications existantes utilisatrices des services OSI. Des applications qui sont développées pour fonctionner sur un système OSI centralisé peuvent continuer à s'exécuter dans l'environnement d'un SD-RL quel que soit leur site d'exécution. Ceci est vrai, à condition que la mise en œuvre réussisse à garder les mêmes interfaces originales d'accès aux services des différentes couches (cette

condition dépend fortement de la nature de ces interfaces).

- à première vue, le fonctionnement de DOSIS, basé sur l'acheminement des primitives d'interface par le système de communication local, peut paraître pénalisant pour les performances. En réalité dans certains cas , on peut, au contraire s'attendre à un certain gain dans les performances. C'est d'autant plus vrai que le niveau de communication externe est élevé. En effet, le niveau de communication externe est indépendant du niveau de communication interne dans DOSIS. Ainsi, pour réaliser une communication externe au niveau application il suffit d'utiliser une communication interne au niveau transport par exemple, alors que pour la solution de conversion, les deux communications interne et externe doivent être de niveau application. Le nombre total de couches traversées est donc supérieur par rapport au cas de DOSIS. Nous pensons cependant que l'aspect performance de DOSIS nécessite une étude plus approfondie.

- un effet secondaire de DOSIS, mais néanmoins intéressant, est la possibilité de réaliser une communication interne au moyen des services OSI. En réalité, dans ce cas, les protocoles OSI se déroulent seulement entre les deux agents serveurs représentant des programmes communicants sur le site serveur. Ceci peut être cependant utilisé pour réaliser des tests des programmes utilisant les services OSI.

En face des avantages présentés de l'approche DOSIS, les inconvénients suivants peuvent être évoqués :

La critique qui peut être adressée à l'égard du serveur OSI porte sur le fait que l'interface de communication externe (interface OSI) est différente de l'interface de communication interne, alors que pour la technique de conversion l'utilisateur a une seule et unique interface pour les deux communications.

A notre avis, çeci ne constitue pas un problème majeur car le service de communication externe est un service indépendant et visible à l'utilisateur, alors que le service de communication interne est un service de base dans un SD-RL intégré. Il n'a pas à être souvent directement accédé par l'utilisateur concepteur d'application. Ce sont deux services de nature et d'utilisation différentes. D'autre part, grâce à l'entité agent client, il est possible de

présenter une interface différente de l'OSI et en particulier une interface semblable à celle offerte par le système de communication interne. C'est à la charge de l'agent client alors de faire la correspondance entre l'interface présentée et celle de l'OSI. C'est une sorte de conversion d'interface réalisée dans le site client et non pas dans la passerelle comme dans la technique de conversion.

On peut noter aussi que DOSIS implique l'implémentation d'un logiciel (l'agent client) sur chaque site client, ce qui n'est pas le cas pour la technique de conversion. Ceci est vrai, mais il convient de noter que les ressources nécessaires pour l'agent client sont beaucoup moins importantes que celles nécessaires à l'implémentation des protocoles OSI.

6. Réalisations

Afin de valider l'approche DOSIS, nous avons été amenés à réaliser deux maquettes de DOSIS. La première est orientée vers la version centralisée, et avait un caractère expérimental. La deuxième est réalisée de façon à permettre une extension vers la version distribuée, et elle est plus orientée vers une exploitation dans un environnement réel.

L'environnement dans lequel se placent ces deux réalisations est constitué d'un réseau local de type Ethernet. Ce réseau connecte plusieurs systèmes dont la plupart sont de type station de travail. La quasi-totalité de ces systèmes supporte UNIX™(essentiellement les versions : System V de ATT et 4.2 BSD ou 4.3 BSD). L'architecture de communication qui permet l'échange entre les stations du réseau local est base sur les protocoles TCP/IP et UDP/IP.

Dans cet environnement, DOSIS permet à des stations, dont la majorité ne supporte pas les protocoles OSI, de communiquer avec des systèmes distants (sur d'autres réseaux) au moyen de services de communication OSI.

La réalisation d'un DOSIS complet tel qu'il a été présenté implique la disponibilité de l'ensemble de l'architecture OSI sur une des stations connectées au réseau local. Ceci n'est pas le cas dans notre environnement. C'est pourquoi nous avons limité notre réalisation à un DOSIS partiel, qui offre seulement les services de niveau transport et les services de niveau session.

Dans la suite de cette partie, nous décrivons les points essentiels de l'implémentation réalisée, les outils logiciels que nous avons utilisés et les problèmes posés lors de la mise en œuvre de l'architecture spécifiée dans un environnement réel.

6.1. Le serveur transport-ROSE

La première implémentation réalisée avait pour but de mettre en œuvre un serveur transport. Ce serveur offre à tous les utilisateurs, quelle que soit leur localisation sur le réseau local, l'accès au service transport. Le site serveur est connecté au monde extérieur par le réseau public X25. Nous avons donc, d'une part, les sites clients qui supportent chacun les protocoles de communication TCP/IP et UDP/IP, et d'autre part le site serveur qui supporte les protocoles X25 et le protocole transport ISO mais également les protocoles TCP/IP, UDP/IP. La communication interne sera donc basée sur le protocole TCP/IP qui est en mode connexion.

Le site serveur est constitué d'une machine Mini 6 qui supporte une version hybride de UNIX V7 et UNIX 4.2 BSD. L'indisponibilité du réseau Ethernet à l'époque du développement de cette maquette nous a obligé à simuler le fonctionnement de ce réseau par une communication interne au Mini 6. Cette communication utilisait cependant les services TCP/IP. Cette simulation modifie très peu les paramètres du problème. Dans la suite nous faisons donc abstraction de cette simulation.

6.1.1. Les interfaces transport-ROSE et TCP/IP

Avant de décrire l'implémentation et de dégager les problèmes, nous résumons ci-dessous, les deux services de communication qui concernent l'implémentation du serveur : le transport-ROSE pour la communication externe et TCP/IP pour la communication interne.

Le protocole transport utilisé est celui qui est développé dans le cadre du projet ROSE (cf. chapitre 2). Une des caractéristiques essentielles du transport ROSE est son intégration dans le système UNIX. En effet, le protocole est implémenté au niveau du noyau et il offre une interface de type "driver" UNIX. Cette interface utilisateur est constituée de cinq appels système :

```
open(name, mode)
  /*
     cet appel permet de s'allouer un point d'accès au niveau transport, au travers lequel
     l'utilisateur peut appeler les services du transport
  */

close(fildes)
  /*
     fermeture d'un point d'accès au service transport
  */

ioctl(fildes, function, arg)
  /*
     cet appel permet de réaliser plusieurs fonctions transport. Il permet notamment de
     demander l'ouverture ou la fermeture des connexions transport.
  */

read(fildes, buffer ,nbytes)
  /*
     Cet appel permet la réception des données sur une connexion transport
  */

write(fildes, buffer, nbytes)
  /*
     Cet appel permet l'envoi des données sur une connexion transport
  */
```

Le protocole TCP/IP disponible sur les machines UNIX sera utilisé pour la communication entre l'agent client d'une part et le contrôleur ou l'agent serveur d'autre part. Ce protocole propose un service de niveau transport en mode connexion, il est donc approprié pour supporter le protocole client-serveur de DOSIS. L'interface de ce protocole sous UNIX est l'interface appelée "socket" (le socket est une sorte de point d'accès aux services d'un protocole comme TCP par exemple). Les primitives essentielles suivantes sont utilisées (ce sont des appels système au sens UNIX) :

```
socket(domain, type, protocol)
  /*
     Permet de créer un socket pour accéder au service d'un protocole de communication.
     Il retourne l'identificateur de ce socket
  */

connect(s, addr, addrlen)
  /*
     Permet d'établir une connexion (connexion TCP par exemple) entre un socket local et
     un socket distant.
  */

listen(s, ql)
  /*
  Permet à un serveur de se bloquer dans l'attente de l'arrivée d'une demande d'ouverture
  de connexion.
  */

accept(s, addr, addrlen)
  /*
     Permet d'accepter l'ouverture d'une connexion. Cet appel rend un identificateur d'un
     nouveau socket qui est désormais lié à un socket distant par une connexion.
  */

read(s,buf,nbytes)
write(s,buf,nbytes)
  /*
     Ces appels permettent de recevoir et d'envoyer des données sur une connexion.
  */
```

6.1.2. Le principe de l'implémentation et les choix retenus

Nous avons choisi dans cette maquette de sacrifier la portabilité des applications existantes qui utilisent le transport au profit d'une réalisation simple. Ceci signifie que l'interface utilisateur accessible à travers le serveur de transport sera légèrement différente de l'interface originale du transport. Cependant, ces modifications de l'interface ne nécessitent pas une réécriture

complète des applications. Les modifications à apporter concerne les aspects suivants :

- afin de simplifier la réalisation et de pouvoir valider les idées essentielles du serveur, nous avons choisi de ne pas traiter l'activation et la désactivation des agents serveurs de manière implicite. Les appels `BIND` et `UNBIND` sont visibles au niveau de l'utilisateur du transport. Ainsi l'utilisateur doit faire précéder les appels aux services transport par la primitive `BIND` pour ouvrir une session de service transport. Il doit également terminer la session de service par un appel à la primitive `UNBIND`.

- dans l'implémentation ROSE, les primitives de l'interface sont des appels système au noyau UNIX. Il n'est donc pas possible d'intercepter ces appels par l'agent client sans modifier la bibliothèque des appels système UNIX. Au lieu d'introduire ces modifications dans le noyau, nous avons préféré de modifier la syntaxe de ces appels. Cette modification purement syntaxique consiste à faire précéder l'identificateur de la primitive transport par "s_". Ainsi les nouvelles primitives du transport vues par les clients du serveur transport sont les suivantes :

```
s_open(name,mode)
s_close(fildes)
s_ioctl(fildes,function,arg)
s_read(fildes,buffer,nbytes)
s_write(fildes,buffer,nbytes)
```

Les appels cités ci-dessus en plus des deux appels d'ouverture et fermeture d'une session de service transport constituent les points d'entrée du module agent client.

L'implémentation des trois éléments de l'architecture du serveur qui sont l'agent client, l'agent serveur et le contrôleur est réalisée de la manière suivante :

L'agent client est implémenté sous forme d'un module de la bibliothèque du système sur chaque site client. Ce module doit être lié au programme application par l'utilisateur et forme ainsi un processus UNIX. Le contrôleur

est un processus UNIX qui est lancé à l'initialisation du système et qui s'exécute en mode "background". L'agent serveur transport est implémenté sous forme d'un fichier exécutable à partir duquel le processus agent serveur sera activé au moyen des appels système FORK et EXEC exécutés par le contrôleur.

La coopération entre les différentes entités est résumée de la manière suivante :

Le contrôleur est normalement à l'écoute, au moyen de l'appel LISTEN, sur son "socket" (connu par l'ensemble des agents clients), des demandes de connexion venant des agents clients. A la réception d'une demande de connexion, le processus contrôleur accepte l'ouverture de connexion TCP au moyen de l'appel ACCEPT en récupérant un nouveau socket et se duplique au moyen de l'appel système FORK. Le processus père continue à attendre de nouvelles demandes de connexion sur son socket. Le processus fils demande la réception des données sur la connexion TCP (à travers le nouveau socket) qui le relie désormais à l'agent client qui a initialisé la communication. Cette demande de réception est réalisée grâce à l'appel READ.

La réception d'un PDU du protocole client-serveur de type ouverture d'une session de service par le contrôleur fils provoque l'exécution de l'appel système EXEC par ce dernier. Cet appel porte sur le fichier agent serveur transport, ce qui a pour effet de remplacer le code du processus contrôleur fils par le code du processus agent serveur. A partir de ce moment, l'agent client et l'agent serveur s'échangent des messages TCP contenant les requêtes-réponses des services transport (les PDU du protocole client- serveur).

A la détection d'un des événements : connexion TCP rompue ou réception d'une demande de fermeture de session de service, l'agent serveur-transport se termine.

Ce schéma de fonctionnement est inspiré du schéma général du fonctionnement des serveurs UNIX. Il est donc adapté à notre environnement d'implémentation. La création dynamique des processus agents serveurs est facilitée par les mécanismes FORK et EXEC, alors que la création de multiples agents serveurs est encouragée par le fait que les processus UNIX identiques peuvent partager le même code.

Cette implémentation expérimentale a été testée au moyen d'un programme de transfert de fichiers utilisant les services du transport-ROSE. Le renseignement le plus important tiré de cette expérience est la simplicité de la mise en œuvre de la solution DOSIS. Ceci apparaît clairement en comparant cette solution à la solution de conversion entre le protocole transport TCP et le protocole transport OSI présentée dans le premier chapitre [Green 86]. L'objectif des deux solutions est le même : il s'agit de communiquer avec un système OSI au niveau transport à partir d'un système connecté à un réseau local supportant le protocole TCP. La solution de conversion est plus complexe à concevoir et à réaliser. Elle impose des restrictions sur l'utilisation de certains services de l'un ou l'autre des protocoles, alors que la simplicité de notre solution vient du fait qu'il n'existe pas de rapport entre les services des deux protocoles.

6.2. Le serveur distribué session-transport

Nos objectifs pour cette seconde maquette consistent à améliorer les moyens de la mise en œuvre de DOSIS en s'appuyant sur les renseignements tirés de la première maquette et de valider les idées concernant l'approche distribuée. Il est à noter que la communication entre le client et le serveur n'est plus simulée, elle passe réellement par le réseau local Ethernet.

Nous disposons pour cette maquette d'une machine Sps7 sous SPIX 22 reliée au réseau local. Elle contient, d'une part, les protocoles X25 et le protocole transport OSI et, d'autre part, les protocoles DARPA (en particulier le protocole TCP). Cette machine peut donc jouer le rôle de site passerelle. Il est à noter que le protocole transport OSI supporté est propre au Sps7, les appels aux primitives transport respectent une interface générale appelée "Méthode d'Accès Distribuée" (MAD). Le site client correspond à la machine Vax sous Unix 4.3 BSD, d'autres machines peuvent être aussi des sites clients. Nous disposons en plus du logiciel Session OSI développé dans le cadre du projet ROSE et pouvant interfacer le Transport-MAD du Sps 7.

Nous visons dans cette maquette l'expérimentation de la configuration DOSIS suivante (Fig 3.9) :

- un serveur transport accessible à partir des sites clients.

- l'utilisation de la session comme une application (client) du serveur transport. Ceci permet de tester l'aspect distribution des couches proposé dans DOSIS. La session sera donc sur un site différent de celui de la passerelle.

- un serveur session accessible à partir des sites clients. Ce serveur peut aussi, à son tour, utiliser le serveur transport (c'est une configuration de serveur distribué où le site qui contient la session joue le rôle de site intermédiaire).

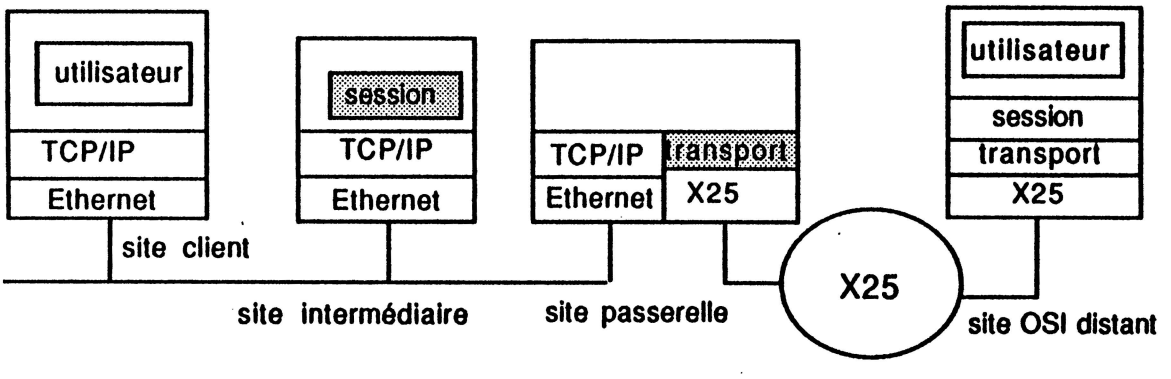


Fig 3.9 Configuration de la maquette de DOSIS distribué

Ceci implique la réalisation :

- d'un agent serveur transport sur le site passerelle ;
- d'un agent client transport sur les sites clients et notamment sur le site de la couche session ;
- d'un agent client session sur les sites clients ;
- d'un agent serveur session sur le site de la couche session.

6.2.1. Les choix retenus pour l'implémentation du serveur session-transport

Dans cette implémentation, nous avons fait les deux choix suivants :

- l'activation et la désactivation des agents-serveurs ne sont plus visibles à l'utilisateur mais gérées automatiquement par l'agent client selon la stratégie que nous avons définie dans 4.1.2. Ceci évite la modification de l'interface originale du transport et de la session et permet par

conséquent la portabilité des applications existantes.

- l'implémentation de la première maquette nous a montré qu'une partie du code écrit était consacrée au codage et décodage des primitives OSI en messages envoyés par TCP, et à la gestion du mécanisme requête-réponse synchrone utilisé dans notre protocole client-serveur. Cette partie du code peut être économisé en utilisant un protocole d'appel de procédure à distance existant et approprié pour réaliser notre protocole client-serveur, sans modifier l'architecture que nous avons définie.

Dans la suite, nous nous intéressons seulement à l'aspect concernant l'utilisation du logiciel Courier [Cooper 83] d'appel de procédure à distance (RPC) pour implémenter l'architecture du serveur OSI.

6.2.2. Le logiciel Courier

Courier est un logiciel d'appel de procédure à distance basé sur le modèle de Nelson [Nelson 81]. Il s'exécute dans l'environnement UNIX et utilise le protocole TCP pour la communication.

Une application utilisant Courier est décomposée en deux modules : le client et le serveur qui peuvent être situés sur des machines différentes. Le client peut appeler des procédures définies dans le serveur. Le concepteur de l'application n'a pas à se soucier des aspects de communication et de synchronisation. Un appel du client a la même syntaxe que l'appel de procédure local. La transparence de la distribution entre le client et le serveur est assurée grâce à deux éléments du logiciel Courier :

- le protocole d'appel de procédure à distance,
- le langage Courier pour la description des interfaces et le compilateur Courier pour la génération automatique des "stubs" (voir ci-dessous).

Le protocole d'appel de procédure à distance est réalisé au-dessus du protocole TCP. Il a pour fonctions de coder et décoder les paramètres des procédures selon un format de présentation indépendant de la machine et d'assurer la synchronisation nécessaire entre le client et le serveur.

La transparence de la distribution entre le client et le serveur est assurée grâce

à deux modules appelés "stub client" et "stub serveur". Ces deux modules sont générés automatiquement de la façon suivante :

Le concepteur de l'application décrit avec le langage Courier l'interface exportée par le serveur, c'est-à-dire les procédures qui peuvent être appelées par le client ainsi que leurs paramètres. Le compilateur Courier génère à partir de cette description les deux modules stub client et stub serveur. Le module stub client doit être lié au client pour constituer le processus client et le module stub serveur doit être lié au serveur pour constituer le processus serveur. Le stub client contient l'interface des procédures exportées par le serveur, le corps de ces procédures étant composé des instructions de codage et de décodage ainsi que l'appel au protocole d'appel de procédure à distance pour demander l'exécution de ces procédures et la réception des paramètres de retour. Ainsi le stub client représente le serveur sur le site client. Le stub serveur effectue des appels au protocole d'appel de procédure à distance permettant ainsi de recevoir les appels du client, de les décoder, d'effectuer ces appels dans l'environnement du serveur, d'intercepter les paramètres de retour pour les envoyer vers le client. Ceci est également transparent au serveur.

L'activation du processus serveur est dynamique. Mais ceci implique l'appel par le client de la procédure :

```
Bind(nom du processus serveur, nom de la machine);
```

Cet appel nécessaire de la part du client montre que la transparence de la distribution n'est pas complète. L'activation est réalisée par un processus "daemon"(au sens UNIX) qui est toujours à l'écoute du réseau du côté serveur et qui crée le processus serveur à la réception de l'appel `Bind`.

6.2.3. Réalisation à l'aide de Courier du serveur session-transport

Le logiciel Courier permet l'exécution d'appels de procédure à distance. Il n'est pas pour autant utilisable directement entre le client et la couche OSI (transport ou session dans ce cas), parce qu'il n'est pas capable de traiter les paramètres de type référence. Cependant, l'architecture que nous avons définie permet d'exploiter ce logiciel de manière intéressante. Il sera utilisé comme moyen de communication (protocole client-serveur) entre l'agent

client et l'agent serveur. Nous examinons ci-dessous brièvement l'utilisation de Courier pour le développement du serveur.

Les entités de DOSIS sont construites à l'aide des entités Courier de la manière suivante :

- l'agent client et l'agent serveur de DOSIS sont respectivement le client et le serveur selon le modèle Courier.

- le processus daemon de Courier implémente l'entité contrôleur.

- la fonction `Bind` de Courier est appelée par l'agent client pour réaliser la primitive `Bind` que nous avons définie.

Il suffit maintenant pour chaque primitive OSI de définir une procédure dans l'agent serveur. Cette procédure sera exportée au sens Courier, c'est-à-dire appelée à distance par l'agent client. Elle a les mêmes paramètres que la primitive OSI correspondante mais en remplaçant les paramètres de type référence par les valeurs référencées. L'ensemble des procédures ainsi définies constitue l'interface exportée de l'agent serveur. A partir de la spécification de cette interface par le langage Courier, le compilateur Courier génère le stub agent client et le stub agent serveur qui sont respectivement liés à l'agent client et l'agent serveur qui seront chargés de la réalisation du protocole d'appel de procédure à distance proprement dit. Ainsi l'appel à une primitive OSI par l'utilisateur est intercepté par l'agent client qui effectue la transformation nécessaire des paramètres et appelle à distance la procédure correspondante dans l'agent serveur. L'exécution de cette procédure par l'agent serveur consiste à reconstruire la primitive OSI (transformation inverse des paramètres) et à appeler la primitive en question (appel local).

L'utilisation de Courier a permis de définir un processus de développement systématique des entités de DOSIS ainsi que la génération automatique d'une partie du code.

Nous avons réalisé selon la méthode de développement décrite ci-dessus un serveur de transport (la couche transport réside sur le Sps 7) qui est actuellement opérationnel et un serveur session (la couche session réside sur le Vax) qui est en cours de réalisation. Le serveur transport permet à des

programmes communicants qui s'exécutent sur le Vax d'utiliser les services transport OSI. Le logiciel Courier a simplifié cette implémentation, les problèmes techniques rencontrés étaient surtout liés à la contrainte que nous avons fixée et qui consistait à garder la même interface originale du transport lors de l'accès à partir des sites clients. Ceci était dû à l'interface particulière du transport (l'interface MAD). Ce serveur transport a été validé par l'utilisation de la couche session comme client à partir du Vax, ce qui avait également pour effet de tester l'aspect concernant la distribution des couches.

Conclusion

Nous nous sommes intéressés dans cette thèse au problème de l'ouverture et la coopération entre les systèmes distribués sur réseaux locaux. Il n'était ni réaliste, ni souhaitable, de réaliser cette coopération sous forme d'intégration pour former un "super système distribué". Nous nous sommes alors orientés vers une approche qui met en œuvre cette coopération sous forme de communication explicite et visible au concepteur d'application.

L'examen des caractéristiques de cette communication externe a permis de constater que les services de communication OSI sont fonctionnellement adaptés aux besoins de la communication entre les SD-RL. Considérer les services OSI comme un moyen de communication entre les SD-RL avait un double intérêt : d'une part, ces services constituent une norme internationale, ce qui est un facteur important pour assurer l'ouverture des SD-RL. D'autre part, ces services ont été définis pour fonctionner dans des environnements hétérogènes, ce qui est le cas car chaque système distribué est géré par un système d'exploitation à priori différent des autres.

Réaliser les services OSI implique la mise en œuvre des protocoles OSI. La question qui s'est alors posée consistait à savoir si les protocoles OSI sont utilisés à l'intérieur des systèmes distribués. L'observation de la réalité a donné une réponse, en général, négative. Nous avons vu que ceci était dû à des problèmes de disponibilité sur le plan de la spécification ou sur le plan de l'implémentation des protocoles OSI. Les aspects liés aux performences et aux ressources sont aussi un facteur important de cette constatation. Enfin, les services OSI (même au niveau application), qui sont plus des fonctions de communication que des fonctions de type système d'exploitation, ne sont pas adaptés aux environnements intégrés.

A partir de ce constat, il nous a fallu trouver une solution assurant une communication entre des systèmes distribués en tenant compte du fait que leurs systèmes de communication respectifs sont différents de l'OSI et, par conséquent, hétérogènes.

Le problème a été alors ramené à un problème d'interconnexion de systèmes de communication hétérogènes au moyen des services OSI. Nous avons donc étudié les techniques d'interconnexion de réseaux. Nous avons vu que les techniques basées sur le principe d'encapsulation sont adaptées à des réseaux partiellement hétérogènes et que les techniques basées sur la conversion peuvent permettre l'interconnexion de systèmes de communication totalement hétérogènes. Mais ces techniques de conversion sont difficiles à réaliser, non systématiques, et parfois impossibles à mettre en œuvre lorsque le sous-ensemble commun des fonctions des protocoles à convertir est vide.

L'approche que nous avons proposée pour assurer l'ouverture des systèmes distribués est basée sur un serveur OSI distribué (DOSIS). Ce serveur permet l'utilisation d'un système de communication spécifique à l'intérieur du système distribué, tout en donnant à l'extérieur une vision de ce système qui est celle d'un système OSI unique. Ceci représente à notre avis, sur le plan intellectuel et sur le plan pratique, la base d'un compromis dans le débat entre partisans et adversaires du modèle OSI.

L'architecture et le fonctionnement de DOSIS ont été définis pour tirer avantage de l'environnement distribué des SD-RL. Ceci a été réalisé par l'accès et le partage d'une instance unique des couches OSI dans le système distribué, et par la possibilité de distribuer les couches OSI sur plusieurs sites du système distribué. Ceci offre, à chaque utilisateur dans le SD-RL, la vision de supporter l'ensemble des couches OSI sur son site local.

L'intérêt fondamental de l'approche DOSIS réside dans la séparation entre les protocoles de communication interne et les protocoles de communication externe, ce qui est un avantage important par rapport à la technique de conversion. Ainsi, l'approche DOSIS est applicable quel que soit le niveau de communication externe sans imposer des contraintes sur le service de communication interne.

Nous pensons que l'idée de considérer les services et les protocoles OSI comme un moyen pour la communication entre les systèmes distribués continuera à présenter beaucoup d'intérêt avec le développement des systèmes distribués.

DOSIS constitue, par ailleurs, une solution intéressante pour l'interconnexion de réseaux hétérogènes dans des cas où les techniques habituelles d'encapsulation et de conversion se révèlent inapplicables ou difficiles à mettre en œuvre.

La mise en œuvre de l'approche DOSIS a été réalisée à travers deux maquettes. Dans la première, à caractère expérimental, nous avons développé un serveur de niveau transport OSI en nous appuyant sur un système de communication basé sur l'architecture DARPA et en particulier sur le protocole TCP. Les renseignements tirés de cette maquette ont été exploités pour le développement d'une deuxième maquette dans un environnement réel de réseau local Ethernet et des protocoles DARPA. Dans cette maquette nous avons validé nos idées sur l'aspect lié à la distribution des couches à travers la réalisation d'un serveur transport accessible par le niveau session. Ces services, actuellement opérationnels, vont être complétés par un serveur de niveau session qui est en cours de réalisation. La réalisation de cette maquette a donnée lieu à l'utilisation d'un logiciel d'appel de procédure à distance (Courier) qui s'est révélé parfaitement adapté à une mise en œuvre simple et systématique de DOSIS.

Nous pensons que la suite du travail que nous avons commencé doit traiter deux aspects :

Le premier aspect doit concerner la réalisation d'un serveur pour l'accès à d'autres niveaux que ceux du transport et de la session et, en particulier, le niveau d'application, car c'est à ce niveau là que les limites de la technique de conversion apparaissent le plus clairement. C'est à ce niveau aussi que certains points d'interrogation concernant DOSIS peuvent être évoqués. En effet, les structures des données qui sont échangées au niveau interface deviennent complexes et volumineuses (des fichiers dans le cas de FTAM par exemple). Il est donc important de vérifier que ceci n'amènera pas plus de complexité dans la mise en œuvre du protocole client-serveur entre l'agent client et l'agent serveur.

Nous pensons que plusieurs applications OSI peuvent être concernées par l'approche DOSIS. Les plus importantes sont l'accès et le transfert de fichiers FTAM, la messagerie X400 et l'accès aux bases de données distantes RDA.

Le deuxième aspect concerne la spécification formelle et l'évaluation des performances. En effet, une spécification formelle de l'architecture de DOSIS et du protocole client-serveur permettra ensuite d'évaluer les performances de la communication OSI globale. Cette évaluation est nécessaire pour pouvoir comparer notre approche avec l'approche de conversion. Elle permettra également de savoir si la solution DOSIS, qui est basée sur l'accès à distance aux services OSI, peut se révéler plus performante que celle qui consiste à supporter localement, sur chaque station, les protocoles OSI.

Annexe

Les modèles d'interconnexion : ISO, DARPA, et CCITT

1. Introduction

Nous avons examiné dans le chapitre 2 les deux principales approches pour l'interconnexion de réseaux. L'objet de cette annexe est d'étudier le modèle d'interconnexion dans trois différentes architectures de communication, c'est-à-dire voir comment les approches étudiées dans le deuxième chapitre sont appliquées et intégrées dans les architectures de communication. La première étude concerne l'architecture OSI de l'ISO qui représente les travaux de normalisation dans le domaine d'interconnexion. Nous étudions ensuite le modèle d'interconnexion utilisé dans la famille des protocoles DARPA. L'exemple OSI permet de présenter les solutions proposées par les instances de normalisation concernant le problème d'interconnexion, alors que le choix des protocoles DARPA est surtout dicté par la large diffusion de ces protocoles, que ce soit aux Etats-unis ou en Europe. Ces protocoles représentent un standard "de fait" surtout depuis leur intégration dans la plupart des systèmes UNIX. Le dernier modèle d'interconnexion est celui utilisé par le CCITT pour interconnecter des réseaux publics X25.

Les trois cas qui seront étudiés concernent l'interconnexion des réseaux partiellement hétérogènes. Le modèle d'interconnexion, dans chacun de ces cas, est intégré dans l'architecture de communication concernée.

2. Le modèle d'interconnexion OSI

Le modèle OSI traite le problème d'interconnexion des réseaux hétérogènes ou homogènes (appelés sous-réseaux dans ce contexte) à travers des passerelles appelées *InterWorking Unit (IWU)*. Le niveau d'interconnexion considéré est le niveau réseau. Ceci signifie que :

- le modèle d'interconnexion suppose que l'hétérogénéité qui peut exister entre les différents sous-réseaux à interconnecter ne dépasse pas le niveau réseau. Les niveaux supérieurs (à partir du transport) sont identiques sur l'ensemble des sous-réseaux et compatibles avec les spécifications de l'ISO. Ces sous-réseaux peuvent être, soit des réseaux à grande distance, soit des réseaux locaux.

- le modèle OSI ne traite pas des solutions d'interconnexion au niveau inférieur au réseau, comme par exemple l'interconnexion de réseaux locaux hétérogènes au niveau MAC au moyen d'une passerelle de type pont. Ainsi un ensemble de réseaux interconnectés par des ponts est considéré par l'OSI comme étant un seul sous-réseau.

Les fonctions d'interconnexion étant placées au niveau réseau, nous nous proposons d'étudier la structure de la couche réseau de l'OSI et de montrer comment elle réalise l'interconnexion des réseaux hétérogènes ou homogènes.

Le modèle OSI définit deux types de services au niveau réseau, le premier est un service de communication en mode connexion [ISO/DIS 8348], le deuxième est un service en mode sans connexion [ISO/DIS 8348/DAD1]. L'idée est de prendre en compte les caractéristiques des différents types de réseau. Cependant, l'OSI n'impose pas l'utilisation de l'un ou l'autre service dans un environnement donné. En simplifiant (l'étude du problème sort du cadre de l'interconnexion), on peut dire que pour des raisons de performance, la tendance est à l'utilisation des services sans connexion dans les réseaux locaux. Il est très important de noter que dans un environnement d'interconnexion, un des deux services doit être choisi pour représenter le service réseau sur l'ensemble des sous-réseaux interconnectés.

Il s'agit donc d'interconnecter des sous-réseaux au moyen de passerelles, et de construire ainsi un réseau global fournissant aux couches transport un service

réseau OSI. Trois configurations peuvent se présenter :

a) certains (ou la totalité) des sous-réseaux ne supportent aucun des services réseau OSI définis ci-dessus.

b) certains des sous-réseaux supportent le service en mode connexion alors que d'autres supportent le service en mode sans connexion.

c) la totalité des sous-réseaux supportent le même service réseau OSI, que ce soit en mode connexion ou en mode sans connexion.

2.1. Structure de la couche réseau

Afin de prendre en compte les trois cas décrits ci-dessus, la couche réseau a été structurée en trois sous-couches contenant chacune un groupe de fonctions [ISO/DIS 8648] (Fig A.1) :

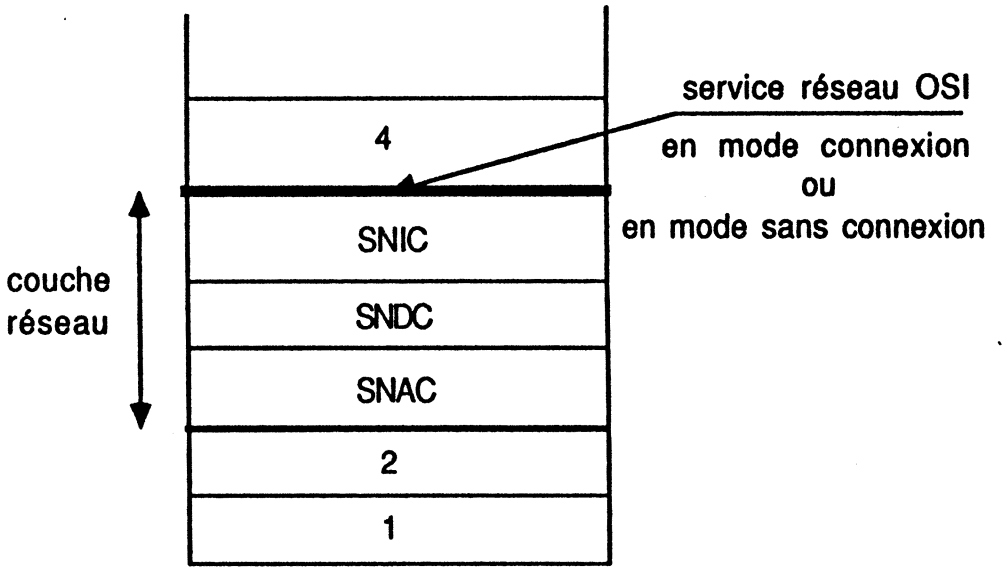


Fig A.1 Strucure de la couche réseau dans le modèle OSI

1) SubNetwork ACcess (SNAC) :

Ce groupe de fonctions constitue la partie inférieure de la couche réseau. Il offre le service d'accès à un sous-réseau particulier permettant la communication entre les stations connectées à ce sous-réseau au moyen du protocole SNAP (Subnetwork Access Protocol). Le modèle OSI ne fait aucune hypothèse sur les services fournis par cette sous-couche. Un cas particulier se présente lorsque les services SNAC correspondent aux services réseau OSI. La couche réseau contiendra alors uniquement le SNAC, les deux autres sous-couches seront vides. Un exemple de protocole SNAP est le protocole X25 niveau paquet. Pour un sous-réseau de type réseau local, cette sous-couche est assimilée à la couche inférieure, car les fonctions d'accès et de communication à travers un réseau local sont assurées par la couche liaison de données.

2) SubNetwork Dependent Convergence (SNDC)

Le groupe de fonctions SNDC constitue la sous-couche intermédiaire de la couche réseau. Son rôle est de compléter les services d'accès aux sous-réseaux offerts par le SNAP pour permettre de supporter le protocole réseau OSI. La réalisation de cette fonction dépend du service d'accès à chaque sous-réseau. Par conséquent, la portée du protocole SNDCP qui réalise les services SNDC est limitée à un sous-réseau. Il existe donc autant de protocoles SNDCP que de protocoles SNACP différents. Les fonctions de la sous-couche SNDC s'annulent sur un sous-réseau qui offre déjà le service réseau OSI. Lorsque les services offerts par la sous-couche SNAC sont suffisants pour supporter les services réseau OSI, la sous-couche SNDC est alors assimilée à la sous-couche SNAC (une correspondance directe existe entre les services de l'une et l'autre). C'est le cas quand la sous-couche SNDC est implémentée au dessus de la couche LLC classe 1 ou classe 2 (mode sans connexion) dans un réseau local pour offrir le service attendu par le protocole réseau en mode sans connexion [ISO/DIS 8473/DAD1].

3) SubNetwork Independant Convergence (SNIC)

SNIC est la sous-couche supérieure de la couche réseau. Sa fonction est de fournir le service réseau OSI complet à travers le réseau global qui est constitué de l'ensemble de sous-réseaux interconnectés. La réalisation de cette

fonction est indépendante des sous-réseaux sous-jacents. Ceci est rendu possible grâce à la sous-couche SNDC. Le service offert par SNIC peut-être soit le service en mode connexion, soit en mode sans connexion. Le protocole au niveau de cette sous-couche est commun sur le réseau global. Cette sous-couche peut être absente (plus exactement assimilée à la couche inférieure) dans deux cas :

a) la sous-couche SNAC offre le service réseau OSI qu'on souhaite avoir sur le réseau global.

b) le service de la sous-couche SNAC peut être modifié par une sous-couche SNDC pour atteindre directement le service réseau OSI du réseau global. Le document [ISO/DP 8878] décrit un protocole SNDCP à implémenter au dessus du protocole X25 niveau paquet version 1980, qui n'offre pas le service réseau mode connexion, pour obtenir le protocole X25 niveau paquet version 1984 qui est capable d'offrir le service réseau en mode connexion.

En ce qui concerne les passerelles, elles sont vues dans le modèle d'interconnexion OSI comme des commutateurs inter-réseaux. Ils assurent deux fonctions essentielles : le Relais et le Routage*"R+R"*. La structure de la passerelle (Fig A.2) est constituée des couches réseaux et les couches inférieures correspondant à chaque sous-réseau en plus du module R+R réalisant les fonctions de relais et de routage. Généralement, les sous-couches SNDC et SNAC sont différentes d'un sous-réseau à un autre. Cependant une interface commune est offerte à la sous-couche SNIC qui est identique sur le réseau global et qui fournit le service réseau OSI en mode connexion ou sans connexion. La présence des trois sous-couches n'est pas toujours nécessaire comme nous l'avons vu précédement. Les fonctions du module R+R et l'interface de ce module avec les sous-couches SNIC seront examinées plus loin.

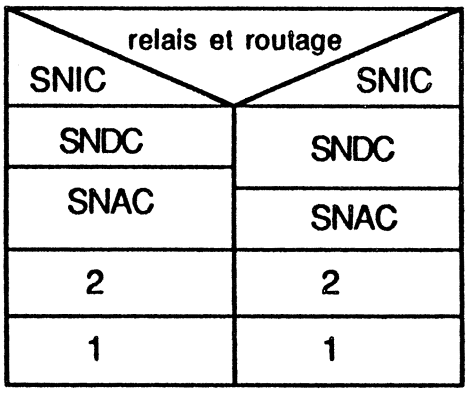


Fig A.2 Structure générale d'une passerelle OSI

2.2. Les approches OSI pour l'interconnexion

La structure de la couche réseau décrite précédemment a permis à l'OSI d'adopter deux approches différentes pour l'interconnexion. Le choix entre les deux approches se fait en fonction du niveau de service offert par les sous-réseaux à interconnecter. Les deux approches correspondent en réalité aux deux techniques d'interconnexion que nous avons présentées : l'encapsulation et le relais (cas particulier de la conversion).

2.2.1. L'approche "relais"

Le principe de relais représente un cas particulier de la conversion où les protocoles à interconnecter sont quasiment identiques. Il existe donc une correspondance un-pour-un entre les services de chaque protocole. La fonction du module de conversion se réduit à relayer les données arrivant d'un côté de la passerelle sur l'autre côté. Cette approche suppose que chacun des sous-réseaux fournit le service réseau OSI. Si ce n'est pas le cas, il faut d'abord réaliser une harmonisation et implémenter sur les sous-réseaux qui ne supportent pas les services OSI les sous-couches appropriées. Ceci afin de transformer le service de communication sur ces sous-réseaux en un service OSI. Le service OSI désigne ici un service spécifique pour la communication à travers le réseau global, soit en mode connexion, soit en mode sans connexion.

Un des deux services seulement est considéré, dans une configuration d'interconnexion donnée, comme étant le représentant du service réseau OSI.

Selon le service offert sur un sous-réseau quelconque, une des trois actions suivantes doit être entreprise :

a) le service SNAC est équivalent au service réseau OSI, nous avons donc : SNAC = service réseau OSI = SNIC ; aucune harmonisation n'est à faire.

b) le service SNAC ne correspond pas exactement au service OSI mais il y est suffisamment proche, une harmonisation par addition des fonctions SNDC au dessus de SNAC permet d'obtenir le service OSI. Nous obtenons donc :
(SNAC + SNDC) = service réseau OSI = SNIC.

c) le service SNAC est très différent du service OSI, deux étapes d'harmonisation sont alors nécessaires. Nous obtenons donc :
(SNAC + SNDC + SNIC) = service réseau OSI.

Pour simplifier la présentation de l'approche relais, nous faisons abstraction de cette harmonisation en considérant l'existence sur chaque sous-réseau d'une couche SNIC offrant le service réseau OSI.

La figure (Fig A.3) montre une passerelle de type relais qui interconnecte deux réseaux R_a et R_b. Le module R+R doit relayer les PDU entre les deux sous-couches $SNIC_a$ et $SNIC_b$ (ce module correspond au module adaptateur selon l'architecture générique des passerelles présentée dans le chapitre 2). Il permet ainsi la concaténation des deux communications de type intra-réseau pour former une communication inter-réseaux. Cette communication apparaît aux utilisateurs des niveaux réseau dans les stations hôtes comme étant de bout en bout.

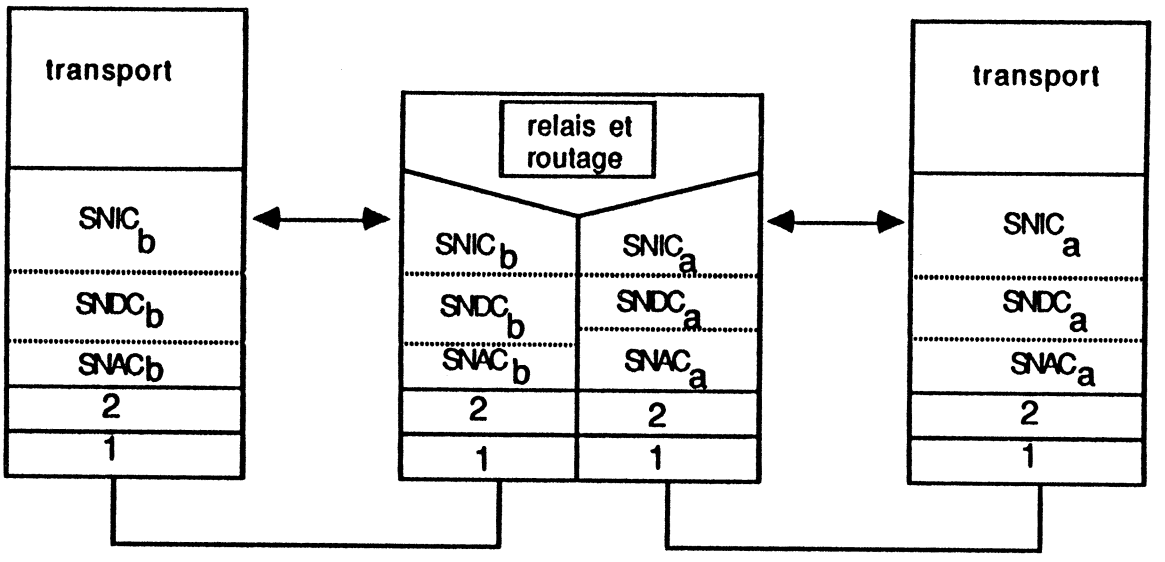


Fig A.3 Interconnexion OSI basée sur l'approche "Relais"

Les principales actions effectuées par le module R+R sont les suivantes :

- en mode connexion, le module R+R doit gérer les cascades d'ouvertures et de fermetures de communication inter-réseaux.

- il réalise éventuellement certaines adaptations des paramètres de communication intra-réseau :

  - adaptation de la longueur des données en effectuant la segmentation et le réassemblage nécessaires.

  - adaptation des tailles des fenêtres d'émission et de réception qui constituent les paramètres de contrôle de flux.

- il détermine la route en fonction de l'adresse destinataire globale et il adapte éventuellement des structures d'adresses pour le prochain sous-réseau traversé par les données.

- lorsque la fonction relais agit au niveau interface, une correspondance entre les primitives d'interface des différentes sous-couches SNIC doit être éventuellement réalisée par le module R+R. Ceci est nécessaire quand ces primitives d'interfaces ne sont pas identiques pour les différentes SNIC. On peut noter qu'il s'agit simplement d'une correspondance syntaxique car les services offerts par les sous-couches SNIC sont identiques par définition (ils correspondent au service réseau OSI).

Cette approche est adaptée dans le cas où la majorité des sous-réseaux fournissent déjà le service réseau OSI (les sous-couches SNDC et SNIC sont vides). L'idée donc est de concaténer les différentes communications à travers chaque sous-réseau pour former la communication à travers le réseau global. Pour cela, il suffit d'implémenter dans les passerelles les modules R+R qui assure le relais. Il est à noter que le module R+R ne fait l'objet d'aucune normalisation et reste donc spécifique à chaque implémentation.

2.2.2. L'approche "internet"

Cette approche est basée sur le principe d'encapsulation. Il s'agit ici d'offrir une interface harmonisée au niveau transport. Cette interface est fournie par la sous-couche SNIC sur toutes les stations hôtes et les passerelles. Cette sous-couche est implémentée dans chaque sous-réseau au moyen de son protocole d'accès au sous-réseau spécifique SNAC et éventuellement la sous-couche d'adaptation SNDC. Les structures PDU de SNIC qui sont émises par la station hôte traversent les différents sous-réseaux au moyen des protocoles SNAC de chaque sous réseau. Ces PDU sont routés dans les différentes passerelles à partir des informations de routage contenues dans leur en-tête. La figure A.4 schématise l'interconnexion de réseaux selon l'approche "internet" de l'ISO.

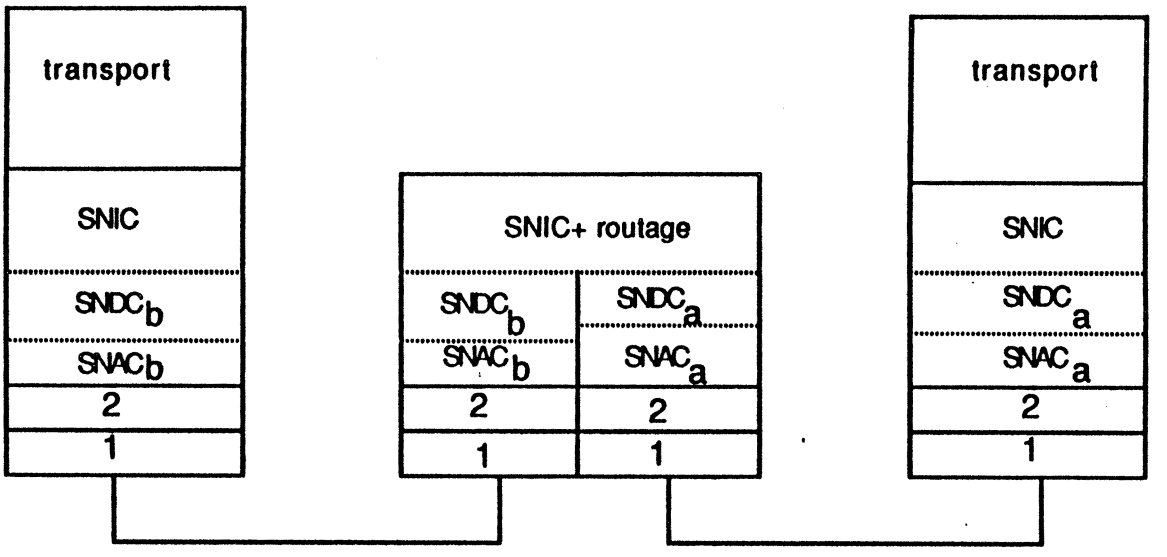


Fig A.4 Interconnexion OSI basée sur
l'approche "Internet"

Cette approche est particulièrement intéressante lorsque la plupart des sous-réseaux à interconnecter ne supportent pas le service réseau OSI. Cette hétérogénéité est cachée par la couche SNIC implémentée sur l'ensemble des sous-réseaux.

2.2.3. Application des approches internet et relais

Comme nous l'avons déjà vu, le modèle OSI fournit au niveau réseau deux types de service, l'un en mode connexion [ISO/DIS 8348] et l'autre en mode sans connexion [ISO/DIS 8348/DAD1]. Les protocoles qui offrent ces services sont décrits, respectivement pour chaque service, dans les références [ISO/DIS 8208] et [ISO/DIS 8473].

Théoriquement, les deux approches d'interconnexion peuvent être associées indifféremment aux protocoles en mode connexion ou en mode sans connexion. Lorsqu'on pose le problème d'interconnexion entre des réseaux OSI qui offrent tous l'un ou l'autre des services réseaux, il n'y a pas de différence essentielle entre la fonction d'encapsulation et la fonction de relais.

Les deux approches ainsi que les protocoles d'interconnexion associés correspondent en réalité à deux solutions pour deux cas concrets d'interconnexion :

1) l'ensemble des sous-réseaux à interconnecter supporte le protocole X25 niveau paquet au niveau réseau. La réalisation de la fonction relais assure l'interconnexion de ces sous-réseaux. Cette configuration correspond à l'interconnexion de réseaux X25 privés et publics. Cependant, l'interconnexion entre des réseaux X25 et des réseaux locaux peut être basée sur l'approche relais, à condition que les réseaux locaux supportent le protocole X25 niveau paquet. Cette possibilité est prévue par l'ISO [ISO/DP 8881].

2) chaque sous-réseau supporte un protocole spécifique au niveau réseau. L'approche relais ne peut pas s'appliquer directement. L'interconnexion est alors réalisée selon l'approche d'encapsulation (l'approche internet) qui permet de masquer l'hétérogénéité. Cette solution répond au problème d'interconnexion entre réseaux locaux et réseaux X25. Le protocole d'encapsulation utilisé dans ce cas est le protocole réseau en mode sans connexion, un protocole en mode connexion n'étant pas approprié à l'encapsulation pour les raisons que nous avons citées lors de la présentation de l'approche d'encapsulation. Le protocole réseau OSI en mode sans connexion (appelé souvent *OSI Internet Protocol "OSI-IP"*) est plutôt conçu comme un protocole d'encapsulation dans un environnement d'interconnexion que comme un protocole offrant simplement le service réseau sans connexion. Cela explique sa similitude avec le protocole IP dans l'architecture DARPA (pour cette raison, nous ne détaillons pas le protocole d'encapsulation OSI). Pour avoir une idée des principaux mécanismes utilisés, on peut se rapporter au protocole IP présenté dans le paragraphe 3.3.

On peut noter, dans le cas d'interconnexion entre réseaux locaux et réseaux X25, que le choix important à faire concerne l'utilisation ou non du protocole X25 niveau paquet dans les réseaux locaux. Le choix de supporter X25 présente les caractéristiques suivantes :

- il simplifie la réalisation de l'interconnexion avec le réseau X25.

- il permet une communication avec les stations attachées au réseau X25 qui supportent uniquement les protocoles X25. Cela signifie que deux types d'interconnexion sont réalisés à la fois : la première est entre réseaux locaux à travers le réseau X25, la seconde est entre les réseaux locaux d'un côté et le réseau X25 de l'autre.

- l'utilisation d'un protocole de transport classe 2 ou classe 3 est suffisant dans ce cas.

- supporter X25 peut engendrer une dégradation des performances de la communication à l'intérieur du réseau local (intra-réseau).

Le second choix consiste à ne pas supporter X25 niveau paquet dans les réseaux locaux, le besoin d'interconnexion implique alors l'utilisation du protocole réseau en mode sans connexion sur l'ensemble des réseaux connectés. Cela implique les conséquences suivantes :

- l'utilisation du protocole transport classe 4 pour la détection des erreurs et la reprise.

- les performances de la communication intra-réseau (pour les réseaux locaux) sont moins affectées que dans le cas précédent, grâce à la simplicité du protocole en mode sans connexion.

- une communication entre une station attachée à un réseau local et une autre attachée au réseau X25 n'est plus possible. Seule la communication entre des stations attachées au différents réseaux locaux est possible (interconnexion des réseaux locaux à travers le réseau X25).

2.3. Scénarios d'interconnexion

Nous montrons dans ce paragraphe, à travers deux scénarios d'interconnexion entre réseaux à grande distance et réseaux locaux, l'application des deux méthodes d'interconnexion OSI dans un environnement d'interconnexion réel.

2.3.1. Scénario basé sur l'approche orientée connexion

Cette approche consiste à supporter au niveau réseau d'un réseau local un protocole en mode connexion. L'interconnexion s'effectue alors selon le principe de relais.

Il s'agit donc d'interconnecter deux réseaux locaux et un réseau public X25 au moyen du protocole X25 PLP (niveau paquet) version 1984 (noté X25 PLP OSI) qui offre le service réseau en mode connexion. Ce protocole est implémenté dans les deux réseaux locaux comme représentant du niveau réseau.

L'ISO a défini dans le document [ISO/DP 8208] l'utilisation du protocole X25 PLP OSI (niveau paquet) en mode point à point pour offrir le service réseau OSI en mode connexion. Le document [ISO/DP 8881] définit l'utilisation de ce protocole dans les réseaux locaux normalisés par le projet IEEE 802.

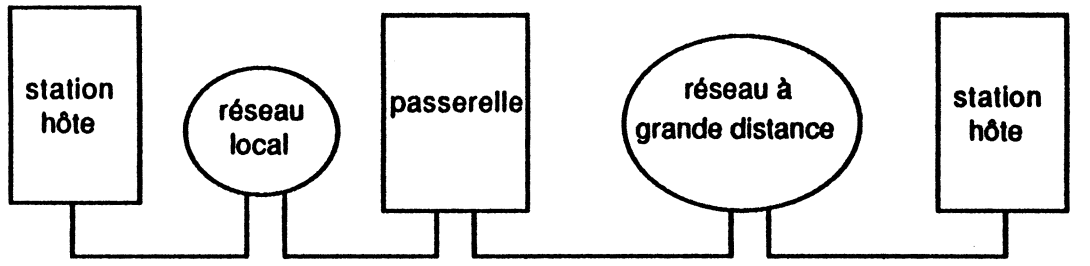
La figure (Fig A.5) présente la structure d'interconnexion des réseaux locaux à un réseau X25 utilisant le protocole X25 PLP OSI pour offrir le service réseau OSI. Chacun de ces deux réseaux est interconnecté au moyen d'une passerelle au réseau public X25 qui lui supporte au niveau paquet le protocole X25 PLP version 1980 qui n'offre pas le service réseau OSI en mode connexion. Le module communicateur du côté réseau local supporte les couches MAC, LLC et X25 PLP OSI. Du côté réseau X25, le module communicateur supporte les protocoles X25 niveau 1, 2(LAPB) et 3(PLP) auxquels il faut rajouter la sous-couche d'harmonisation SNDC qui élève le protocole X25 PLP au niveau X25 PLP OSI. Les fonctionnalités de cette sous-couche sont définies dans le document [ISO/DP 8878]. Ainsi l'ensemble des sous-réseaux interconnectés fournit le service réseau OSI en mode connexion. Le module relais réalise la concaténation des circuits virtuels, ce qui donne une apparence de circuit virtuel de bout en bout.

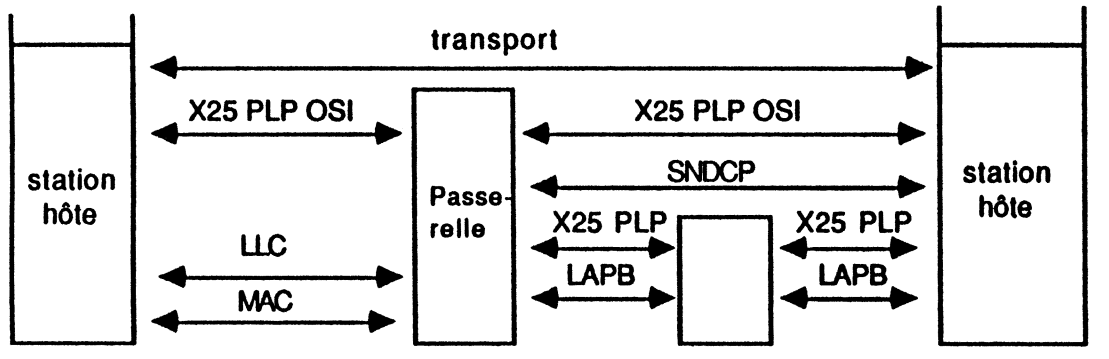
2.3.2. Scénario basé sur l'approche orientée sans connexion

Dans ce scénario, le niveau réseau offre un service en mode sans connexion. Chacun des réseaux interconnectés implémente le protocole réseau en mode sans connexion défini dans [ISO/DIS 8473]. Ce protocole est conçu particulièrement pour permettre l'interconnexion selon le principe

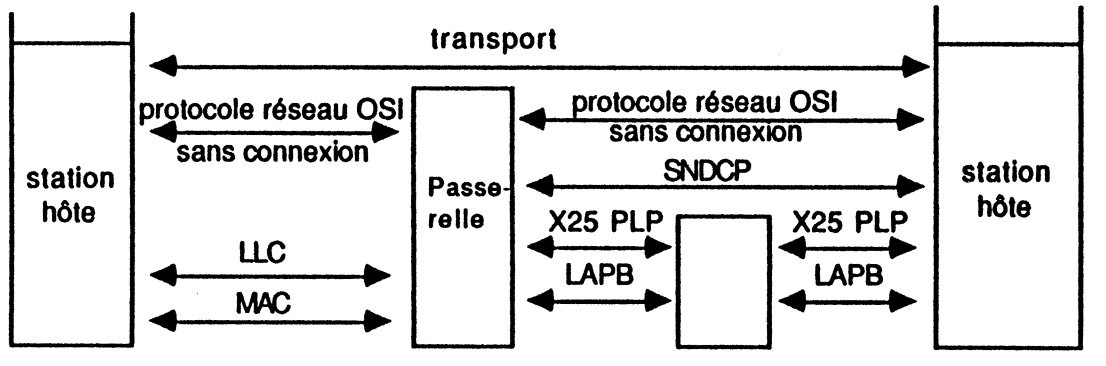
d'encapsulation. Il est très similaire au protocole IP défini dans le cadre de l'architecture DARPA. Le document [ISO/DP 8473/DAD1] décrit les services qui doivent être assurés par une couche de convergence SNDC. Cette couche permet aux réseaux locaux basés sur l'architecture IEEE.802 et aux réseaux basés sur le X25 PLP de supporter le protocole d'interconnexion réseau en mode sans connexion.

La figure Fig A.5 montre l'architecture d'interconnexion des réseaux locaux à un réseau X25 basé sur l'utilisation du protocole réseau en mode sans connexion. Ce protocole représente dans ce cas la couche SNIC qui opère sur l'ensemble des réseaux interconnectés (réseaux locaux et réseau X25). La passerelle supporte d'un côté les protocoles IEEE.802 MAC et LLC, et de l'autre côté les protocoles X25. La couche d'encapsulation est assurée par le protocole réseau en mode sans connexion. La communication de bout en bout en mode connexion est assurée par le protocole de transport. La classe du protocole transport qui est généralement choisie dans ce contexte est la classe 4. Le service de détection d'erreurs et de reprise assuré par cette classe permet de remédier à la faible fiabilité du protocole réseau en mode sans connexion.





Interconnexion en mode connexion



Interconnexion en mode sans connexion

Fig A.5 Les alternatives OSI pour l'interconnexion des réseaux locaux et des réseaux public

2.4. Adressage et routage

L'adressage dans le modèle OSI est basé sur la notion de *point d'accès au service (N)-SAP*. Cette notion est utilisée à tous les niveaux. A un niveau N, un (N)-SAP permet de spécifier une entité communicante qui utilise les services de la couche N sur une station hôte. Chaque (N)-SAP est adressable au moyen de son identificateur (N)-SAP-id. Nous nous intéressons dans ce paragraphe aux structures d'adressage des (N)-SAP au niveau réseau (appelées NSAP comme Network Service Access Point). Ceci parce que l'interconnexion selon le modèle OSI se fait au niveau réseau. Les NSAP représentent donc, le premier niveau d'adressage sur le réseau global, ce qui implique l'unicité des adresses NSAP.

Il est important de dissocier la notion de NSAP de l'adresse physique d'une station. Cette adresse est appelée selon l'ISO *point d'attachement à un sous-réseau (Subnetwork Point of Attachement (SNPA))*. Celle-ci permet de désigner une station quelconque à l'intérieur d'un sous-réseau selon le protocole spécifique d'accès à ce réseau. Cela correspond, par exemple, à l'adresse DTE (X121) dans un réseau X25, ou à l'adresse MAC (adresse machine) dans un réseau IEEE.802 CSMA/CD. L'adresse NSAP est donc visible au niveau du réseau global alors que le SNPA a un sens à l'intérieur d'un sous-réseau. En plus, le NSAP ne désigne pas seulement une station hôte mais aussi une entité communiquante dans cette station. Cependant, il est possible de structurer l'adresse NSAP pour y coder l'adresse physique de la station hôte destinatrice finale.

Un des enrichissements qui ont été introduits au protocole X25 niveau paquet pour offrir le service réseau OSI en mode connexion, consiste à supporter les adresses NSAP en plus des adresses X 121.

Les adresses NSAP ont une structure hiérarchique (Fig. A.6). Cette structuration utilise les notions suivantes [Langlois 85] [ISO/DP 8348/DAD2] :

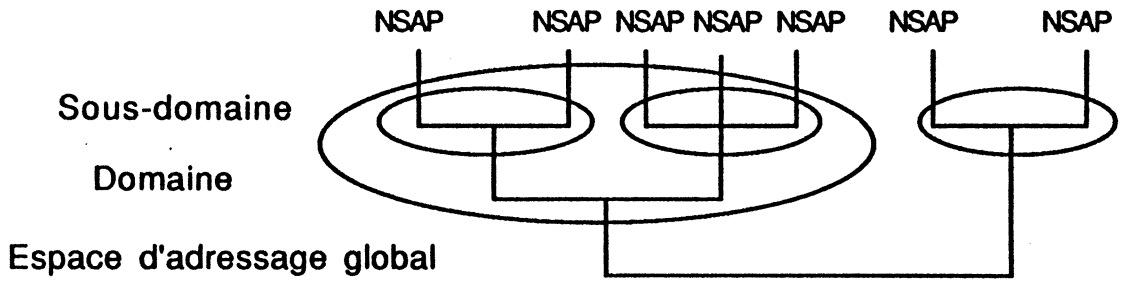


Fig A.6 Structure hiérarchique des NSAP

Domaine : sous-ensemble de l'environnement OSI représentant un espace d'adressage uniforme.

Sous-Domaine : représente un sous-espace d'adressage d'un domaine. La notion de sous-domaine est récursive.

Autorité : responsable de l'allocation des identificateurs uniques à l'intérieur d'un domaine ou d'un sous-domaine.

Le réseau global sera ainsi découpé en sous-domaines. Ce découpage peut être fait de manière à refléter la topologie d'interconnexion. Cela peut faciliter par la suite la fonction de routage. L'inconvénient dans ce cas réside dans la nécessité de modifier l'adresse d'une entité communicante lorsque sa localisation change.

Basé sur les notions présentées ci-dessus, le document [ISO/DP 8348/DAD2] définit la structure hiérarchique des adresses NSAP du point de vue syntaxique et sémantique. Le principe est simple : à chaque niveau hiérarchique appelé *domaine*, une partie initiale de l'adresse sert à identifier sans ambiguïté ce domaine. Le reste de l'adresse est alloué par l'adminstration du sous-domaine concerné et permet d'identifier un NSAP dans ce sous-domaine ou dans un des sous-domaines de niveau hiérarchiquement inférieur (définition récursive). Nous obtenons donc la structure suivante :

<NSAP> = <IDP> <DSP>
<IDP> = <AFI> <IDI>

- *IDP (Initial Domain Part)* : cette partie de l'adresse identifie un sous-domaine, elle est décomposée en deux parties :

- *AFI (Authority and Format Identifier)* : cet identificateur spécifie l'autorité responsable de l'allocation de la valeur de l'identificateur IDI, son format, et la syntaxe abstraite du champ DSP. Deux principales syntaxes sont définies, binaire et décimale. La valeur de AFI est représentée par deux digits décimaux appartenant aux valeurs [00,99]. Certaines valeurs sont réservées à l'usage des normes ISO DCC, CCITT X.121, E.163 (PSTN), E.164 (Telex)...etc. Les valeurs 48 et 49 sont réservées pour des environnements locaux où il n'y a pas encore d'autorité habilitée à gérer l'allocation des IDI, ce dernier champ est alors nul.

- *IDI (Initial Domain Identifier)* : cet identificateur spécifie un sous-domaine dans lequel la valeur du DSP sera significative.

- *DSP (Domain Specific Part)* : cet identificateur permet de spécifier un NSAP dans le sous-domaine identifié par IDP. Il peut à son tour être structuré hiérarchiquement. La valeur de DSP est allouée par l'autorité qui gère le sous-domaine spécifié par IDI.

A titre d'exemple, nous présentons la structure NSAP utilisée dans l'architecture MAP version 2.2 (Manufacturing Automation Protocol). La structure décrite ici est celle qui est appelée format global. Elle est composée des champs suivants (Fig A.7) :

- AFI = 37 indique que l'identificateur de domaine (IDI) est géré par le CCITT selon X.121 et que la syntaxe décimale sera utilisée dans le champ DSP.

- la valeur du champ IDI désigne un domaine MAP.

- le champ DSP est structuré hiérarchiquement pour refléter la structure d'interconnexion MAP. Un octet est réservé à l'identification d'un sous-réseau SNI (SubNetwork Identifier). Sept octets sont réservés à l'adresse physique de la station destinatrice finale, ils correspondent soit à une adresse X121, soit à une adresse MAC IEEE selon que le sous-réseau final est un réseau X25 privé ou un réseau local IEEE. Le dernier octet est

réservé au champ NSS (Network Service Selector). Celui-ci permet de sélectionner une entité communicante utilisatrice du service réseau (entité transport) sur la station hôte destinatrice.

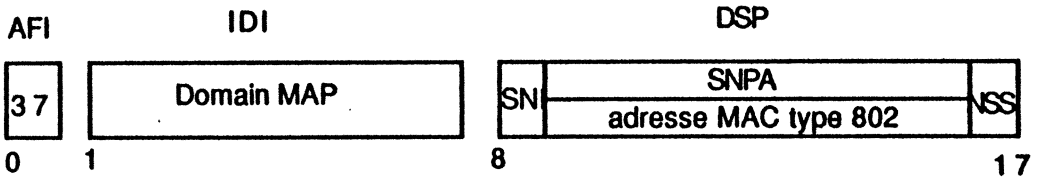


Fig A.7 Format des NSAP selon MAP

En ce qui concerne le routage, le modèle OSI ne définit pas de stratégie de routage inter-réseaux en terme de choix d'une route parmi plusieurs possibles (tout comme dans l'architecture DARPA). Cette stratégie est donc spécifique à chaque réseau global. En revanche, la structure hiérarchique des NSAP peut être exploitée afin d'en extraire une partie des informations concernant le routage. La fonction de routage opère au niveau SNIC dans les stations hôtes ainsi que dans les passerelles. Les adresses NSAP destinataires sont analysées dans la station source et dans chaque passerelle traversée. Le parcours de la structure hiérarchique des NSAP permet la résolution de l'adresse de proche en proche jusqu'à l'arrivée du PDU à la station hôte destinatrice. Comme dans le protocole IP de l'architecture DARPA, le protocole réseau OSI en mode sans connexion offre les mécanismes de routage source et d'enregistrement de route.

Il est clair qu'une partie des travaux de l'ISO en matière d'interconnexion (en particulier ceux concernant la définition d'un protocole réseau en mode sans connexion) a été influencé par l'expérience réussie du modèle d'inter-connexion DARPA. Ce modèle fera l'objet des prochains paragraphes.

3. Le modèle d'interconnexion DARPA

Sous l'égide de Defense Advanced Research Projects Agency (DARPA), le département de la défence américaine a financé depuis 1970 le développement de plusieurs réseaux de types différents : réseaux locaux, réseaux de satellite, réseaux à grande distance, réseaux de radio, etc..., dont le premier était le réseau ARPANET. L'architecture de communication commune à ces réseaux est souvent appelée l'architecture DARPA. Le problème de l'interconnexion de ces différents réseaux s'est donc posé pour aboutir vers le début des années 80 au développement d'un protocole d'interconnexion *(internet protocol)*. Ce protocole a été intégré dans l'architecture de communication DARPA.

Il nous a paru nécessaire d'inclure le modèle d'interconnexion DARPA dans notre présentation pour les deux raisons suivantes :

- l'architecture et les protocoles DARPA sont devenus une norme de fait depuis l'intégration de ces protocoles dans le système UNIX et la très large diffusion que connait ce système.

- le modèle d'interconnexion DARPA représente un exemple typique, opérationnel et très diffusé de l'approche d'encapsulation.

3.1. L'architecrure DARPA

La technique d'interconnexion utilisée dans l'architecture DARPA est basée sur l'approche d'encapsulation. La couche encapsulation (appelée "internet") est intégrée dans l'architecture DARPA entre les protocoles spécifiques d'accès aux sous-réseaux interconnectés et le protocole transport. La figure A.8 montre l'architecture DARPA et la place de la couche d'encapsulation ainsi que la correspondance entre les couches OSI et les couches DARPA. Cette architecture est décomposée en quatre niveaux :

1) le premier niveau inclut les protocoles spécifiques à un réseau. Ce niveau contient, par exemple, les protocoles Ethernet pour un réseau Ethernet ou les protocoles X25 niveau physique, ligne et paquet pour un réseau X25. Ce niveau correspond aux couches physique, liaison de données, et une partie de la couche réseau (SNAC+SNDC) dans le modèle OSI. L'architecture DARPA ne définit pas de protocole à ce niveau.

2) le deuxième niveau contient la couche permettant l'interconnexion des réseaux pour constituer un réseau global. C'est la couche d'encapsulation, qui réalise le protocole IP (Internet Protocol). Son objectif est de présenter une interface commune à la couche transport supérieure. Ce niveau correspond à la sous-couche supérieure (SNIC) de la couche réseau dans le modèle OSI.

3) le troisième niveau est le niveau transport qui permet une communication inter-réseaux de bout en bout. A ce niveau, deux protocoles sont définis : un protocole en mode connexion appelé TCP (Transmission Control Protocol) et un protocole en mode sans connexion appelé UDP (User Datagram Protocol).

4) le quatrième et dernier niveau contient des protocoles d'application répartis comme le protocole de transfert de fichiers FTP (File Transfert Protocol), le protocole d'accès au terminal Telnet, le protocole de serveur de noms NSP (Name Server Protocol), etc...

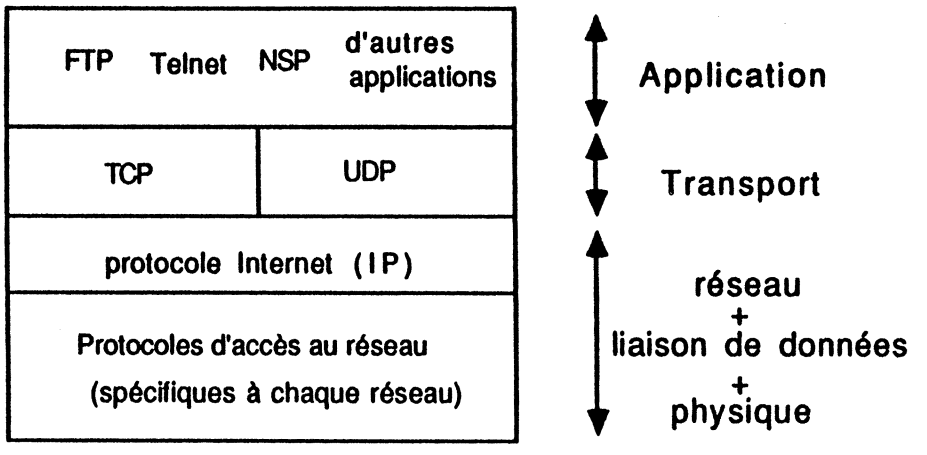


Fig A.8 L'architecture de communication DARPA
et la position du protocle Internet

3.2. Le principe d'interconnexion

Une collection de sous-réseaux interconnectés utilisant le protocole IP comme couche d'encapsulation constitue un réseau global appelé *Catanet* (une concaténation des communications au niveau IP forme une communication inter-réseaux globale). Suivant le principe d'encapsulation, ces différents sous-réseaux sont interconnectés par des passerelles. Chaque passerelle implémente le premier niveau (les protocoles spécifiques au sous-réseau) de chaque sous-réseau auquel elle est attachée ainsi que la couche IP (Fig A.9). La passerelle n'est pas un système dédié mais une station hôte qui respecte l'archirecture nécessaire pour l'interconnexion. Les stations hôtes supportent la couche IP.

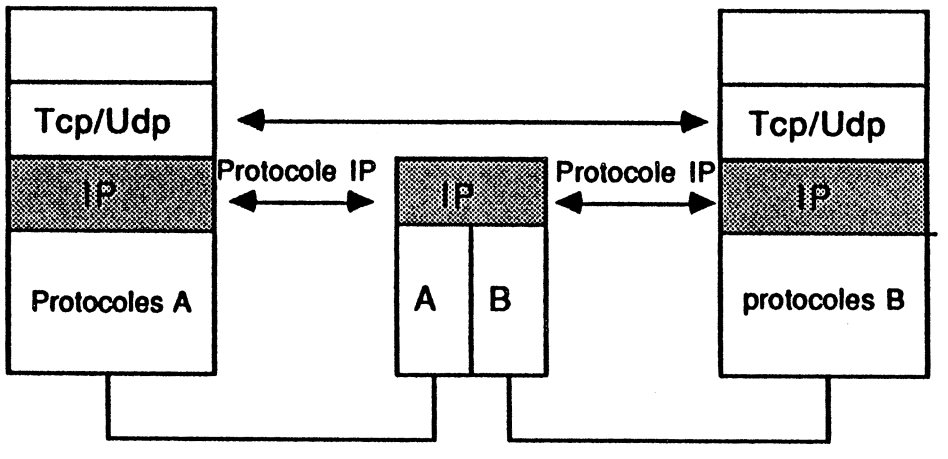


Fig A.9 Le principe d'interconnexion DARPA

Au moyen d'un protocole en mode sans connexion, la couche IP transfère les données de proche en proche jusqu'à leur destination finale. La couche supérieure qui réalise un protocole transport de bout en bout (TCP ou UDP) utilise un schéma d'adressage global qui permet de désigner tout correspondant sur le réseau global. Les unités de données du protocole transport TPDU sont délivrées à la couche IP dans la station hôte. La couche IP colle au TPDU l'en-tête IP qui contient notamment l'adresse destinataire globale et forme ainsi un *datagramme* (c'est le nom donné généralement aux unités de données d'un protocole en mode sans connexion au niveau réseau). La couche IP dans la station hôte identifie la localisation du destinataire final.

Si le destinataire n'est pas sur le sous-réseau local, le datagramme sera transmis à la passerelle appropriée au moyen de protocoles spécifiques au sous-réseau local. Le datagramme sera véhiculé ainsi d'une passerelle à une autre jusqu'à ce qu'il atteigne la station hôte destinatrice. La couche IP, dans cette station, reçoit le datagramme, décapsule l'en-tête IP et délivre le TPDU au protocole transport approprié.

3.3. Le protocole IP

L'objectif de ce paragraphe est de présenter avec certains détails le protocole IP [Postel 81], d'une part, pour identifier la particularité d'un protocole d'interconnexion de type internet surtout en matière de simplicité, et d'autre part, parce qu'il est très diffusé et opérationnel sur la majorité des systèmes UNIX. Enfin, le protocole IP présente beaucoup de similitudes (si on excepte les structures d'adresses) avec le protocole réseau OSI en mode sans connexion [ISO/DIS 8473] qui joue essentiellement le rôle d'un protocole internet.

Après la description du fonctionnement général présenté ci-dessus, nous étudions dans ce paragraphe les éléments et les fonctions réalisées par le protocole IP. Trois fonctions principales IP sont identifiées :

1) l'envoi et la réception des datagrammes de proche en proche en mode sans connexion.

2) une fonction de routage qui est réalisée dans la station hôte source et dans les stations passerelles. Cette fonction est basée sur la structure d'adressage définie au niveau IP qui est appelée structure d'adressage internet.

3) l'adaptation de la longueur des datagrammes aux longueurs des données acceptées par les différents sous-réseaux traversés au moyen d'un mécanisme de segmentation et de réassemblage.

3.3.1. Format des datagrammes IP

Le protocole IP est un protocole en mode sans connexion qui a été conçu dans un souci de simplification afin de ne pas compromettre les performances des communications inter-réseaux et intra-réseau dans lesquelles il intervient. Un seul type d'unité de données de protocole est défini, c'est le datagramme, il est

composé de deux parties : la partie en-tête, qui contient des informations de gestion du protocole IP ; la partie données, qui contient un PDU (ou un segment de PDU) de la couche supérieure. La figure A.10 montre le format de l'en-tête d'un datagramme IP, dont les champs principaux sont les suivants :

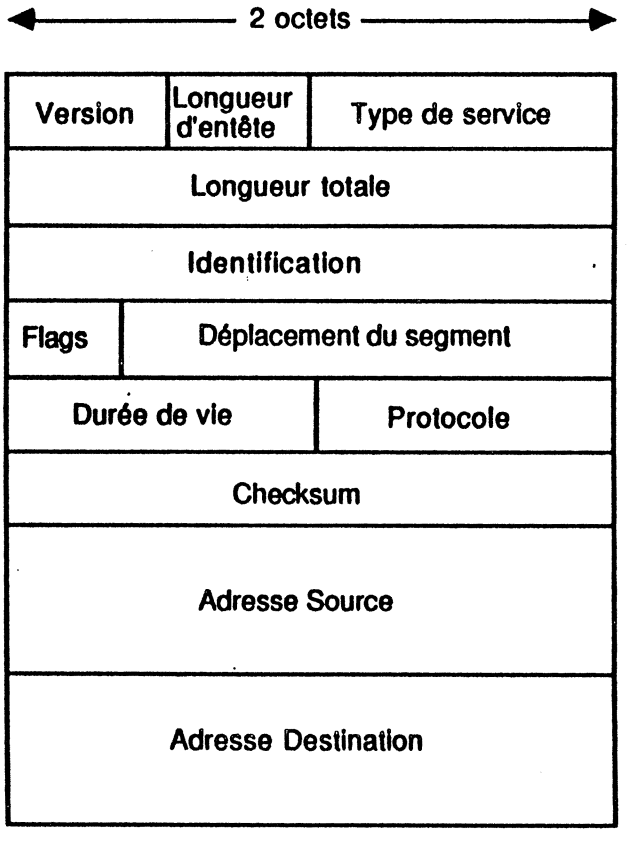


Fig A.10 Format de l'en-tête IP

Type de service *(Type of service)* :

Ce sont des paramètres qui permettent de décrire une qualité de service abstraite demandée au protocole IP. Ces paramètres sont utilisés par le protocole IP dans les passerelles pour choisir les paramètres de service qui concernent les communications au niveau de chaque sous-réseau traversé. Le protocole IP définit seulement une abstraction des services qui peuvent être demandés, l'exploitation réelle de ces informations est définie au moment de l'implémentation en fonction du protocole sous-réseau.

Duré de vie *(Time To Live)* :

Ce champ indique au départ la durée maximale de l'existence du datagramme dans le réseau global (Catenet). Ce temps est décrémenté dans chaque station recevant le datagramme. Quand la valeur de ce champ devient nulle, le datagramme doit être ignoré (éliminé). L'idée est d'économiser des ressources en éliminant des datagrammes devenus probablement inutiles. En effet, le protocole transport qui est de bout en bout effectue des retransmissions sur temporisation. Les "vieux" datagammes sont ainsi dupliqués, leur éliminiation du réseau global est donc souhaitable et même nécessaire dans certains cas. Ce paramètre permet aussi d'éviter le bouclage infini de certains datagrammes qui peut être provoqué par des erreurs de routage.

Identificateur de protocole *(Protocol)* :

Le protocole IP traite les datagrammes de manière indépendante les uns des autres. Ces datagrammes peuvent envelopper des PDU des différents protocoles transport (TCP ou UDP par exemple). L'identificateur de protocole dans un datagramme permet de désigner le protocole de niveau supérieur qui est concerné par le contenu de ce datagramme.

Contrôle de checksum *(Header Checksum)* :

Le checksum est calculé seulement sur l'en-tête du datagramme, laissant la responsabilité de la vérification du contenu du datagramme à la couche supérieure. Le contrôle checksum est fait à chaque couche IP traversée par le datagramme (de proche en proche) car certains paramètres comme la durée de vie ou les paramètres de segmentation sont modifiés dans chaque passerelle.

Autres champs :

Il existe aussi d'autres champs dans l'en-tête IP comme le numéro de version du protocole, la longueur de l'en-tête et la longueur totale du datagramme, ceci en plus des champs qui concernent le mécanisme de segmentation et réassemblage ou l'adressage. Ces derniers sont examinés dans le prochain paragraphe.

3.3.2. Eléments du protocole IP

Adressage :

Le modèle DARPA utilise au niveau internet une structure d'adressage hiérarchique codée sur quatre octets. Le datagramme IP contient l'adresse internet source ainsi que l'adresse internet destination. L'adresse est formée de deux parties : la première correspond à l'identificateur du réseau (nous utiliserons le terme "réseau" à la place de "sous-réseau" par commodité, le réseau qui résulte de l'interconnexion des sous-réseaux est appelé "réseau global" pour éviter l'ambiguïté) ; la deuxième partie correspond à l'identificateur de la station hôte. Afin de prendre en compte différentes configurations d'interconnexion, la longueur de champ allouée à chaque identificateur n'est pas figée, mais elle est fonction de ce qui est appelé la classe du réseau, exprimée dans un troisième champ de l'adresse. Ainsi nous obtenons la structure d'adressage suivante:

<ad-internet> = <cl-réseau><id-réseau><id-hôte>

où :

ad-internet : est l'adresse internet qui permet de désigner une station hôte dans le réseau global.

cl-réseau : est la classe du réseau.
id-réseau : l'identificateur du réseau qui permet de désigner un réseau parmi l'ensemble des réseaux interconnectés.
id-hôte : l'identificateur d'une station hôte attachée au réseau spécifié dans le champ id-réseau.

Trois classes de réseaux ont été définies :

Classe A :

la longueur du champ cl-réseau est de 1 bit, sa valeur en binaire est "0". Dans cette classe, il y a 7 bits alloués au champ id-réseau et 24 bits alloués au champ id-hôte, ce qui permet d'interconnecter potentiellement 2^7 réseaux et 2^{24} stations hôtes par réseau.

Classe B :

la longueur du champ cl-réseau est de 2 bits, sa valeur binaire est "10." Dans cette classe, il y a 14 bits alloués au champ id-réseau et 16 bits alloués au champ id-hôte, ce qui permet d'interconnecter potentiellement 2^{14} réseaux et 2^{16} stations hôtes par réseau.

Classe C :

la longueur du champ cl-réseau est de 2 bit, sa valeur binaire est "11". Dans cette classe, il y a 22 bits alloués au champ id-réseau et 8 bits alloués au champ id-hôte, ce qui permet d'interconnecter potentiellement 2^{22} réseaux d 2^{8} stations hôtes par réseau.

Le choix de la classe appropriée est la responsabilité de l'administrateur de réseau. Ce choix se fait en fonction de la configuration d'interconnexion.

A noter les remarques suivantes concernant l'adressage IP :

- le partitionnement de l'espace d'adressage en deux niveaux hiérarchiques est directement exploité par les passerelles afin de prendre la décision de routage appropriée.

- la notion de réseau correspond à la visibilité au niveau internet. Ainsi un composant réseau peut être lui-même composé d'un ensemble de réseaux interconnectés, vus au niveau adressage et routage internet comme un seul réseau. C'est le cas quand on interconnecte plusieurs réseaux Ethernet par des répéteurs, cet ensemble est alors vu comme étant un seul composant réseau du réseau global.

- la correspondance entre l'adresse internet d'une station hôte ou d'une passerelle et son adresse physique (son adresse au niveau inférieur) est réalisée par la couche IP mais elle ne fait pas partie de la spécification IP. Ceci est donc spécifique à chaque implémentation qui applique la méthode de correspondance appropriée en fonction du réseau traversé. A titre d'exemple nous citons les trois méthodes suivantes :

  - le protocole ARP (Adresse Resolution Protocol) qui est utilisé dans le

réseau Ethernet et qui consiste à diffuser un message demandant l'adresse Ethernet qui correspond à une adresse internet donnée.

- l'utilisation des tables de correspondances entre l'adresse internet et l'adresse physique. Ces tables peuvent exister sur toutes les stations ou dans un sous-ensemble auquel cas on y accède à travers un protocole particulier.

- la troisième méthode est la plus économique, elle consiste à donner au champ id-hôte une sémantique (un codage particulier), permettant de retrouver l'adresse physique au moyen d'une fonction de correspondance : ad-physique = f(ad-hôte). Cette méthode n'est pas toujours applicable, en particulier dans les réseaux type Ethernet où les adresses physiques des stations sont codées sur 48 bits.

Routage :

La décision de routage est une fonction essentielle du protocole IP, cependant l'algorithme de routage lui-même ne fait pas partie de la spécification du protocole. Cette décision de routage est basée sur l'analyse du champ d'adresse destination dans le datagramme et en particulier du sous-champ qui correspond à l'identificateur réseau. Comme l'adresse internet destination contenue dans le datagramme est celle de la destination finale (de bout en bout) et non l'adresse internet de la prochaine passerelle, elle n'est donc pas modifiée lors des passages à travers les différentes passerelles. La décision de routage est prise de proche en proche (dans la couche IP de chaque passerelle traversée) jusqu'à l'arrivée du datagramme vers la destination finale. A la réception d'un datagramme par la couche IP d'une station source ou d'une passerelle, les actions suivantes sont effectuées :

- le champ adresse destination est analysé, si l'identificateur réseau correspond à un réseau auquel la station est directement connectéee, le datagramme sera envoyé directement à sa destination en utilisant le protocole de communication spécifique au réseau. L'adresse destination selon ce protocole (c'est le protocole au niveau directement inférieur à IP) est obtenue à partir de l'adresse internet au moyen d'une des méthodes présentées ci-dessus (table de correspondances, protocole ARP, etc...).

- si l'identificateur réseau ne correspond pas à un réseau auquel la station est directement connectée, le datagramme est envoyé vers une passerelle accessible à travers un réseau auquel la station est directement connectée. L'adresse internet de cette passerelle est obtenue au moyen d'un algorithme de routage (typiquement en accédant à une table qui contient des associations entre identificateur réseau et adresse internet d'une prochaine passerelle permettant de l'atteindre). La transmission du datagramme vers la passerelle utilise le protocole spécifique au réseau qui connecte la station vers cette passerelle. L'adresse de la passerelle, selon ce protocole, est également obtenue au moyen des mécanismes décrits ci-dessus.

- si l'identificateur réseau n'est pas connu (pas de chemin connu pour l'atteindre) le datagramme sera tout simplement ignoré.

La construction et la gestion des tables de routage ne sont pas définies dans le protocole internet, elles sont donc dépendantes d'une implémentation spécifique. Cependant deux protocoles appelés EGP (Exterior Gateway Protocol) [RFC 904] et GGP (Gateway to Gateway Ptotocol) [RFC 823] ont été définis dans le cadre de l'architecture DARPA pour la communication entre les passerelles. L'objet de cette communication est l'échange d'informations concernant le routage entre les passerelles, afin de réaliser une gestion dynamique du routage.

Le protocole IP offre deux mécanismes optionnels supplémentaires concernant le routage :

1) *Source Routing (routage source)* :

Ce mécanisme permet à la source de spécifier la route que doit prendre le datagramme. Afin de supporter ce mécanisme, le datagramme IP contient un champ optionnel dans lequel sont positionnées les adresses internet des passerelles à traverser. Deux possibilités existent :

- le routage source strict : selon ce routage, la source détermine l'ensemble des passerelles intermédaires qui doivent être visitées, avec l'obligation de passer par les passerelles spécifiées et seulement celles-ci. La route est donc strictement définie.

- le routage source faible : selon ce routage, la source détermine quelques passerelles devant être visitées, mais d'autres passerelles intermédaires peuvent être traversées par le datagramme. La route est donc partiellement définie.

2) *Route Recording (enregistrement de la route)* :

Chaque passerelle doit insérer son adresse internet dans chaque datagramme qui passe par elle et qui comporte l'option "source routing". Ainsi le datagramme reçu par la couche IP, dans la station hôte destinatrice, contient une suite d'adresses internet représentant la route traversée par ce datagramme.

Segmentation et réassemblage :

La segmentation et le réassemblage constituent la deuxième fonction essentielle effectuée par le protocole IP. La segmentation utilisée est de type segmentation inter-réseaux. Sur la route empruntée par un datagramme, chaque passerelle peut effectuer une segmentation si cela est nécessaire. La segmentation d'un datagramme donne lieu à de nouveaux segments datagrammes qui peuvent à leur tour être segmentés si nécessaire. Le réassemblage de l'ensemble de segments n'est effectué qu'au niveau de la couche IP de la station hôte destinatrice. Trois champs dans l'en-tête du datagramme sont destinés au mécanisme de segmentation et de réassemblage :

- le champ identification (Identification (ID)) sur 2 octets.

- le champ déplacement du segment (Fragement Offset (FO)) sur 15 bits.

- le champ des indicateurs (flags) qui contient deux indicateurs: MF (More Fragments) et DF (Don't Fragment).

L'indicateur DF d'un datagramme est positionné lorsque sa segmentation n'est pas permise. Ceci peut obliger une passerelle à rejeter le datagramme si sa taille ne lui permet pas de traverser le prochain réseau. L'indicateur MF est positionné dans tout datagramme ne constituant pas le dernier segment d'un datagramme initial.

Les actions suivantes sont effectuées dans la couche IP qui reçoit un datagramme nécessitant une segmentation :

- découpage des données et constitution de segments en ajoutant l'en-tête IP à chaque segment. Le découpage doit se faire à la frontière d'un octet.

- la valeur du champ identification est identique dans tous les segments, elle est aussi identique à sa valeur dans le datagramme initial. Le champ permet à l'arrivée (dans la couche IP destinataire finale) d'identifier tous les segments qui doivent être réassemblés.

- si MF = 0 dans le datagramme initial, alors MF = 1 dans tous les segments sauf le dernier qui aura MF = 0. Si MF = 1 dans le datagramme initial tous les segments auront MF = 1.

- la valeur FO dans chaque segment correspond à la position du début des données de ce segment (en nombre d'octets) par rapport au début des données du datagramme initial.

- pour chaque segment il faut calculer la nouvelle valeur du checksum et modifier en conséquence le champ correspondant.

A partir des informations positionnées lors de la segmentation, le processus de réassemblage peut s'effectuer sans difficulté dans la couche IP destinatrice finale. Le réassemblage échoue si un des segments d'un datagramme n'est pas correct (erreur de checksum, durée de vie expirée, etc...). Dans ce cas, tous les segments qui composent le datagramme initial sont ignorés.

Contrôle d'erreurs et contrôle de flux :

Le protocole IP n'effectue aucun contrôle de flux, les datagrammes qui ne peuvent pas être traités sont ignorés. Le protocole de niveau supérieur (le transport) effectuera les retransmissions nécessaires.

La seule action effectuée par la couche IP qui reçoit un datagramme erroné est de le rejeter. Cependant un protocole spécifique a été défini dans la couche internet de l'architecture DARPA. Il a pour rôle essentiel de notifier à la couche IP source un compte-rendu dans certains cas d'erreur. Ce protocole est

appelé Internet Control Message Protocol (ICMP) [RFC 792]. Il utilise le service de communication IP pour envoyer un datagramme de compte-rendu vers la station source. Ceci afin de communiquer à la source le rejet d'un datagramme par le protocole IP, le diagnostic d'erreur et la partie de datagramme qui a provoquée l'erreur. Les principaux cas d'erreurs qui sont reportés sont les suivants :

- contrôle de checksum négatif,
- destination impossible à atteindre,
- rejet d'un datagramme qui n'accepte pas la segmentation,
- erreur de protocole (erreur syntaxique ou sémantique).

3.4. L'utilisation du protocole IP dans une configuration d'interconnexion de réseaux locaux à travers un réseau X25

Un exemple typique d'une configuration d'interconnexion, utilisant l'approche IP, est celle qui permet la communication entre des réseaux locaux de type Ethernet supportant les protocolesTCP/IP, UDP/IP et leurs applications à travers un réseau public X25. Nous décrivons dans ce paragraphe le principe d'une telle interconnexion, les problèmes et le choix de solutions correspondantes. Le terme "réseau local" signifie dans ce paragraphe un réseau local Ethernet supportant les protocoles DARPA

3.4.1. Principe d'interconnexion

Le principe d'interconnexion de plusieurs réseaux locaux à travers un réseau X25, n'est qu'une application directe de l'approche d'encapsulation "internet" pour l'interconnexion de réseaux. Les messages au niveau transport sont encapsulés par les datagrammes IP. Ces datagrammes sont véhiculés d'un réseau local vers un autre en utilisant les protocoles X25 qui représentent les protocoles d'accès au réseau X25. Il suffit pour cela d'introduire une passerelle entre chaque réseau local et le réseau X25. Cette passerelle contient d'une part le protocole Ethernet et d'autre part les protocoles X25. Elle contient en plus la couche IP, qui représente la couche d'encapsulation et qui interface à la fois le protocole Ethernet et le protocole X25 niveau paquet (Fig A.11).

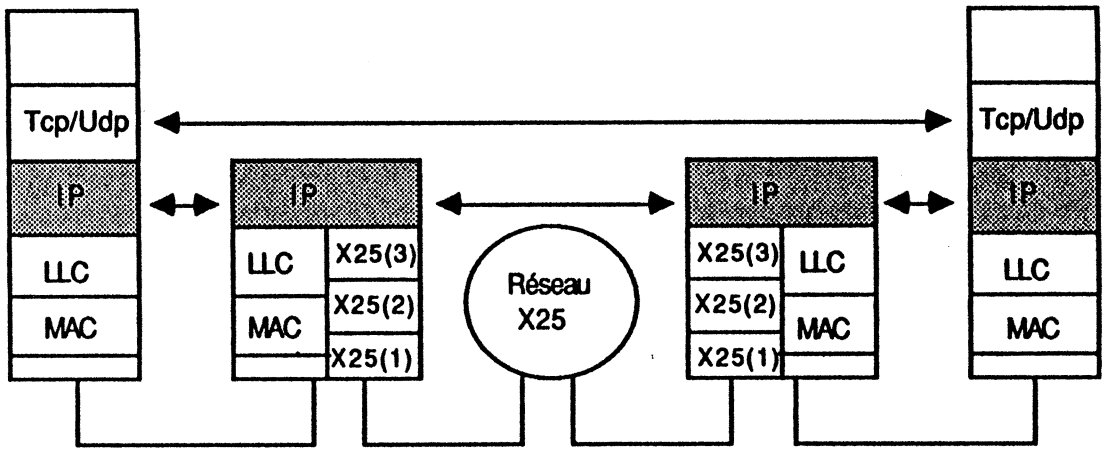


Fig A.11 L'interconnexion des réseaux locaux à travers un réseau X25 au moyen du protocole IP

D'autres approches peuvent a priori être envisagées pour assurer cette interconnexion, comme par exemple une conversion au niveau protocole ou au niveau service entre le protocole TCP et le protocole X25. Une telle approche n'est pas réellement appropriée dans ce cas de figure pour les deux raisons suivantes :

1) une telle conversion peut s'avérer complexe à réaliser à cause de la disparité de services entre X25 et TCP [Comer 83].

2) il s'agit dans cette configuration d'assurer l'interconnexion entre des réseaux locaux homogènes supportant les protocoles DARPA. Seul le chemin qui les relie supporte des protocoles hétérogènes. L'approche de conversion impliquerait dans ce cas une double conversion. Elle peut présenter un certain intérêt si le but était de permettre la communication entre des entités TCP et d'autres utilisant X25.

Dans la suite nous présentons la mise en œuvre du principe d'interconnexion basé sur la couche d'encapsulation IP. Cela montrera la simplicité relative de cette approche et les choix différents qui peuvent être faits lors de la réalisation.

3.4.2. La mise en œuvre de IP sur X25

La mise en œuvre du principe d'interconnexion cité ci-dessus, consiste essentiellement à développer une interface entre la couche IP et le protocole X25 dans les passerelles. Cela signifie qu'il faut assurer l'envoi et la réception des datagrammes IP entre les passerelles à travers des circuits virtuels X25. Plusieurs implémentations de l'interface IP/X25 ont été réalisées, la plupart sur des systèmes UNIX [Comer 83] [Bilting 86].

L'implémentation de la couche IP dans le système UNIX supporte la notion d'interface. Il s'agit de l'interface de IP avec les couches inférieures. Pour une station quelconque, IP a autant d'interfaces différentes que de réseaux différents auxquels la station est connectée. Dans la configuration étudiée ici, la couche IP dans une passerelle possède deux interfaces différentes, l'une avec le protocole Ethernet et l'autre avec le protocole X25. A une interface, on associe les informations, les structures de données et les procédures suivantes :

- deux files d'attente, une en sortie, l'autre en entrée. Les éléments de ces files sont des datagrammes IP qui doivent être envoyés ou ont été reçus de l'interface en question.

- une adresse (au sens adressage internet) réseau auquel l'interface est attachée. Ainsi les datagrammes, qui sont dans la file d'attente en sortie, ont été aiguillés par la fonction de routage réalisée par IP. Cette fontion consiste à déterminer, à partir de l'adresse réseau destinataire dans les datagrammes et l'adresse réseau associée à chaque interface, vers quelle interface il faut envoyer les datagrammes.

- la taille des segments datagramme acceptée en fonction du réseau traversé. Une fois l'interface appropriée déterminée par la fonction de routage, la segmentation appropriée est réalisée. La file d'attente en sortie contient des datagrammes segmentés.

- un ensemble de procédures qui utilisent les primitives du protocole associé à cette interface (X25 dans l'exemple présent) pour envoyer et recevoir les datagrammes IP.

3.4.3. Les fonctionnalités de l'interface IP/X25

Les fonctions principales effectuées par l'interface IP/X25 sont les suivantes :

1) correspondance entre les adresses internet et les adresses X25

La première fonction consiste à déterminer, à partir de l'adresse internet d'un datagramme IP, l'adresse X25 de la passerelle vers laquelle le datagramme est envoyé. En effet, à chaque réseau local, un identificateur réseau au sens adressage internet est attribué. Dans chaque passerelle une table de correspondance est utilisée pour stocker les associations entre les identificateurs réseau et les adresses X25 des passerelles correspondantes. Cette table sera consultée pour l'envoi de chaque datagramme IP.

2) ouverture de circuits virtuels et envoi de datagrammes

Une fois l'adresse X25 de la passerelle distante déterminée, une ouverture de circuit virtuel avec cette passerelle est demandée, à moins qu'un tel circuit ne soit déjà ouvert. Dans ce cas, il suffit d'envoyer le datagramme IP sur ce circuit.

3) fermeture des circuits virtuels

L'algorithme de fermeture des circuits virtuels est un point important car il influence d'un part, le débit d'informations et, d'autre part, le coût dû à l'utilisation des circuits virtuels (surtout pour un réseau X25 public). Deux stratégies peuvent être utilisées :

- les circuits virtuels restent en principe ouverts. Si le nombre maximal permis de circuits virtuels ouverts est atteint et qu'il y a besoin d'ouvrir un nouveau circuit, un circuit sélectionné sera fermé. L'algorithme de sélection consiste, par exemple, à choisir le circuit qui est resté le plus longtemps inactif. Cette stratégie privilégie le paramètre de la performance.

- la deuxième stratégie consiste à définir une durée maximale d'inactivité. Au bout de cette durée le circuit virtuel est fermé. Le choix de cette durée est dicté par le souci de trouver un compromis entre le coût et la

performance.

4) fonctionnalités supplémentaires

Afin d'augmenter le débit de la communication à travers le réseau X25, des fonctionnalités supplémentaires sont envisagées :

a) segmentation des datagrammes

La couche IP effectue une segmentation en fonction d'une taille définie pour chaque interface. Cette segmentation consiste à découper le datagramme en plusieurs segments en dupliquant l'en-tête IP dans chaque segment. La taille du segment pour l'interface X25 est de 128 octets (c'est la taille maximale des données permises dans un paquet de données X25) dont 20 octets sont pris par l'en-tête IP. Afin d'éviter la duplication des en-têtes et améliorer le débit utile, la taille de segment demandée à IP est choisie généralement deux ou trois fois supérieure. La segmentation est alors réalisée au niveau de l'interface X25 en utilisant le flag "More Data" pour le réassemblage.

b) multi-circuits

Certaines implémentations proposent l'utilisation de multi-circuits virtuels entre deux passerelles quand la cadence de l'échange des datagrammes IP commence à provoquer la perte des ces derniers (pas de contrôle de flux au niveau IP). Ce mécanisme est favorisé par le fait que le protocole IP, qui est en mode sans connexion, n'exige pas le séquencement des datagrammes.

4. Le modèle d'interconnexion CCITT

Ce modèle concerne l'interconnexion des réseaux publics qui utilisent le protocole d'accès réseau X25. L'approche d'interconnexion CCITT est un exemple intéressant du problème de l'interconnexion des réseaux à grande distance. Il s'agit dans ce cas d'un ensemble de réseaux qui présentent la même interface d'accès aux stations hôtes. Cette interface d'accès est basée sur les protocoles X25 niveau physique, ligne et paquet normalisés par le CCITT. Le réseau global constitué par l'interconnexion des réseaux X25 représente un des plus grands réseaux d'ordinateurs par son étendue géographique. Le besoin de

définir un protocole d'interconnexion entre les réseaux X25 est lié à deux contraintes :

1) les réseaux X25 utilisent la même interface d'accès basée sur les protocoles X25. Mais les protocoles internes sont spécifiques à chaque réseau. Par conséquent, il n'est pas possible d'intégrer ces réseaux dans un seul réseau global à commutation de circuits. Il est donc nécessaire de définir un protocole commun permettant de véhiculer les paquets X25 d'un réseau vers un autre.

2) pour des raisons d'administration et de sécurité, il est nécessaire d'établir clairement les frontières entre les différents réseaux publics.

4.1. Principe d'interconnexion

L'interconnexion entre deux réseaux X25 est basée sur l'utilisation de deux demi-passerelles appelées STE (Signalling Terminal Equipment). Chaque demi-passerelle est attachée à un réseau X25. Un STE est un nœud de commutation particulier dans un réseau X25. Il joue le rôle d'un nœud de raccordement avec d'autres réseaux X25 et fait donc partie du réseau X25. Ces choix architecturaux caractérisent la solution CCITT par rapport à d'autres possibles comme l'utilisation d'une passerelle "entière" ou d'une passerelle de type station hôte [Cerf 78].

Les deux STE communiquent à l'aide du protocole X75 qui est un protocole en mode connexion très proche de X25. Le protocole X75 est composé aussi de trois niveaux : physique, liaison de données et paquet. Le niveau liaison de données supporte la procédure LAPB, et il permet l'utilisation d'une procédure multiligne afin d'augmenter le débit de communication entre les deux réseaux interconnectés. Au niveau paquet le protocole X75 est similaire au protocole X25 niveau paquet, avec quelques paramètres supplémentaires pour la gestion de l'interconnexion entre les réseaux.

L'interconnexion entre les protocoles X25 et le protocole X75 est basée sur l'approche relais. Ainsi dans chaque demi-passerelle (STE), le module adaptateur réalise les fonctions de relais et de routage nécessaires. La figure A.12 montre le principe d'interconnexion.

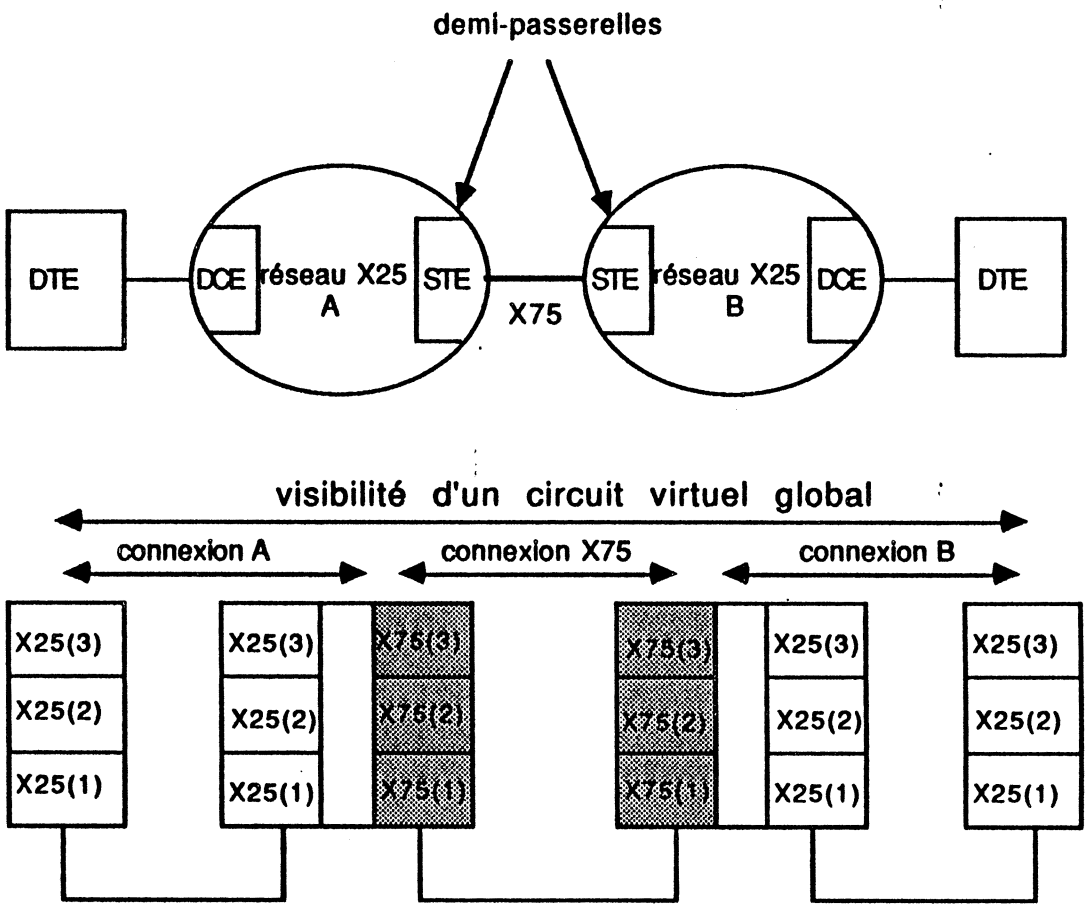


Fig A.12 Interconnexion des réseaux X25
selon l'approche CCITT

La communication entre une station hôte (appelé DTE dans la terminologie CCITT) attachée au réseau X25 R_a et une autre station hôte attachée au réseau X25 R_b est décomposée en cinq connexions partielles :

- la connexion entre la station hôte DTE_a et le nœud de raccordement (appelé DCE selon la terminologie CCITT) sur laquelle la communication se déroule selon la procédure X25 ;

- la connexion entre le nœud DCE_a et le nœud STE_a qui supporte une communication selon les protocoles internes au réseau X25 R_a ;

- la connexion entre les deux demi-passerelles STE_a et STE_b sur laquelle

une communication se déroule selon la procédure X75 à travers un circuit virtuel ;

- la connexion interne au réseau R_b entre le nœud STE_b et le nœud DCE_b ;

- et enfin, la connexion de type X25 entre le nœud de raccordement DCE_b et la station hôte DTE_b.

La concaténation de ces connexions forme le circuit virtuel X25 qui relie les deux stations hôtes. Comme pour la communication interne dans un réseau X25, le circuit virtuel X75 est transparent par rapport aux stations hôtes. Tout se passe comme si les deux stations hôtes étaient connectées au même réseau X25.

Le protocole X75 offre les mécanismes nécessaires pour la création, l'utilisation et la fermeture des circuits virtuels entre deux STE. La fonction d'un relais consiste à établir les circuits virtuels X75 nécessaires à la création du chemin reliant la station DTE appelante et la station DTE appelée. Il faut ensuite relayer les données arrivant sur le réseau X25 pour les réinjecter sur le circuit X75 et vice versa. Enfin, la demande de fermeture de connexion effectuée par l'une des stations DTE doit engendrer la fermeture de l'ensemble des connexions qui constituent le circuit virtuel entre l'appelant et l'appelé, et en particulier le circuit virtuel X75.

Décrivons les opérations effectuées lors des trois phases : l'établissement d'un circuit virtuel entre les stations hôtes DTE_a et DTE_b attachées aux deux réseaux X25 interconnectés par X75, le transfert de données entre ces stations hôtes, et enfin la fermeture du circuit virtuel.

1) établissement du circuit virtuel

La réception du paquet "Call Request" en provenence de DTE_a à travers DCE_a par STE_a provoque la procédure d'ouverture du circuit virtuel X75 entre STE_a et STE_b. L'adresse destinataire contenue dans ce paquet est celle du destinataire final (DTE_b) codée selon la structure d'adressage CCITT définie dans X121. STE_a génère donc un paquet "Call Request" pour demander l'ouverture d'un circuit virtuel X75 avec STE_b. Les paramètres de contrôle de flux positionnés dans ce paquet sont spécifiques à la communication

STE_a-STE_b. A la réception de ce paquet, STE_b émet un paquet "Call Request" selon la procédure X25 vers le DCE_b qui, à son tour, délivre au DTE_b (station hôte destinataire) le paquet "Incoming Call". Si le destinataire (DTE_b) accepte l'ouverture de la communication, il émet un paquet "Call Accepted". Ce paquet traverse le même chemin que le paquet d'ouverture. A son arrivée dans STE_a, le circuit virtuel X75 entre les deux demi-passerelles devient ouvert. STE_a relaie ce paquet vers DCE_a qui envoie le paquet "Call Connected" à la station DTE_a, le circuit virtuel X25 global (DTE_a-DTE_b) devient à son tour ouvert.

2) transfert de données

Les paquets de données envoyés par DTE_a et reçus par STE_a sont relayés sur la connexion X75. Une segmentation de ces paquets peut être effectuée pour traverser cette connexion. Les paramètres de contrôle de flux, de numérotation, et l'identificateur de la connexion sont repositionnés à des valeurs spécifiques à la connexion X75.

3) fermeture de circuit virtuel

D'une façon similaire à la phase d'ouverture, les paquets de demande de fermeture et de confirmation de fermeture "Clear Request" et "Clear Confirmation" du circuit virtuel DTE_a-DTE_b sont relayés par STE_a et STE_b. Ceci provoque la fermeture de la connexion X75 reliant les deux demi-passerelles et la libération de toutes les ressources associées à cette connexion.

La fonction de relais assurée dans les demi-passerelles consiste donc à réaliser la correspondance : (X25)-PDU <=> (X75)-PDU.

Cette opération est assez simple grâce à la similitude syntaxique et sémantique entre le protocole X75 et celui de X25. Seules les valeurs de certains champs dans les paquets sont modifiés pour prendre en compte la spécificité des paramètres de chaque connexion.

4.2. L'adressage CCITT

La recommandation X121 du CCITT définit la structure d'adressage à utiliser dans les réseaux publics basés sur l'interface X25. Ce modèle d'adressage prend en compte l'interconnexion de réseaux X25 entre eux au moyen du protocole X75. Une adresse X121 doit identifier une station hôte (DTE) attachée à un réseau X25 quelconque faisant partie de l'ensemble des réseaux X25 interconnectés. La structure des adresses X121 est basée sur le modèle hiérarchique. Trois niveaux de hiérarchie sont pris en compte : un pays, un réseau public dans ce pays, et enfin, une station hôte connectée à ce réseau. La structure des adresses X121 est donc la suivante :

<adresse X121>:=<identificateur réseau><identificateur hôte>
<identificateur réseau>:= <code pays><code réseau>

Le code d'un pays utilise 3 digits décimaux et supporte au maximum 600 codes différents. Le code d'un réseau utilise 1 digit décimal et permet ainsi de supporter 10 réseaux différents par pays. Une extention de ce code est possible. L'identificateur de la station hôte (DTE) est codé sur 10 digits décimaux. Il permet d'identifier une station hôte unique attachée au réseau identifié par l'identificateur réseau. La gestion et la structure de l'identificateur hôte ne sont pas définies par le CCITT. Seuls les réseaux publics peuvent être identifiés par le code réseau. Il existe donc un problème d'identification des réseaux privés attachés à un réseau public. Cela ne rentre pas dans le cadre de la normalisation CCITT. Cependant le problème est suffisamment important pour essayer d'y apporter une solution générale. Deux méthodes sont proposées :

- la structuration de l'identificateur hôte de façon à pouvoir adresser les réseaux privés.

- l'utilisation du champ "données utilisateur" dans le paquet d'ouverture de circuit virtuel pour supporter les adresses des réseaux privés.

On peut remarquer que l'adressage X121 permet d'identifier un DTE destinataire, mais ne permet pas d'identifier une entité utilisatrice spécifique située sur ce DTE. C'est une des raisons pour laquelle le protocole X25 (niveau paquet) version 1980 n'est pas considéré comme étant un protocole qui offre le

service réseau ISO en mode connexion. Cela a été modifié dans la version X25 1984 qui supporte les identificateurs NSAP source et destinataire dans le paquet de demande d'ouverture de circuit virtuel "Call Request".

Concernant le routage inter-réseaux, on peut noter les points suivants :

- un STE est un commutateur inter-réseaux. Sa fonction de routage est similaire à celle d'un commutateur interne. Cette fonction est réalisée selon des algorithmes de routage spécifiques à chaque réseau qui ne font pas partie de la norme d'interconnexion X75.

- un circuit ou une cascade de circuits X75 est ouvert pour le compte d'un circuit virtuel X25 global liant les DTE source et destinataire. Le mode orienté connexion qui caractérise le protocole d'interconnexion implique que tout le trafic concernant le circuit vituel global passe par la même suite de STE.

- le routage source inter-réseaux n'est pas offert dans le modèle d'interconnexion CCITT. Ceci peut constituer un inconvénient au niveau de la sécurité dans un environnement d'interconnexion de réseaux publics.

4.3. Caractéristiques du protocole d'interconnexion X75

Afin de permettre la continuité de circuit virtuel à travers deux ou plusieurs réseaux X25, le protocole X75 a deux caractéristiques essentielles :

- le service offert sur la portion de la communication X75 est similaire au service attendu de bout en bout, c'est un service en mode connexion.

- le protocole X75 est très similaire à celui de X25, les paquets X25 ne subissent ainsi que peu de modifications lors de leurs passages entre les STE (la portion X75).

Les fonctions de segmentation/réassemblage ainsi que celles de contrôle de flux sont équivalentes à celles offertes dans X25. Les deux STE communicants gèrent ces fonctions sur la connexion X75 indépendamment des autres connexions qui forment le circuit virtuel global. Ainsi, les longueurs de

segments et les paramètres de contrôle de flux (réalisé par le mécanisme de fenêtrage) peuvent avoir des valeurs différentes sur la connexion X75 et sur les portions de communication X25.

L'élément essentiel supplémentaire dans le protocole X75 par rapport à celui de X25, est le champ appelé "utilities". Ce champ existe dans les paquets d'ouverture, d'acceptation et de fermeture de la connexion X75. L'intérêt de ce champ est de supporter des informations utiles pour l'adminstration de l'interconnexion entre les différents réseaux publics X25, comme par exemple l'identification des réseaux traversés, des informations concernant le tarif et le trafic, etc...

5. Conclusion

Cette étude de l'interconnexion de réseaux partiellement hétérogènes permet de dégager les éléments suivants :

- le niveau d'interconnexion est placé généralement au niveau réseau afin d'assurer une communication réellement de bout en bout au niveau transport.

- deux approches sont utilisées :

    - l'approche d'encapsulation qui implique l'introduction d'une couche internet sur l'ensemble des sous-réseaux interconnectés. Cette couche fait abstraction des protocoles spécifiques utilisés dans chaque sous-réseau. Elle utilise un protocole simple en mode sans connexion.

    - l'approche relais qui assure la concaténation des communications intra-réseau. Ces communications utilisent des protocoles identiques qui sont en mode connexion.

- la structure d'adressage utilisée est souvent une structure hiérarchique, afin de pouvoir en extraire des informations nécessaires au routage.

Bibliographie

[Algayrès 86]
Distributed Systems : Internetworking Techniques
System Development Corporation, California, 1986.

[Almes 85]
The EDEN System: a Technical Review
G.T.Almes, A.P.Black, E.D.Lazowska
IEEE Transactions on software Engineering SE-11
January 1985.

[Apple 85]
Inside Appletalk Manual
Apple Computer Inc, 1985.

[Balter 87]
Principe de conception du système d'exploitation réparti Guide
R.Balter et al.
Rapport Guide - R1, Rapport de Recherche IMAG
Grenoble avril 1987.

[Bilting 86]
ARPA IP using X25 in Unix 4.2 BSD
U.Bilting et al.
EUUG Autumn'86, Manchester, September 1986.

[Black 86]
Object structure in the Emerald System
A.Black et al.
Technical report 86-04-03
University of Washington, April 1986.

[Braden 83]
A distributed approach to the interconnection
of heterogeneous computer networks
R.Braden, R.Cole, P.Higginson and P.Lloyd
SIGCOMM 83, Austin, Texas March 1983.

[Brownbridge 82]
The Newcastle Connection or Unixes of the world unite!
D.R.Brownbridge, L.F.Marshall, B.Randell
Software Practice and Experience, Vol. 12, July 1982.

[Burg 84]
Of Local Networks, Protocols, and The OSI Reference Model
F.M.Burg, C.T.Chen, H.C.Folts
Data communications, November 1984.

[Callon 83]
Internetwork Protocol
R.Callon
Proc. of IEEE, Vol.71, No.12, December 1983.

[Cerf 78]
Issues in Packet Network Interconnection
V.G.Cerf, P.T.Kirstein
Proc. of IEEE, Vol.66, No.11, November 1987.

[Cheriton 83]
Local Networking and Internetworking in the V-System
D.R.Cheriton
8th Data Communications Symposium, Massachusetts, October 1983.

[Comer 83]
CSNET Protocol Software : The IP-to-X25 interface
D.E.Comer, J.T.Korb
SIGCOMM 83, Austin, Texas March 1983.

[Cooper 83]
Writing Distributed Programs with Courier
E.C.Cooper
University of California, Berkeley, March 1982.

[Danthine 82]
Network Interconnection
A.S.Danthine
Symposium on local computer networks, IFIP, Florence, April 1982.

[ECMA/14 82]
Network Layer Principles
ECMA, Technical Report 13, September 1983.

[ECMA/21 84]
Local area networks interworking units for distributed systems
ECMA, Technical Report 21, March 1984.

[ECMA 87]
Basic Remote Procedure Call (RPC) using OSI remote operations
ECMA, Draft 1, March 87.

[François 83]
Some Methodes for Providing OSI Transport in SNA
P.François, A.Potocki
IBM Journal of Research and Development, Vol.27, September 1983.

[Gien 79]
Design Principles for Network Interconnection
M.Gien, H.Zimmerman
Proceeding of 6th Data Communications Symposium
Pacific Grove, California, November 1979.

[Green 86]
Protocol Conversion
P.E.Green
IEEE Transactions on Communications, Vol.34, No.3, March 1986.

[Groenbaek 86]
Conversion between the TCP and ISO Transport Protocols as a Methode for Achieving Interoperability between Data Communication Systems
I.Groenbaek
IEEE J. Select Areas Communications Vol. SAC-4, No.2, 1986.

[Haj Houssain 86]
OSI server for LAN's
S.Haj Houssain, C.Roisin
IFIP International Symposium on LAN and PBX, Toulouse, 1986.

[Haj Houssain 87]
DOSIS : A new approach for interconnecting distributed systems
S.Haj Houssain
EFOC/LAN 87, Basel, June 1987.

[Hawe 84]
An Achitecture for Transparently Interconnecting IEEE 802 Local Area Networks
B.Hawe et al.
IEEE Standards Activity Documents, 1984.

[ISO/IS 7498]
Open System Interconnection - Basic Reference Model
IS 7498, 1984.

[ISO/DIS 8348]
Network Service Definition
DIS 8348, July 1985.

[ISO/DIS 8648]
Internal Organisation of the Network Layer
DIS 8648, February 1986.

[ISO/DIS 8208]
X25 Packet Level Protocol for DTE
DIS 8208, 1984.

[ISO/DIS 8473]
Protocol for providing The Connectionless Mode Network Service
DIS 8473, 1985.

[ISO/DIS 8073]
Transport protocol specification
DIS 8073, September 1984.

[ISO/DP 8348/DAD1]
Addendum to the Network Service Definition covering Connectionless mode transmission
DP 8348/DAD1, October 1983.

[ISO/DP 8348/DAD2]
Addendum to the Network Service Definition covering Network Layer addressing
DP 8348/DAD2, October 1984.

[ISO/DP 8473/DAD1]
Protocol for provoding The Connectionless Mode Network Service - Provision of the underlying service assumed by DIS 8473
DP 8473/DAD1, 1985.

[ISO/DP 8881]
Use of the X25 Packet Level Protocol in local area networks
DP 8881, 1985.

[ISO/DP 8880]
Specification of protocols to provide and support the OSI network service
DP 8880/1, DP 8880/2, DP 8880/3, 1985.

[ISO/DP 8878]
Use of X25 to provide the OSI Connection-Oriented Network Service
DP 8878, 1985.

[ISO/DP 8571]
File Transfer, Access, and Management Protocol
DP 8571, Avril 1984.

[Juanole 86]
On Architectures for Internetworking
G.Juanole, A.Onodi
IEEE Computer Networking Symposium, Washington 1986.

[Juanole 87]
Sur l'interconnexion des réseaux hétérogènes
G.Juanole, A.Onodi
4eme Congrès De Nouvelles Architectures pour les Communications
Paris, 1987.

[Jones 86]
Mach and Matchmaker : kernel and language support for object-oriented distributed systems
M.B.Jones, R.F.Rashid
SIGPLAN Notices, Vol.21, November 1986.

[Langlois 85]
Une architecture de réseau local OSI et son implantation sous UNIX
S. Langlois
Thèse de 3ème cycle - Université de Paris 6, 1985.

[Liskov 84]
Overview of the Argus Language and System
B.Liskov
Programming Methodology Group, memo 40, MIT, 1984.

[Lyon 84]
Overview of the Sun Network File System
B.Lyon et al.
SUN Microsystems Inc.,October 1984.

[Nelson 81]
Remote Procedure Call
B.J.Nelson
PHD, Report No CMU-CS-81-119
Carnegie-Mellon University, 1981.

[Postel 80]
Internetwork Protocol approaches
J.B.Postel
IEEE Transactions on Communications, Vol. 28, No 4, April 1981.

[Postel 81]
The ARPA Internet Protocol
J.B.Postel, C.A.Sunshine, D.Cohen
Computer Networks 5, 1981.

[Pujolle 85]
Réseaux et Télématique
G.Pujolle et al.
Eyrolles, 1985.

[Rashid 81]
Accent, A Communication Oriented Network Operating System Kernel
R.Rashid, G.Robertson
Proc. of the 8th ACM SOSP/SIGOPS, 1981.

[RFC 791]
Internet Protocol
DARPA Internet Program
September 1981.

[RFC 823]
The DARPA Internet Gateway
R.Hinden, A.Sheltzer
Bolt Beranek and Newman Inc., Mass., September 1982.

[RFC 904]
Exterior Gateway Protocol formal specification
D.L.Mills
April 1984.

[RFC 792]
Internet Control Message Protocol
DARPA Internet Program
September 1981.

[ROSE 85]
ROSE Technical Specifications
Version 2, 1985.

[Seret 85]
Evaluation de performances des protocoles de communication - application aux réseaux locaux et aux transmissions par satellite
D.Seret
Thèse de Doctorat d'Etat, Université de Paris 6, 1985.

[Shoch 78]
Inter-network naming, addressing and routing
J.F.Shoch
Proc. COMPCON, 1978.

[Shoch 79]
Packet fragmentation in inter-network protocols
J.F.Shoch
Computer Networks, No.3, 1979.

[Sunshine 80]
Interconnection of computer networks
C.A.Sunshine
Computer networks, No.1, 1977.

[Svobodova 87]
The role of OSI in Distributed Computing
L.Svobodova
7th International Conference on Distributed Computing Systems, Berlin, 1987.

[Tanenbaum 81]
Computer Networks
A.S. Tanenbaum
Printice Hall 1981.

[Tanenbaum 85]
Distributed operating systems
A.S.Tanenbaum, R.V.Resse
Computing Surveys, Vol.17, No.4, December 1985.

[Walker 83]
The LOCUS Distributed Operating System
Walker et al.
ACM-SGOPS, Vol.17, No.5, 1983.

[Zimmerman 81]
Basic concepts for the support of distributed systems: the CHORUS approach
H.Zimmerman et al.
Proc of the 2nd IEEE International Conference, Versailles, 1981.

[Zoline 85]
An approach for interconnecting SNA et XNS networks
K.O.Zoline, W.P.Lidinsky
Proc. of the 9th Data Communications Symposium
Canada, September 1985.

RESUME

Nous nous sommes intéressés dans cette thèse au problème d'ouverture des systèmes distribués au monde extérieur. Cette ouverture est assurée au moyen d'un service de communication externe. Ce service permet aux applications s'exécutant dans le système distribué local de communiquer à plusieurs niveaux de protocoles avec d'autres applications s'exécutant sur des systèmes distants distribués ou centralisés.

L'examen des caractéristiques de la communication externe permet de constater que les services et les protocoles de communication OSI sont adaptés à cette communication. Le modèle OSI peut donc être considéré comme un modèle pour la communication entre les systèmes distribués.

Assurer la communication entre les systèmes distribués implique l'interconnexion de leurs systèmes de communication respectifs qui doivent être supposés totalement hétérogènes. L'analyse des techniques d'interconnexion basées sur l'encapsulation ou sur la conversion montre leurs limites pour assurer une solution satisfaisante. Nous proposons donc une nouvelle approche pour assurer la communication entre les systèmes distribués. Cette approche est basée sur un serveur OSI distribué (DOSIS). Ce serveur permet l'utilisation des systèmes de communication spécifiques à l'intérieur de chaque système distribué, tout en donnant à l'extérieur une vision de ce dernier qui est celle d'un système OSI unique.

L'architecture et le fonctionnement de DOSIS ont été définis pour permettre l'accès et le partage d'une instance unique des couches OSI dans le système distribué ainsi que la possibilité de distribuer ces couches sur plusieurs sites. Chaque utilisateur a l'impression de supporter l'ensemble des couches OSI sur son site local.

L'intérêt fondamental de l'approche DOSIS réside dans la séparation entre les protocoles de communication interne et les protocoles de communication externe dans un système distribué. Ainsi cette approche est applicable quel que soit le niveau de communication externe sans imposer de contraintes particulières sur le service de communication interne.

MOTS CLES

systèmes distribués, ouverture, communication, serveur de communication, modèle OSI, interconnexion de réseaux, passerelle.