

## Projet de Réseaux

### Mise en œuvre d'un service connecté de réseau

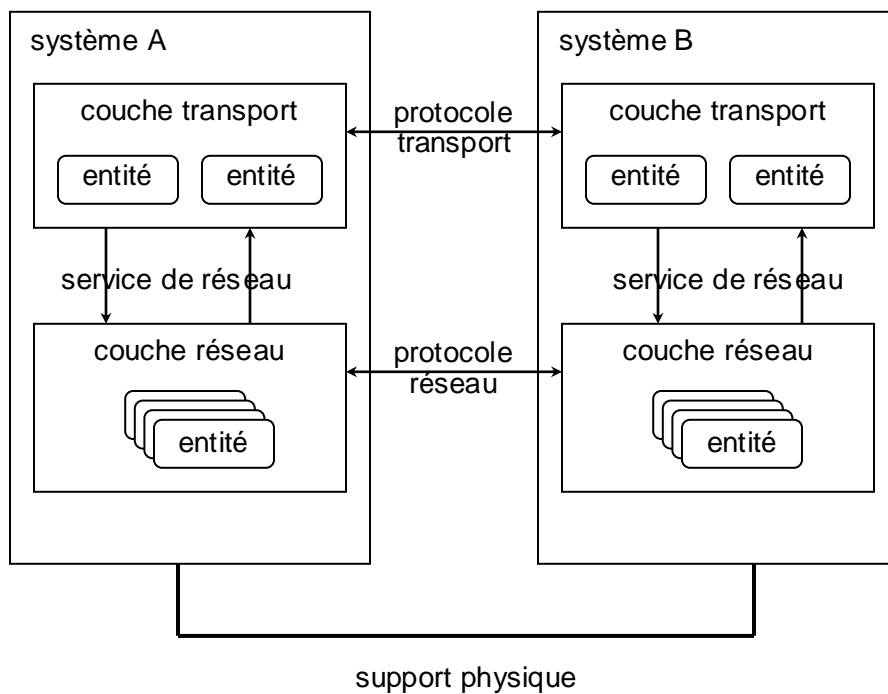
#### PRÉSENTATION GÉNÉRALE

Les étudiants étudieront le texte suivant et le mettront en œuvre sur machines UNIX. Pour effectuer cette simulation, ils pourront se regrouper par 4 ou 5, et rédigeront un dossier de quelques pages décrivant la manière de mettre en œuvre cette simulation et contenant les programmes, les fichiers d'essai et de résultats commentés. À la fin, ils feront une démonstration de fonctionnement.

Le projet sera sanctionné par une note qui représentera **30 %** de la note totale.

#### 1 Présentation du service de réseau considéré pour le projet

Soit la configuration suivante :



Une entité de couche transport du système A voulant communiquer avec son entité homologue de couche transport du système B applique les règles définies par le protocole de transport. Celui-ci se base sur le service fourni par la couche réseau. Nous choisissons un service de réseau en mode connecté : une communication se déroule en trois phases, établissement de connexion, transfert de données et libération de connexion. Pour fournir ce service de réseau, la couche réseau met en œuvre le protocole réseau.

Le fournisseur du service de réseau (i.e. la couche réseau) et l'utilisateur de ce service (i.e. la couche transport) interagissent au moyen de primitives de service. Comme le service de réseau est en mode connecté, il y a trois groupes de primitives, un pour chaque phase. L'ensemble de ces primitives est décrit ci-après.

Phase	Primitive	Paramètres
Établissement de connexion de réseau	demande de connexion de réseau <b>N_CONNECT.req</b>	adresse source, adresse destination
	indication de connexion de réseau <b>N_CONNECT.ind</b>	adresse source, adresse destination
	réponse à une demande de connexion de réseau <b>N_CONNECT.resp</b>	adresse en réponse
	confirmation de connexion de réseau <b>N_CONNECT.conf</b>	adresse en réponse
Transfert de données	demande de transfert de données de réseau <b>N_DATA.req</b>	données utilisateur du service de réseau
	indication de transfert de données de réseau <b>N_DATA.ind</b>	données utilisateur du service de réseau
Libération de connexion de réseau	demande de libération de connexion de réseau <b>N_DISCONNECT.req</b>	adresse en réponse
	indication de libération de connexion de réseau <b>N_DISCONNECT.ind</b>	adresse en réponse, raison

## 1.1 L'établissement de connexion de réseau

Ce service est de mode confirmé, et est donc constitué de quatre primitives, requête, indication, réponse et confirmation. L'utilisateur A de service de réseau utilise la primitive de requête pour demander au fournisseur du service de réseau d'établir une connexion de réseau. Celui-ci va alors dialoguer avec son entité homologue du système B via le protocole réseau et cette entité va adresser une primitive d'indication à l'utilisateur B de service de réseau. Dans le cas où B accepte la connexion, il répond par la primitive de réponse de demande de connexion au fournisseur de service de réseau et ce dernier adresse alors à l'utilisateur A une primitive de confirmation de connexion de réseau (Figure 1).

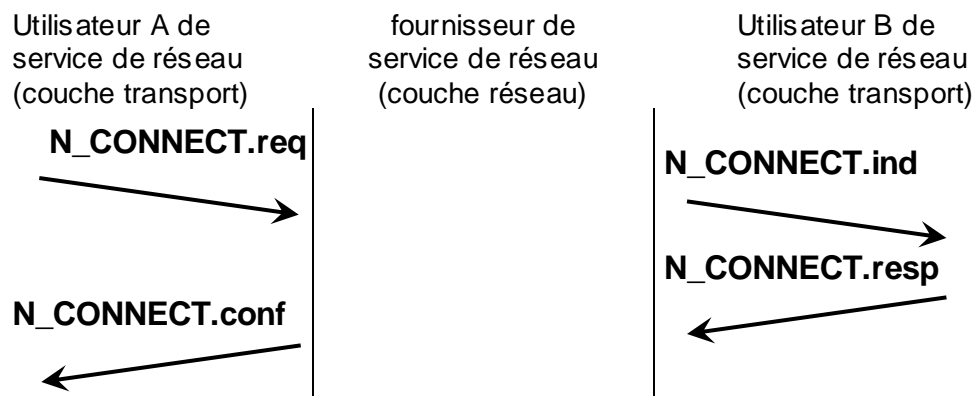


Figure 1: Établissement de connexion réussie

L'utilisateur distant B, ou le fournisseur, peuvent refuser d'établir la connexion (manque de ressources, mauvais fonctionnement local, ou autre raison). Les figures 2 et 3 illustrent un refus d'établissement de connexion de la part de l'utilisateur, respectivement du fournisseur.

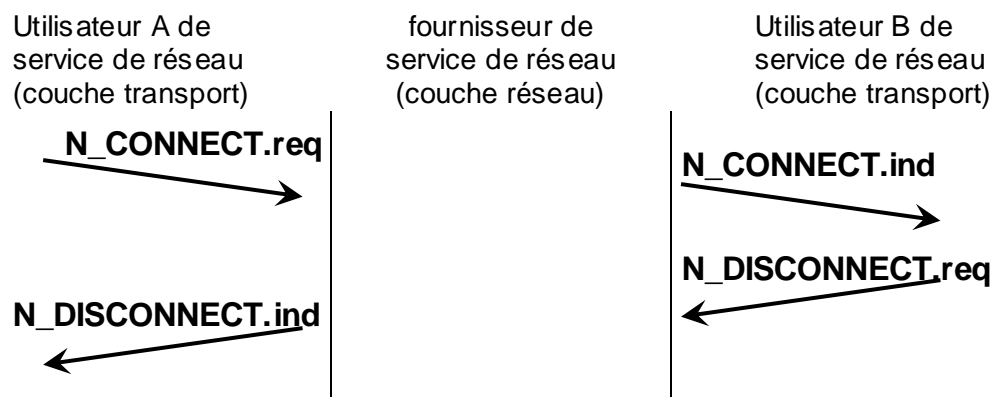


Figure 2 : Refus d'établissement de connexion par l'utilisateur distant

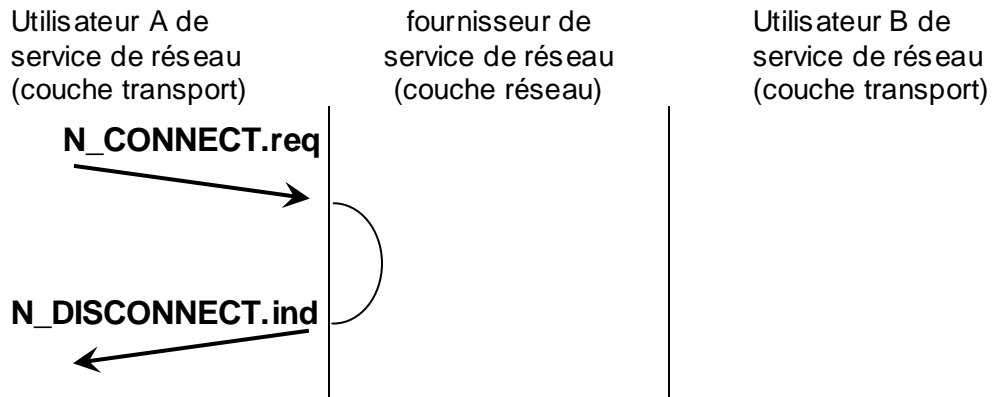


Figure 3 : Refus d'établissement de connexion par le fournisseur

## 1.2 Le transfert de données

Le transfert de données est un service non confirmé. Les primitives utilisées sont donc au nombre de deux : N\_DATA.req et N\_DATA.ind (Figure 4)

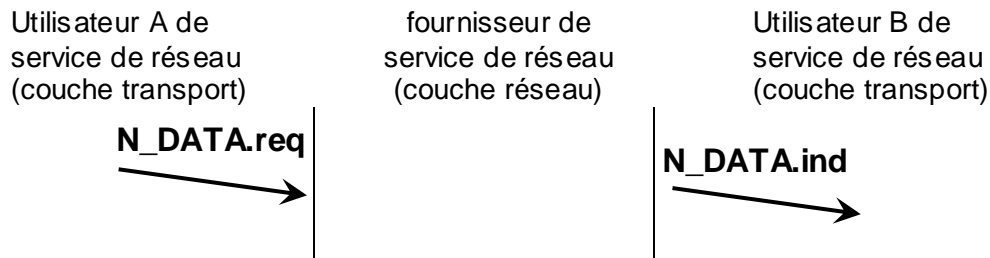


Figure 4: Transfert de données

## 1.3 La libération de connexion

La libération de connexion est un service non confirmé. Il y a donc deux primitives associées à ce service. La libération peut intervenir sur demande de l'un des deux utilisateurs du service de réseau (Figure 5), ou bien sur demande du fournisseur de service (Figure 6).

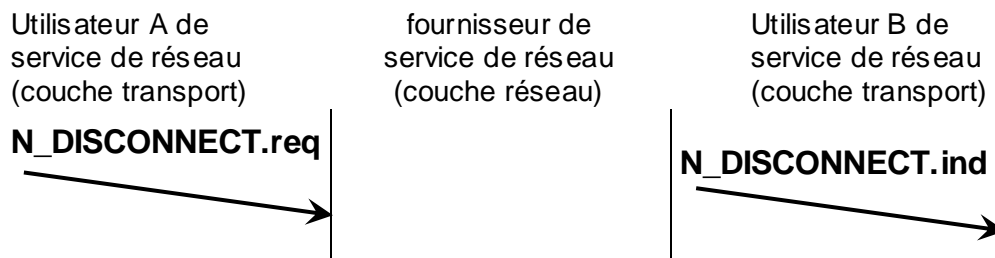


Figure 5 : Libération de connexion par l'un des utilisateurs

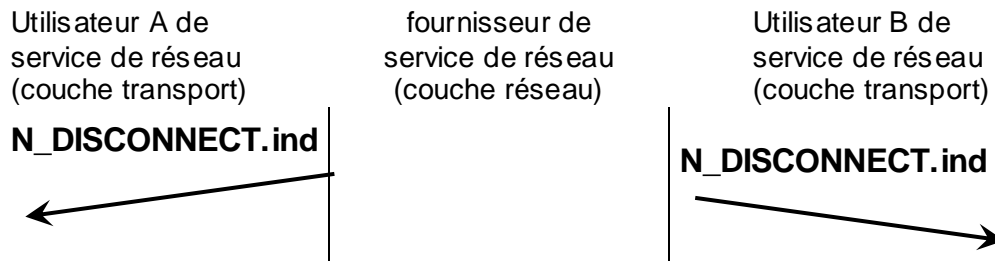


Figure 6 : Libération de connexion par le fournisseur

## 2 Présentation du protocole réseau considéré pour le projet

Le protocole réseau réalise le service de réseau. Il permet l'échange d'unités de données de protocole réseau (NPDU — Network Protocol Data Unit) entre entités homologues de couche réseau.

Les NPDU, appelées paquets, sont classées en trois familles. Il y a les paquets utilisés lors de la phase d'établissement, les paquets utilisés pour le transfert de données et ceux utilisés lors de la phase de libération.

Les actions décrites dans les paragraphes suivants sont celles mises en œuvre par les entités de la couche réseau **côté appelant**. De même pour les paquets. D'autre part, seul l'appelant émet des données.

### 2.1 La phase d'établissement

Lorsqu'un utilisateur de service de réseau utilise la primitive de requête d'établissement de connexion, le fournisseur du service de réseau, donc une entité de la couche réseau, met en œuvre les actions suivantes :

- attribution d'un numéro de connexion logique à la demande. Ce numéro sera ensuite utilisé dans tous les paquets pour identifier la connexion;
- construction d'un paquet d'appel;
- choix d'une route en fonction de l'adresse destinataire;
- utilisation du service de liaison pour que la couche liaison achemine le paquet;
- récupération via la couche liaison de la réponse à la demande d'établissement de connexion, qui peut être :
  - un paquet de communication établie, si l'entité réseau homologue distante a répondu positivement à la demande d'établissement de connexion;
  - un paquet d'indication de libération si l'entité distante (ou le fournisseur de service de réseau) a répondu négativement à la demande d'établissement de connexion;
  - selon la réponse, utilisation de la primitive **N\_CONNECT.conf** ou **N\_DISCONNECT.ind** pour prévenir l'utilisateur du service de réseau.

### Format des paquets

<b>Paquet d'appel</b>	8	1	<b>Paquet de communication établie</b>	8	1
	numéro connexion			numéro connexion	
	type paquet			type paquet	
	adresse source			adresse source	
	adresse destination			adresse destination	

Le type de paquet a la valeur **00001011**.

Le type de paquet a la valeur **00001111**.

<b>Paquet d'indication de libération</b>	numéro connexion	
	type paquet	
	adresse source	
	adresse destination	
	Raison	

Le type de paquet a la valeur **00010011**. La raison peut prendre les valeurs suivantes :

- **00000001** : le disant refuse la connexion (absent, occupé, manque de ressources);
- **00000010** : le fournisseur de service de réseau refuse la connexion.

## 2.2 La phase de transfert de données

Lorsque l'utilisateur de service de réseau est prévenu que la connexion est établie, il peut alors utiliser la primitive de requête de transfert de données pour que le fournisseur du service de réseau achemine ses informations. L'entité de la couche réseau alors met en œuvre les actions suivantes :

- identifier le numéro de connexion concernée par la demande;
- segmenter en plusieurs paquets si c'est nécessaire;
- construire le (ou les) paquet(s) de données;
- utilisation du service de liaison pour acheminer le (ou les) paquet(s);
- attente d'un paquet d'acquittement entre l'émission de chaque paquet (la fenêtre d'anticipation est de 1). Si le paquet d'acquittement n'arrive pas (expiration d'un temporisateur) ou si c'est un paquet d'acquittement négatif, réémettre le paquet concerné. Il n'y aura qu'une tentative de réémission.

### Format des paquets

<b>Paquet de données</b>	8	1
	numéro connexion	
	type paquet	
	données	

Le champ "type de paquet" est de la forme :

8	7	6	5	4	3	2	1
p(r)			M	p(s)			0

**p(s) et p(r)** sont les numéros de séquence des paquets représentant respectivement le numéro du paquet envoyé et le numéro du prochain paquet attendu en réception. La numérotation est faite modulo huit (sur 3 bits).

Le **bit M** est lié à la **segmentation**. Il y a segmentation lorsque la taille du message délivré par la couche transport est supérieure à la taille maximale du champ de données d'un paquet (128 octets). Dans ce cas, plusieurs paquets sont nécessaires pour transférer le message. Le bit M est à 1 pour indiquer qu'un paquet est une partie d'un message. Il est à 0 pour indiquer que le paquet est le dernier (ou l'unique) segment du message.

Le champ de données d'un paquet est limité à 128 octets.

	8	7	6	5	4	3	2	1		8	7	6	5	4	3	2	1
<b>Paquet d'acquittement</b>	numéro connexion								<b>Paquet d'acquittement négatif</b>	numéro connexion							
	p(r)		0	0	0	0	0	1		p(r)		0	1	0	0	0	1

Le deuxième octet, type du paquet, contient le numéro p(r) qui désigne le numéro du prochain paquet attendu en réception par le distant.

### 2.3 La phase de libération

Lorsque la libération est initiée par l'utilisateur du service de réseau par l'envoi de la primitive de demande de libération, l'entité de la couche réseau met en œuvre les actions suivantes :

- identification du numéro de connexion concernée par la demande;
- construction du paquet de demande de libération;
- utilisation du service de liaison pour acheminer le paquet;
- fermeture de la connexion, c'est-à-dire libération des ressources qui lui étaient allouées.

#### Format des paquets

	8							1
<b>Paquet de demande de libération</b>	numéro connexion							
	type paquet							
	adresse source							
	adresse destination							

Ce paquet est de même type que celui d'indication de libération (valeur 00010011).

Si la libération est initiée par le fournisseur de service de réseau, l'entité de la couche réseau d'une part utilise la primitive **N\_DISCONNECT.ind** pour prévenir l'utilisateur du service de réseau et d'autre part construit un paquet d'indication de libération qui est envoyé à l'entité distante via l'utilisation du service de liaison. Le format de ce paquet est donné au paragraphe 2.1.

### 3 Travail demandé

On vous demande de programmer en C les actions mises en oeuvre par une entité de couche réseau côté appelant qui réalise le service de réseau. Seul l'appelant émet des données.

#### 3.1 Entités de transport et réseau

Une entité de la couche transport utilisatrice du service de réseau est un processus ET. Une entité de la couche réseau fournissant le service de réseau par utilisation du service de liaison est un processus ER.

Les deux processus ET et ER communiquent par **tubes et par signaux**. Ces communications représentent les échanges de primitives de service de réseau entre les couches transport et réseau.

Les processus ET et ER seront programmés indépendamment et leurs codes chargés par une commande de type *exec*. Il faudra donc constituer un processus d'initialisation qui créera au moyen de la primitive ***fork*** deux processus : l'un représentant ET et l'autre ER. Ce processus d'initialisation aura précédemment créé deux tubes pour la communication entre ET et ER, un pour chaque sens.

Chaque processus devra, d'un côté, communiquer avec la couche supérieure, de l'autre avec la couche inférieure. Les informations représentant la couche supérieure à ET seront *stockées* dans deux fichiers **S\_lec** et **S\_ecr**. Le fichier S\_lec représentera le jeu d'essai et S\_ecr une partie des résultats. Les informations envoyées vers la couche liaison seront écrites dans un fichier **L\_ecr**. Les informations émanant de la couche liaison seront tirées de manière aléatoire (voir paragraphe 3.5). Elles seront également copiées dans un fichier **L\_lec**, afin de permettre de tester le programme.

Les deux processus dans l'état *attente* boucleront sur l'examen (passant) de leurs interfaces supérieures et inférieures.

#### 3.2 Interface entre couches

De manière à ne pas être bloqué sur une lecture, les processus liront le contenu du tube sur réception d'un signal émis par l'écrivain une fois l'information écrite dans le tube. On représentera l'interface entre les 2 couches par 4 fonctions : *lire\_de\_reseau*, *ecrire\_vers\_reseau*, *lire\_de\_transport*, *ecrire\_vers\_transport*, qui concerneront les primitives de service échangées entre les couches. Ces fonctions seront non bloquantes et rendront transparente la réalisation pratique de l'interface programmée au moyen des tubes et des signaux.

Un "message" (au sens écriture en un seul ***write*** sur le tube) est de taille variable et est précédé de la taille du message sur un entier court (2 octets). Les procédures de gestion des signaux liront un "message" d'un unique *read*, les procédures *lire\_de\_...* retourneront soit un code indiquant qu'il n'y a rien sur l'interface, soit la partie du message qui concerne le lecteur.



La simulation sera terminée lorsqu'il n'y aura plus une seule communication en cours, ni aucune demande à traiter. Un processus de terminaison indiquera la fin, tous les processus étant terminés.

### 3.3 Adressage des stations

Le réseau est constitué de  $N$  stations,  $N \leq 255$ . L'adresse d'une station est un entier compris entre 0 et 249. Lors d'une connexion entre deux stations, il est nécessaire de disposer de deux adresses, celle de la station source et celle de la station destinataire. **Ces deux adresses sont déterminées par un tirage aléatoire de deux nombres qui doivent être distincts et compris entre 0 et 249.**

### 3.4 Routage

Le fournisseur de service de réseau détermine le routage du paquet d'appel à l'établissement de la connexion. Le routage consiste à partir d'une station d'adresse source et en fonction de l'adresse de la station destination, à déterminer une route dans le réseau. Pour la station source, cela consiste à identifier une adresse d'acheminement, à partir de la consultation de sa table de routage (routage statique). Les tables de routage des stations sont fixées de la façon suivante :

Pour une station source dont l'adresse résultant du tirage est comprise entre 0 et 99 :

- si l'adresse de destination est comprise entre 0 et 99  
alors l'adresse d'acheminement est égale à l'adresse de destination ;
- si l'adresse de destination est comprise entre 100 et 199  
alors l'adresse d'acheminement est égale à 255 ;
- si l'adresse de destination est comprise entre 200 et 249  
alors l'adresse d'acheminement est égale à 254 ;

Pour une station source dont l'adresse résultant du tirage est comprise entre 100 et 199

- si l'adresse de destination est comprise entre 0 et 99  
alors l'adresse d'acheminement est égale à 250 ;
- si l'adresse de destination est comprise entre 100 et 199  
alors l'adresse d'acheminement est égale à l'adresse de destination ;
- si l'adresse de destination est comprise entre 200 et 249  
alors l'adresse d'acheminement est égale à 253 ;

Pour une station source dont l'adresse résultant du tirage est comprise entre 199 et 249

- si l'adresse de destination est comprise entre 0 et 99  
alors l'adresse d'acheminement est égale à 251 ;
- si l'adresse de destination est comprise entre 100 et 199  
alors l'adresse d'acheminement est égale à 252 ;
- si l'adresse de destination est comprise entre 200 et 249  
alors l'adresse d'acheminement est égale à l'adresse de destination ;

### 3.5 Simulation du service de liaison

L'utilisation du service de liaison par un processus ER pour acheminer les paquets est simulée par des écritures dans le fichier L\_echr. L'envoi d'un paquet à une entité réseau distante par utilisation du service de liaison correspond alors à écrire dans le fichier.

En réception, les paquets provenant du distant peuvent être les suivants :

- paquet de communication établie, réponse positive à une demande de connexion;
- paquet d'indication de libération, refus d'établissement de connexion;
- paquet d'acquittement, positif ou négatif;
- paquet d'indication de libération, rupture de connexion établie. Cette catégorie n'est pas considérée dans la suite.

Les trois types de paquets considérés sont des réponses aux émissions faites par l'entité réseau locale. D'autre part, il peut arriver que ces émissions restent sans réponse. Les réponses reçues ou l'absence de réponse sont déterminées aléatoirement de la façon suivante :

- Pour toute demande de connexion émise par le processus ER :
  - la demande ne reçoit pas de réponse si l'adresse de la station source est un multiple de 19;
  - le paquet reçu en réponse est un refus de connexion de la part du distant si l'adresse de la station source est un multiple de 13;
  - dans les autres cas, la réponse est un paquet d'acceptation.
- Pour tout paquet de données émis :
  - L'émetteur ne reçoit pas de paquet d'acquittement si l'adresse de la station source est un multiple de 15;
  - Le paquet reçu en réponse est un acquittement négatif si le numéro p(s) du paquet émis est égal à un nombre tiré aléatoirement dans l'intervalle (0,7) :
  - Dans les autres cas, la réponse est un paquet d'acquittement positif.

Afin de tester le programme, les réponses provenant de la couche liaison (donc les résultats des tirages aléatoires) seront écrites dans un fichier L\_lec, préfixées par le numéro de la connexion concernée.

### 3.6 Déroulement d'une communication

Cette description est celle perçue et mise en œuvre du côté appelant. Seul l'appelant émet des données.

Suite à une demande de la couche supérieure, une entité de couche transport entre en communication avec une entité homologue. Cela correspond à une lecture du fichier S\_lec par le processus ET, et à l'envoi d'une demande de connexion du processus ET vers le processus ER contenant :

- le numéro de demande, appelé identifiant d'extrémité de connexion réseau (le processus ET gère les numéros des demandes qu'il émet);
- la primitive N\_CONNECT.req. Les paramètres de la primitive sont les adresses source et destination.

Le processus ET gère l'état des connexions en cours. Pour cela, il utilise une table qui pour chaque identifiant d'extrémité de connexion, indique l'état de la connexion. Les deux états possibles sont *en attente de confirmation d'établissement* et *connexion établie*. On suppose que toute demande de connexion reçoit une réponse (pas forcément positive) de la part du processus ER.

Le processus ER, sur une demande de connexion, peut accepter ou refuser (refus de connexion par le fournisseur de service). La décision d'accepter ou de refuser est déterminée aléatoirement selon une procédure analogue à celle du refus de connexion de la part du distant : si l'adresse de la station source est un multiple de 27, le fournisseur refuse la connexion. Dans ce cas, il doit indiquer la libération au processus ET. Celui-ci arrête alors tout traitement pour cette demande, libère les ressources et écrit dans le fichier S\_ecr le résultat.

Si le processus ER accepte, il attribue à la demande un numéro de connexion. Il construit ensuite le paquet d'appel. Il détermine l'adresse à laquelle le paquet doit être acheminé en examinant sa table de routage. Il mémorise les informations nécessaires relatives à cette connexion : numéro de connexion, adresse source, adresse destinataire, adresse d'acheminement, état de la connexion (en cours d'établissement ou établie), identifiant d'extrémité de connexion réseau. Il utilise le service de liaison pour acheminer le paquet d'appel (écriture du paquet dans le fichier L-ecr). Il se met ensuite en attente de la réponse du distant. S'il n'y a pas de réponse ou si celle-ci est un refus de connexion (respectivement une acceptation) (cf. 3.3), il envoie au processus ET une indication de libération de connexion contenant l'identifiant d'extrémité de connexion réseau et la primitive N\_DISCONNECT.ind (respectivement une confirmation d'établissement de connexion contenant l'identifiant d'extrémité de connexion réseau et la primitive N\_CONNECT.conf.). Celui-ci arrête alors tout traitement pour cette demande, libère les ressources et écrit dans le fichier S\_ecr le résultat.

Sur réception d'une primitive N\_CONNECT.conf., le processus ET peut alors modifier l'état de la connexion (connexion établie) et délivrer au processus ER les données à émettre. Pour cela, le processus ET adresse au processus ER une demande de transfert de données contenant l'identifiant d'extrémité de connexion réseau et la primitive DATA.req.

Le processus ER construit le paquet de données correspondant (voire les paquets de données si une segmentation est nécessaire) et l'envoie via le service de liaison (écriture dans le fichier L-ecr). La réponse reçue (ou l'absence de réponse) est déterminée selon la procédure décrite au paragraphe 3.3.

Lorsque le processus ET veut libérer la connexion (le cas de la libération par l'entité distante ne sera pas traité), il envoie au processus ER une demande de libération de connexion contenant l'identifiant d'extrémité de connexion réseau et la primitive N\_DISCONNECT.req. Le processus ER utilise le service de liaison pour adresser au distant un paquet d'indication de libération. Les deux processus peuvent alors libérer les ressources qui étaient utilisées pour la gestion de la connexion.