

LA COUCHE LIAISON

Objectif de la couche liaison de donnée

Le but de cette couche est de faire communiquer deux machines distantes reliées par un canal de transmission de type coaxial, ou ligne téléphonique, etc... Cette couche fait le lien entre la couche réseau et la couche physique et apporte un service liaison qui est rendu par un protocole de niveau 2. Cette couche réalise un certain nombre de fonctions spécifiques :

- ☞ Traitement contre les erreurs de transmissions
- ☞ Contrôle de flux pour régulariser la quantité de données échangées
- ☞ Gestion de liaison

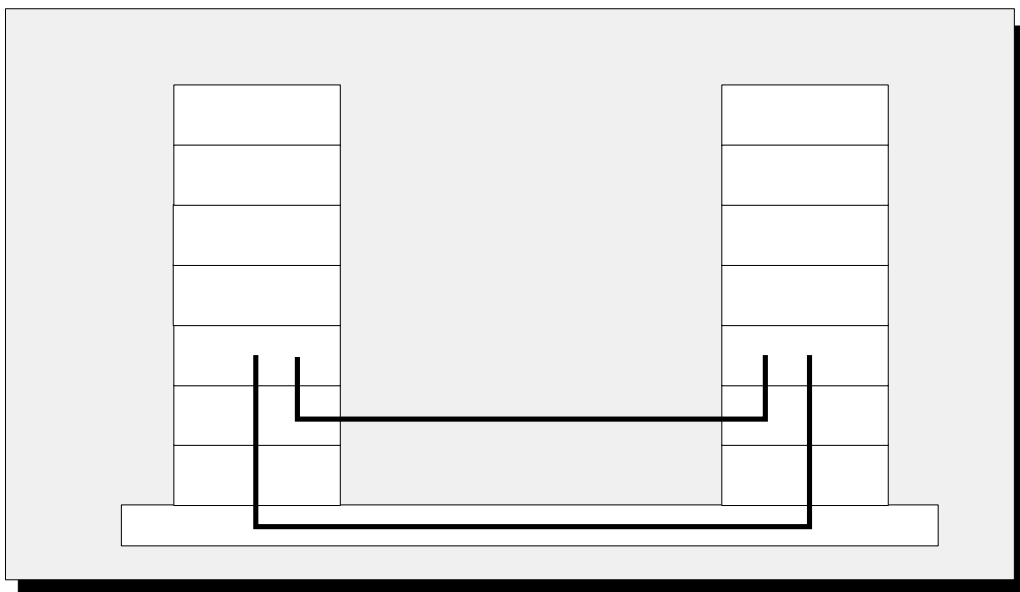


Fig.1

La communication entre les couches Réseau et Liaison se fait par l'intermédiaire de primitives de service. Elles représentent :

la demande : utilisée par la couche réseau pour demander à la couche liaison de faire quelque chose (envoi d'une trame de donnée, de gestion etc...)

la réponse : utilisée pour répondre à une indication.

l'indication : pour informer la couche réseau qu'un événement vient de se produire (une demande de connexion ou de déconnexion vient d'être faite par une autre machine etc...)

la confirmation : utilisée par la couche réseau pour connaître le succès ou l'échec d'une demande faite précédemment.

Ces primitives sont illustrées par le schéma suivant de deux manières différentes (Fig. 2)

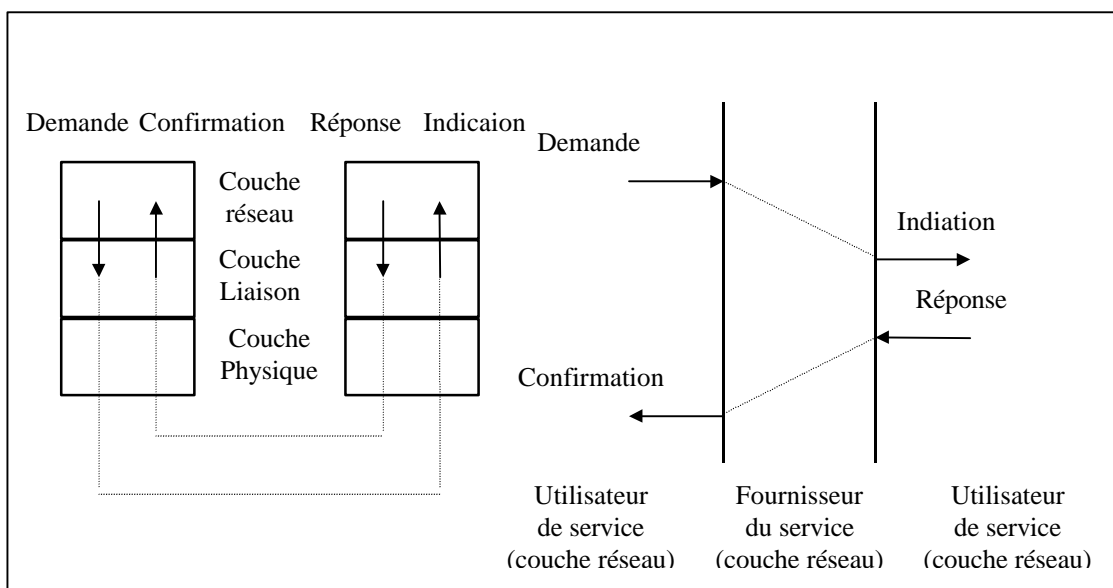


Fig. 2

Protection contre les erreurs

Nous avons vu dans le chapitre précédent que les lignes de transmission n'étaient pas parfaites et que des erreurs pouvaient être introduites sur les symboles d'une suite binaire. Les origines de ces erreurs peuvent être recensées et dépendent :

- du type de relation (locale, inter urbaine)
- du nombre de répéteur
- du support de transmission
- du débit, du type de codage, modulation ...

Ce taux varie de 10^{-7} à 10^{-9} .

L'objet de cette partie est de décrire les méthodes couramment utilisées pour protéger les informations émises des erreurs de transmission. Ces méthodes mettent en œuvre un matériel placé entre la source, qui délivre les informations, et un collecteur qui reçoit une suite de symboles binaires.

La stratégie d'utilisation du codeur et du décodeur dépend de l'ensemble du système de transmission de données.

- Cette stratégie d'utilisation est une simple détection lorsque les symboles sont restitués au collecteur avec une alarme dès que le décodeur détecte une erreur. Cette erreur peut faire l'objet d'une correction d'erreurs mais deux cas apparaissent :

- Le décodeur corrige automatiquement certaines erreurs Correction d'erreur directe.
- Le décodeur ne peut que détecter les erreurs et il est donc nécessaire de retransmettre une partie des données pour réaliser la correction. Cette stratégie est une correction par retransmission ARQ.

Il existe 3 types différents :

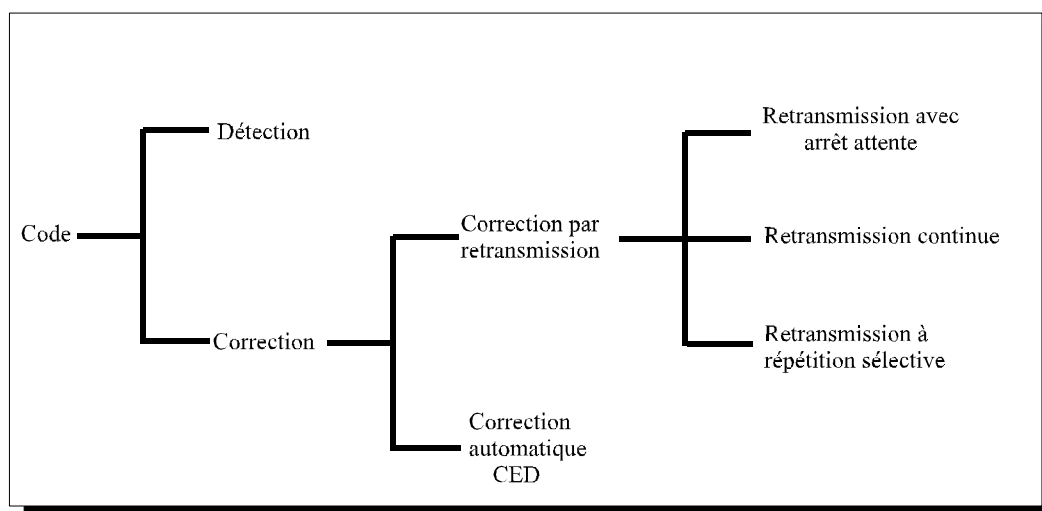


Fig. 3

Classification

Nous pouvons considérer une suite binaire découpée en blocs de k bits consécutifs et à chacun de ces blocs le codeur associe r bits appelés bits de contrôles. Ce type de codage est dit systématique :

Si les bits de contrôle sont calculés uniquement à partir des bits d'informations, le code est appelé code par bloc. Si les bits de contrôles dépendent des blocs précédents le code est dit convolutionnel ou récurrent. Dans la suite nous ne développerons que les codes en blocs.

Méthodes simples de détection d'erreur

le mode Echoplex

Ce mode permet dans le cas d'un mécanisme de transmission le plus élémentaire d'échanger des informations entre deux machines informatiques avec un taux d'erreur nul. Lors d'une communication, les caractères transmis d'une machine vers une autre sont retransmis à l'arrivée vers la machine émettrice. De ce fait, si une erreur se produit dans la transmission ou la retransmission le caractère en erreur est automatiquement détecté et retransmis. Le mécanisme des échanges est montré par la figure ci-dessous.

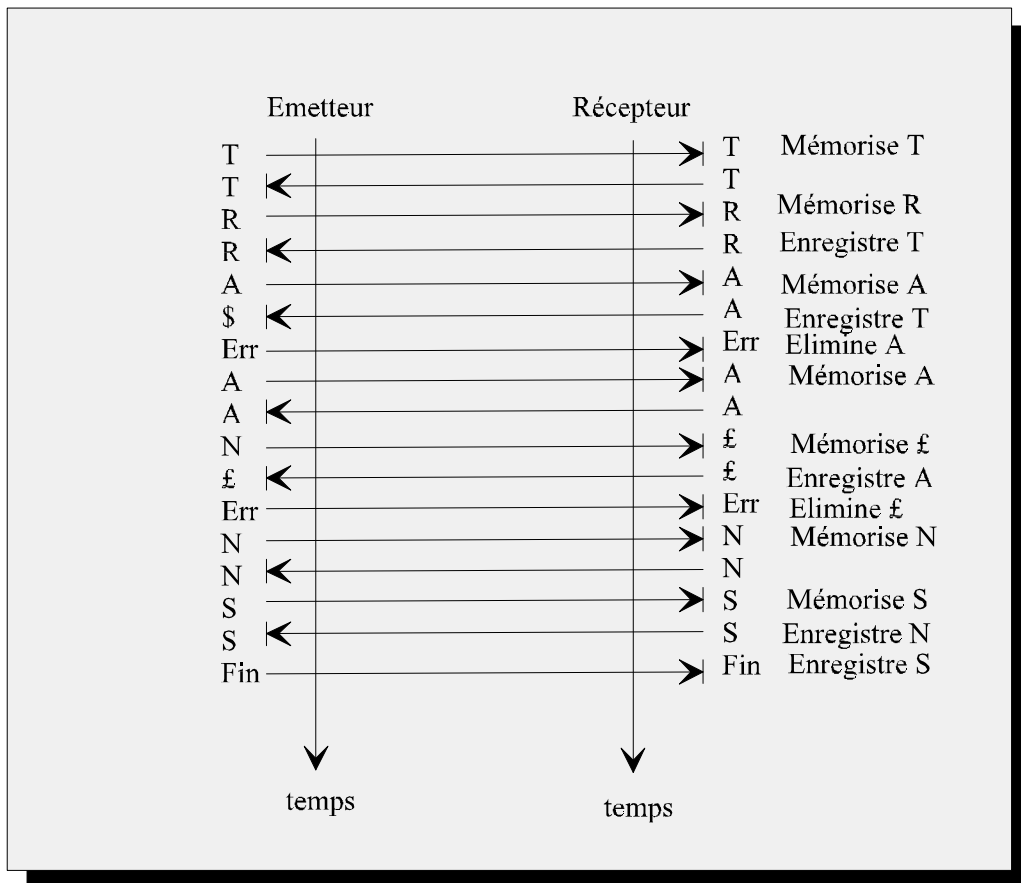


Fig. 4

La détection d'erreur par vérification de la parité par caractère

Dans ce mode de fonctionnement, les caractères transmis sont automatiquement codés avec un bit de contrôle relevant d'une parité paire ou impaire.

Dans le cas d'une parité paire, ce bit est positionné à 0 si le nombre d'éléments binaires positionnés au niveau 1 dans le mot est paire et il est positionné à 1 si ce nombre est impaire.

Exemple : En utilisant un code 7 bits du CCITT (code n° 5), on peut utiliser le 8^{ième} bit comme bit de parité.

	Suite binaire sur 7 bits	Parité paire	parité impaire
A	100001	0 100001	1 100001
E	101001	1 101001	0 101001
V	0110101	0 0110101	1 0110101

Cette solution permet de détecter une erreur simple sur chacun des mots transmis, mais dès que ce nombre est excédé il est impossible de détecter des erreurs avec certitude.

Cette méthode ajoute 1 bit tous les 7 bits transmis. Le rendement de la transmission est donc de :

$$h = \frac{7}{8} = 87,5 \%$$

exemple :

	Sans erreur	Avec 1 erreur	Avec 2 erreurs	Avec 3 erreurs
On veut transmettre A	1000001	1000001	1000001	1000001
Calcul du bit de parité	0 1000001	0 1000001	0 1000001	0 1000001
Transmission	0 1000001	0 1000001	0 1000001	0 1000001
Réception	0 1000001	0 1010001	0 1010000	0 1001111
Suppression du bit de parité	1000001	1000001	1000001	10001111
Calcul le bit de parité	0 1000001	1 1000001	0 1010000	0 1001111
Les nombres sont égaux	OK	Erreur	OK	Erreur

Une amélioration à ce système peut être apportée en transmettant tous les n mots un mot de contrôle sur les n précédents : Contrôle de parité vertical.

A	1000001	0 1000001
B	1000010	0 1000010
C	1000011	1 1000011
D	1000100	0 1000100
E	1000101	1 1000101
F	1000110	1 1000110
G	1000111	0 1000111
H	1001000	0 1001000
Contrôle	0001000	10001000

Fonctionnement de cette méthode :

On transmet le n + 1 mot du bloc
01000001 01000010 11000011 01000100 11000101 11000110 01000111 01001000 10001000

A la réception on supprime tous les tests de parité
 1000001 1000010 1000011 1000100 1000101 1000110 1000111 1001000

et on les recalcule.
01000001 01000010 11000011 01000100 11000101 11000110 01000111 01001000 10001000
 Si les mots reçus sont en tout point identique à ceux recalculés il n'y a pas d'erreur sinon il y a erreur.

0 erreur	1 erreurs	2 erreurs	2 erreurs	2 erreurs
01000001	01000001	01000001	01000001	01000001
01000010	1 1010010	1 1010010	01000010	01000010
11000011	11000011	11000011	11000011	11000011
01000100	01000100	01000100	01000100	01000100
11000101	11000101	0 1000100	11000101	11000101
11000110	11000110	11000110	11000110	11000110
01000111	11000111	11000111	01000111	01000111
01001000	01001000	01001000	01001000	01001000
10001000	10011000	10011000	10001000	10001000

Cette méthode ajoute 1 bit tous les 7 bits transmis et 8 bits tous les 8 mots transmis. Le rendement de la transmission est donc de :

$$h = \frac{7 * 8}{(7 + 1) * 8 + 8} = 77,7\%$$

Remarque : dans le cas d'une détection avec parité PAIRE le LRC de l'octet de contrôle du bloc (le VRC) vérifie à la fois l'octet VRC et Les différents LRC.

Dans le cas d'une parité IMPAIRE il n'y a pas de correspondance.

Définitions générales

- Opérations

$$\begin{array}{cccc} 0 + 0 = 0 & 0 + 1 = 1 & 1 + 0 = 1 & 1 + 1 = 1 \\ 0 \cdot 0 = 0 & 0 \cdot 1 = 0 & 1 \cdot 0 = 0 & 1 \cdot 1 = 1 \end{array}$$

- Rendement

$$R = \frac{k}{k+r}$$

- $n = k + r$

n représente la longueur du mot du code

k représente l'information réellement utile

r représente l'information de contrôle

Il existe donc dans un code 2^k mots différents composés de n bits.

Ces codes sont généralement noté Code(n,k) avec $r = n-k$.

- Le décodage : Opération qui permet d'associer à chaque vecteur reçu un mot de code unique.

- Poids de Hamming : Le poids de Hamming d'un vecteur représente le nombre de 1 qu'il contient.

- Distance de Hamming : C'est le nombre de bits en lesquels 2 mots d'un code diffèrent.

- Le poids minimum : c'est le plus petit des mots non nul.

Les codes linéaires :

Codes pour lesquels les bits de contrôle dépendent linéairement des bits d'informations.

A tout vecteur d'information i de k bits

$$i = (i_1, i_2, i_3, \dots, i_k), i_j \in \{0,1\}$$

nous associons un mot de code c de longueur n

$$c = \{ i_1, i_2, \dots, i_k, c_{k+1}, \dots, c_n \} \quad c_j \in \{0,1\}$$

par la transformation linéaire suivante

$$c = i \cdot G \text{ avec } G = [I_k, P]$$

I_k : Matrice identité de dimension k

P matrice de k lignes et n-k colonnes

L'ensemble C des 2^k mots de code c est appelé code linéaire.

La matrice G est la matrice génératrice de ce code.

On dit que c est un mot du code si et seulement si

$$cH^t = 0 \text{ avec } H = [P^t, I_{n-k}]$$

Exemple : Code(7,4) de matrice génératrice :

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (I_k P)$$

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = (P^t I_{n-k})$$

$$(i_1 \ i_2 \ i_3 \ i_4 \ c_5 \ c_6 \ c_7) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (i_1+i_2+i_3+c_5 \ i_2+i_3+i_4+c_6 \ i_1+i_2+i_4+c_7)$$

$$c_5 = i_1 + i_2 + i_3$$

$$c_6 = i_2 + i_3 + i_4$$

$$c_7 = i_1 + i_2 + i_4$$

Tableau représentant les mots du code :

c_7	c_6	c_5	i_4	i_3	i_2	i_1
0	0	0	0	0	0	0
0	0	1	0	0	0	1
1	1	1	0	0	1	0
0	1	0	0	0	1	1
0	1	1	0	1	0	0
0	1	0	0	1	0	1
1	0	0	0	1	1	0
0	0	1	0	1	1	1
1	1	0	1	0	0	0
0	1	1	1	0	0	1
0	0	1	1	0	1	0
1	0	0	1	0	1	1
0	0	1	1	1	0	0
1	0	0	1	1	0	1
0	1	0	1	1	1	0
1	1	1	1	1	1	1

Propriétés

Syndrome

Soit un vecteur x de longueur n

$$x = (x_1 \ x_2 \ \dots \ x_k \ \dots \ x_n)$$

$$S(x) = x \cdot H^t$$

Un vecteur x est un mot du code si et seulement si son syndrome est nul.

$$x \in C \Leftrightarrow S(x) = 0$$

La distance minimum entre les mots du code conditionne la capacité de correction d'un code. Les propriétés suivantes permettent de connaître cette distance minimum.

La distance minimum entre les mot d'un code linéaire est égale au poids minimum de ces mots.

Dans un code linéaire de distance minimum m il existe au moins un ensemble de m colonne de la matrice H dont la somme est égale à 0.

Aucun sous ensemble de moins de m colonnes de H n'a de somme égale à 0.

Si c est un mot de code de poids min m $c \cdot H^t = 0$

$S(x) = 0 \Rightarrow x$ est un mot du code

$S(x) \neq 0 \Rightarrow$ une erreur s'est produite

on peut déclencher une alarme

on peut réaliser une correction par retransmission

on peut corriger l'erreur directement

Dans cette dernière solution on choisit comme vecteur d'erreur un vecteur de syndrome $S(x)$ de poids minimal.

Tableau standard associé à un code.

On considère un code linéaire (n,k) . Il existe 2^{n-k} syndromes auxquels nous associons la classe des vecteurs ayant ce syndrome.

En 1er ligne, nous avons tous les mots de code positionné en ordre croissant.

En première colonne sont répertoriées toutes les erreurs de syndrome de $2n-k$ en commençant par 0 et en les mettant dans l'ordre croissant des erreurs par colonne.

Exemple : Soit un code $(4,2)$ avec pour matrice génératrice $G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$

$$H = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \quad H^t = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$(i_1 \ i_2 \ c_3 \ c_4) H^t = (i_1 \ i_2 \ c_3 \ c_4) \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = 0$$

$$\begin{cases} c_3 = i_2 \\ c_4 = i_1 + i_2 \end{cases}$$

Syndrome	$c_4 c_3 i_2 i_1$	$c_4 c_3 i_2 i_1$	$c_4 c_3 i_2 i_1$	$c_4 c_3 i_2 i_1$	
00	0000	1001	0111	1110	Mots appartenant au code
01	1000	0001	1111	0110	Mots n'appartenant pas au code est de syndrome 01
11	0100	1101	0011	1010	Mots n'appartenant pas au code est de syndrome 11
10	0010	1011	0101	1100	Mots n'appartenant pas au code est de syndrome 10

Ce tableau est un dictionnaire qui est utilisé comme suit :

Dès que l'on reçoit un mot on calcul son syndrome . S'il n'est pas nul, on verifie sa place dans le dictionnaire et on prend la valeur se trouvant sur la première ligne.

Codes cycliques redondants

Ces codes, font appel à une représentation du message par un polynôme. La formule générale d'un message binaire est un polynôme de la forme :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x^1 + a_0 x^0$$

Les coefficients a_n ne peuvent prendre que les valeurs 0 ou 1 en binaire et les x^n sont les puissances successives de 2.

Si nous divisons ce polynôme par un autre polynôme binaire, le reste de la division constitue le caractère de contrôle.

exemple :

Soit à transmettre le message **1011011**.

le polynôme correspondant à la forme :

$$P(x) = x^6 + x^4 + x^3 + x + 1$$

Si le polynôme diviseur est égal à : $x^4 + x + 1$

Ce qui donne en binaire : **10011**

Il faut donc diviser **1011011** par **10011**.

Cette division est exécutée de la façon suivante :

- Nous ajoutons au dividende, et à sa droite, autant de 0 que le diviseur comporte de bit moins un (ici 5-1 -> 4) ce qui donne **10110110000**.

- Puis nous exécutons la division par soustractions successives (soustractions grâce au OU-exclusif) :

$$\begin{array}{r}
 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 1\ 1\ 1 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 0\ 0\ 0 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 0\ 1\ 1\ 0\ 0
 \end{array}$$

- la dernière ligne de la division donnera le reste donc le CRC; c'est lui qui sera ajouté au message que l'on va transmettre.

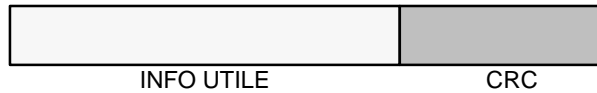


Fig. 5

A la réception, un calcul semblable s'effectue sur le mot reçu, mais il faut, ici, que le reste soit nul. Dans le cas contraire, c'est qu'une erreur est survenue en cours de route.

Pour procéder à la vérification nous effectuons comme suit :

a) sans erreur

$$\begin{array}{r}
 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 1\ 1\ 1 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 0\ 1\ 1 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

b) avec erreur

$$\begin{array}{r}
 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 1\ 1\ 0 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 1\ 1\ 1 \\
 -\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 0\ 0\ 0
 \end{array}$$

Si le reste est différent de 0 , le transfert s'est mal effectué.

Les principaux polynômes diviseurs sont :

LRCC-8 :	$x^8 + 1$
LRCC-16 :	$x^{16} + 1$
CRC 12 :	$x^{12} + x^{11} + x^3 + x^2 + x + 1$

$$\begin{aligned} \text{CRC 16 Forward :} & \quad x^{16} + x^{15} + x^2 + 1 \\ \text{CRC 16 Backward :} & \quad x^{16} + x^{14} + x + 1 \\ \text{CRC CITT Forward :} & \quad x^{16} + x^{12} + x^5 + 1 \\ \text{CRC CITT Backward :} & \quad x^{16} + x^{11} + x^4 + 1 \end{aligned}$$

Le CRC-12 est appliqué aux systèmes à mots de 6 bits.

Le CRC-CITT est le standard européen, pour système à mots de 8 ou 16 bits; il procure des détections d'erreurs au-delà de 12 bits de message avec une fiabilité de 99%.

Les réseaux locaux et réseaux métropolitains

Ces types de réseaux sont de tailles réduites et couvrent un domaine privé. Cette limitation engendre un certain nombre de caractéristiques :

- Les distances sont plus courtes donc le taux d'erreur plus faible,
- les temps de propagation des signaux sont plus élevés,
- les débits peuvent être augmentés.

De manière générale, un réseau local peut être caractérisé par :

- son support de transmission,
- sa topologie,
- sa méthode de contrôle d'accès au support.

L'architecture I.E.E.E. (voir chapitre I)

Le besoin d'une normalisation au niveau des réseaux locaux s'est fait sentir dès les années 1970, et c'est en 1980 que l'I.E.E.E. a créé le comité 802 afin d'élaborer et de proposer des spécifications relatives à un réseau local standard. L'approche adoptée est comme pour le modèle OSI de l'I.S.O. une structure en couche, permettant uniquement à partir des 2 couches les plus basses de faire communiquer des informations sur des réseaux hétérogènes. Les standards produit correspondent donc à une implémentation particulière des couches 1 et 2.

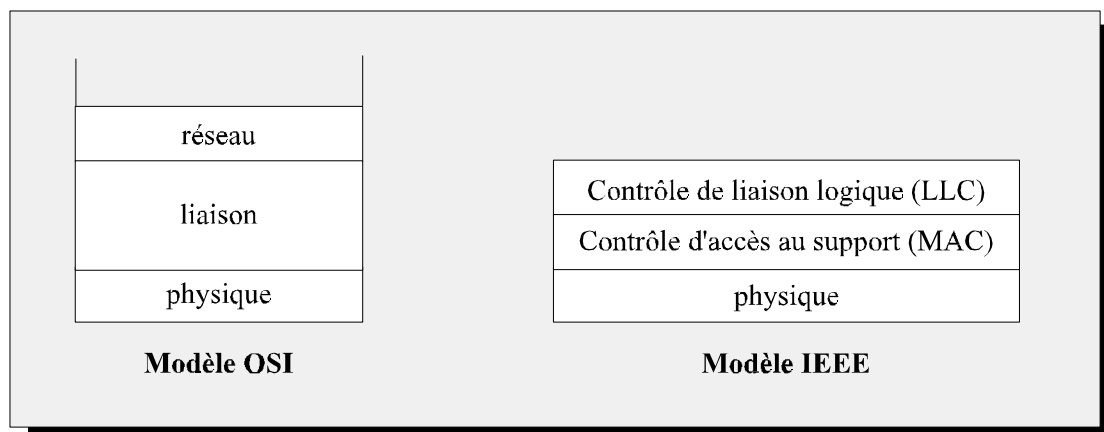


Fig. 11

La couche physique dans les deux références à pour but de transmettre des bits entre les deux entités communicantes.

La couche liaison du modèle OSI est divisée en 2 sous couches :

La couche MAC (Medium Access Control) son rôle est de partager le support entre tous les utilisateurs.

la couche LLC (Logical Link Control) son rôle est de gérer les communications.

Le comité 802 n'a pas donné lieu à un standard unique, mais à une série de standard pour couvrir l'ensemble des besoins. C'est ainsi que plusieurs sous comité ont été créés chacun traitant d'un sujet bien particulier dans lequel un standard a été élaboré.

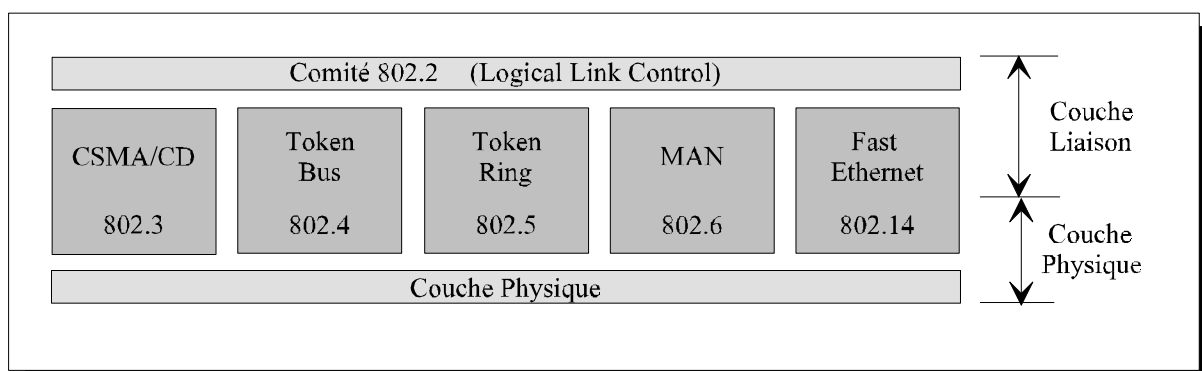


Fig. 12

802.1	:	Interfaces couches hautes (réseau)
802.2	:	Logical Link Control
802.3	:	CSMA/CD
802.4	:	Jeton sur bus
802.5	:	Jeton sur anneau
802.6	:	MAN (Normalisation pour les réseaux locaux urbains)
802.7	:	Large Bande (Broadband TAG)
802.8	:	Fibre Optique (Fiber Optic TAG)
802.9	:	IVD (Integrated Voice/Data)
802.10	:	Secure data exchange - Security and privacy
802.11	:	
802.12	:	Any Lan
802.13	:	
802.14	:	Fast Ethernet

Toutes ces normes IEEE 802.x ont été reprises et adoptées par l'ISO sous les numéros 8802.x.

La norme MAC (Medium Access Unit)

La norme 802.3

Cette norme est née chez Xérox en 1972 et a commencé une carrière commerciale à partir de 1980. C'est cette norme qui est utilisée dans les réseaux *Ethernet* et la méthode utilisée pour accéder au canal de transmission est dite à compétition. Ce protocole d'accès est aussi connu sous le nom de CSMA/CD (Carrier Sens Multiple Access with Collision Detect).

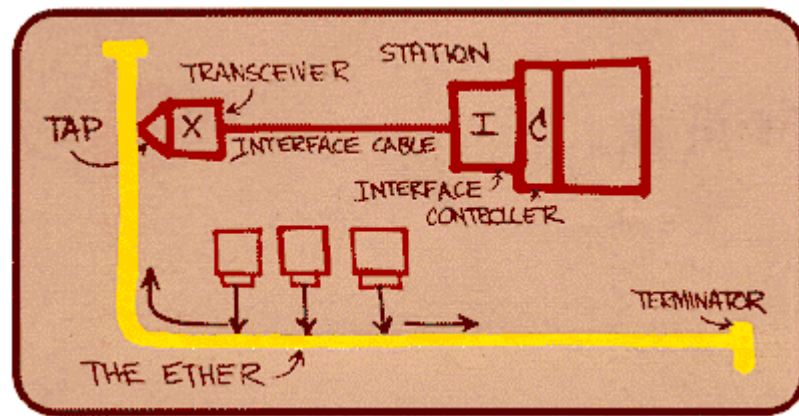


Schéma d'origine du 802.3

Les méthodes d'accès aléatoires

Ces méthodes ont pour origine la méthode ALOHA, testée au début des années 1970 sur un réseau reliant les différentes îles Hawaï.

La méthode ALOHA

Lorsqu'un coupleur a une information à transmettre sur le réseau, il envoie sa trame sans se préoccuper si un autre coupleur est en transmission. Il en résulte que des collisions d'informations se produisent et que des données soient perdues. En cas de perte d'informations, les trames entrées en collision sont régénérées à partir des coupleurs en faute et renvoyées au bout d'un temps aléatoire (voir figure suivante).

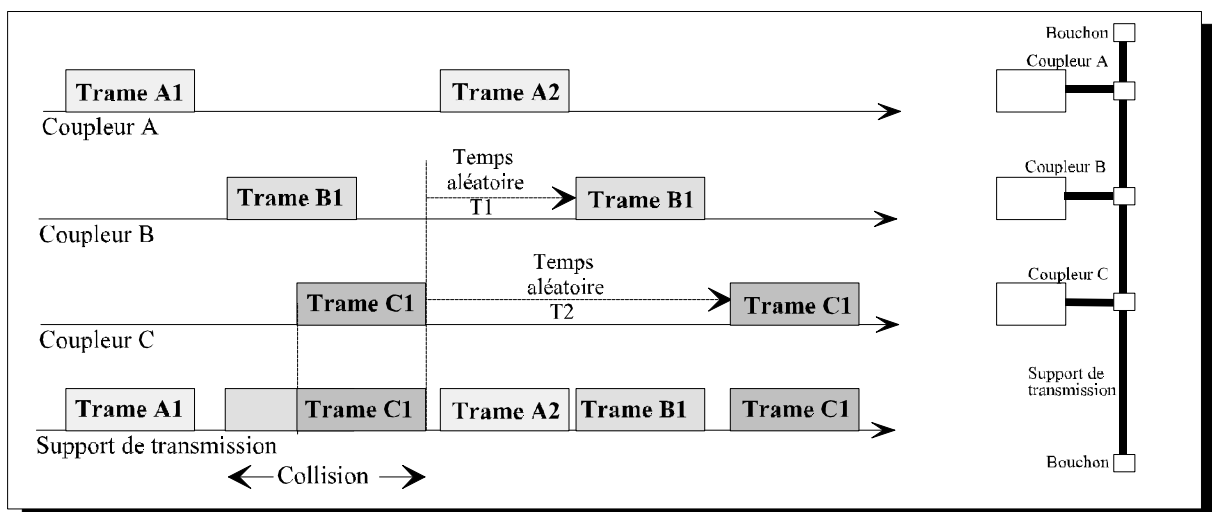


Fig. 13

La méthode ALOHA en tranche

Une amélioration de la méthode ALOHA consiste à synchroniser chacun des coupleurs sur une horloge commune, qui permet de générer des tranches de temps (ou quanta de temps) pour lesquels un coupleur n'est autorisé à transmettre une trame d'information uniquement à leur commencement.

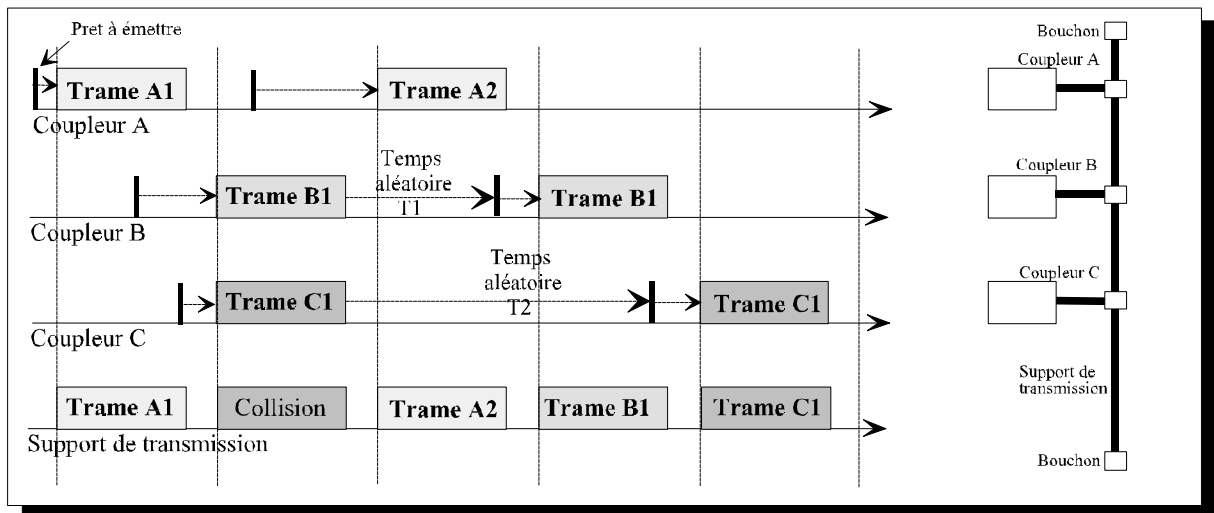


Fig. 14

La méthode CSMA non-persistant

Le coupleur qui a quelque chose à transmettre écoute préalablement le canal. Si le canal est libre de toute communication, il émet sa trame, sinon il génère un temps aléatoire au bout duquel il refera une tentative.

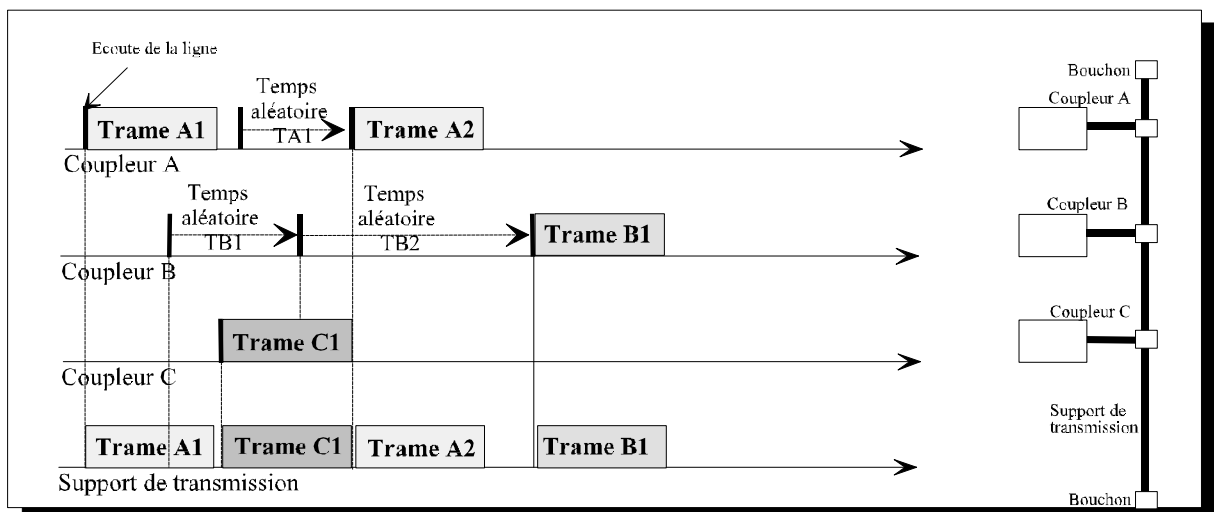


Fig. 15

Un problème peu néanmoins subsister lorsque deux coupleurs distants détectent au même instant une absence de communication sur le support de transmission, et entre tous les deux en émission. Il y a alors collision.

La méthode CSMA persistant

Le coupleur qui a quelque chose à transmettre écoute préalablement le canal. Si le canal est libre de toute communication, il émet sa trame, sinon il attend que le canal soit libre pour transmettre sa trame.

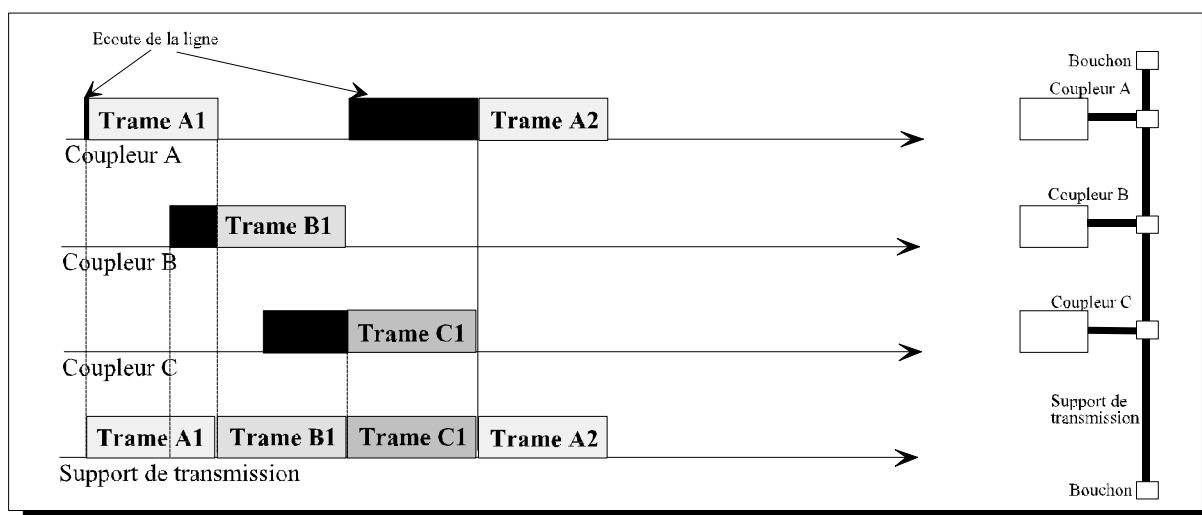


Fig. 16

La méthode CSMA/CD

Cette technique est actuellement la plus utilisée parmi les disciplines d'accès aléatoire. A l'écoute préalable du réseau (CSMA persistant) s'ajoute l'écoute pendant la transmission. De ce fait dès qu'une collision est détectée il avorte immédiatement la transmission en envoyant Le coupleur qui a quelque chose à transmettre écoute préalablement le canal. Si le canal est

Structure de la trame :

La taille minimale de la trame est de 64 octets ce qui correspond pour un débit de 10 Mbits/s à un temps de propagation aller retour sur 2,5 Km.

La taille maximale est de 1518 octets.

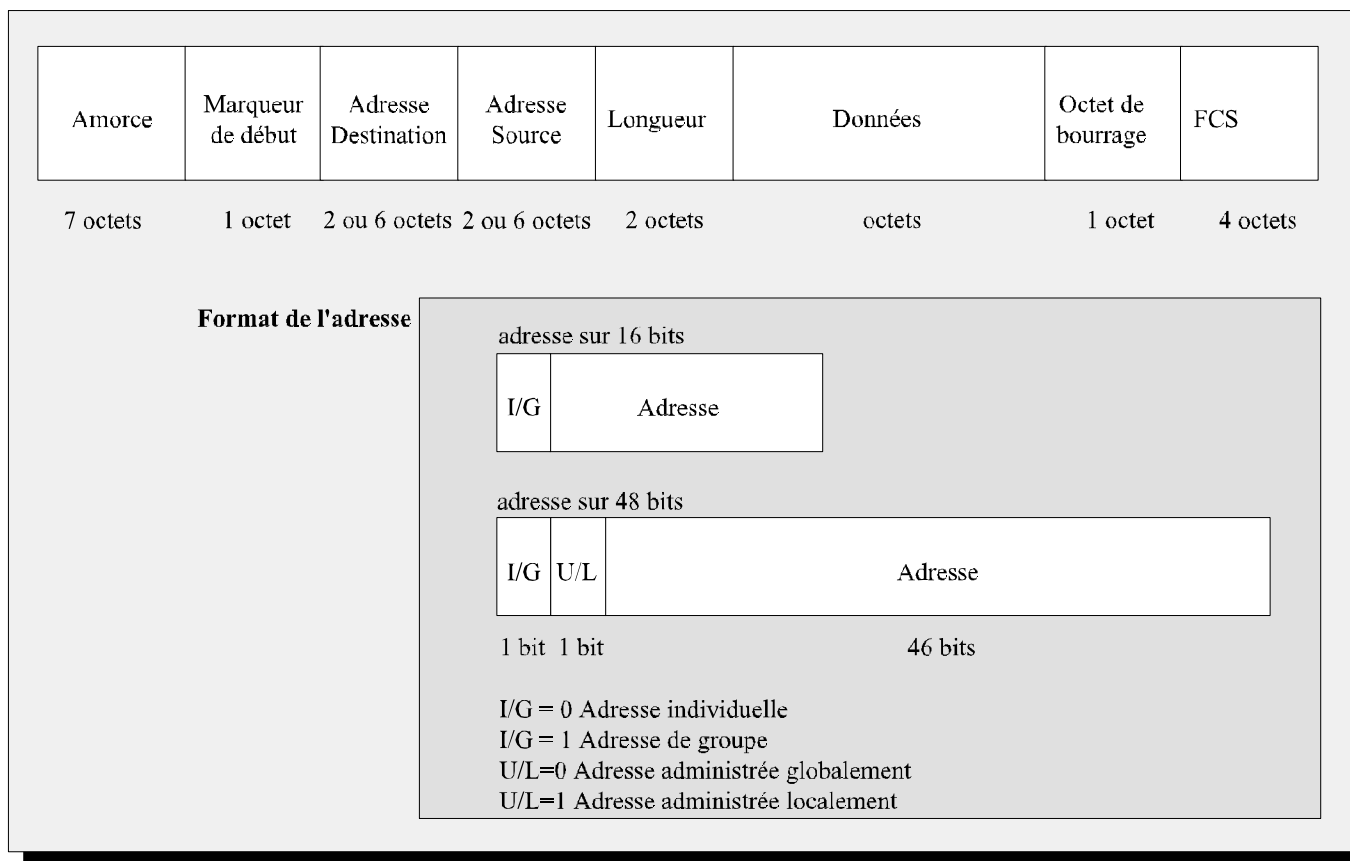


Fig.17

Amorce : Représente le début de chaque trame et elle est composée de 7 octets positionnés à 10101010. Cette amorce permet de synchroniser les stations réceptrices.

Marqueur de début de trame : (Start Frame Delimiter) Cet octet début de la trame et a pour valeur 010101011.

Adresse destination (Destination Address) Adresse source (Source Address) : Ce sont les adresses physiques du réseau codées sur 2 ou 6 octets. Les adresses 16 bits sont résolues localement par contre les adresses sur 48 bits sont résolues soit localement soit généralement.

Longueur du champ d'information (Length) : Ce champ indique sur 2 octets la longueur des données LLC. Ce nombre est compris entre 0 et 1500 octets.

Données (Data) : Champ de données LLC contenant entre 0 et 1500 octets (Logical Link Control).

Bourrage (PAD) : Octets de bourrage ajoutés si la trame ne contient pas 46 octets pour satisfaire la taille minimale d'une trame 802.3 (entre 0 et 46 octets).

FCS : (Frame Control Sequence) Constitué d'un mot de 32 bits, ce champ représente le code de vérification d'erreur sur la trame. Sa portée s'effectue sur tous les champs exceptés :

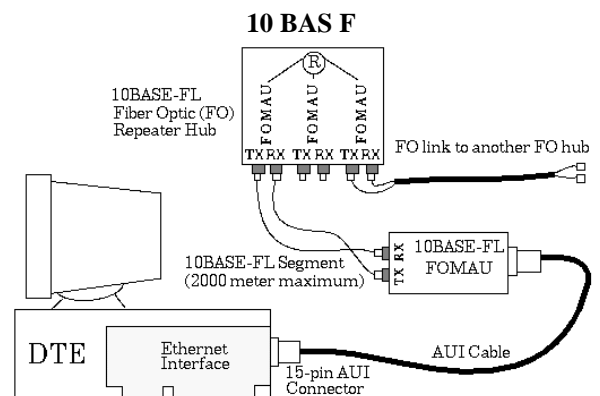
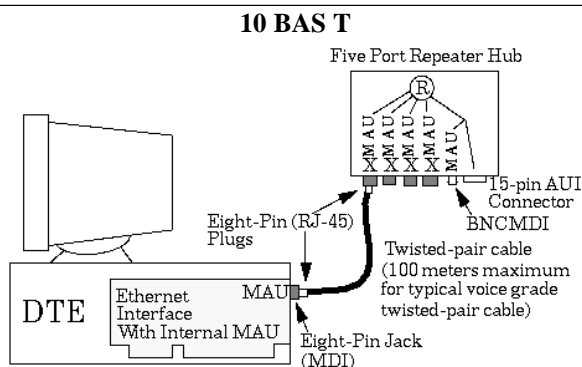
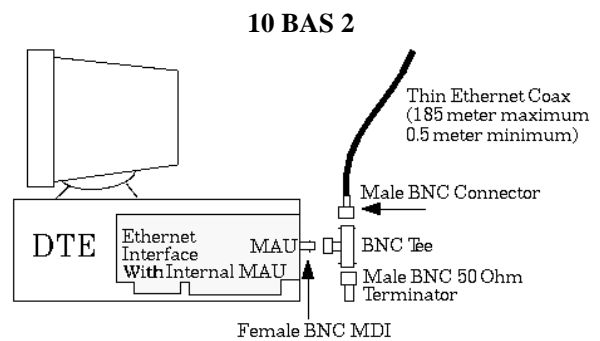
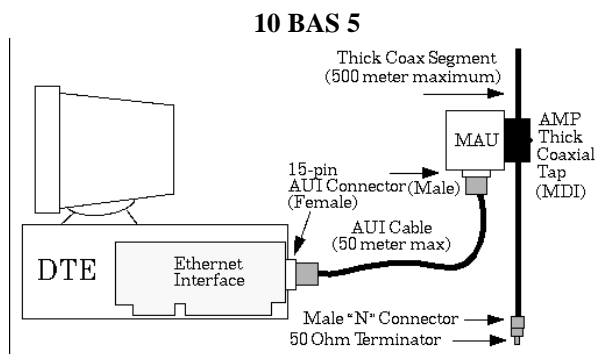
- le préambule,
- le délimiteur de début de trame,
- le FCS.

Polynôme générateur d'une trame 802.3 :

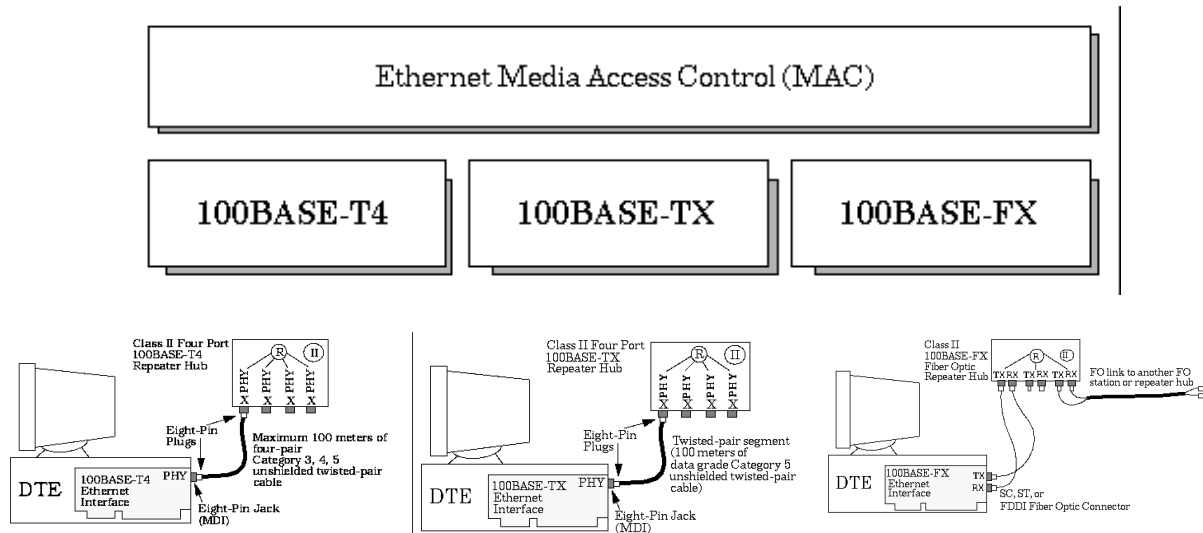
$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$$

Description des normes

Norme	10 BAS 5	10 BAS 2	10 BAS T	10 BROAD 36	10 BAS F
Support	Coaxial 50 Ω Câble jaune	Coaxial 50 Ω Câble noir RG58	Paire torsadée	Coaxial 50 Ω Type CATV	Fibre optique
Vitesse	10 Mbits/s	10 Mbits/s	10 Mbits/s	10 Mbits/s	10 Mbits/s
Longueur d'un segment	500 m	200 m	100 m	1875 m	1 km
Taille du réseau	2,5 km	925 m	5 HUBS en cascade	3675 m	-
Distance min inter-station	2,5 m	0,5 m	250 m max	-	-
nombre de station par segment	100 max.	30 max.	-	1024	-
Codage	Manchester	Manchester	Manchester	Modulation	-
Topologie	bus	bus	bus	étoile	étoile
câble	Semi-rigide avec rayon de courbure 30 cm	souple	-	-	-
repiquage	prises piquées	T vissés BNC	-	-	-
Remarques		MAU intégré sur la carte contrôleur, compatible 10 BAS 5			utilise un HUB



Le FAST ETHERNET



Norme 802.5

Cette norme, appelée aussi Token Ring (ou anneau à jeton), a été initialement commercialisée par IBM. Elle est également normalisée par l'ISO sous la référence IS8802-5.

Chaque station est connectée en mode point à point à deux autres stations et l'interconnexion totale de l'ensemble des machines crée un anneau unidirectionnel où circule une seule trame (voir Fig. 18).

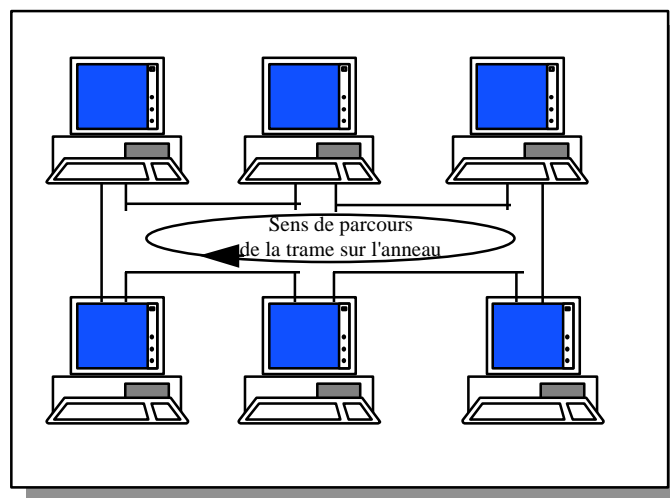


Fig. 18

Méthode d'accès par jeton

Dans un anneau à jeton, les machines sont connectées en série les unes aux autres. L'information est toujours transmise séquentiellement et bit à bit entre deux stations (Une station active vers la station suivante, les autres restants inactives). Une seule trame vide ou pleine peut circuler sur l'anneau.

Principe de fonctionnement :

- ☞ Une trame vide circule de station en station sur l'anneau avec un jeton positionné à libre.
- ☞ Une station recevant la trame avec le jeton libre souhaite transmettre de l'information; Elle positionne le jeton a occupé, elle remplit la trame avec les données à transmettre, elle indique le destinataire, et renvoiera la trame vers la station suivante.

- ☞ La trame circule de station en station jusqu'à ce qu'elle atteigne le destinataire.
- ☞ La station destinatrice recopie les données nécessaires, et renvoie la trame dans l'anneau après avoir positionné certains indicateurs.
- ☞ La station émettrice reçoit la trame, la vide, remet le jeton à libre et redépose la trame dans le réseau.
- ☞ La trame circulera dans l'anneau jusqu'à ce qu'une station le capture pour transmettre quelque chose.

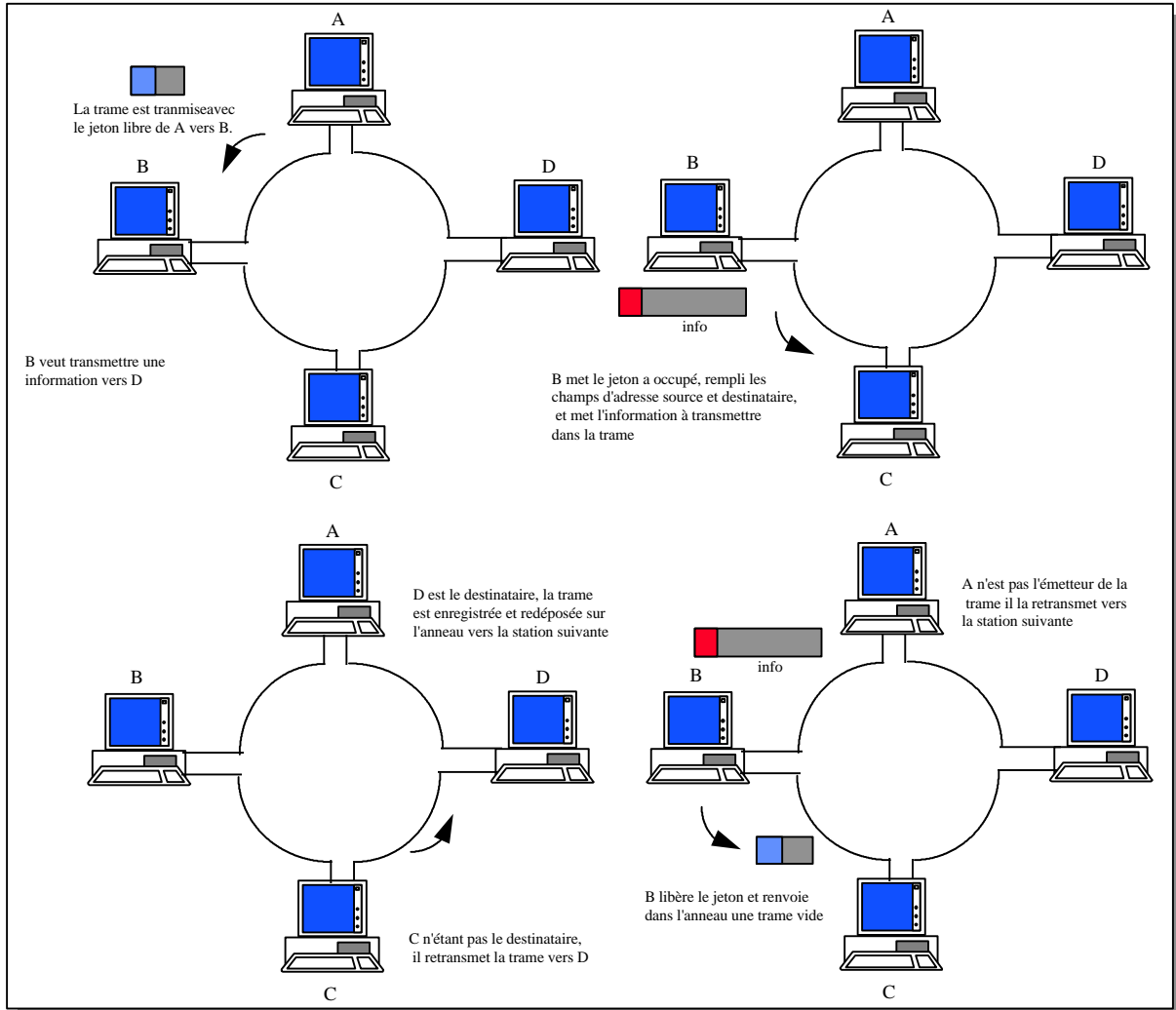


Fig. 19

Format des trames

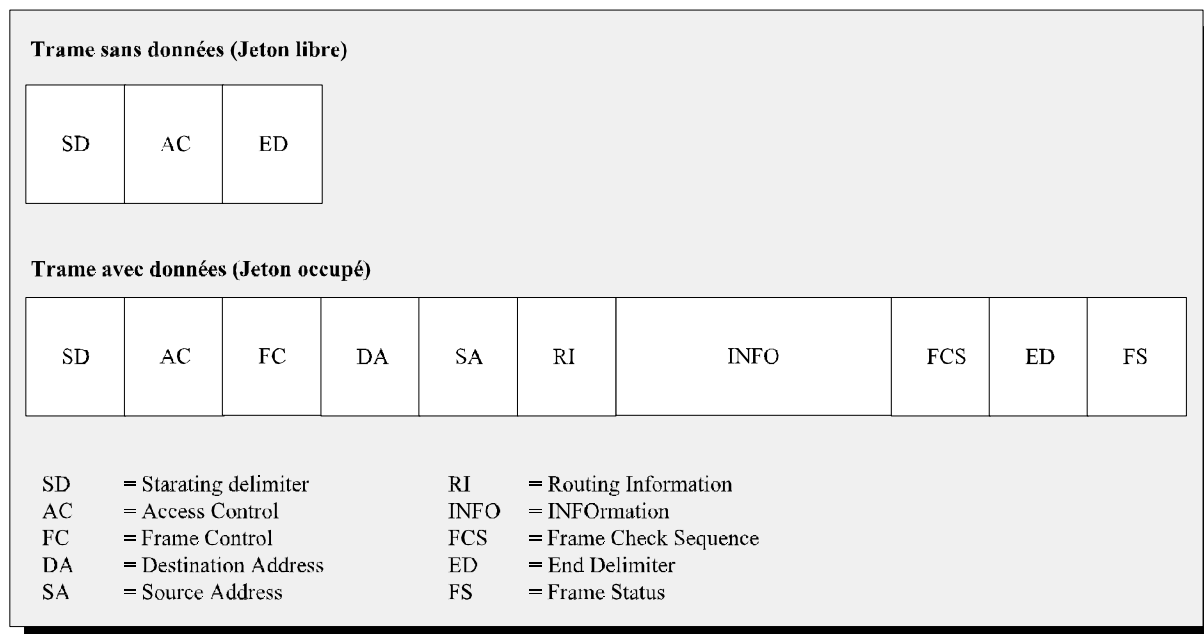
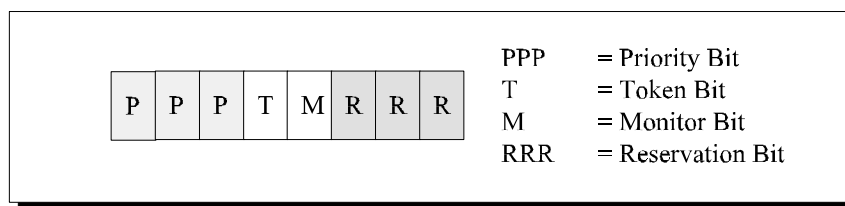


Fig. 20

SD : (*Starting delimiter*), codé sur 1 octet, il sert à délimiter le début d'une trame ou d'un jeton. So format est représenté par la valeur :

JK0JK000 J et K ont des valeurs binaires ne représentant ni un 0 ni un 1.

AC : (*Access Control*) 1 octet dont la structure est de la forme suivante :



PPP : Indique la priorité de la trame (de 0 à 7)

T : Permet aux stations d'accéder au support pour les transferts d'information.

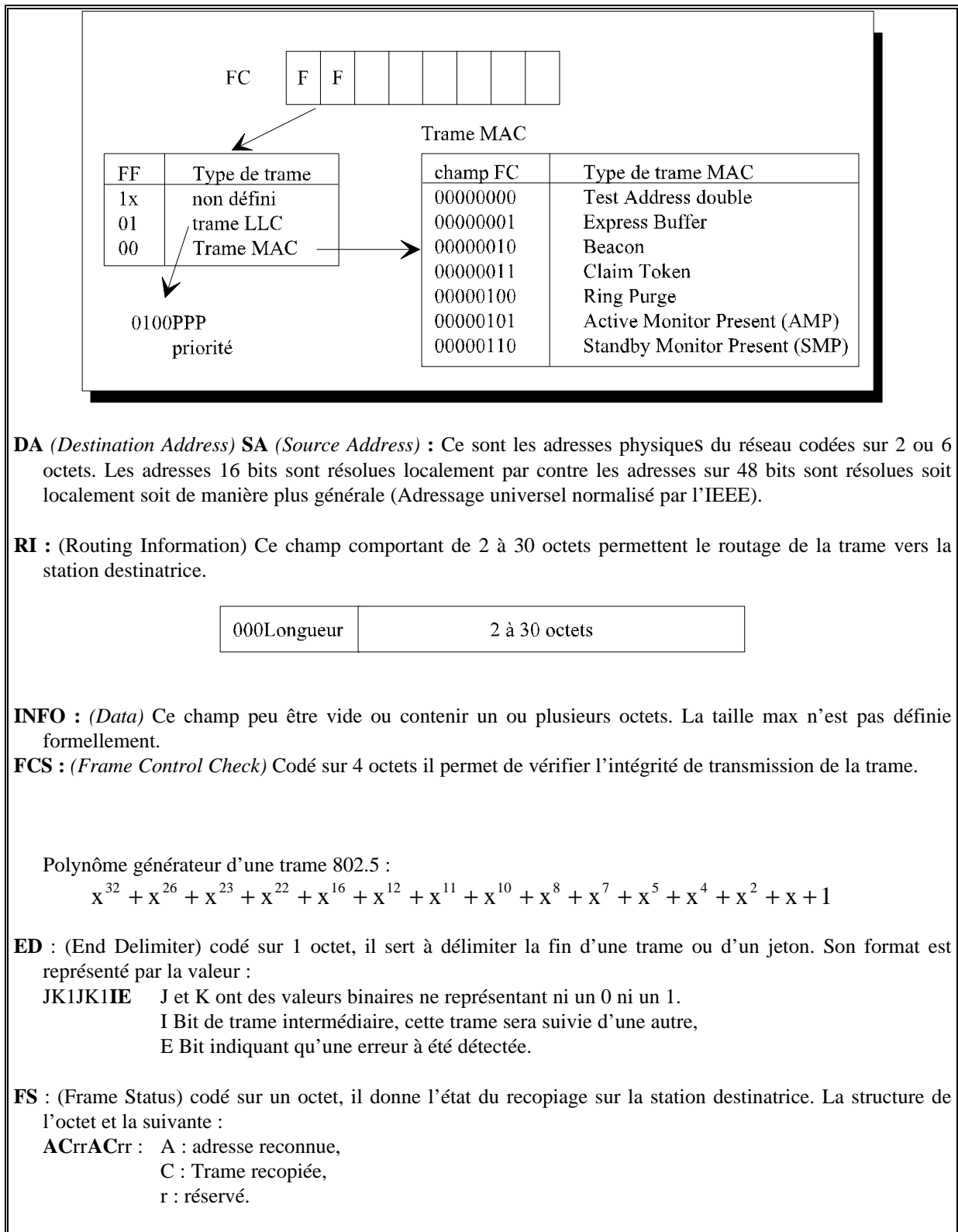
T=0 jeton libre

T=1 jeton occupé

M : Monitor bit, positionné par la station 'moniteur' du réseau il permet d'éviter que les trame fasse plusieurs fois le tour du réseau.

RRR : Bits de réservation de plus grande priorité pour augmenter la priorité du prochain jeton.

FC : (*Frame Control*) : 1 octet permettant de définir le type de la trame.



DA (Destination Address) SA (Source Address) : Ce sont les adresses physiques du réseau codées sur 2 ou 6 octets. Les adresses 16 bits sont résolues localement par contre les adresses sur 48 bits sont résolues soit localement soit de manière plus générale (Adressage universel normalisé par l'IEEE).

RI : (Routing Information) Ce champ comportant de 2 à 30 octets permettent le routage de la trame vers la station destinatrice.

000Longueur	2 à 30 octets
-------------	---------------

INFO : (Data) Ce champ peut être vide ou contenir un ou plusieurs octets. La taille max n'est pas définie formellement.

FCS : (Frame Control Check) Codé sur 4 octets il permet de vérifier l'intégrité de transmission de la trame.

Polynôme générateur d'une trame 802.5 :

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

ED : (End Delimiter) codé sur 1 octet, il sert à délimiter la fin d'une trame ou d'un jeton. Son format est représenté par la valeur :

JK1JK1IE J et K ont des valeurs binaires ne représentant ni un 0 ni un 1.
I Bit de trame intermédiaire, cette trame sera suivie d'une autre,
E Bit indiquant qu'une erreur a été détectée.

FS : (Frame Status) codé sur un octet, il donne l'état du recopiage sur la station destinatrice. La structure de l'octet et la suivante :

ACrACrr : A : adresse reconnue,
C : Trame recopiée,
r : réservé.

Protocole élémentaire

Il existe 3 protocoles de transmissions de données avec contrôle d'erreur. Ces principes sont basés sur l'utilisation de numéro de séquence, d'acquittement et de retransmissions automatiques. Chaque trame reçue peut être acquittée positivement ou négativement par l'émission de trame d'acquittement ACK ou NACK.

Ces indications peuvent être envoyées individuellement dans une trame ou intégrées dans une trame de données en sens inverse.

Pour un fonctionnement correct des ces ACK/NACK il faut adopter une stratégie sinon il risque de se produire des situations attentes interminables (type étreintes fatales ou Dead Locks).

protocole SEND and WAIT

Dans ce protocole une seule trame est émise à la fois puis l'émetteur arme un réveil et se met en attente d'un signal d'acquittement. Si ce signal n'arrive pas avant le déclenchement du réveil, la trame est émise de nouveau. chaque trame envoyée est stockée temporairement dans un tampon qui sera libéré à la réception d'un ACK.

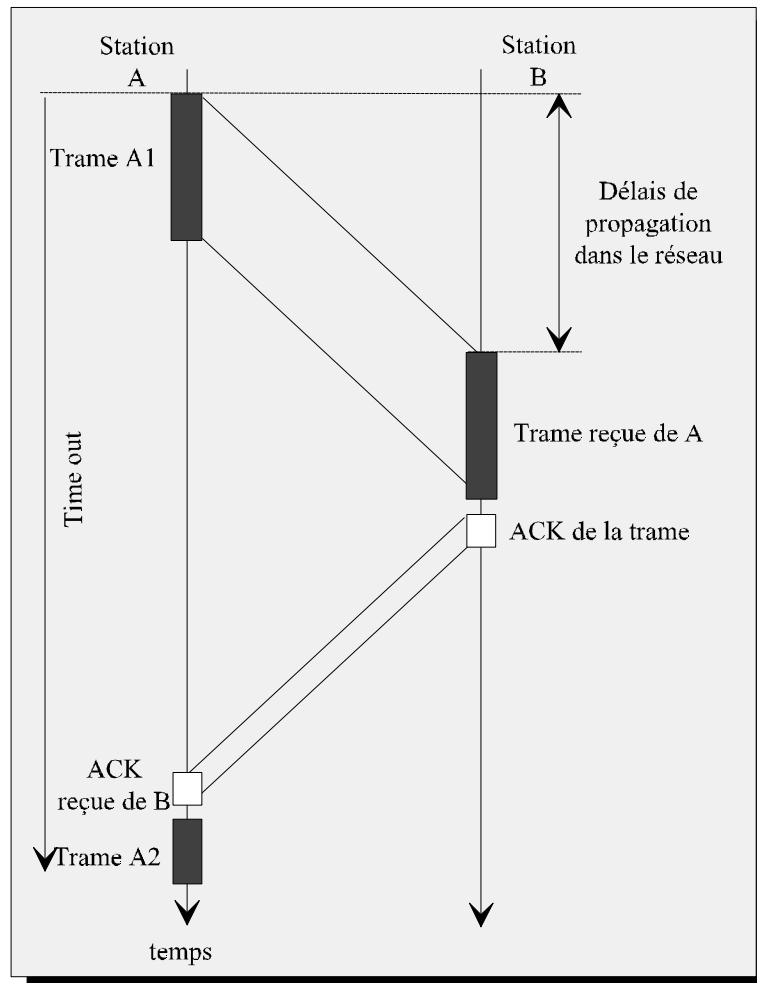


Fig. 7

protocole Go-Back n

L'émetteur peut envoyer continuellement des trames tant qu'il en a transmettre sans d'acquittement. Chaque fois qu'une trame est envoyée un réveil est armé. Si un ACK n'arrive pas avant l'expiration de timer ou qu'un NACK arrive la trame et toutes les suivantes sont envoyées à nouveau.

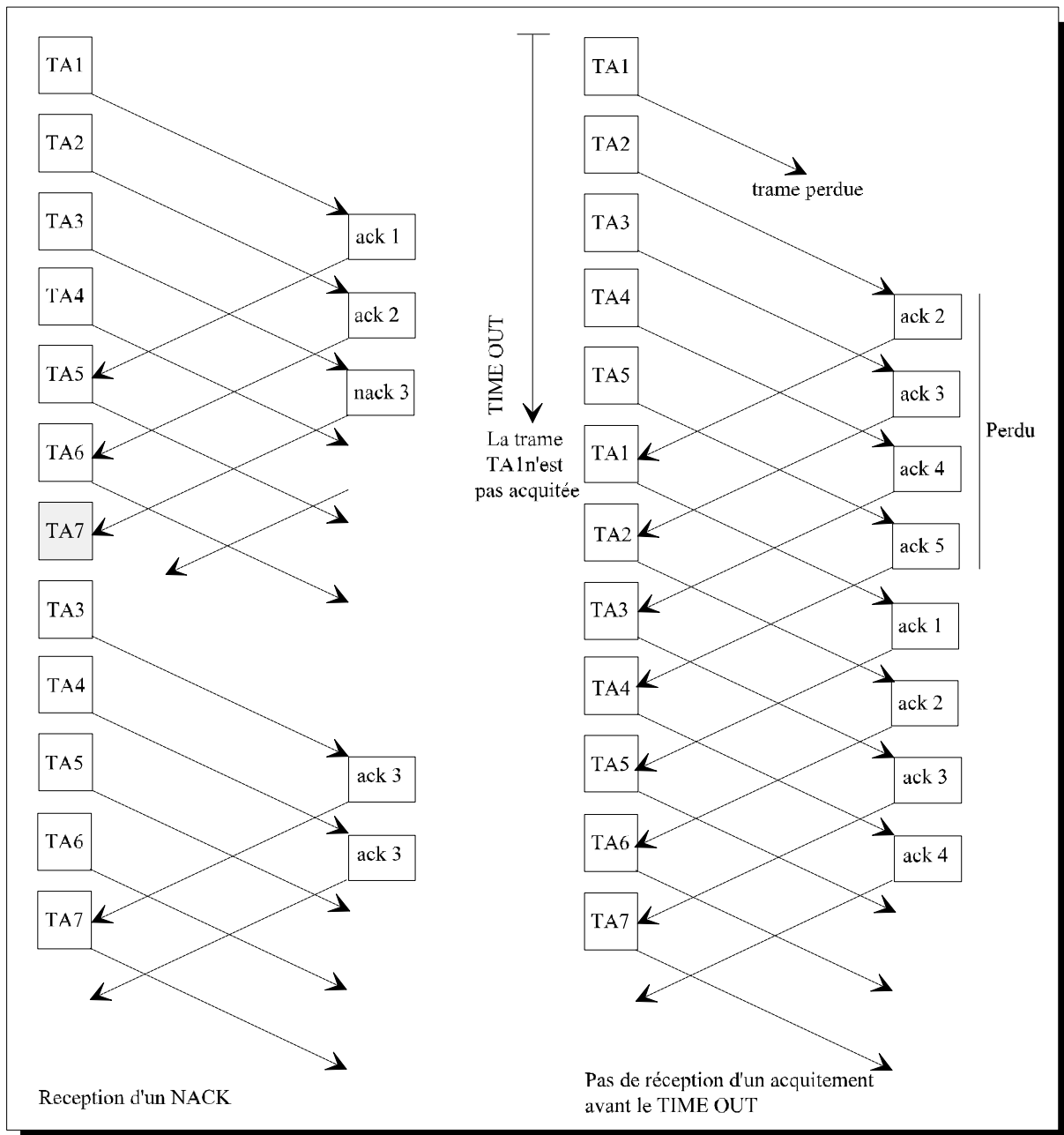


Fig. 8

La norme LLC (*Logical Link Control*)

Cette couche représente la deuxième division de la couche liaison dans le modèle OSI. Elle se situe au dessus de la sous couche MAC (*Medium Access Control dont le rôle est de gérer le contrôle d'accès au support de transmission*).

Le rôle de cette sous couche LLC est de gérer les communications entre stations et d'assurer l'interface avec les couches supérieures. Ses spécifications sont données dans le standard IEEE 802.2 et sont reprises dans la norme internationale IS 8802-2 de l'ISO.

Cette norme offre à la couche réseau des services :

- sans connexion (LLC1),
- avec connexion (LLC2),
- sans connexion avec acquittement (LLC3).

Cette norme est donc divisée en 3 normes qui sont adaptés à ces modes de fonctionnement différents. Le but de ces protocoles est de fournir une garantie de livraison des messages (appelés LSDU Link Services Data Unit), la détection et reprise sur erreur.

La norme LLC1 :

Le service LLC1 est un protocole de peu complexe.

☞ Le transfert des données peu se faire :

- ☞ - en point à point,
- ☞ en point à multi-points ou diffusion.

☞ les trames sont envoyées :

- ☞ sans séquençement,
- ☞ aucune forme d’acquittement de la part du destinataire,
- ☞ aucun contrôle d’erreur.

Ce protocole simplifié est utile lorsque les couches supérieures fournissent déjà un service de contrôle d’erreur ainsi qu’un séquençement. Dans beaucoup de réseaux locaux, les systèmes d’extrémités utilisent un service datagramme au niveau réseau et un service connecté au niveau transport (TCP/IP). Il n’est pas alors nécessaire de dupliquer ce service au niveau LLC.

Primitive de connexion en LLC1

L_DATA.request (@locale, @distante,LSDU, classe de service)
 L_DATA.indication ((@locale, @distante,LSDU, classe de service)

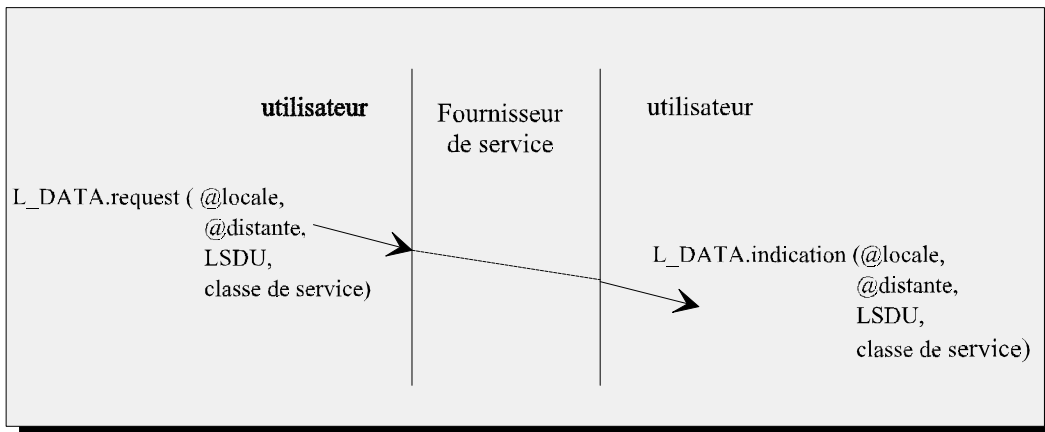


Fig.21

Une machine source envoie des trames à la machine destination sans recevoir de cette dernière un accusé de réception.

Aucune connexion n’est établie avant l’envoi d’une trame, et aucune déconnexion n’est faite après leur envoi.

Cette norme a été conçue en considérant que les informations issues de la couche physique sont données avec un taux d’erreur inférieur à 10⁻¹⁰. Si une erreur survient, rien n’a été prévu pour y remédier dans cette couche et la trame se trouve donc erronée ou perdue. Il faut donc que les couches supérieures prennent en compte ce type de problème.

La norme LLC2 :

Ce service utilise les 3 phases classiques de fonctionnement du mode connecté :

- Phase d'établissement de la connexion,
- Phase de transfert de l'information,
- Phase de libération de la connexion.

LLC2 garantit que toutes les données sont correctement délivrées à leur destinataire :

- pas erreur
- pas de perte,
- pas de duplication,
- pas de déséquence.

Si ce n'était pas le cas LLC2 le signalerait au destinataire.

Primitive de connexion en LLC2

L_CONNECT.requête (@locale, @distante, classe de service)
 L_CONNECT.indication (@locale, @distante, classe de service)
 L_CONNECT.réponse (@locale, @distante, classe de service)
 L_CONNECT.confirmation (@locale, @distante, classe de service)

L_DISCONNECT.indication (@locale, @distante, raison)
 L_DISCONNECT.requête (@locale, @distante)

L_DATA_CONNECT.requête (@locale, @distante, LSDU)
 L_DATA_CONNECT.indication (@locale, @distante, LSDU)

L_RESET.requête (@locale, @distante)
 L_RESET.indication (@locale, @distante)
 L_RESET.réponse (@locale, @distante)
 L_RESET.confirmation (@locale, @distante)

L_CONNECT_FLOW_CONTROL.requête (@locale, @distante, valeur)
 L_CONNECT_FLOW_CONTROL.indication (@locale, @distante, valeur)

Le transfert de données :

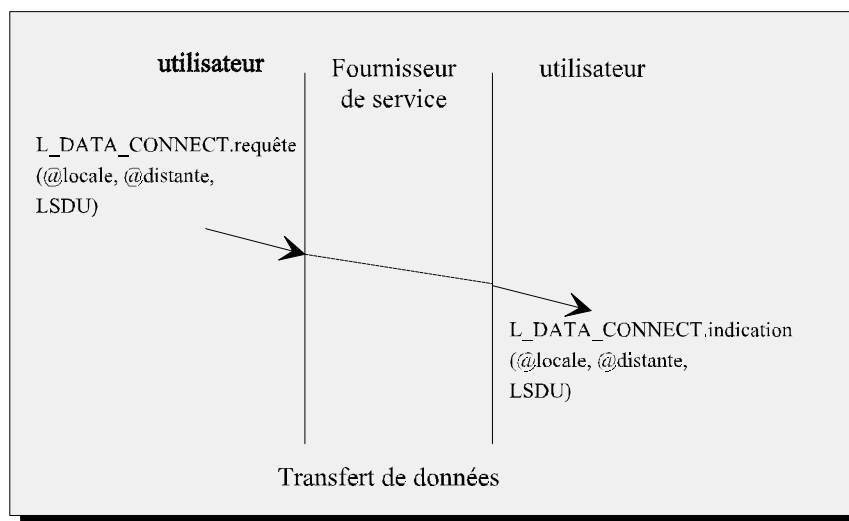


Fig. 22

Une confirmation de l'utilisateur n'est pas nécessaire puisque le service est garanti la remise correcte des données. En cas de problème l'utilisateur est averti par des fonctions de déconnexion ou réinitialisation.

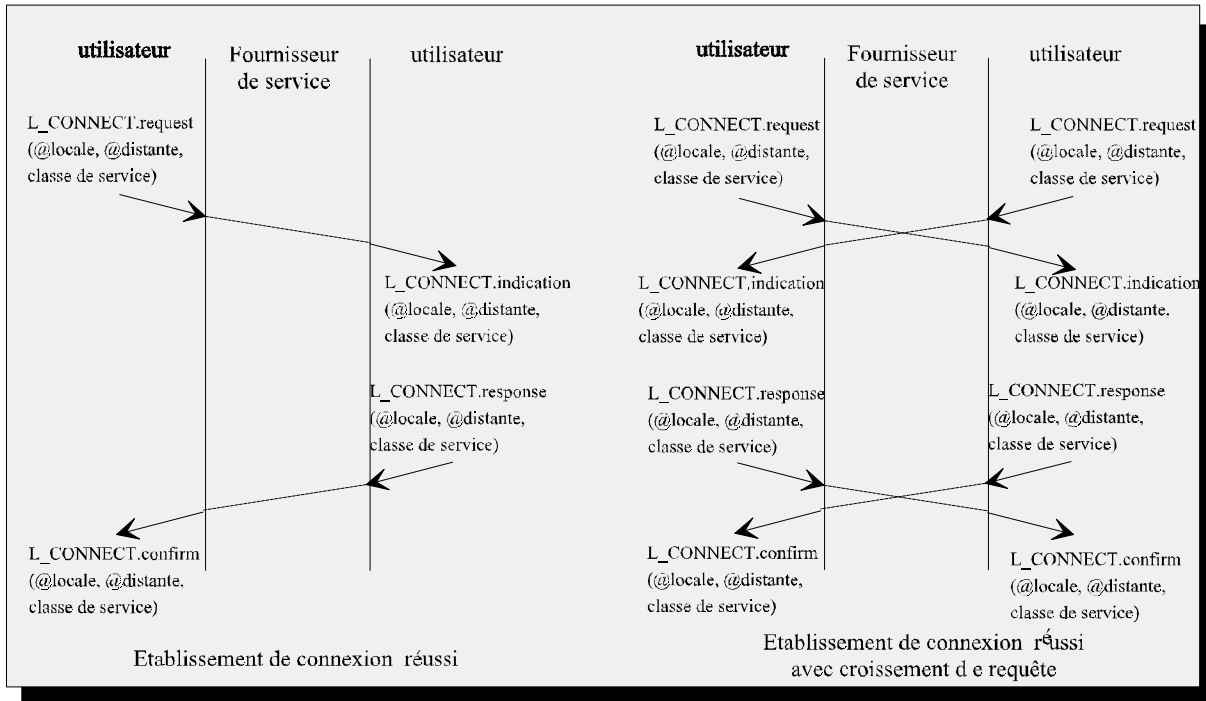


Fig. 23

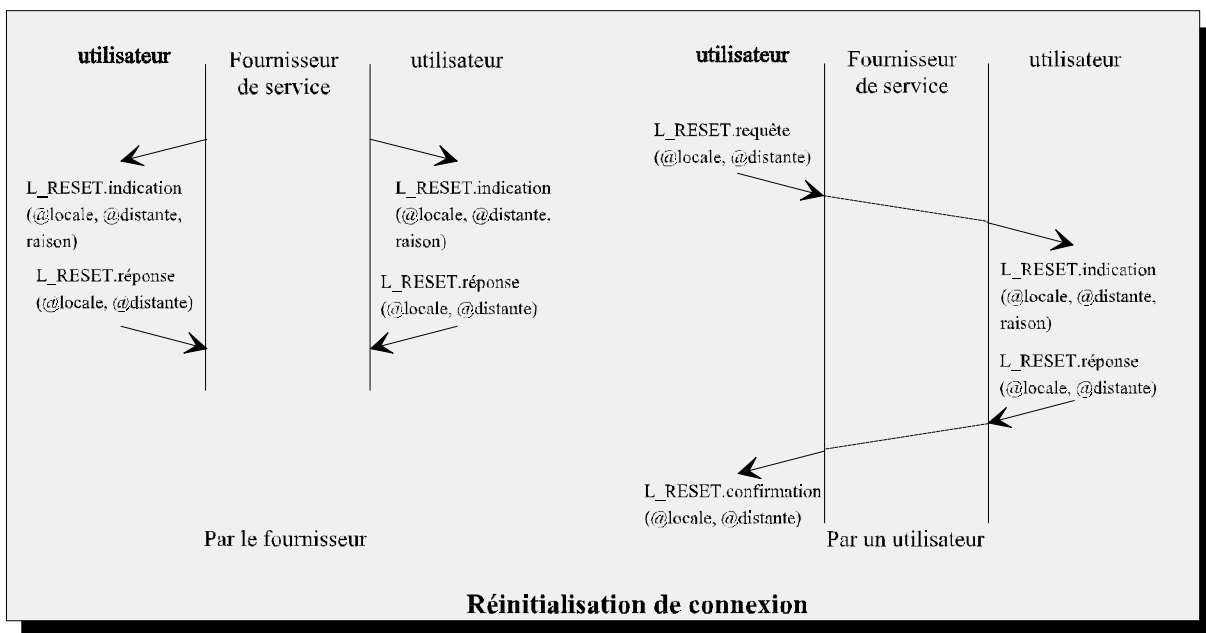


Fig. 24

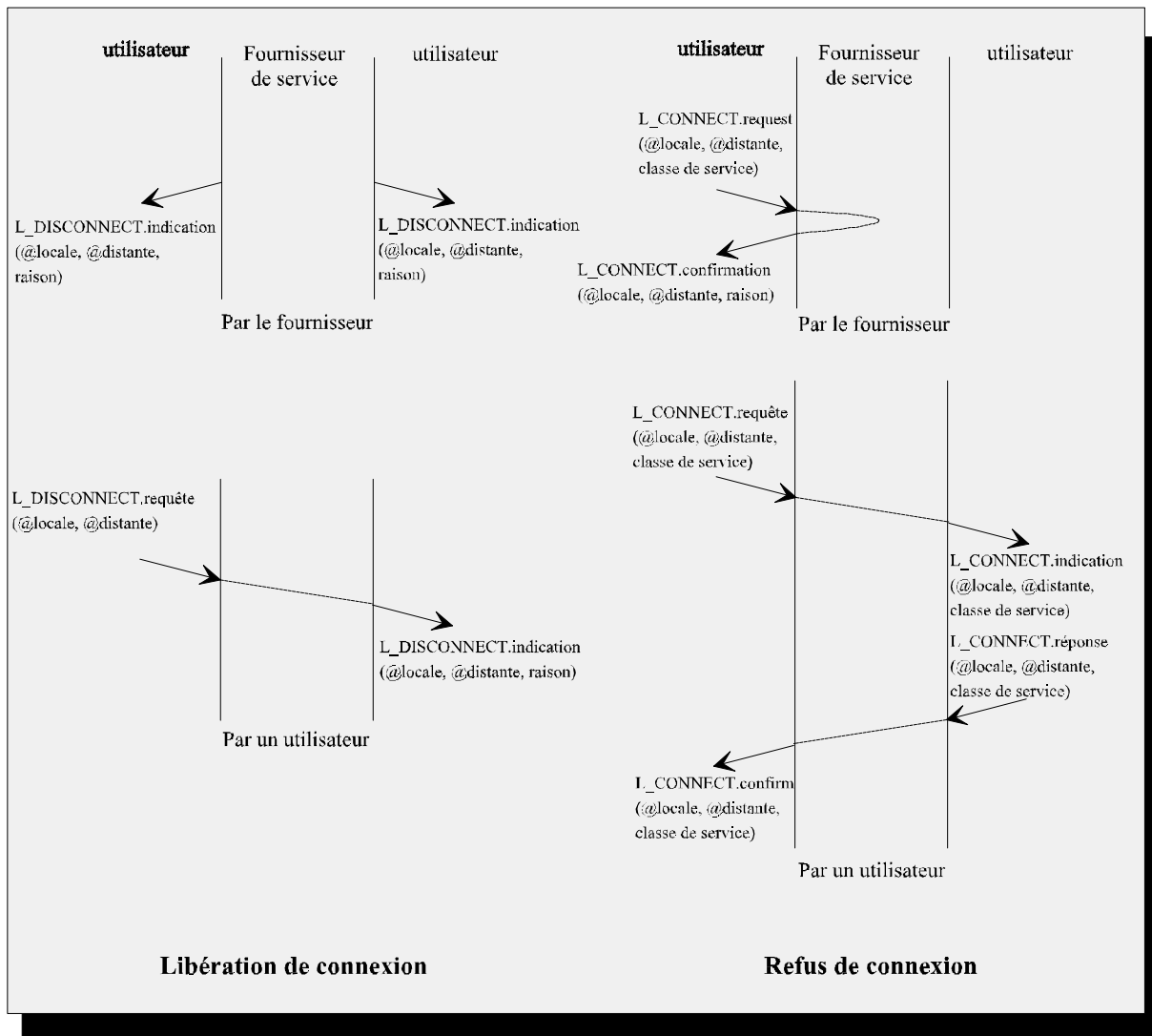


Fig. 25

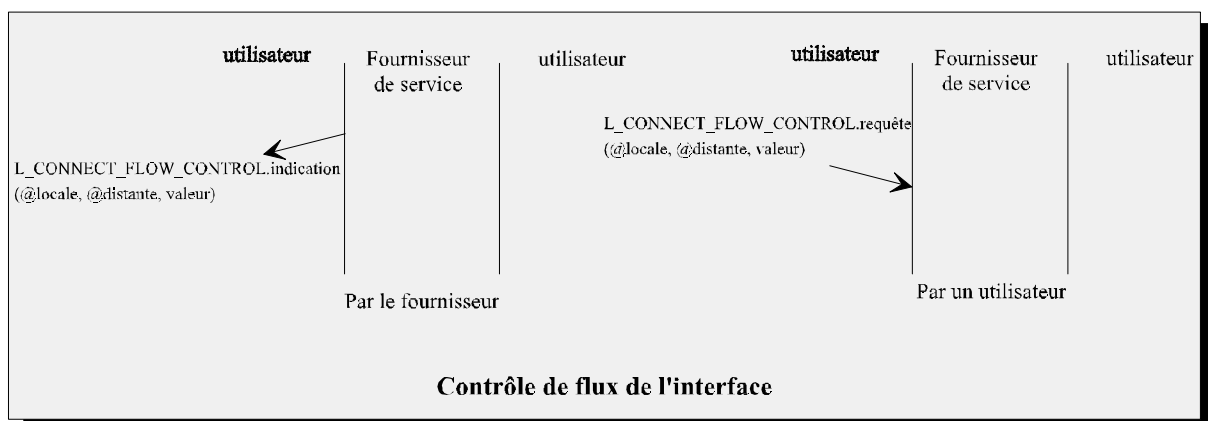


Fig. 26

Résumé

Cette norme est exploitée en mode connexion avec acquittement des trames envoyées. Ce service est le plus fiable de tous. Ce service garantit donc que toutes les trames émises ont été reçues et ceci dans l'ordre d'émission.

Lorsqu'une trame est envoyée, il y a établissement d'une connexion, transmission des trames, et déconnexion à la fin de la transmission de toutes les trames. Un contrôle de flux peut-être réalisé, et des procédures de reprises sur erreurs peuvent-être utilisées.

La norme LLC3 :

Ce service est utilisé uniquement en mode point à point.

Cette norme est exploitée en mode sans connexion avec acquittement des trames envoyées. Ce service est donc plus fiable que le LLC1. Aucune connexion n'est établit avant l'envoi d'une trame par contre chaque trame envoyée doit fournir dans un temps imparti un signal d'acquiescement. Si ce n'est pas le cas, la trame est réémise à l'échéance de ce temps.

Primitive de connexion en LLC3

L_DATA_ACK.requête (@locale, @distante, LSDU, priorité, classe_de_service)
 L_DATA_ACK.indication (@locale, @distante, LSDU, priorité, classe_de_service)
 L_DATA_ACK_STATUS.indication (@locale, @distante, LSDU, priorité, classe_de_service)

L_REPLY.requête(@locale, @distante, LSDU, priorité, classe_de_service)
 L_REPLY.indication (@locale, @distante, LSDU, priorité, classe_de_service)
 L_REPLY_STATUS.indication (@locale, @distante, LSDU, priorité, classe_de_service, status)
 L_REPLY_UPDATE.requête (@locale, LSDU)
 L_REPLY_UPDATE_STATUS.indication (@locale, status)

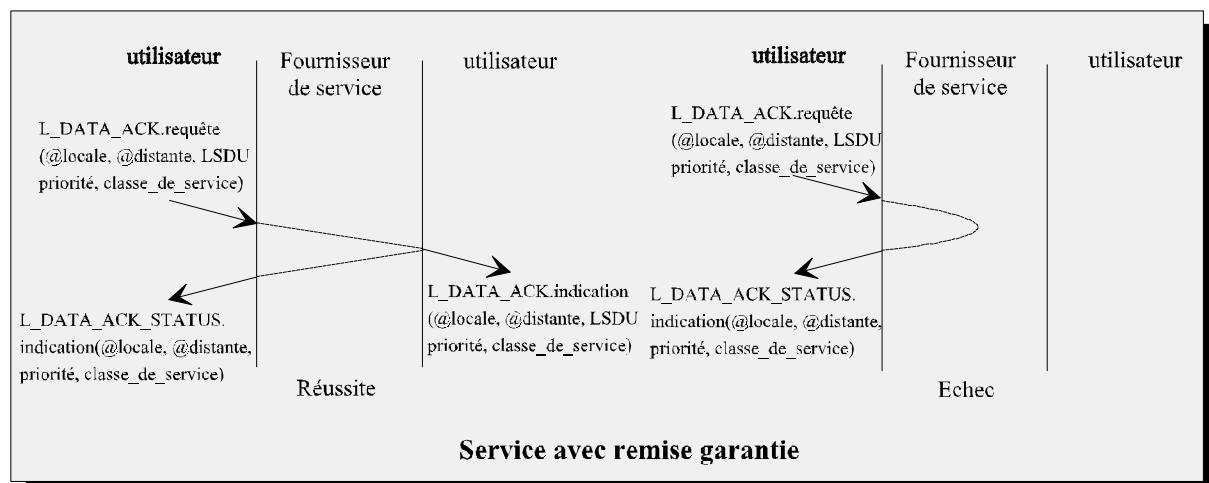


Fig. 27

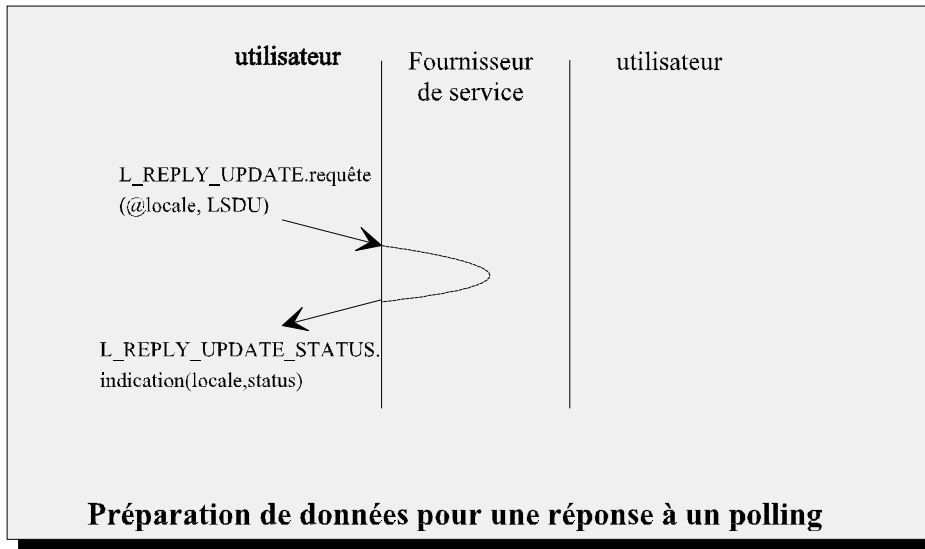


Fig. 28

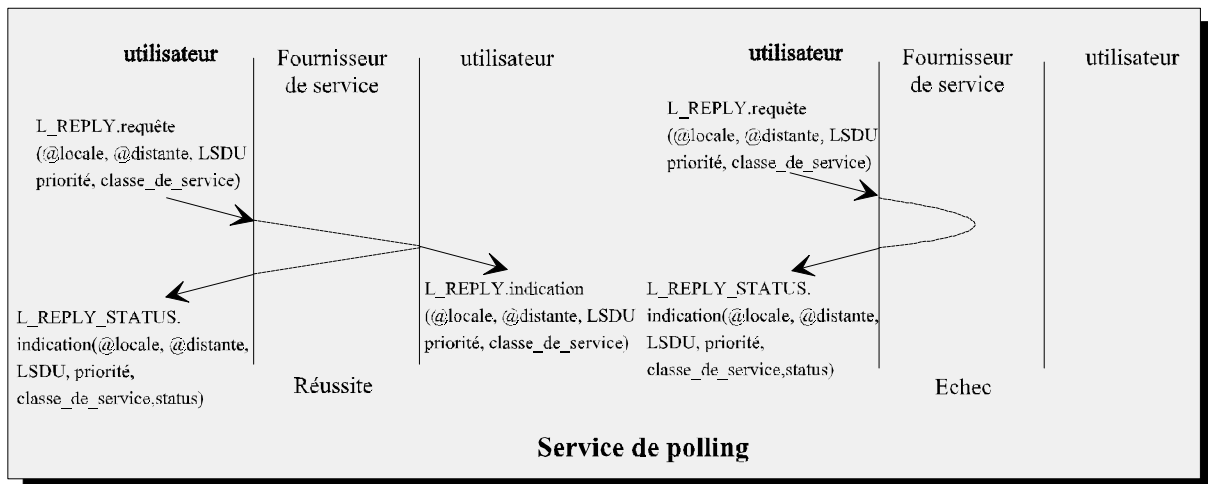


Fig. 29