I. GENERALITES

A. A quoi cela sert-il ?

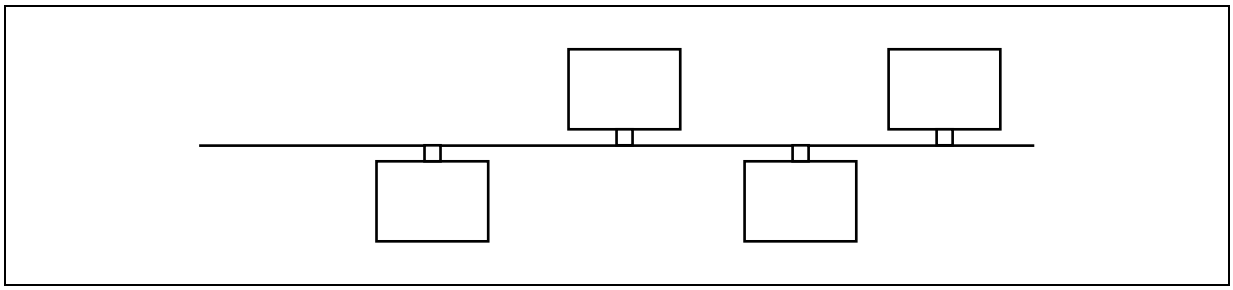
- **Communications entre services, personnes, machines**
- **Accès à des ressources partagées**
- **Accès à des données partagées**
- **Communications performantes et fiables**

Lors de leur apparition, les réseaux locaux s'étendaient sur quelques mètres à quelques kilomètres (d'où leur nom) mais aujourd'hui ils peuvent atteindre des dizaines de km ou même 100 km.

Dans le cas des réseaux locaux, le débit se situe entre 50 kbit/s et 100 Mbit/s ou 1 Gbit/s ou même plus encore.

B. Topologie

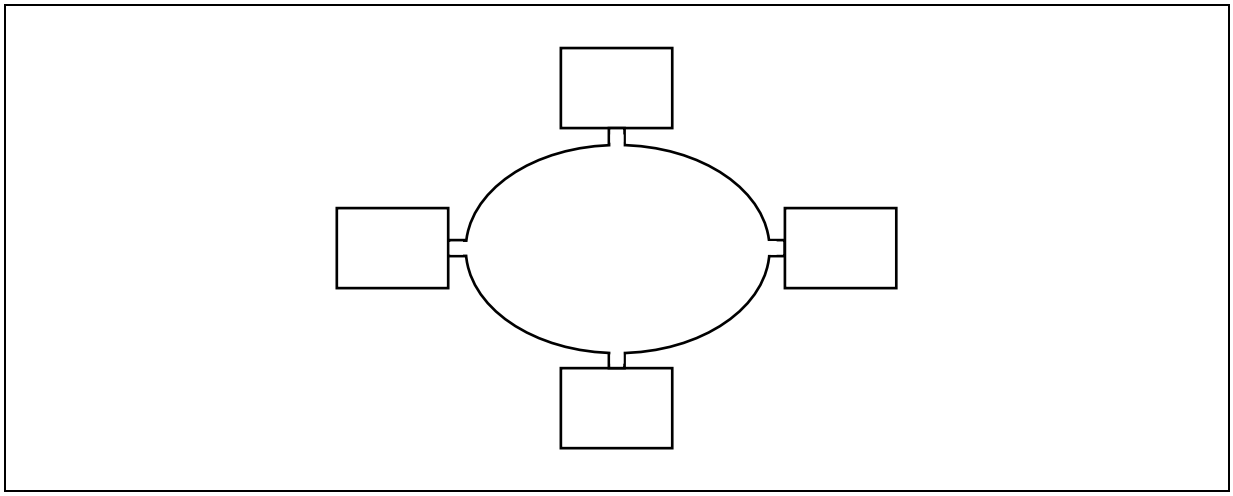
1. Bus



Caractéristiques

- Propagation bidirectionnelle des informations
- Contrôle centralisé ou distribué dans chaque station
- Mise en oeuvre et extensions simples à réaliser
- Incidents faciles à identifier et isoler
- Moins de câble
- Attention aux extrémités (terminateurs)

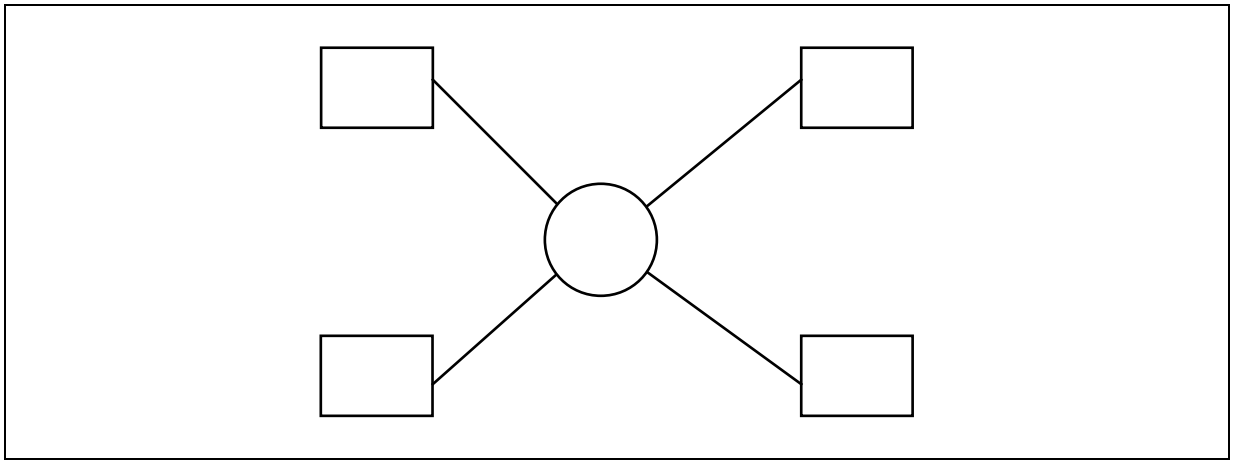
2. Anneau



Caractéristiques

- Chaque station reçoit et renvoie l'information
- Unidirectionnel
- Répétition des signaux par chaque noeud
- Extensions faciles
- Coupures simples à identifier
- Dans le cas d'un anneau, une coupure entraîne un non-fonctionnement du réseau
- Plus de câble que le bus
- Pas d'extrémités
- Fibre Optique adaptée

3. Etoile

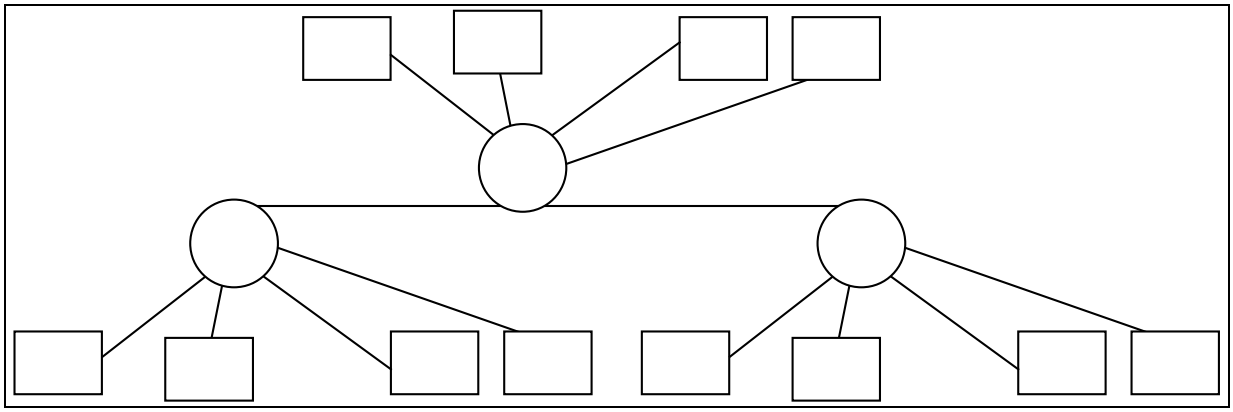


Caractéristiques

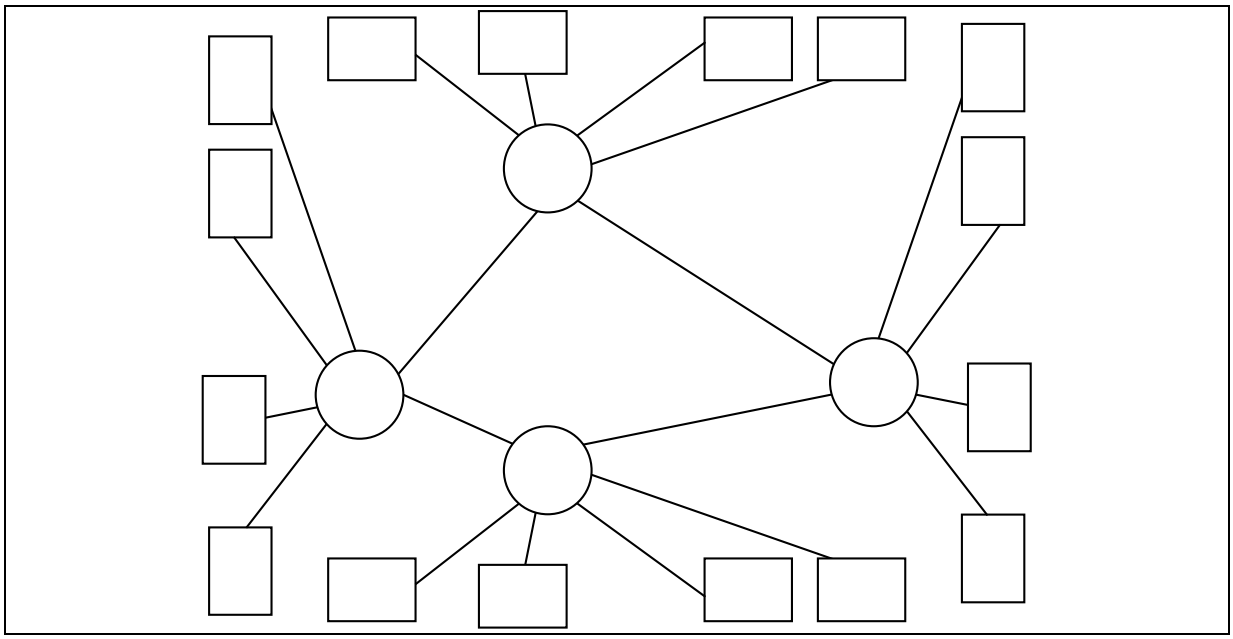
- Toute communication passe par le noeud central
- Extensions limitées mais faciles
- Solution onéreuse, beaucoup de câble
- Une coupure de câble empêche simplement la communication au départ du noeud en défaut
- Point sensible est le contrôleur central

4. Hybrides

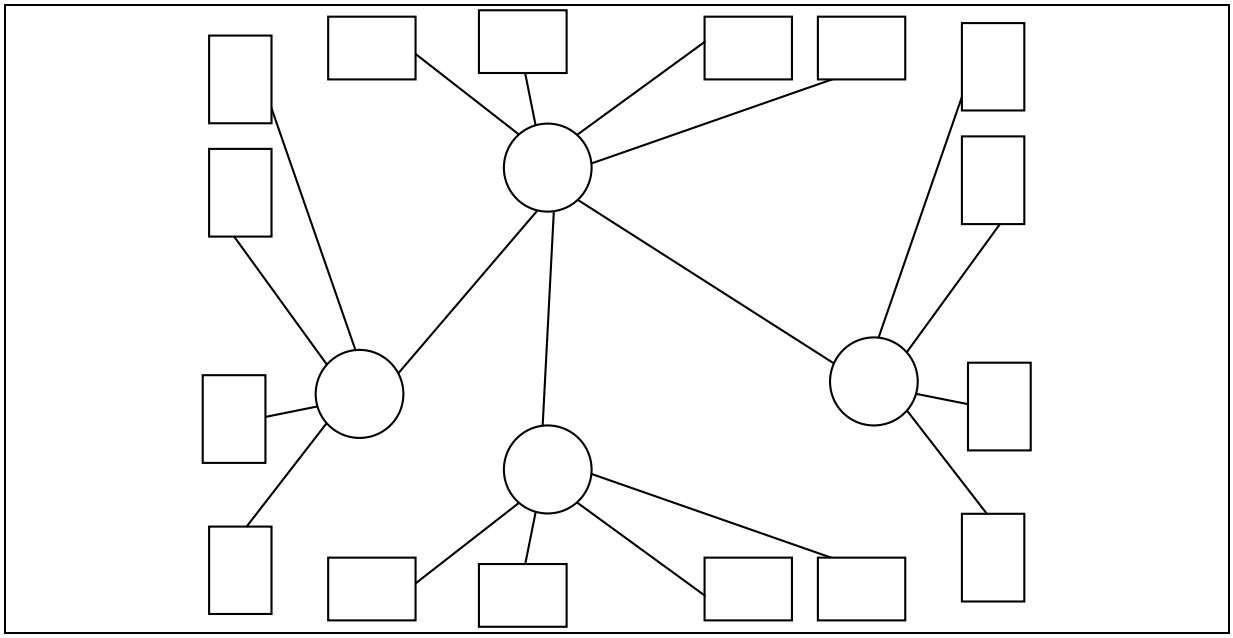
a) Bus et étoiles



b) Anneau et étoiles



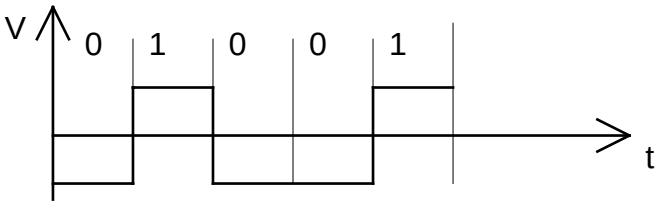
c) Arbre



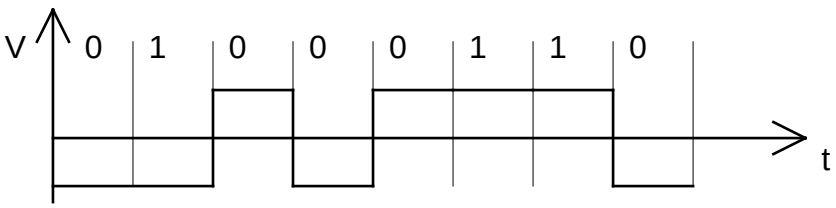
C. Mode de transmission

1. Codage du signal

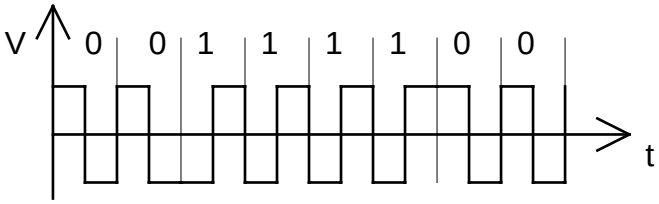
a) Codage NRZ (Non Return to Zero)



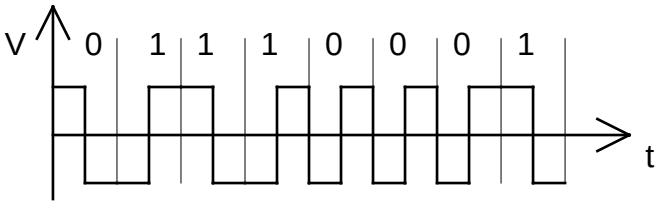
b) Codage NRZI (Non Return to Zero, Invert on one)



c) Codage Manchester



d) Codage Manchester différentiel



2. Type signal

a) Bande de base (base band)

- 1 seul canal qui utilise toute la bande passante du support physique
- Transmission de signaux numériques sans aucune modulation particulière
- Faible coût

b) Bande porteuse (carrier band)

- 1 modem
- 1 seul canal qui utilise toute la bande passante du support physique
- 1 signal analogique
- Modulation en amplitude, fréquence, phase
- Coût moyen

c) Large Bande porteuse (Broadband Band)

- 1 modem
- Plusieurs canaux / multiplexage fréquentiel du support physique
- Modulation en amplitude, fréquence, phase
- Coût très élevé

- Gestion du multiplexage de fréquence par station nécessaire
- Attention à la maintenance du modem

D. Support de transmission

Voici les principaux supports de transmission utilisés dans le cas des réseaux locaux.

Paire torsadée	Débit fonction de la distance TP cat 5 100 MHz / 100m	Immunité faible Pose facile Raccord aisé
Câble coaxial Petit diamètre (ϕ <1 cm)	Bande passante jusqu'à 50 MHz Utilisable pour signal en bande de base	Raccord facile Bonne immunité
Câble coaxial Gros diamètre (ϕ >1 cm)	Bande passante > 400 MHz utilisable pour signal large bande	Raccord facile Bonne immunité
Fibre optique Multimode (réflexion du signal lumineux)	Très large bande passante	Immunité totale aux P.E.M. Isolation galvanique Coût élevé
Fibre optique monomode (pas de réflexion)	Bande passante de plusieurs GHz	Immunité totale aux P.E.M. Coût très élevé Installation difficile

E. Méthode d'Accès

1. Maître-Esclaves

Le maître gère le droit de parole des différents partenaires du réseau. Système déterministe.

2. Jeton

Une station du réseau a le droit de parole si elle dispose du jeton. Le jeton passe par les différents partenaires du réseau. Système déterministe.

3. Accès Aléatoire

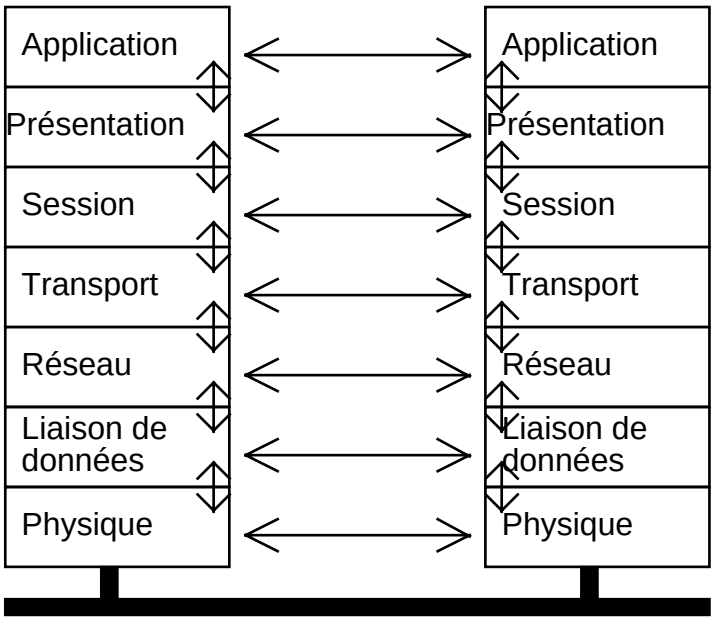
Si le support est libre, un partenaire du réseau peut l'utiliser pour communiquer. Système probabiliste.

II. LE MODELE OSI DE ISO (ISO 7498)

Le modèle de référence OSI (Open Systems Interconnection) de ISO (International Standardisation Organisation) est un modèle standardisé destiné à permettre la communication entre systèmes hétérogènes. Il fournit les bases pour créer un système de communication ouvert. Il définit l'architecture en 7 couches normalisées pour les réseaux télématiques.

Son but étant de permettre l'interconnexion de systèmes ouverts, les concepteurs de systèmes de communications ont donc intérêt à utiliser ce modèle de référence et à utiliser les normes ISO associées pour permettre l'ouverture de leurs systèmes vers d'autres vendeurs.

Chaque couche a un rôle bien précis et intervient sur des éléments précis de la communication entre systèmes hétérogènes.



Support Physique

Couches 1 à 4 orientées Communication

Couches 5 à 7 orientées Information

Le modèle de référence OSI de ISO

A chaque couche correspond une ou plusieurs normes ISO. Les normes indiquent les services que la couche doit assurer, le codage des informations, le comportement de la couche, ... Elles n'indiquent pas la manière de mettre en oeuvre les services. Cela est du ressort du concepteur. Ainsi, la mise en oeuvre d'une norme peut être totalement différente d'un concepteur à l'autre mais l'utilisation de la norme lors de la mise en oeuvre permettra la communication entre les deux systèmes si elle a bien été suivie.

L'utilisation de l'architecture en couches permet de modifier une couche sans devoir changer l'ensemble des couches. On peut ainsi s'adapter facilement à l'évolution technique.

Dans ce modèle, il existe une communication virtuelle entre couches homologues des deux partenaires qui dialoguent. Pour obtenir cette communication virtuelle, la couche supérieure utilise la couche directement inférieure. Ainsi, la couche Liaison

de données utilise la couche Physique. Pour que les couches Applications communiquent de manière virtuelle, il faut que l'information passe de la couche 7 à la couche 1 de l'émetteur puis de la couche 1 à la couche 7 du récepteur.

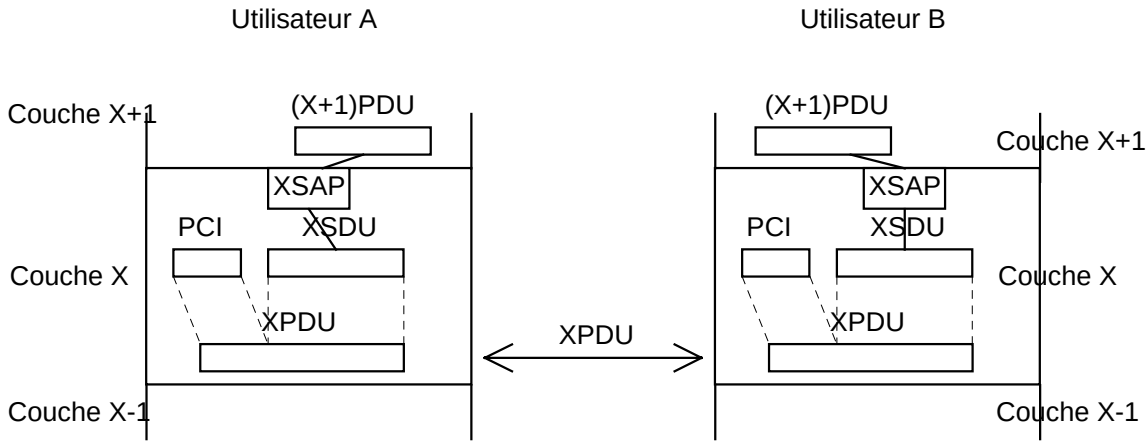
A. Quelques définitions

Les **Protocoles** sont les procédures gérant les communications entre couches homologues.

Les **Services** représentent l'ensemble des moyens fournis à une couche supérieure par une couche inférieure afin que la couche supérieure puisse exécuter ses protocoles.

L'accès par une couche supérieure aux services d'une couche inférieure se fait par l'intermédiaire des **SAP** ou **S**ervice **A**ccess **P**oints.

Les informations que la couche supérieure désire transmettre sont mises dans le **SDU** ou **S**ervice **D**ata **U**nit de la couche inférieure.



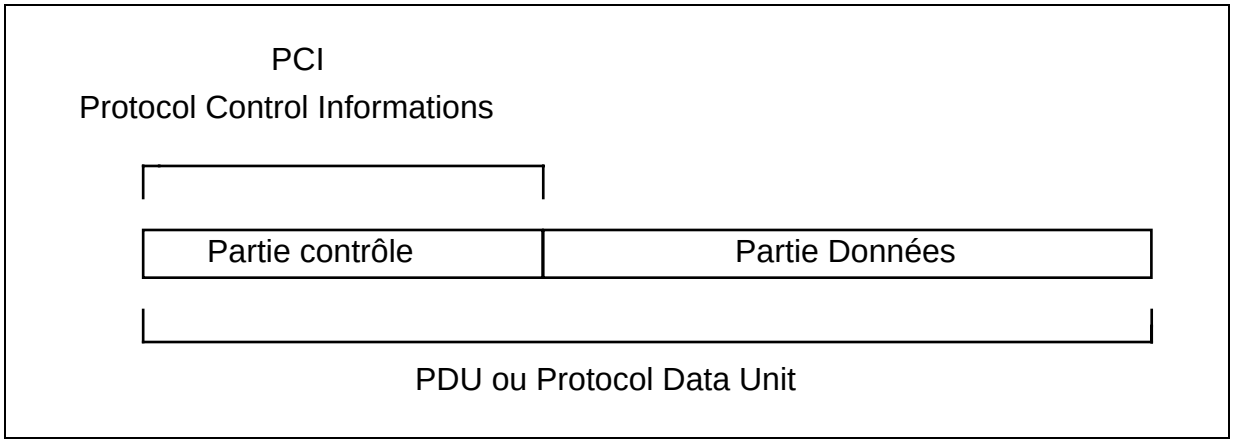
Entre couches homologues s'échangent des Unités de données de Protocole (PDU)

La communication entre couches homologues peut se faire sur base:

en **mode connecté** : une connexion X-1 doit être établie par le service X-1entre un (X-1)SAP de la couche X appelante et (X-1)SAP de la couche X appelée.

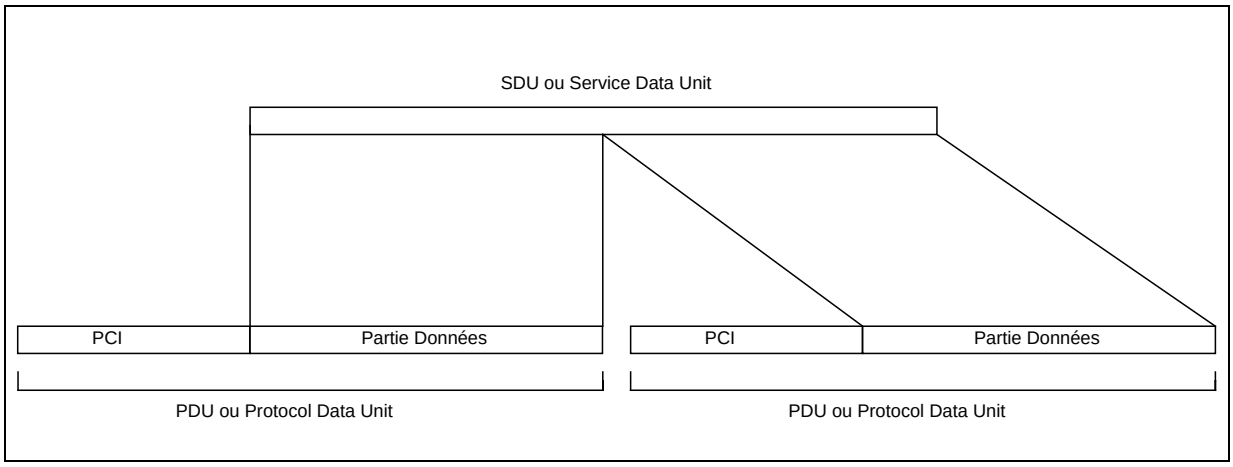
❑ en **mode non connecté**

Les couches homologues s'échangent des messages appelés **PDU** ou **P**rotocol **D**ata **U**nit. Ceux-ci sont composés d'une partie de contrôle (**PCI** ou **P**rotocol **C**ontrol **I**nformations) et d'une partie de données.

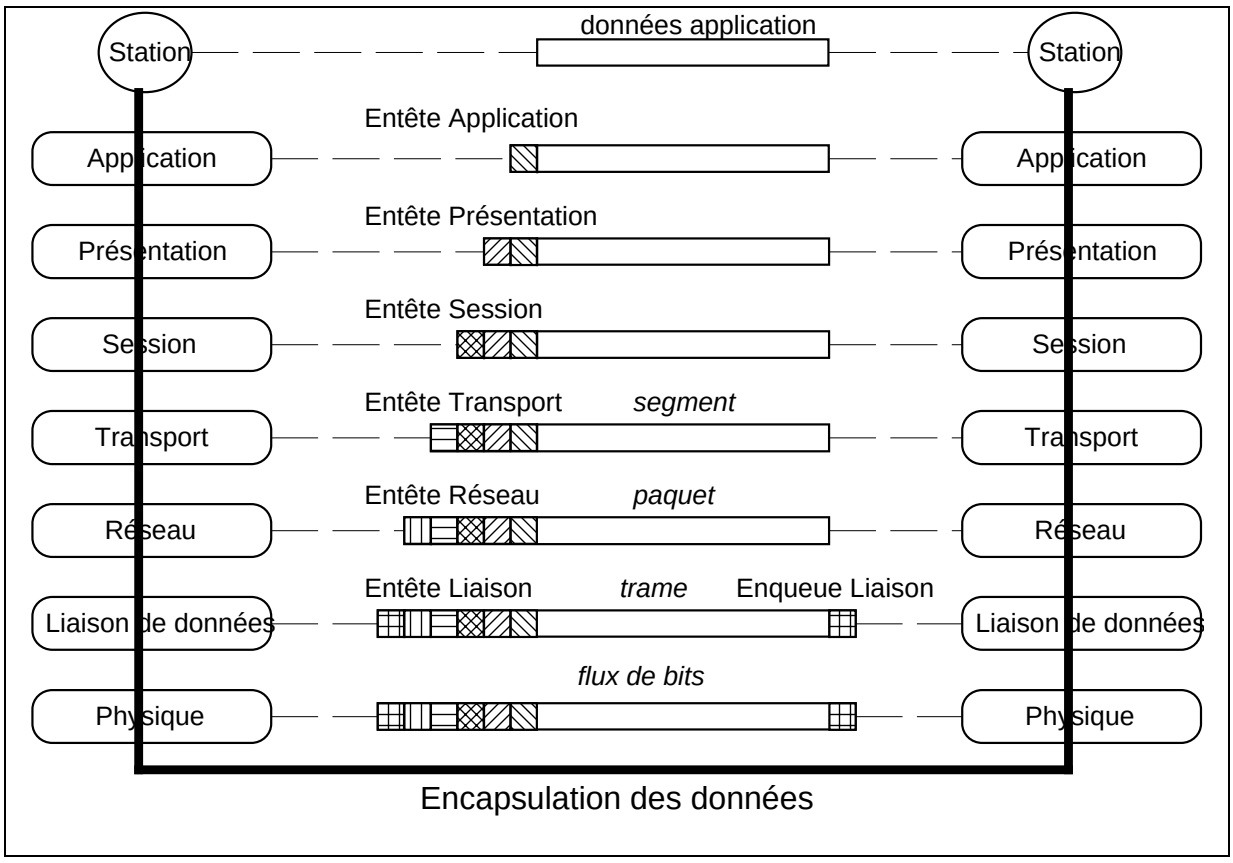


Les informations provenant d'une couche supérieure sont intégrées dans les données de la couche inférieure. La couche inférieure n'interprète donc pas les données de la couche supérieure.

Un SDU peut être transmis par l'intermédiaire de plusieurs PDU lorsque la taille du SDU est plus grande que la partie de données du PDU.



Lors du passage d'une couche supérieure vers une couche inférieure, le nombre d'informations à transmettre augmente puisque des informations de contrôle peuvent être ajoutées par certaines couches. Le nombre d'informations diminue par contre lors du passage d'une couche inférieure vers une couche supérieure.



Encapsulation des données

B. Les 7 couches du modèle OSI

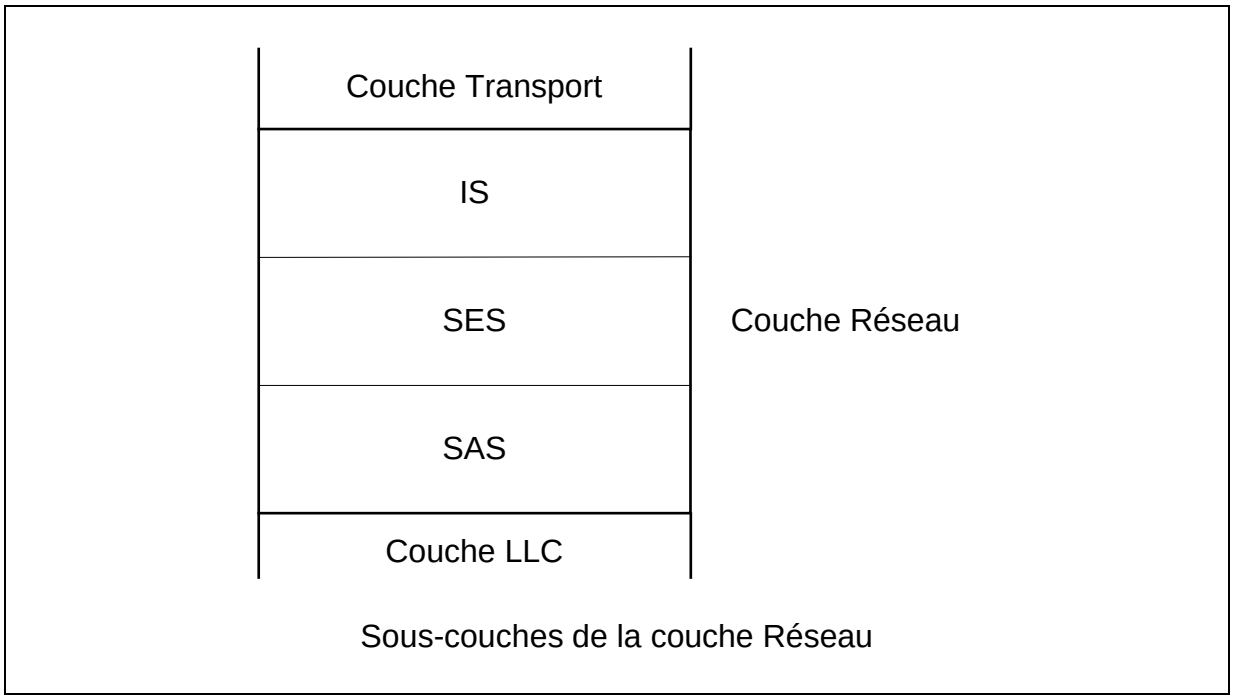
Le modèle se compose donc de 7 couches :

La couche 1 ou couche **physique**. Celle-ci comporte les aspects de câblage du système de communication, de connexions physiques et de transmission sur le support des données. Les informations échangées sont des bits. Exemple : RS232C, X.21

La couche 2 ou couche **liaison de données**. Elle veille à transmettre les informations d'un noeud à l'autre avec détection, correction des erreurs éventuelles. Dans le cas des réseaux locaux, celle-ci est décomposée en deux sous-couches. Ces couches sont la couche 2a ou **Media Access Control** (MAC) et la couche 2b ou **Logical Link Control**. La couche 2a s'occupe de la manière d'accéder au support physique et la couche 2b veille à l'échange des informations entre noeuds. Les informations échangés entre les couches liaison de données d'un réseau de communication sont appelés **trames** (ou **LPDU**).

La couche 3 ou couche **réseau**. Elle se charge du routage et de l'adressage des informations entre systèmes lorsque celles-ci doivent transiter par d'autres noeuds pour passer du système émetteur au système récepteur (rôle réduit ou inexistant dans les réseaux locaux). Les informations circulant entre les couches réseau sont appelées **paquets** (ou **NPDU**). Elle dispose de mécanismes de segmentation/réassemblage qui permettent de transmettre des NSDU en utilisant plusieurs paquets.

La couche Réseau est subdivisée en trois sous-couches :
♦ la couche **SAS** ou **S**ubnet **A**ccess **S**ublayer qui utilise le **SNACP** ou **S**ub**n**etwork **Ac**cess **P**rotocol.
♦ la couche **SES** ou **S**ubnet **E**nhancement **S**ublayer qui utilise **SNDCP** ou **S**ub**n**etwork **D**ependent **C**onvergence **P**rotocol
♦ la couche **IS** ou **I**nternet **S**ublayer qui utilise **SNICP** ou **S**ub**n**etwork **I**ndependent **C**onvergence **P**rotocol



Sous-couches de la couche Réseau

La sous-couche **SAS** gère l'accès au sous-réseau réel particulier et le routage dans le sous-réseau.

La sous-couche **SES** est destinée à fournir une série bien précise de fonctions à la sous-couche IS au départ des fonctions de la sous-couche SAS. Elle assure donc un découplage entre les caractéristiques du sous-réseau réel et le fournisseur de service Réseau OSI.

La sous-couche I**S** fournit le service Réseau OSI à la couche Transport en utilisant une série de fonctions bien précises offertes par la sous-couche SES. On remarque donc que la sous-couche IS est relativement indépendante des caractéristiques du sous-réseau réel.

Cette subdivision en sous-couches est nécessaire lors de l'emploi de sous-réseaux de type différent. Un exemple consiste en l'emploi, par des sous-réseaux ISO séparés géographiquement, de X25 (réseau téléphonique) pour échanger des données.

Vu que la couche réseau assure le routage des paquets, c'est elle qui s'occupe de l'adressage. L'adresse d'un point d'accès au niveau de la couche Réseau s'appelle **NSAP** (**N**etwork **S**ervice **A**ccess **P**oint)

La couche 4 ou couche **transport**. Elle veille au contrôle du transport des informations de bout en bout au travers du réseau. Elle doit donc veiller à l'acheminement sans erreur des messages entre l'émetteur et le récepteur. Les informations échangées entre couches transport sont appelées **TPDU** (**T**ransport **P**rotocol **D**ata **U**nit). Elle dispose de mécanismes de segmentation/réassemblage (TSDU -> TPDUs et TPDUs -> TSDU), de multiplexage et de démultiplexage, de contrôle de flux pour éviter la saturation des systèmes de communications. L'accès aux services de la couche Transport se fait au travers d'un **TSAP** (**T**ransport **S**ervice **A**ccess **P**oint).

La couche 5 ou couche **session**. Elle assure les fonctions d'initialisation, de synchronisation et de terminaison du dialogue entre applications. Par exemple, en cas de coupure de la communication, la couche Session fournit les moyens de reprendre la communication à des endroits bien précis (point de synchronisation). L'accès aux services de la couche Session se fait au travers d'un **SSAP** (**S**ession **S**ervice **A**ccess **P**oint). **On retiendra qu'une connexion Session correspond à une seule connexion Transport**.

La couche 6 ou couche **présentation** fournit les services principaux suivants :

- ♦ services de session
- ♦ transformation de l'information et transcodage, encryptage/c ompression
- ♦ négociation de la syntaxe
- ♦ modification de contexte.

L'accès aux services de la couche Présentation se fait au travers d'un **PSAP** (**P**resentation **S**ervice **A**ccess **P**oint). **De plus, une connexion Présentation correspond à une seule connexion Session.**

Voyons maintenant ces fonctions de manière plus détaillée.

La couche Présentation code le langage spécifique à chaque système en un code standard propre au réseau et retransforme ce code standard en un langage propre à chaque application. Elle assure donc ainsi la compatibilité entre tous les matériels connectés au réseau.

Le code standard est appelé **syntaxe de transfert**. La conversion entre la syntaxe propre à l'utilisateur (syntaxe d'application) et la syntaxe de transfert est effectuée au niveau de la couche de Présentation. La couche Présentation peut supporter plusieurs syntaxes de transfert. Les couches Présentation des correspondants doivent opérer le choix d'une syntaxe de transfert en fonction des syntaxes de transfert disponibles de part et d'autre et en fonction de l'application que les partenaires utilisent (une syntaxe de transfert peut être mieux adaptée à certaines syntaxes d'application).

Dès qu'une syntaxe de transfert a été choisie, il suffit de connaître la syntaxe concrète effectivement employée au niveau de la couche d'application pour définir le convertisseur de syntaxe à utiliser au niveau de la couche de présentation.

L'association entre une syntaxe d'application et une syntaxe de transfert est appelée **contexte de présentation**. La couche Session offre la possibilité à une application de changer de contexte

La couche d'application peut maintenir simultanément plusieurs contextes de présentation. L'ensemble des contextes de présentation définis, **DCS**, (**D**efined **C**ontext **S**et) représente toutes les possibilités de conversion de syntaxe qui sont disponibles à un instant donné sur une connexion de présentation.

En fait, la spécification de la syntaxe concrète utilisée par la couche application n'est pas simple. En effet, la syntaxe concrète d'application dépend entièrement de l'implantation : une même entité application pourrait donc exister sous un grand nombre de variantes possibles.

De plus, au niveau de la couche d'application, les données sont interprétées pour leur contenu sémantique, pour leur signification. Ceci amène à définir à ce niveau des objets nombreux et complexes.

Il est alors intéressant de décrire ces objets de façon abstraite afin que la syntaxe d'application soit facilement comprise par les utilisateurs, et afin que ces derniers n'aient pas à se préoccuper des détails d'implantation.

L'obtention de la syntaxe concrète d'un objet propre à une implantation (une plate-forme) est réalisée par la compilation de la description de l'objet, description qui utilise une syntaxe abstraite d'application.

Toutes les syntaxes abstraites d'application sont construites au départ d'une notation normalisée dite **notation de syntaxe abstraite ASN.1**

Actuellement, il n'existe qu'une seule syntaxe de transfert normalisée(X.209) qui porte le nom de **spécification des règles de codage pour la notation de syntaxe de abstraite ASN.1** (Specification of Basic Encoding Rules for Abstrast Syntax Notation One, ASN.1).

Cette syntaxe de transfert est en fait un ensemble de règles qui spécifient le transcodage d'une syntaxe d'application décrite avec ASN.1. Ces règles précisent comment les différentes structures de données doivent être représentées pour la communication. Dans cet environnement, un contexte représente l'application de ces règles de codage à une syntaxe d'application particulière définie avec ASN.1.

En résumé, on peut dire qu'actuellement, il existe plusieurs protocoles d'applications auxquels correspondent plusieurs syntaxes abstraites d'application. Ces syntaxes abstraites d'application sont représentées par la même notation, ASN.1 et sont converties en une syntaxe de transfert unique.

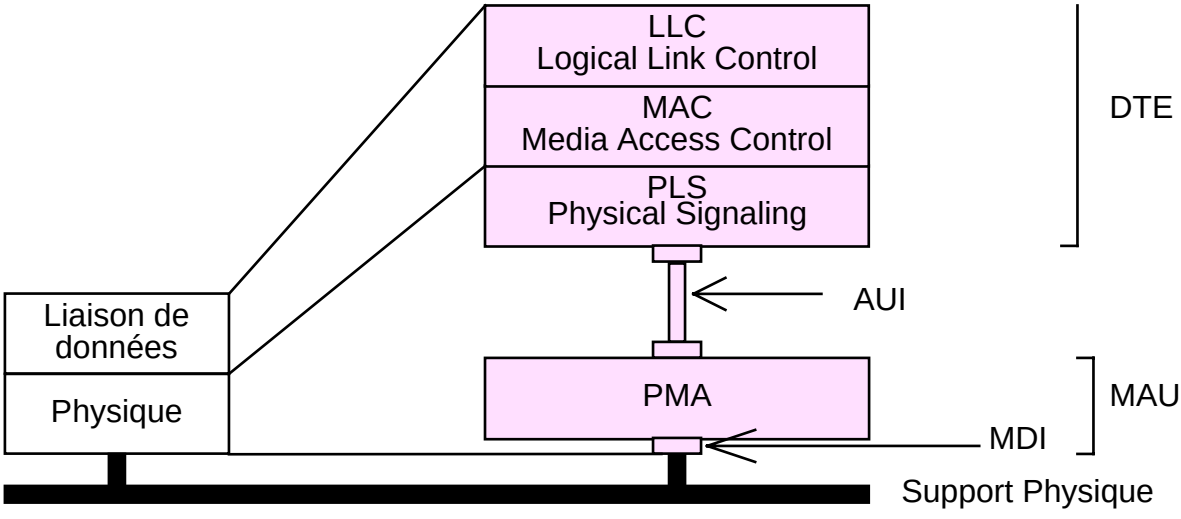
La couche 7 ou couche **application**. Elle fournit des services standardisés pour les systèmes d'application. Par exemple :

- ♦ identification des partenaires communicants et authentification
- ♦ synchronisation des applications coopérantes
- ♦ transfert de données ...

C. Le modèle OSI et les réseaux locaux

Dans le cas des réseaux locaux, les normes ISO reprennent les normes IEEE (Institute of Electrical and Electronics Engineers). Les travaux de l'IEEE ont conduit à la subdivision des couches physique et liaison de données en plusieurs sous-couches.

La norme ISO 8802.3 correspond à IEEE 802.3.



AUI ou Attachment Unit Interface
MAU ou Medium Attachment Unit
MDI ou Medium Dependent Interface
PMA ou Physical Medium Attachment
DTE ou Data Terminal Equipment

Subdivision des couches OSI 1 et 2 dans le cas de ISO 8802.3

La couche **MAU** permet d'attaquer le support physique, de transmettre les signaux entre le support physique et la couche PLS au travers de l'AUI.

Les sous-couches **AUI** et **PLS** permettent la connexion et le transfert d'informations entre le MAU et la couche MAC de l' équipement Terminal de traitement de données (DTE).

D. Les systèmes d'interconnexion de réseaux

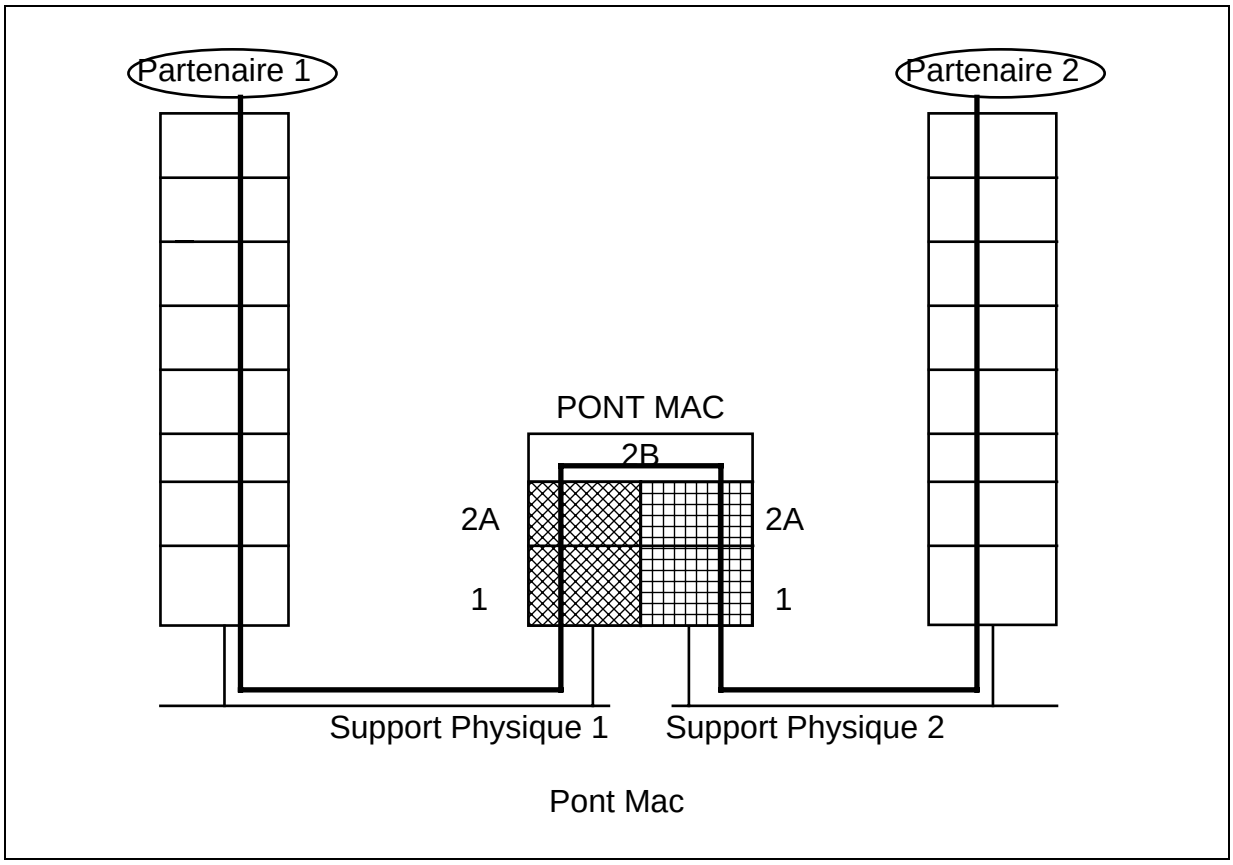
1. Les Répéteurs

Dans les réseau locaux, les systèmes qui régénèrent les signaux sur le support physique sont soit des **Répéteurs** ou soit des **Amplificateurs**. Ce type de système veille à transmettre le signal sur l'ensemble des segments (câbles) qui forment le réseau local. Ce n'est donc pas réellement un système d'interconnexion de réseau. Il n'assure **pas un filtrage** des trames.

2. Les Ponts

On appelle **Pont MAC (MAC bridge)** le système qui relie, au niveau de la couche MAC, des réseaux locaux dont les supports physiques et le principe d'accès peuvent être différents. Le Pont MAC peut donc assurer l'adaptation de couches physique et MAC différentes. Il transpose les trames d'un type MAC vers un autre. Il permet donc la communication entre systèmes qui disposent des mêmes couches 2b et supérieures mais qui n'utilisent pas forcément les mêmes support physique et méthode d'accès au support.

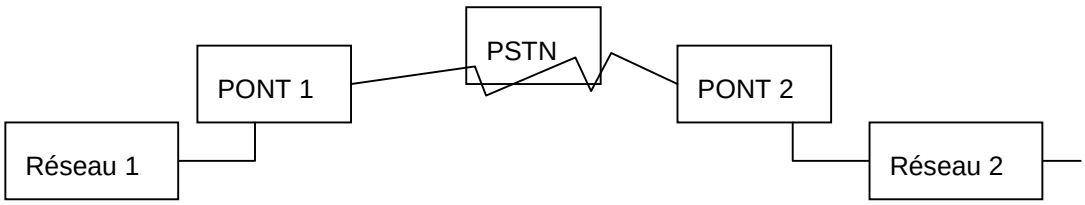
Comme il décode les trames MAC, il dispose donc des adresses source et destination. Et, vu qu'il connaît les adresses des stations connectées sur chaque réseau local, il peut donc en fonction des adresses transmettre ou non la trame d'un support physique vers l'autre. Il assure donc un **filtrage** des trames. **La présence d'un pont est transparente à l'utilisateur**. Il **transmet les trames de diffusion** (broadcast).



Voici quelques exemples de pont ou bridge :

pont qui relie un réseau Ethernet (MAC IEEE802.3) et un réseau Token RING (MAC IEEE 802.5) : les couches MAC diffèrent.

pont qui permet relier des réseaux Ethernet au travers du réseau téléphonique via une ligne analogique, une ligne RNIS.

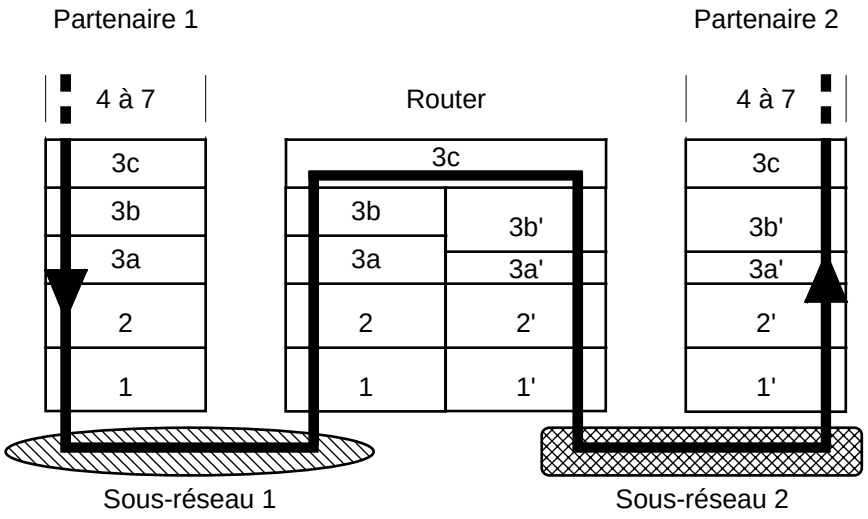


Ici, la création de la liaison entre les réseaux (appel du pont distant au travers du réseau téléphonique) n'a lieu que si des trames doivent être transmises entre le réseau 1 et le réseau 2. Le pont distant appelé est toujours le même.

3. Les Routeurs

On appelle **Routeurs (Router)** les systèmes qui permettent de relier des sous-réseaux et qui veillent à acheminer les paquets depuis l'émetteur jusqu'au destinataire au travers des différents sous-réseaux qui séparent les deux partenaires. Les sous-réseaux sont en communication au niveau de la couche réseau. Le Routeur amène aussi un effet de **filtrage** des paquets et le choix d'un routage optimal. Il est à noter que le Router n'est **pas transparent à l'utilisateur**. Les paquets qui doivent passer d'un sous-réseau vers un autre doivent être envoyés en indiquant le router qui doit être utilisé pour passer vers l'autre réseau.

Le routeur permet aussi de passer d'un type de sous-réseau vers un autre.



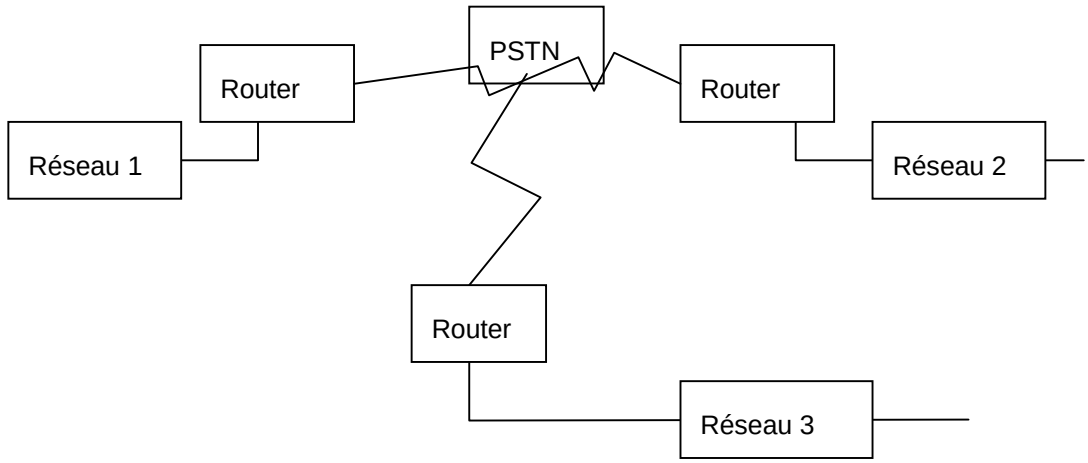
Router reliant des sous-réseaux de types différents

Dans certaines publications, le router porte aussi le nom de passerelle.

Voici quelques exemples de routeur :

un routeur qui relie un réseau Ethernet (MAC IEEE802.3) et un réseau Token RING (MAC IEEE 802.5).

routeur qui permet relier des réseaux Ethernet au travers du réseau téléphonique via une ligne analogique, une ligne RNIS, une ligne louée.

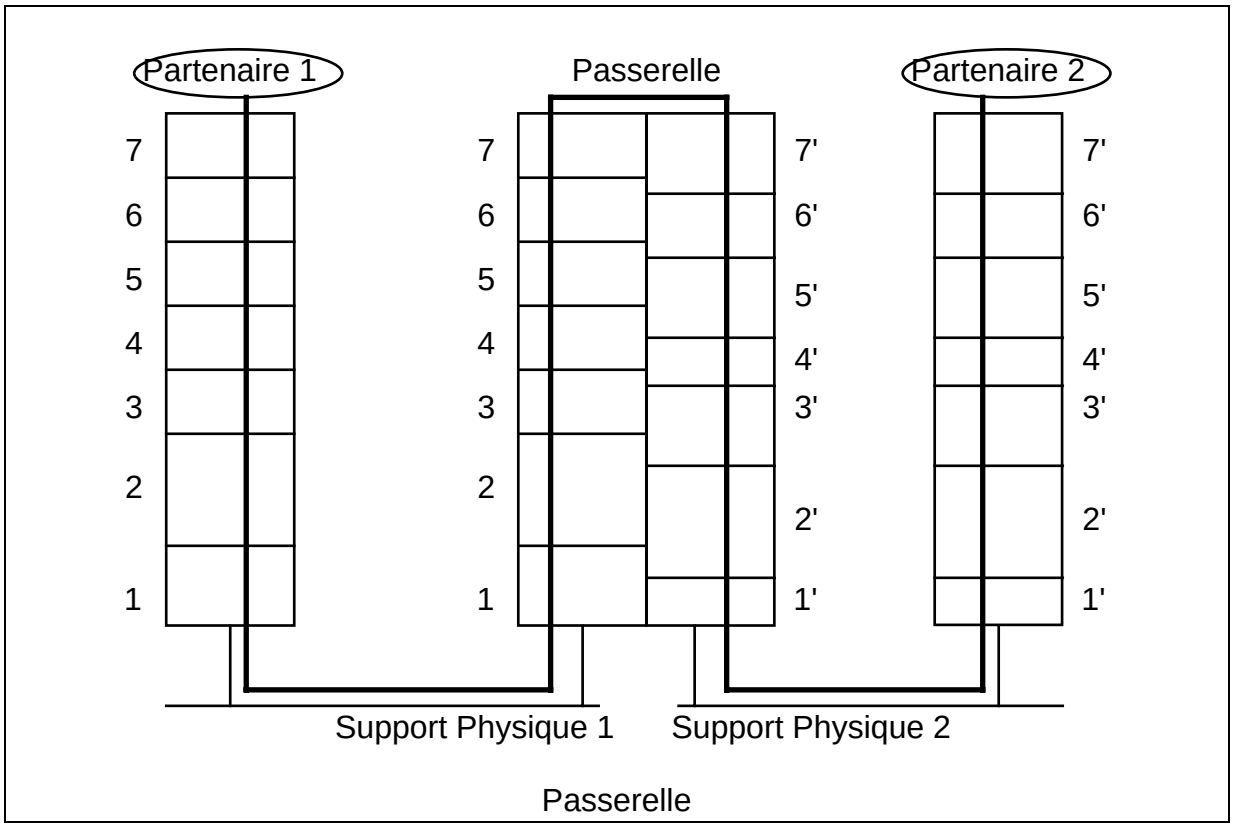


Ici, la création de la liaison entre les réseaux (appel d'un routeur distant au travers du réseau téléphonique) n'a lieu que si des paquets doivent être transmis vers un des autres réseaux. Chaque routeur sait en fonction de l'adresse réseau du destinataire quel routeur distant doit être appelé pour établir une connexion et envoyer le paquet. Le routeur est donc bien plus intelligent que le pont.

4. Les Passerelles

On appelle **Passerelle (Gateway)** le système qui permet la communication entre systèmes qui n'utilisent pas les mêmes protocoles dans les 7 couches OSI. Le système décode les informations au niveau de la couche application. La passerelle doit donc disposer des deux piles de protocoles.

Un exemple de Gateway est le service Netware Gateway de Windows NT Server : il permet à des stations de travail n'utilisant que les protocoles réseau Microsoft d'accéder aux services des serveurs Novell Netware qui demandent normalement disposer des protocoles réseau Netware. Les stations de travail accèdent aux serveurs Novell Netware au travers du serveur Windows NT comme si ce sont des ressources Windows NT.



III.DIFFERENTS TYPES DE RESEAUX

A. Réseau Ethernet ou IEEE 802.3

L'accès au support physique se fait sur base du principe CSMA/CD (Carrier Sense Multiple Access / Collision Detect). Il n'existe pas de maître dans le réseau : émet qui veut.

1. Principe du CSMA/CD

L'émetteur désirant envoyer une trame d'informations écoute la ligne. Si elle est libre, il émet la trame d'informations tout en comparant ce qu'il émet avec ce qu'il entend sur la ligne. Si une différence existe, c'est qu'une autre station a émis en même temps que lui. Il y a collision. L'émetteur arrête l'émission de la trame tout en veillant à émettre un nombre minimum d'octets de bourrage (jam) pour que les autres stations puissent détecter la collision. En effet, il faut tenir compte du temps de propagation de la trame sur le support physique. L'émetteur tentera d'émettre sa trame à nouveau après l'écoulement d'une temporisation dont la valeur varie de manière aléatoire tout en étant que la durée de la temporisation augmente de manière exponentielle en cas de collisions successives.

Le nombre maximum de reprises après collision est de 16. Lorsque ce nombre est dépassé, la couche 2a indique à la couche supérieure qu'elle n'a pas su transmettre ses données.

Lorsqu'un émetteur a plusieurs trames à envoyer, il laissera au moins un délai de 9,6 µs entre l'envoi de deux trames consécutives.

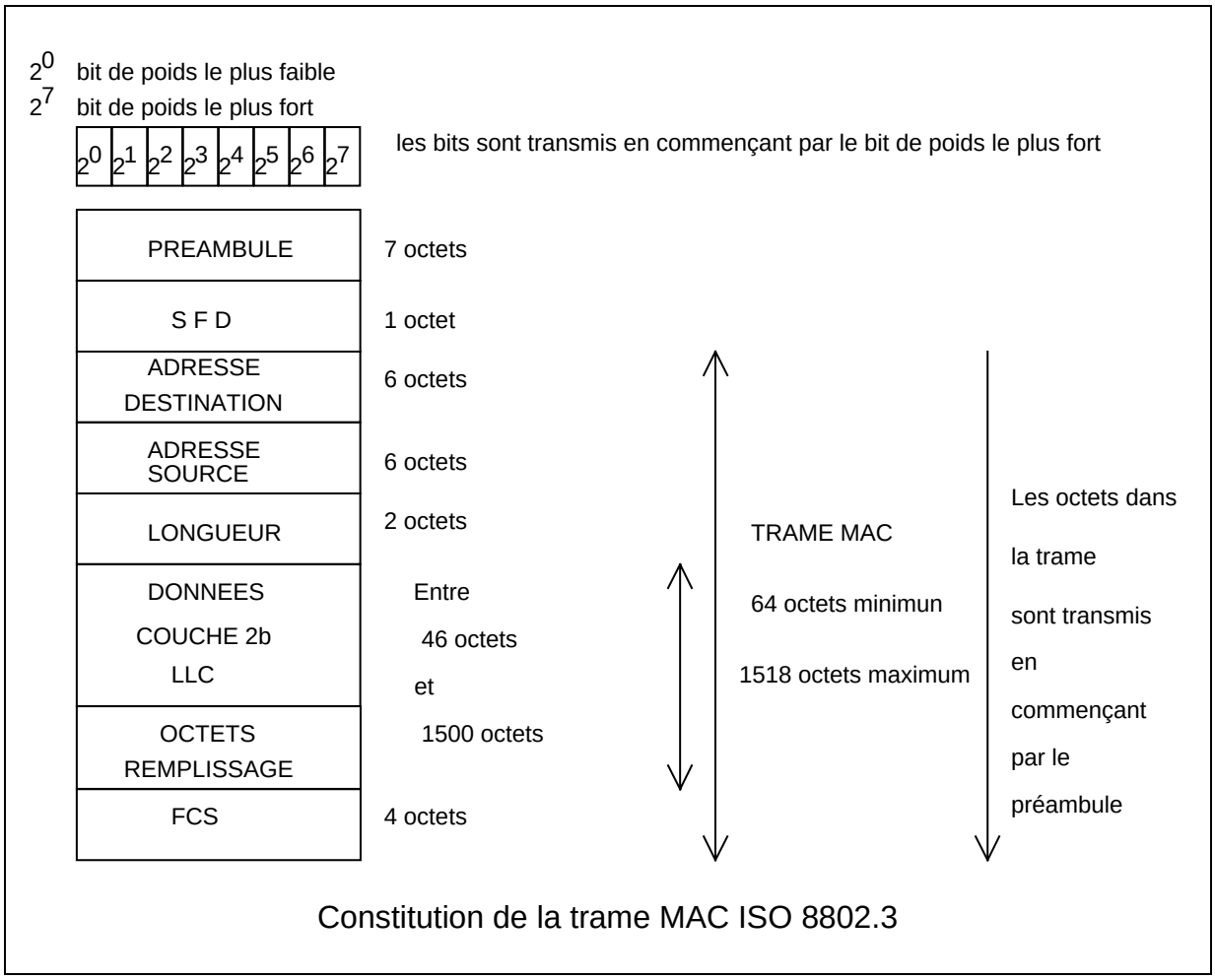
On remarquera que les performances du réseau vont en décroissant pour une charge croissante. De plus, si le réseau est trop chargé, il risque de se présenter trop de collisions qui vont rendre le réseau inutilisable.

Dans ce réseau, les partenaires se connaissent par l'intermédiaire d'une adresse codée 6 octets. Il existe un organisme (IEEE) chargé d'affecter les adresses aux cartes de communications. Cet organisme garantit l'unicité des adresses des cartes de communication.

2. Constitution de la trame

La trame doit avoir une taille minimale inhérente au principe CSMA/CD. La norme prévoit 64 octets.

La taille maximale de la trame a été arrêtée à 1518 octets.



Constitution de la trame MAC ISO 8802.3

La trame MAC se compose de huit parties distinctes :

Le champ préambule

Il est constitué de 7 octets représentant une suite de 10101...10. Le préambule permet au récepteur de se synchroniser sur la fréquence d'émission. Le préambule se termine avec 0.

Le champ SFD (Start Frame Delimiter)

Ce champ indique au récepteur le début des données MAC. Il a pour valeur 10101011.

Le préambule et le SFD ne font pas réellement partie de la trame. Les longueurs minimales et maximales de trames évoquées précédemment correspondent à l'ensemble des 6 champs qui suivent.

Les champs Adresse destination et Adresse source.

Ils ont chacun une taille de 2 ou 6 octets et indiquent donc respectivement le(s) destinataire(s) de la trame et l'émetteur de la trame. Les adresses des stations peuvent être gérées globalement ou localement. En utilisant les adresses globales, on est certain que chaque station dispose d'une adresse différente (prévu par la norme). Quand on utilise les adresses gérées localement, l'utilisateur doit veiller à ce que chaque station ait une adresse différente. Le bit 1 du premier octet de l'adresse indique que l'adresse est gérée localement lorsqu'il vaut 1.

L'adresse destination peut être de trois types :
♦ une adresse individuelle,
♦ une adresse de groupe (le bit 0 du premier octet de l'adresse est à 1) : la trame est envoyée à un groupe de destinataires,
♦ l'adresse broadcast (composée de 1 uniquement).

L'adresse émetteur est toujours individuelle.

Le champ de longueur

Ce champ a une longueur de 2 octets et contient la longueur de la zone de données provenant de la couche 2b ou LLC. Le premier octet représente l'octet de poids fort de la longueur. La longueur varie entre 0 et 1500.

La zone de données

Celle-ci a une longueur minimale de 46 octets et une longueur maximale de 1500 octets. Lorsque le nombre de données à transmettre est inférieur à 46 octets, il y a remplissage de la zone pour obtenir une taille minimale de 46 octets.

Le champ FCS (Frame Check Sequence ou séquence de contrôle de trame)

Ce champ comporte un CRC-32 (Cyclic Redundancy Checksum) contenu dans 4 octets. Il permet de détecter les erreurs de transmission (signal endommagé). Ce CRC est calculé sur les champs d'adresses, de longueur et de zone données grâce au polynôme générateur :

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

3. Supports physiques.

a) Généralités

Le réseau compte 1024 stations au maximum.
Il se compose de segments qui sont interconnectés par des répéteurs (dispositif qui régénère les signaux transmis).

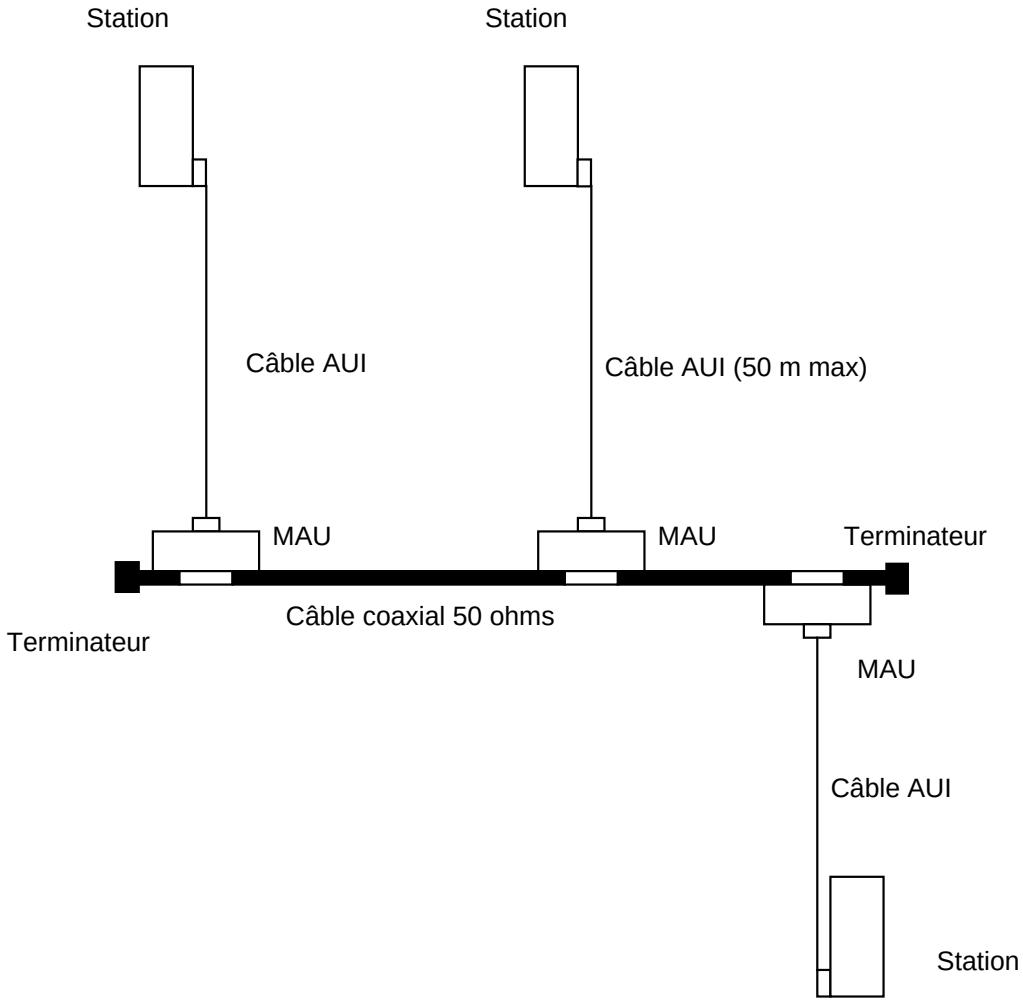
Règle de base

Deux partenaires quelconques du réseau peuvent être séparés par un maximum de 5 segments (dont au maximum 3 sont des segments de type coaxial) et par quatre répéteurs. Les autres segments (Link Segment) sont des segments en fibre optique ou paire torsadée.

b) 10 base 5 (Yellow Cable ou Ethernet industriel)

- 10 Mbit/s
- Câble coaxial 50 ohms. (il existe des fournisseurs qui proposent du câble comportant une enveloppe métallique supplémentaire pour une meilleure protection contre les perturbations électromagnétiques)
- Vitesse de propagation sur le câble > 0,77c
- Segment de 500 mètres au maximum.
- Résistance de terminaison (50 ohms) à chaque extrémité pour éviter les réflexions. Si un segment du réseau est sectionné ou non "terminé", l'ensemble du réseau sera en défaut
- Rayon de courbure minimum 20 cm

- La connexion au segment se fait par l'intermédiaire d'un transceiver (MAU ou Medium Access Unit). La station se connecte au transceiver par l'intermédiaire d'un drop cable (AUI ou Attachment Unit Interface). C'est le transceiver qui détecte les collisions. Le drop cable peut avoir une longueur maximale de 50 m. Il existe deux types de transceivers soit à visser (connecteur N : le réseau est inactif pendant l'ajout de ce type de transceiver) soit vampire (ajout sans interruption du trafic sur le réseau).
- Un Segment peut comporter un maximum de 100 transceivers
- La distance maximale qui peut séparer deux partenaires du réseau est de à peu près 2500 mètres si on ne tient pas compte des drop cable
- Les transceivers doivent être séparés de 2,5 m au moins.
- Les parties Métalliques doivent être isolées.
- Chaque segment doit être mis à la terre en un seul endroit
- Un segment peut être composé de plusieurs morceaux de câbles reliés par des manchons (barrel connector). On veillera à avoir des longueurs de morceaux de câble de 23,4 m, 46,8 m , 70,2 m, ...
- Il existe des concentrateurs qui se connectent à un transceiver pour permettre de connecter 8 stations par l'intermédiaire d'un Transceiver.



Réseau à un segment

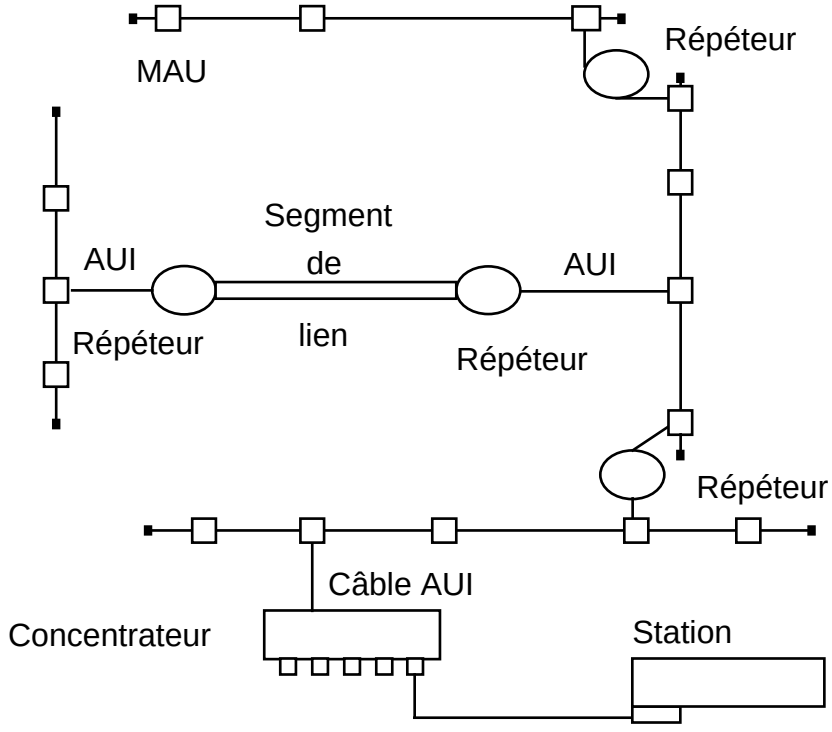
Le MAU se charge de transférer les signaux entre la station et le câble coaxial et de détecter les collisions. Il doit être alimenté par du 15 V puisqu'il comporte des éléments actifs.

Le câble AUI est un câble comportant 4 paires torsadées (signaux émission, réception, détection de collision et alimentation 15 V pour MAU). La norme prévoit une paire de fils supplémentaire optionnelle.

Les correspondants du réseau ne peuvent être séparés l'un de l'autre que:
♦ par deux répéteurs s'il ne sont séparés par aucun segment de lien,
♦ par un répéteur et une ou deux paires de répéteurs (un ou deux segments de lien les séparent).

Ainsi, la distance maximale entre deux stations sera de ± 1,5 km si aucun segment de lien ne les sépare ou de ± 2,5 km si un ou deux segments de lien les séparent.

Il est à noter qu'un concentrateur peut constituer un réseau en lui-même s'il n'est pas connecté à un MAU. Le réseau sera alors composé des stations connectées sur le concentrateur.



Réseau avec répéteurs et concentrateur

c) 10 base 2 (Cheapernet ou Thin Ethernet)

- 10 Mbit/s.
- Câble coaxial 50 ohms (RG-58).
- Vitesse de propagation sur le câble > 0,65c.
- Segment de 185 mètres au maximum.
- Résistance de terminaison (50 ohms) à chaque extrémité pour éviter les réflexions. Si un segment du réseau est sectionné ou non "terminé", l'ensemble du réseau sera en défaut.

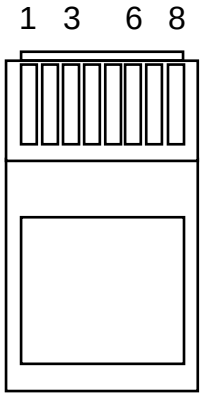
- Rayon de courbure minimum 4 cm.
- La connexion au segment se fait par l'intermédiaire d'un T BNC qui doit directement se fixer sur la carte réseau de la station. Le Transceiver est intégré à la carte réseau.
- Un Segment peut comporter un maximum de 30 connecteurs T ou manchons.
- Les connecteurs T doivent être séparés de 0,5 m au moins.
- Les parties métalliques doivent être isolées.
- Chaque segment doit être mis à la terre en un seul endroit.
- Coût réduit.
- Plus grande sensibilité aux P.E.M.
- Adapté aux environnements bureautiques.

d) 10 Base T (actuel)

- 10 Mbit/s.
- Utilise une double paire torsadée non blindée téléphonique
- Vitesse de propagation 0,585c.
- 100 mètres par segment.
- câble catégorie 5 impédance 100 $\Omega \pm 15\ \Omega$ de 1 à 100 MHz (4 paires)
- Une station par segment.
- Les segments se raccordent à un concentrateur ou hub qui est aussi un répéteur. Ces hubs peuvent être gérables ou non.
- Installation très simple.
- Un maximum de 5 segments ou 4 hubs peuvent séparer deux stations quelconques du réseau.
- Plus grande sensibilité aux P.E.M.
- Adapté au environnement bureautique.
- connecteur RJ-45

Câblage d'un Câble TP sur un connecteur RJ45

N° contact du connecteur	Couleur du fil	Signal
1	Blanc-Orange	TD+
2	Orange	TD-
3	Blanc-Vert	RD+



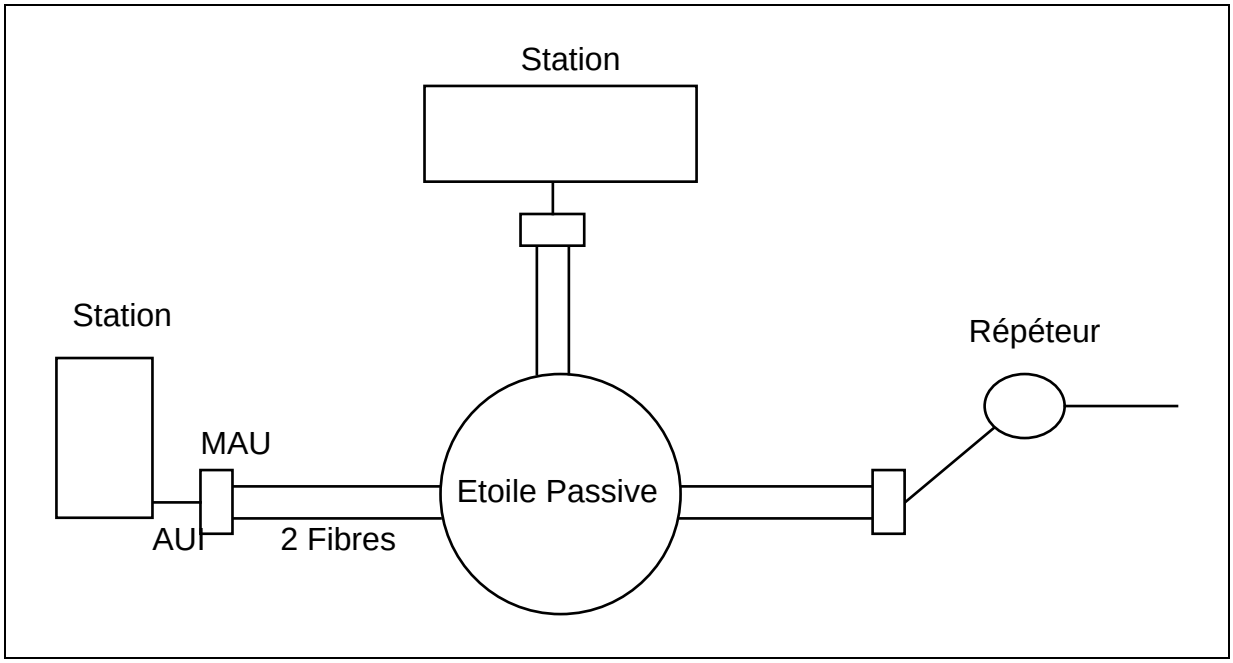
4	Bleu	inutilisé
5	Blanc-Bleu	inutilisé
6	Vert	RD-
7	Blanc-Brun	inutilisé
8	Brun	Inutilisé

e) 10 base F

- Longueur d'onde : 800 -> 910 nm 850 nm nominal.
- Fibre 62,5/125 nm.
- Bande passante : 160 MHz.km à 850 nm.
- Vitesse de propagation : 0,67c.

(1) base FP

Cette norme est utilisée dans le cas d'une étoile passive (pas d'amplification)



Fibre de 500 m maximum entre l'étoile et chaque MAU : on peut connecter un répéteur ou une station.

33 Stations sur une étoile passive.

(2) base FB

Cette norme a été définie pour créer des backbones (dorsale) en FO .

Elle prévoit de relier des répéteurs sur une distance d'au moins 2000m.

A l'état de repos, un signal synchrone à 2,5 MHz est émis sur chaque fibre par chaque répéteur.

En cas d'erreur, un signal à 1,66 MHz est transmis.

(3) base FL

Permet de relier un DTE à un DTE ou un répéteur multiports à une étoile.
Distance minimale 2000 m.

f) FOIRL = Fiber Optic Inter Repeater Link

Cette norme est destinée à permettre de relier des répéteurs sur une distance de 1000 m maximum.

4. Pont IEEE 802.3 Ethernet

L'interconnexion de plusieurs réseaux IEEE 802.3 peut se faire par l'intermédiaire de bridges ou pont. Les bridges comportent plusieurs (2 au moins) interfaces réseau

appelées ports. Les bridges ne se basent que sur les adresses MAC pour connaître la destination d'une trame.

Ces bridges vont utiliser le Transparent Spanning Tree pour assurer leur configuration automatique.

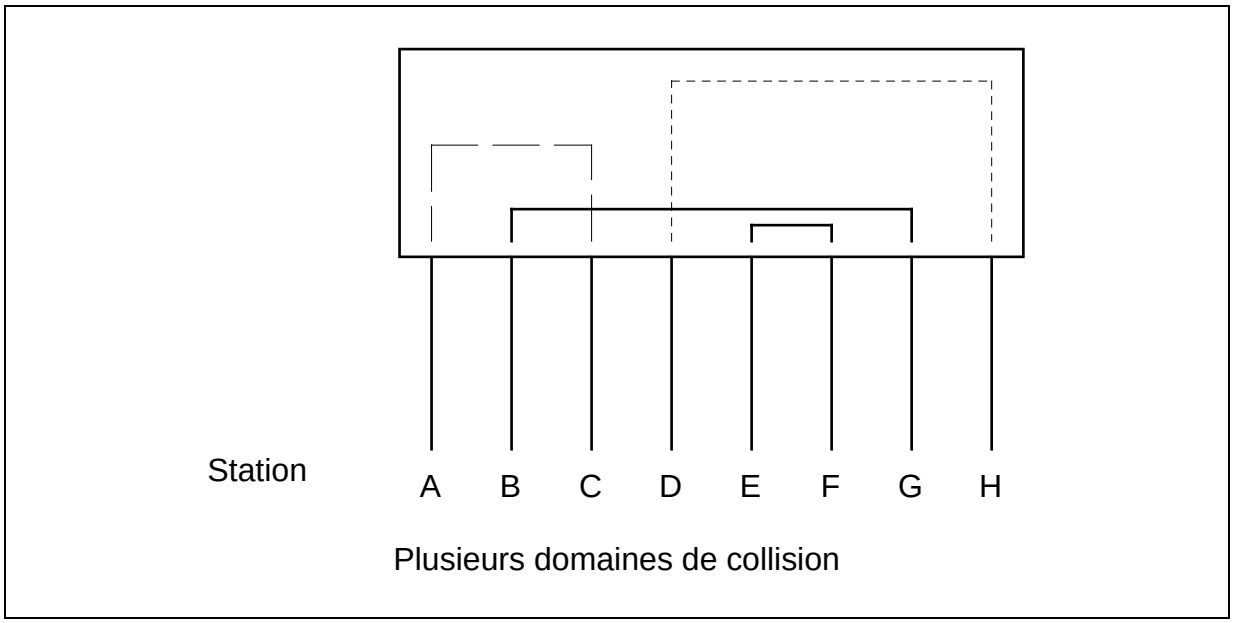
Ce type de bridge utilise le principe de self-learning (auto-apprentissage). Dans chaque bridge, il existe donc une table qui retient l'association entre l'adresse MAC et le port du bridge ainsi que le moment de la mise à jour de chaque ligne de la table. Cette table ne contient pas les adresses "Broadcast" ni "Multicast". Les adresses qui n'ont pas été vues depuis un certain temps sont enlevées de la table. Cette période de temps est appelée "Aging Time" : 10 < Aging Time < 11,57 années par incrément de 1 seconde . La valeur par défaut est généralement 300 secondes. La base de données peut contenir des entrées dynamiques et statiques. Les entrées dynamiques sont celles qui sont gérées par le processus de self-learning. Une entrée statique est définie à l'aide du management du bridge (SNMP par exemple).

Le Transparent Spanning Tree permet de créer des chemins redondants pour interconnecter les réseaux Ethernet. **Ce protocole veille à l'activation d'un seul chemin pour accéder à une station même si plusieurs sont physiquement possibles**. Si ce chemin venait à être détruit (les bridges échanges de trames pour vérifier l'accessibilité des autres bridges), le protocole veillera à activer un autre chemin. Cette reconfiguration des bridges du réseau peut prendre un certain temps et peut amener à la présence de plusieurs chemins possibles pour une destination(=> duplication de trames ou bouclages). En fonction des protocoles couches 3 à 7 utilisés, ce temps de reconfiguration peut être trop élevé pour éviter de perdre les connexions.

5. Les swicth ou commutateurs Ethernet

Face à la saturation d'un réseau Ethernet, une première solution consiste à subdiviser le réseau en plusieurs réseaux reliés par des bridges "classiques". Une nouvelle solution consiste à utiliser des switch ou commutateurs Ethernet. Un switch comporte plusieurs ports (8 par exemple) et crée de manière transparente plusieurs réseaux ethernet (domaine de collision). Dans le cas de la figure ci-dessous, Il est donc possible de faire dialoguer les 8 stations par paire en même temps. La bande

passante du réseau est passée de 10 Mbit/s à 40 Mbit/s. Le switch commute les trames d'un port vers l'autre à une vitesse très élevée : l'apparition de collisions est donc moins fréquente que dans le cas d'un réseau Ethernet ne comportant pas de switch.



B. IEEE 802.5 Token Passing Ring

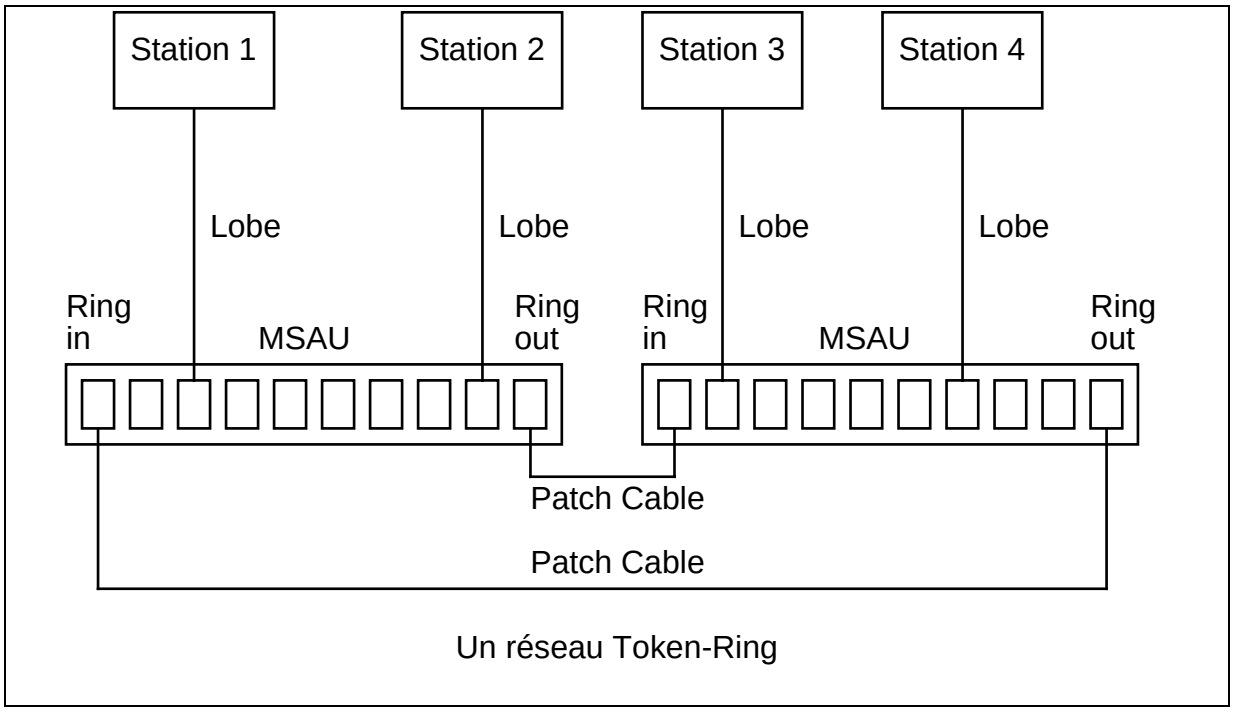
Un anneau physique relie les partenaires. Un jeton qui circule sur l'anneau. Une station doit attendre le jeton pour pouvoir émettre une trame.

La norme prévoit 1 Mb/s, 4 Mb/s et IBM supporte un anneau à 16 Mb/s en plus. IBM continue à développer le Token-Ring et IEEE 802.5 adopte de plus en plus de caractéristiques du Token-Ring IBM.

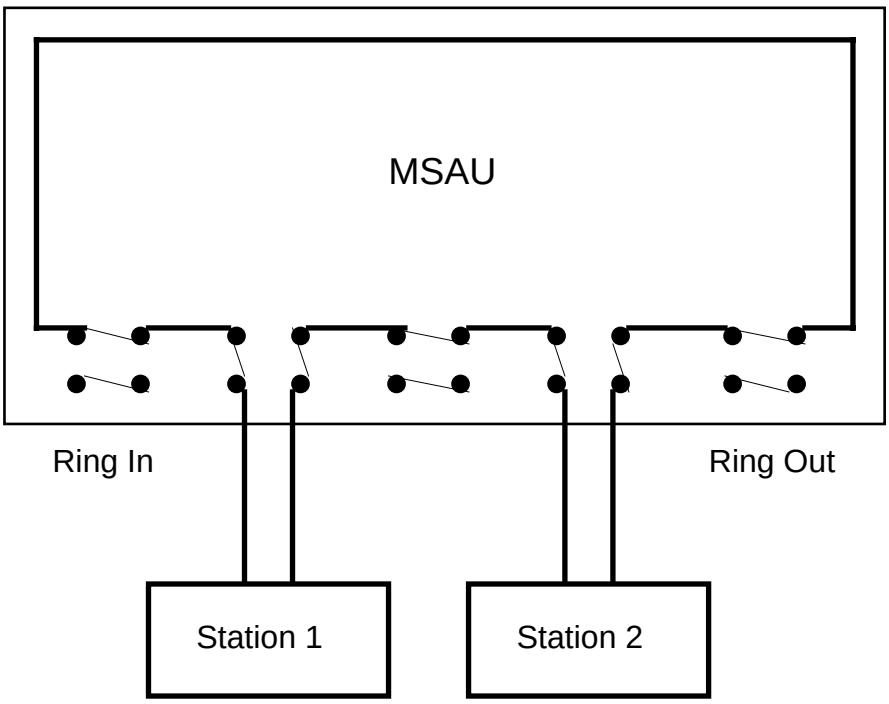
	IEEE 802.5	IBM Token-Ring
Méthode d'accès	Passage de jeton	Passage de jeton
Topologie	Pas spécifiée	"Etoile"
Mode de transmission	Bande de base	Bande de base

Codage du signal	Manchester Différentiel	Manchester Différentiel
Vitesse de transmission	1 Mb/s ; 4 Mb/s	4 Mb/s, 16 Mb/s
Stations/segments	250	260 (STP) 72 (UTP)
support physique	Pas spécifié	Paire torsadée

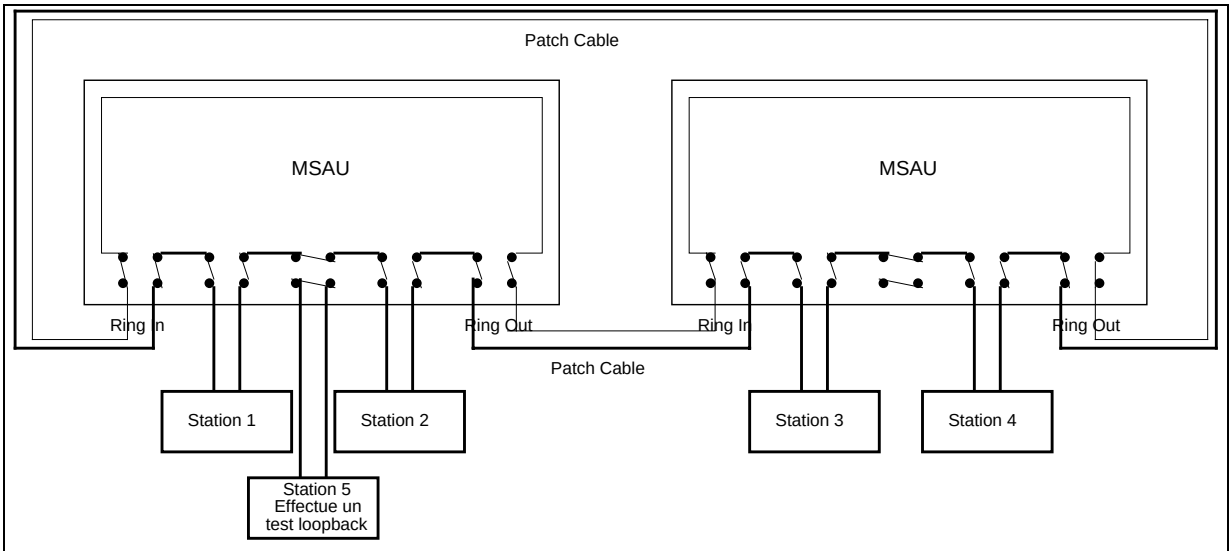
Les stations se relient par un "lobe" (une double paire torsadée blindée ou non) à un MSAU (Multistation Access Unit) (première technologie passive) ou à un LAM (Lobe Access Module) associé à un CUA (Control Access Unit) qui sont des éléments actifs.



Un réseau Token-Ring



Contenu d'un MSAU



Anneau comportant deux MSAU

Chaque station dans le Ring est un répéteur unidirectionnel.
Aussi chaque station reçoit une série de bits de la station précédente, elle les lit puis les retransmet vers la station qui suit dans l'anneau.

Plusieurs MSAU peuvent être connectés ensemble pour former un grand anneau.

Les MSAU comprennent des relais de bypass pour permettre d'enlever des stations de l'anneau. De plus, ils permettent d'augmenter la fiabilité de l'anneau.

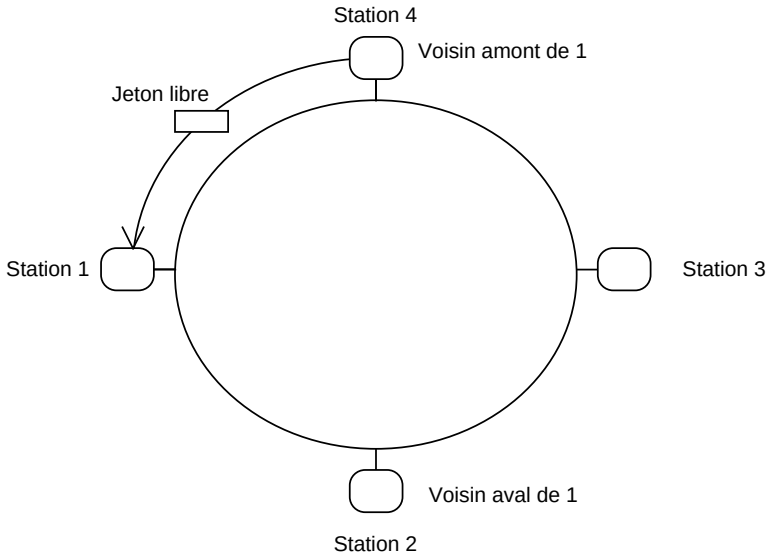
Longueur max entre deux stations 375 m à 16 Mb/s ou 750m à 4 Mb/s avec câble de type 1 (STP). Il existe des abaques fournissant la distance maximale entre stations en fonction du nombre de stations dans l'anneau.

La longueur maximale des trames dépend de l'implémentation mais voici quelques limites standards.

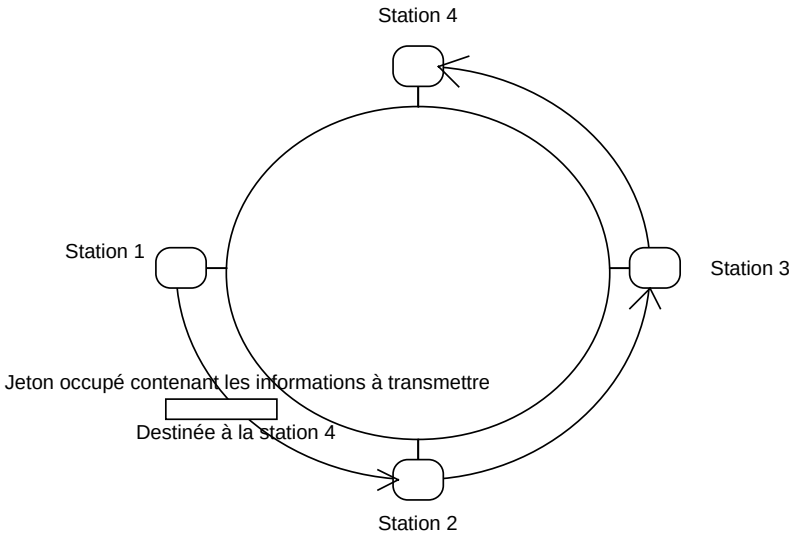
20532 octets, 4472 octets (4 Mb/s), 8172 octets (16 Mb/s), 17800 octets (16 Mb/s pour Chipset Texas Instrument), 64 Kbytes.

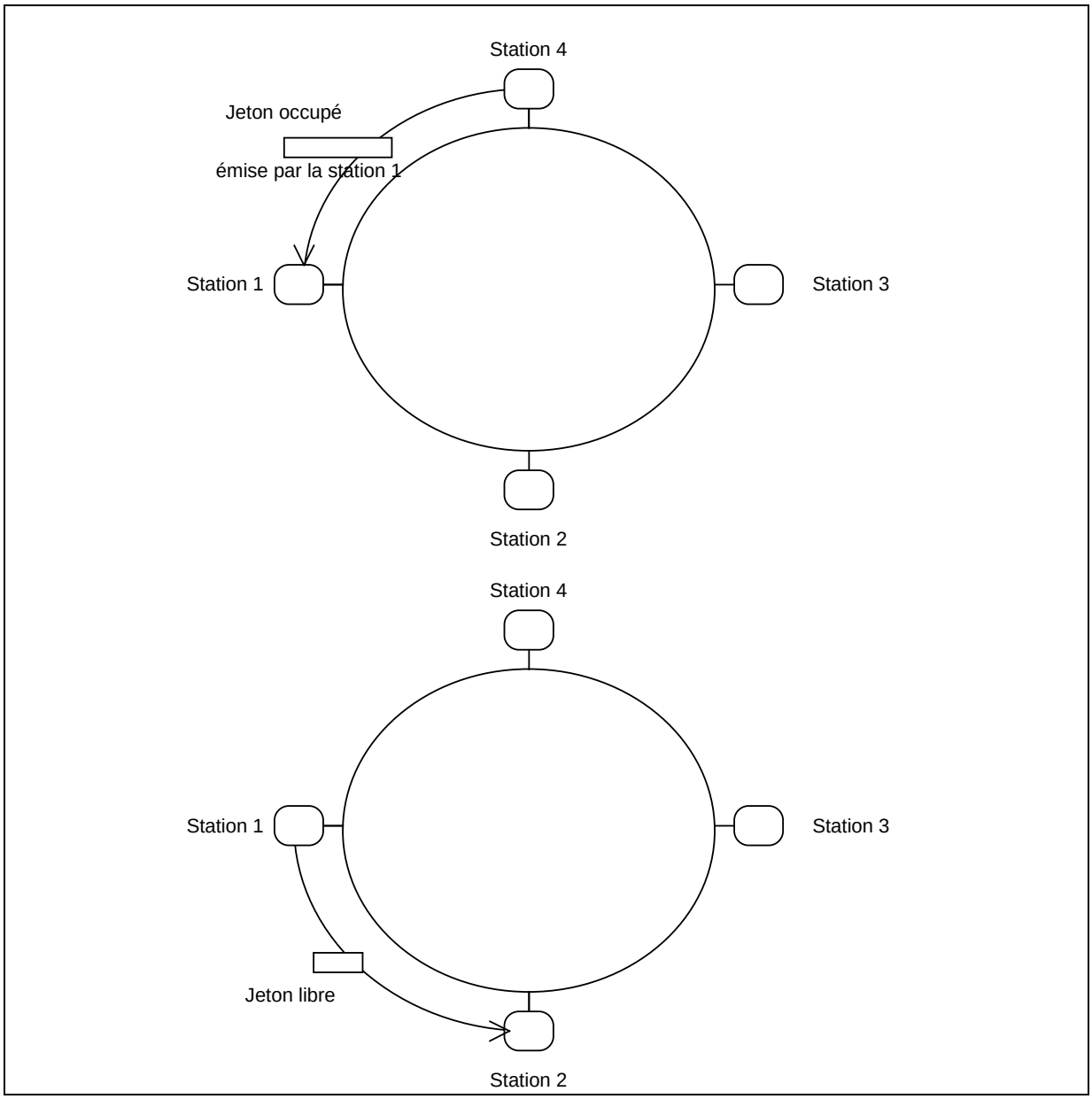
1. Principe d'accès : Passage de jeton

Un jeton donne le droit à transmettre et un seul jeton circule sur l'anneau.



Passage de jeton





Sur les réseaux à plus hauts débits, on peut prévoir la technique Early Token Release. Le Jeton est envoyé à la suite de la trame de données avant même de récupérer la trame.

Le Protocole Token-Ring prévoit une grande partie d'administration du réseau. Dans ce cadre, Token-Ring dispose de plusieurs stations fonctionnelles. Certaines de ces fonctions sont intégrées dans le chipset du contrôleur Token-Ring mais dans certains cas, il faut charger des logiciels supplémentaires sur la station. Voici les stations fonctionnelles standards.

Station fonctionnelle	Adresses Fonctionnelle	Obligatoire/optionnelle
Active monitor	C0 00 00 00 00 01	Obligatoire
Stand-by monitor	Pas applicable	Obligatoire
Ring Parameter Server	C0 00 00 00 00 02	Optionnelle
Ring Error Monitor	C0 00 00 00 00 08	Optionnelle
Configuration Report Server	C0 00 00 00 00 10	Optionnelle

2. Stations fonctionnelles

a) Active monitor

Cette station est la plus importante du réseau. Cette fonctionnalité est intégrée dans le Chipset de l'adaptateur Token-Ring.

Il assure la maintenance de l'anneau :

- Il contrôle le fonctionnement du protocole : il enlève les trames qui circulent indéfiniment.
- Il fournit des services de synchronisation (horloge maître). L'anneau est une boucle fermée sur laquelle il y a toujours un signal d'horloge. S'il n'y a pas de trames alors des 0 circulent. Comme les stations sont synchronisées, le préambule est réduit très fortement.
- Il doit induire un délai de 24 bits dans l'anneau pour pouvoir insérer le jeton en entier. En effet, le jeton n'est pas retiré de l'anneau par les stations de sorte qu'il faut éviter qu'une station répétrice ne reçoive le début du jeton alors qu'elle n'a pas encore terminé d'émettre ce même jeton.
- Il doit lancer une séquence de poll des stations présentes dans l'anneau toutes les 7 secondes. Cette phase Ring Poll ou Neighbor Notification permet à chaque station

de connaître la station qui se trouve directement en amont dans l'anneau et aux nouvelles stations d'entrer dans l'anneau.
• Il redémarre l'anneau après une interruption. Si un jeton est perdu ou corrompu, il commence par une purge de l'anneau.

Toutes les stations peuvent être Active Monitor. Quand aucun Active Monitor n'est présent des procédures automatiques veillent à forcer une station à devenir Active Monitor.

b) Stand-by monitor

Toutes les stations qui ne sont pas Active Monitor sont Stand-by Monitor. Chaque Stand-by Monitor doit surveiller le fonctionnement de la Station Active Monitor. Si la station Active Monitor disparaît, les stations Stand-by Monitor doivent élire une nouvelle station Active Monitor.

c) Ring Parameter Server

La station Ring Parameter Server fournit une méthode pour distribuer les paramètres de l'anneau aux stations qui entrent dans l'anneau. Cette station est optionnelle. Il faut un logiciel supplémentaire sur la station pour pouvoir assumer cette fonction. Il fournit entre autres le numéro de l'anneau local, les priorités d'accès permises, ...

d) Ring Error Monitor

La station Ring Error Monitor a pour seul rôle de collecter les informations concernant des erreurs survenues dans l'anneau. Ces informations lui sont envoyées par les autres stations quand elle décèle une erreur. Cette station est optionnelle. Il faut un logiciel supplémentaire sur la station pour pouvoir assumer cette fonction.

e) Configuration Report Server

La station Configuration Report Server gère les stations de l'anneau et signale les événements qui surviennent au niveau de l'anneau. Cette station peut récupérer mais aussi changer les paramètres des stations de l'anneau. Les paramètres donnés par la Station Ring Parameter Server lors de l'initialisation des stations

peuvent être modifiés par la Station Configuration Report Server. Cette station peut aussi obtenir de toutes les stations de l'anneau leur adresse, leur état. De plus, elle peut demander une déconnexion de station.

Il reçoit aussi les événements qui ont lieu sur l'anneau qui lui sont signalés par les stations :

- Nouvelle Station Active Monitor
- Changement dans les adresses des stations voisines en amont d'une station (entrée de station dans l'anneau ou sortie de station de l'anneau)
- Erreur dans le Pool de l'anneau (le processus de Poll ne s'est pas achevé).
- Erreur de la station Active Monitor.

Il apparaît de tout ceci que le Token-Ring est un réseau qui peut très bien être géré au point de vue des erreurs éventuelles. De plus, comme le protocole essaie envers et contre tout de toujours rétablir l'anneau. L'anneau peut donc être en défaut de manière intermittente sans le savoir.

La création et le maintien du fonctionnement de l'anneau se base sur cinq processus principaux.

3. Phases de fonctionnement d'un Token Ring

a) Monitor Contention (Election de la station Active Monitor)

L'anneau ne peut fonctionner que si une station Active Monitor existe. Lors de la mise sous tension de l'anneau ou lors de la disparition ou d'un dysfonctionnement de la station Active Monitor, il faut qu'une station Active Monitor soit élue.

Ce processus est lancé par la première station qui détecte un problème du point de vue de la station Active Monitor. Cette station est souvent la première station à se connecter dans l'anneau (mise sous tension) ou la station directement en aval de la station Active Monitor dans le cas d'un anneau déjà en fonctionnement. Cette station envoie une trame Claim_Token toutes les 20 msec. Les autres stations vont soit se mettre dans un état de répétition de la trame Claim_Token ou alors elles vont

participer à "l'élection de la station Active Monitor". La nouvelle station Active Monitor sera celle qui a la plus haute adresse sur l'anneau.

La première station continue à répéter les trames Claim_Token jusqu'à ce que :
- elle reçoive une trame Claim_Token avec une adresse source plus élevée que la sienne ou
- elle reçoive trois de ses propres trames Claim_Token. A ce moment, elle est la nouvelle station Active Monitor.

Pour pouvoir être active dans l'élection de la station Active Monitor, la station doit avoir détecté le problème ou être désignée comme participante à l'élection par la Station Ring Parameter Server ou Configuration Report Server lors du chargement du gestionnaire de l'adaptateur.

b) Ring Poll

Toutes les 7 secondes, la station Active Monitor lance le processus Ring Poll. Ce dernier a pour but :
- d'indiquer à toutes les stations Stand-by Monitor que la station Active Monitor est présente,
- d'informer toutes les stations que l'anneau fonctionne correctement,
- de permettre à toutes les stations de connaître la station qui se trouve directement en amont.

Lorsque 7 secondes se sont écoulées, la station Active Monitor attend un jeton libre puis envoie une trame AMP (Active Monitor Present). La station en aval de la station AM (Active Monitor). Cette station sait qu'elle est la première station derrière l'AM car les bits ARI et FCI de la trame sont toujours à 0. Cette station connaît donc l'adresse de la station directement en amont et la retient et lance une temporisation qui permet de vérifier que la station AM est bien active. Cette station met les bits ARI (Address Recognized Indicator) et FCI (Frame Copied Indicator) de la trame à 1. Les autres stations vont simplement répéter la trame. La station AM va retirer la trame AMP et envoyer un jeton libre à la station qui est directement en aval. Celle-ci va envoyer une trame SMP (Standby Monitor Present) à la station qui la suit directement. Cette dernière va remarquer que les bits ARI et FCI de la trame sont toujours à 0. A ce moment, elle connaît donc l'adresse de la station directement en

amont et la retient. Cette station met les bits ARI et FCI de la trame à 1. Les autres stations vont simplement répéter la trame. Et on recommence ces opérations jusqu'à ce la station AM récupère une trame SMP avec les bits ARI et FCI à 0.

c) Initialisation de Station

L'initialisation d'une station se fait en six étapes :

- Charger les valeurs par défaut pour l'adaptateur
- Vérifier le support Lobe (connection en loopback)
- Vérifier la présence de la station AM. Elle se connecte dans l'anneau et doit recevoir endéans 18 secondes une trame AMP ou SMP ou une trame Ring Purge
- Vérifier son adresse. Elle envoie une série de trames Duplicate Address Test. Si elle reçoit deux trames consécutives avec les bits ARI et FCI à 1, elle suppose qu'une autre station a son adresse et se retire de l'anneau. Si elle reçoit deux trames consécutives avec les bits ARI et FCI à 0, elle suppose que son adresse est unique et passe à la phase suivante.
- Participer dans le processus Ring Poll pour connaître l'adresse de la station directement en amont.
- Demander son initialisation. Elle envoie un série de trame Request Initialization vers l'adresse fonctionnelle du Ring Parameter Server ou Configuration Report Server. Si les trames restent quatre fois sans réponse, la station conserve ses paramètres par défaut.

d) Purge de l'anneau

La station AM utilise le processus de Ring Purge pour relancer l'anneau au cas où elle vient de devenir station AM ou si elle a détecté une perte de jeton ou une trame corrompue.

A ce moment, la station AM envoie une trame Ring Purge toutes les 4 ms sans attendre un jeton libre et continue jusqu'à recevoir une trame Ring Purge non corrompue. A ce moment, elle arrête d'émettre ces trames et envoie un nouveau jeton libre. Ces trames sont suivies de trames qui rapportent les erreurs aux stations Ring Error Monitor ou Configuration Report Server.

e) Beacon

Ce processus est la dernière tentative pour rétablir l'anneau dans le cas d'une erreur hardware persistante. Si ce processus réussit, l'anneau va isoler le problème et fonctionner à nouveau correctement; sinon il faut une intervention de techniciens.

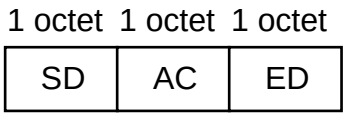
Une station (A) qui détecte une erreur sur son câble de réception suppose d'abord qu'elle n'est pas en cause et rejette la faute sur la station (D) en amont. A commence à transmettre une trame Beacon contenant D comme la station en erreur toutes les 20 ms sans attendre le jeton. Les autres stations passent en mode répétition de trame Beacon dès qu'elles reçoivent ce type de trame. La station D reçoit les trames Beacon indiquant qu'elle est fautive. A la suite de la réception de 8 trames Beacon, D se déconnecte de l'anneau et fait un test en loopback (comme pour l'initialisation). A ce moment, trois issues sont possibles pour ce test :

- l'autotest du hardware de D (Lobe et adaptateur) indique qu'il existe un défaut et la station D reste déconnectée de l'anneau. Comme D s'est retiré de l'anneau, A reçoit ses trames Beacon et passe au processus Monitor Contention
- l'autotest du hardware de D (Lobe et adaptateur) indique que tout est correct. D se reconnecte à l'anneau et passe en mode répéteur de trames Beacon. Si la station A n'a pas reçu de trame Beacon endéans 16 secondes, elle se déconnecte de l'anneau et fait un autotest de son hardware. Si A détecte une erreur, elle reste hors de l'anneau et le problème est résolu. La station B qui suit directement la station A a commencé à envoyer des trames Beacon quand A s'est retiré de l'anneau. Comme B reçoit ses propres trames Beacon, elle passe au processus Monitor Contention.
- Si A n'a pas détecter d'erreur dans son hardware, il se reconnecte à l'anneau et n'arrête plus de transmettre des trames Beacon. L'anneau ne peut être reactivé sans intervention de techniciens dans ce cas.

4. Les différentes trames

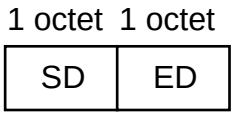
Toutes les trames commencent par l'octet Start Delimiter (SD) qui prévient le récepteur qu'une nouvelle trame arrive. Dans ce champ se trouvent des signaux qui distinguent cet octet de tous les autres car il ne respecte pas l'encodage Manchester Différentiel. Les trames se terminent par l'octet End Delimiter (ED) qui utilise aussi un encodage spécial qui ne respecte pas l'encodage Manchester Différentiel.

a) Le jeton



Jeton

b) La séquence Abort

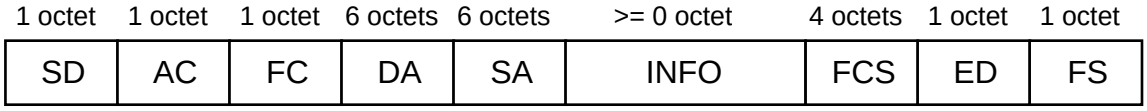


Abort

La séquence Abort peut être envoyée pour indiquer une terminaison prématurée de la transmission d'une trame.

c) La trame

Sur l'anneau deux types de trames circulent : des trames contenant des données utilisateur (LLC) et des trames destinées à la gestion du protocole (Celles-ci ont été évoquées précédemment mais leur structure exacte ne sera pas décrite ici.)



(1) L'octet Start Delimiter (SD)

Un octet contenant des violations du codage Manchester Différentiel (J,K) et permettant ainsi d'identifier le début de la trame.

J	K	0	J	K	0	0	0

(2) L'octet AC (Access Control)

Il contient trois bits de priorité (P) et trois bits de réservation (R).

P	P	P	T	M	R	R	R

Seules les stations qui ont une priorité supérieure ou égale à la priorité indiquée dans le jeton peuvent le prendre. Lorsque qu'une trame de donnée/commande passe chez les différentes stations, celles-ci peuvent mettre les bits de réservation pour essayer de réserver le prochain jeton. Si une station de priorité supérieure suit la première qui essaie de réserver et que celle de priorité supérieure réserve aussi alors c'est cette dernière qui aura le prochain jeton si aucune station de priorité supérieure ne la suit et réserve le jeton. Les données ont toujours une priorité = 0. Les trames de commande de gestion l'anneau ont des priorités supérieures.

Lorsque le jeton est généré, il aura la priorité de la réservation. Les Stations qui élèvent la priorité du jeton sont responsables de la remettre à son niveau précédent.

Le bit T de AC indique qu'il s'agit d'une trame de données quand T = 1. Si T=0 alors on est en présence d'un jeton.

Le bit M est destiné à la station Active monitor. Quand le bit M est à 0 et qu'il s'agit d'une trame ou d'un jeton de priorité différentes de 0, il le met à 1. Quand la station Active Monitor reçoit une trame avec le bit M à 1, il la retire du ring car l'émetteur de la trame ne l'a pas enlevée.

(3) L'octet FC Frame Control

Cet octet indique si la trame contient des données LLC ou des informations de contrôle et de gestion de l'anneau.

F1	F0	0	0	PCF3	PCF2	PCF1	PCF0

SI F1, FO = 00 => Trame de gestion de l'anneau
SI F1, FO = 01 => Trame de données
SI F1=1 => Réservé

Dans le cas des trames de gestion de l'anneau, les bits PCF sont utilisés. Ce sont ces bits qui indiquent s'il s'agit d'une trame Claim_Token, Purge, Beacon, Active Monitor Present, Standby Monitor Present.

(4) Le champ DA

6 octets d'adresse pour indiquer un adresse simple, multicast (bit 8 du premier octet à 0 et à 1 respectivement) . Les adresses peuvent être administrées localement ou universellement (bit 7 du premier octet à 1 et 0 respectivement).

Le bit 8 du troisième octet indique s'il s'agit d'une adresse fonctionnelle (0 si fonctionnelle).

Token-Ring prévoit deux adresses de Broadcast FF FF FF FF FF FF et C0 00 FF FF FF FF pour un broadcast vers les stations fonctionnelles.

(5) Le champ SA

6 octets d'adresse pour indiquer un adresse simple
Le bit 8 du premier octet (RII) permet d'indiquer que de l'information de routage est prévue dans la partie INFO (quand il est à 1) . Les adresses peuvent être administrées localement ou universellement (bit 7 du premier octet à 1 et 0 respectivement).

(6) Le Champ INFO

Il contient les données qui doivent être transmise vers les couches supérieures. La longueur des trames est limités par la durée maximale de rétention du jeton des stations. Dans le cas des trames de gestion de l'anneau, ils contiennent les informations de gestion.

(7) Le champ FCS

FCS : Frame Check Sequence (4 octets). CRC calculé sur FC, DA, SA et INFO.

(8) Le Champ ED

ED End Delimiter. Il indique la fin de la trame. Il contient un bit (ED) indiquant si une erreur a été détectée dans la trame par le destinataire. De plus , le destinataire peut aussi indiquer par un bit (IF) que c'est la dernière d'une séquence logique. Ce signal est rendu unique par l'inclusion de signaux qui violent les règles d'encodage (J, K).

J	K	1	J	K	1	IF	ED

(9) L'octet FS : Frame Status

Cet octet comporte les bits ARI (Address Recognized Indicator) adresse reconnue et les bits FCI : Frame Copied Indicator. Ces bits sont positionnés par le destinataire. L'émetteur les met à 0. S'ils sont toujours à 0 lors du retour de la trame chez l'émetteur, c'est que le destinataire n'est pas là.

ARI	FCI	0	0	ARI	FCI	0	0

Si le destinataire a su reprendre la trame alors ils sont tous à 1.

Si le destinataire n'a pas su copier (tampon rempli) alors les bits ARI sont à 1 et les bits FCI sont à 0.

5. Pont Token-Ring

Pour interconnecter plusieurs réseaux Token-Ring, des bridges sont utilisés et ceux-ci utilisent le **source routing** pour déterminer le chemin à suivre pour faire parvenir une trame au destinataire.

La station émettrice envoie une trame de diffusion pour déterminer le chemin à suivre pour atteindre une machine se situant derrière un ou plusieurs bridges. Les bridges vont répéter cette trame sur leurs différents ports en indiquant dans le trame qu'elle a transité chez eux. Quand la première trame arrive chez le destinataire, celui-ci la renvoie en inversant le chemin parcouru précédemment. L'émetteur va récupérer la

trame et cette dernière contient donc un chemin pour arriver au destinataire. Les prochaines trames envoyées au destinaire contiennent le chemin à suivre.

Un maximum de 7 ponts peuvent séparer deux stations quelconques du "réseau Token-Ring global" (la trame ne prévoit que 7 zones pour retenir le chemin).

Chaque anneau est identifié par un nombre unique sur 12 bits et chaque bridge d'un anneau est identifié par un nombre sur 4 bits.

6. Remarques

L'Implémentation IBM ne prévoit que l'envoi d'une trame par obtention de jeton.

La norme IEEE 802.5 prévoit que l'on peut transmettre plusieurs trames d'information jusqu'à ce qu'il n'y aie plus de trames en attente ou le Token Hold Timer est écoulé (durée de rétention du jeton écoulée).

C. IEEE 802.4 Token-Bus

1. Généralités

Jeton adressé sur un bus. Le jeton est adressé pour permettre de créer l'anneau logique des stations. Toutes les stations disposent de l'"intelligence" pour mettre l'anneau logique en fonctionnement. Chaque station doit connaître la liste des stations actives et les stations qui se trouvent directement en amont et en aval. Une trame peut comporter un maximum 8191 octets.

	IEEE 802.4
Méthode d'accès	Passage de jeton
Topologie	Bus

Mode de transmission et vitesse de transmission	• Codage Manchester avec violation à 1MB/s (bande de base) (Système non directionnel). Le câble principal du réseau peut avoir une longueur maximale entre 1280 m et 7600 m en fonction de la qualité du câble coaxial. Pas de nombre limite de drop câble. • Modulation de porteuse à un canal à 5 ou 10 Mbit/s (**système non directionnel**) • Modulation large bande : 1Mbit/s pour canal 1,5 MHz 5 Mbit/s pour canal 6 MHz 10 Mbit/s pour canal 12 MHz (**Système directionnel**) (Amplificateur bidirectionnel) Câble CATV (télédistribution)
Codage du signal	Manchester Différentiel
support physique	Câble coaxial 75 Ohms

2. Trame

1 octet	1 octet	6 octets	6 octets	>= 0 octet	4 octets	1 octet
SD	Type	DA	SA	INFO	FCS	ED

D. FDDI (ISO 9314)

1. Généralités

Fiber Distributed Data Interface

FDDI = anneau à passage de jeton à haut débit sur Fibre Optique. C'est donc un réseau à accès déterministe.

FDDI est utilisé comme :
- dorsale ou Backbone pour relier différents réseaux
- réseaux pour local d'ordinateurs (connexion mainframe, mini et périphériques dans une même pièce)
- réseau locaux haut débit pour application avec vidéo ou station de travail (CAO ou CFAO)

Méthode d'accès	Passage de jeton
Topologie	Anneau, étoile
	Bande de base
Codage du signal	NRZI 4b/5b
Vitesse de transmission	100 Mb/s
Station /segments	1000
Support physique	Fibre optique multimode 62,5 µm
	Fibre optique monomode
Adressage	16 ou 48 bits
Longueur de fibre totale	200 km

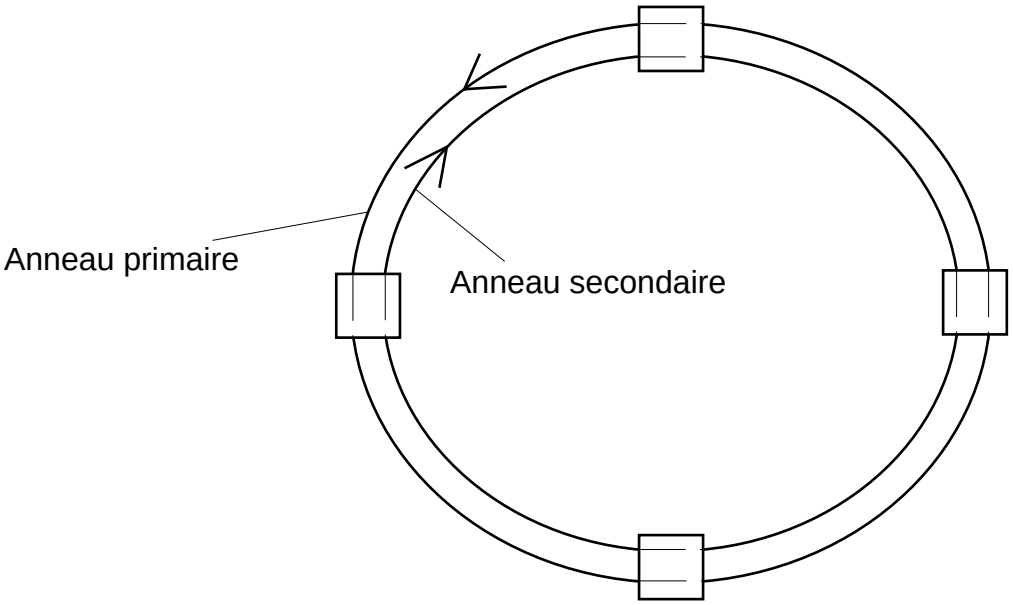
FDDI permet des communications synchrones et la plage qui reste est utilisée par les communications asynchrones.

2. Topologie

FDDI repose sur deux anneaux à contre-sens.

Anneau configuré en topologie étoile pour avoir une meilleure résistance. En réalité, il s'agit de deux anneaux sur lesquels les données circulent dans un sens sur le premier anneau et dans l'autre sens sur le second anneau. Dans les conditions

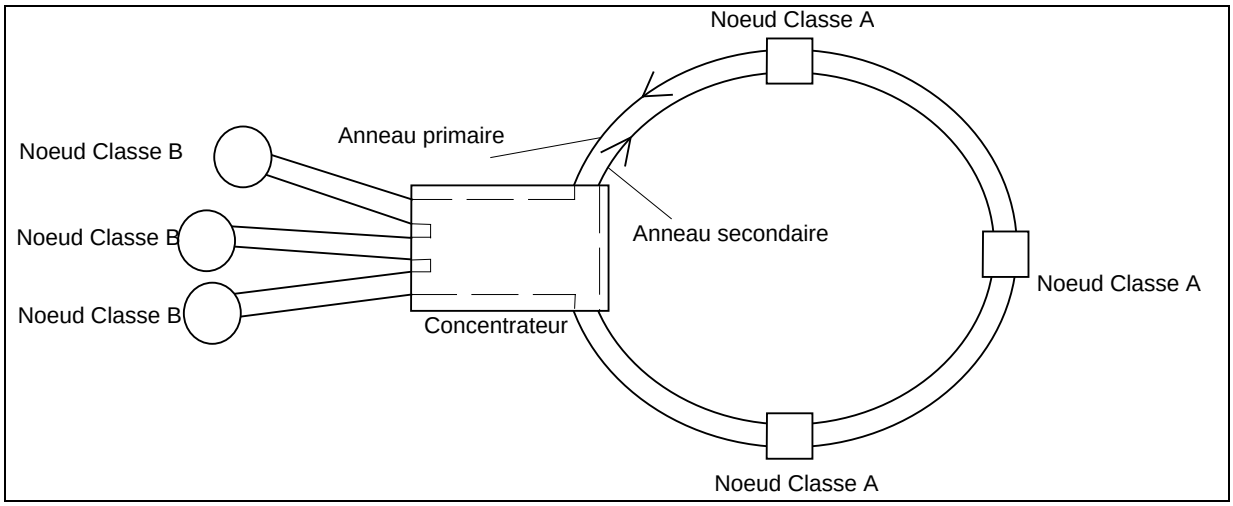
normales, les informations circulent uniquement sur l'anneau primaire. L'anneau secondaire n'entre en action que s'il y a une coupure sur l'anneau primaire.



Il existe deux types de noeuds :

- Les noeuds qui sont connectés sur les deux anneaux ou Noeud de Classe A ou DAS (Dual Attachment Station) : station ou concentrateur
- Les noeuds qui sont connectés sur l'anneau primaire uniquement ou Noeud de Classe B ou SAS (Single Attachment Station) : station.

En cas de problème hardware sur l'anneau, les stations de Classe A peuvent participer à la reconfiguration de l'anneau pour le rendre fonctionnel à nouveau.

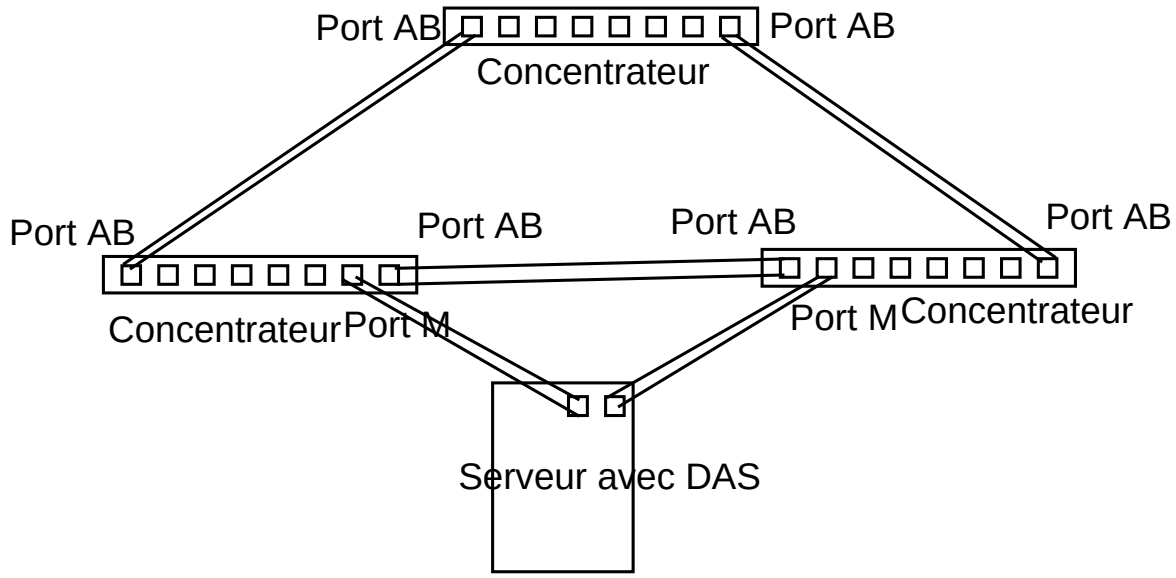


Les concentrateurs comportent des ports M et un port A et B. Les ports A et B sont ceux qui font partie de l'anneau principal.

Les Stations utilisent :

des cartes S : Single Attachment
des cartes AB : carte double avec redondance

Une station DAS peut être raccordée sur les ports M de deux concentrateurs différents. Ainsi cette station disposera d'un chemin redondant pour se connecter sur le réseau. Cette configuration est intéressante pour les serveurs



FDDI est limité à 1000 stations et 200 km mais en réalité 500 stations et 100 km si rupture d'un segment.

Il faut une station ou un répéteur tous les 2 km au minimum.

La gestion de l'anneau repose sur le principe Early Token Release. Le Jeton est émis par la station qui en a pris possession directement à la suite de ses trames de données sans attendre leur retour. Ceci est intéressant pour permettre de mieux utiliser la bande passante offerte par l'anneau en fibre optique. Aussi il est possible que plusieurs trames circulent en même temps sur l'anneau.

3. Trames

La longueur maximale d'une trame est de 4500 octets (préambule exclu). Un octet d'information est découpé en quartet. Chaque Quartet est encodé sur un symbole de 5 bits. Comme FDDI repose sur le principe d'accès de l'anneau à jeton, il utilise deux types de trames : jeton et trame de données ou commandes

a) Jeton

A	B	C	D

Le jeton comprend quatre parties :
A : Préambule de 16 symboles ou plus
B : Start Delimiter comportant 2 symboles
C : Frame Control comportant 2 symboles
D : End Delimiter comportant 2 symboles

b) Trames de données/commandes

Un trame de données ou de commandes comporte, quant à elle, 9 parties :
A : Préambule de 16 symboles ou plus
B : Start Delimiter comportant 2 symboles
C : Frame Control comportant 2 symboles
E : Adresse destination comportant 4 ou 12 symboles
F : Adresse source comportant 4 ou 12 symboles
G : Champ des données comportant 0 ou plusieurs symboles
H : Frame Check Sequence comportant 8 symboles
I : End Delimiter comportant 1 symbole
J : Frame Status comportant au moins 3 symboles qui indiquent si la trame a été copiée par le destinataire, si le destinataire y a décelé une erreur.

4. Allocation de largeur de bande

FDDI supporte l'affectation en temps réel de la largeur de bande. Pour ce faire FDDI prévoit deux types de trafic sur le réseau :
- le trafic synchrone
- le trafic asynchrone

La largeur de bande synchrone correspond à une portion des 100 Mb/s destinée au trafic synchrone (voix et vidéo). La largeur de bande restante est destinée au trafic asynchrone. Le protocole SMT (Station ManagemenT prévu dans FDDI) permet cette affectation dynamique. L'allocation de largeur de bande asynchrone repose sur un système à 8 niveaux de priorité. Si toute la largeur de bande est utilisée par des stations ayant des allocations de largeur de bande synchrone et de haute priorité, les stations qui n'utilisent pas de largeur de bande synchrone et des allocations de largeur de bande asynchrone de faible priorité risquent de ne pas pouvoir

transmettre. Il est même possible à des stations en communication de s'attribuer l'ensemble de la largeur de bande asynchrone pendant un temps certain.

5. CDDI = Copper Distributed Data Interface

CDDI est simplement l'adaptation de FDDI au support de transmission TP cat 5.

6. FDDI-II

Cette version de FDDI est isochrone et permet ainsi le transport de voix.

E. FAST ETHERNET ou IEEE 802.3u

1. Généralités

Fast Ethernet est une version plus rapide de Ethernet 10 Mbit/s. Fast Ethernet fonctionne à une vitesse de 100 Mbit/s. Cette norme a été adoptée en mai 1995.

Cette technologie utilise le même principe d'accès que Ethernet c'est-à-dire le principe CSMA/CD. Les formats des trames sont identiques. Les mêmes limites de tailles sont d'application. Dans cette technologie-ci, le temps qui doit séparer l'émission de deux trames consécutives par une station est de 0,96 µs (9,6 µs /10).

Le principe d'accès au support CSMA/CD se base sur la possibilité de détecter des collisions. Afin d'être certain de pouvoir observer ces dernières, le " diamètre " du réseau (distance maximale séparant deux stations quelconques du réseau) est fonction du slot time qui vaut 5,2 µs (51,6 µs /10).

Le temps nécessaire pour que le signal électrique parcoure le double de la distance maximale séparant deux stations quelconques du réseau doit rester inférieur à ce slot time. Dans le cas contraire, la détection d'une collision ne peut être garantie (ce qui est relativement gênant).

Dans le cas de Fast Ethernet, on remarquera que le diamètre du réseau est fortement réduit. Fast Ethernet utilise la paire torsadée ou la fibre optique comme support de transmission.

Le réseau est un réseau en étoile comportant des Hubs ou concentrateurs et les stations.

Voyons plus en détails, dans la partie suivante, les différents supports utilisés.

2. Supports de transmission

Fast Ethernet prévoit deux types de supports différents.

a) La paire torsadée

Remarque

Au niveau des paires torsadées, il existe différents types de paires. On distingue :
***UTP** : Unshielded Twisted Pair ou Paire torsadée non blindée*
***STP** : Shielded Twisted Pair ou Paire torsadée blindée (tresse). Ce blindage protège des pertubations basse fréquence.*
***FTP** : Foiled Twisted Pair ou Paire torsadée avec feuillard. Ce feuillard protège des pertubations haute fréquence et éviter le rayonnement du câble vers son environnement.*
***SFTP** : Shielded Foiled Twisted Pair*

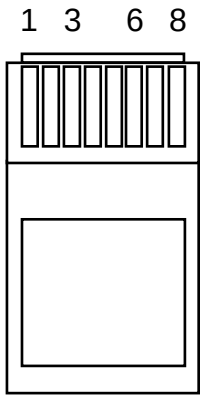
La catégorie de la paire torsadée rend compte de ses caractéristiques électriques tellles que impédance, atténuation en fonction de la fréquence, ... Une paire torsadée Catégorie 5 permet de véhiculer un signal à 100 Mhz par exemple.

*Du point de vue protection de l'environnement contre les rayonnements Electromagnétiques, les cables sont rangés dans des classes : La **classe A** et la **classe B**. Toute nouvelle pose de câble TP dans un environnement bureautique doit utiliser du câble de classe B. Les câbles SFTP sont des câbles de classe B. En fonction du fabricant, un câble UTP ou FTP peut ou non être de classe B.*

Câblage d'un Câble TP sur un connecteur RJ45

N° contact du connecteur	Couleur du fil

1	Blanc-Orange
2	Orange
3	Blanc-Vert
4	Bleu
5	Blanc-Bleu
6	Vert
7	Blanc-Brun
8	Brun



(1) BASE TX

Cette norme correspond à un câble UTP catégorie 5. La distance maximale entre le concentrateur et la station ne peut dépasser 100 mètres. Elle utilise deux des quatre paires du câble. Le connecteur utilisé est un RJ-45. Elle utilise un codage 4B/5B.

(2) BASE T4

Cette norme correspond à un câble UTP catégorie 3,4,5. La distance maximale entre le concentrateur et la station ne peut dépasser 100 mètres. Elle utilise les quatre paires du câble. Lors d'une transmission, 3 paires sont utilisées pour émettre les données et la 4e paire sert de canal de réception pour la détection de collision. Le connecteur utilisé est un RJ-45.

b) La fibre optique

(1) BASE FX

Les fibre multimode 62,5/125 et monomode peuvent être utilisées. Elle permet une distance de connexion de 150 mètres.

3. MII ou Media-Independant Interface

Fast Ethernet prévoit aussi les spécifications pour "Media-Independant Interface" ou MII. Ces spécifications permettent d'adapter n'importe quelle carte Fast Ethernet au type de support physique utilisé. Une carte comportant l'interface MII pourra donc se connecter sur une support Fibre ou 100 Base FX ou T4 par l'utilisation du transceiver ad hoc. Il permet une longueur de câble de 0,5 mètre entre l'interface MMI et le transceiver Connecteur à 40 broches. Il est similaire à un connecteur SCSI (mais plus petit).

4. Les éléments actifs : concentrateur ou switch

a) Les concentrateurs

Il existe deux classes de concentrateurs :
Les concentrateurs ou répéteurs de classe I : retard de 0,7 µs au moins
Les concentrateurs ou répéteurs de classe II : retard de 0,4 µs au maximum.

Dans le cas d'un réseau composé de TP uniquement, deux stations quelconques d'un réseau pourront être séparées par un répéteur de classe I (les stations sont connectées sur le même répéteur=> diamètre réseau 200 mètres) ou par deux répéteurs classe II (diamètre réseau 205 m= 100m+5m+100m)

Dans le cas d'un réseau comportant de la fibre :

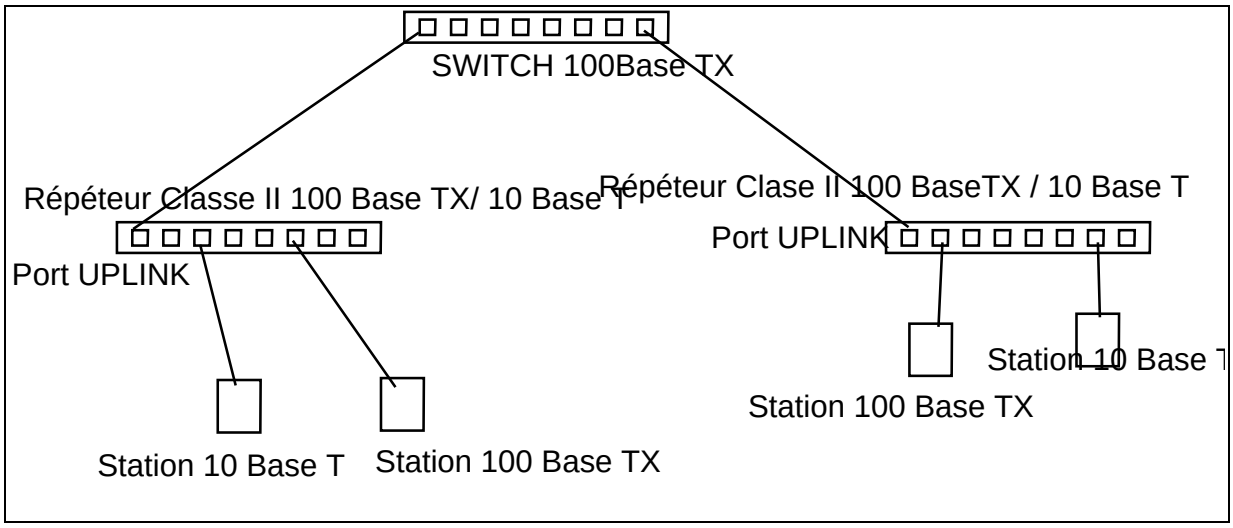
Installation avec répéteur classe 1 : 2 liaisons sont permises dont la somme ne peut pas excéder 272m

Installation avec répéteur classe 2 :

2 liaisons + 1 liaision inter-répéteur, la somme ne peut pas excéder 228m

b) Les switch

Switch to switch : longueur max 412m ou 2000m



F. ATM Asynchronous Transfer Mode

1. Généralités

Les réseaux locaux actuels ont des problèmes de temps de transmission. Pour les données, cela ne pose pas de problèmes mais bien pour de la vidéo et du son.

ATM doit permettre le transfert de fichiers mais aussi de vidéo. Pour cela, ATM utilise des petites cellules de taille fixe pour transmettre les informations.

ATM se compose de commutateurs qui permettent à chaque liaison de fonctionner indépendamment.

Chaque noeud est relié par une liaison point-à-point à un commutateur ATM.

ATM crée des liaisons virtuelles (pas de routage de chaque paquet de données) et un adressage hiérarchique.

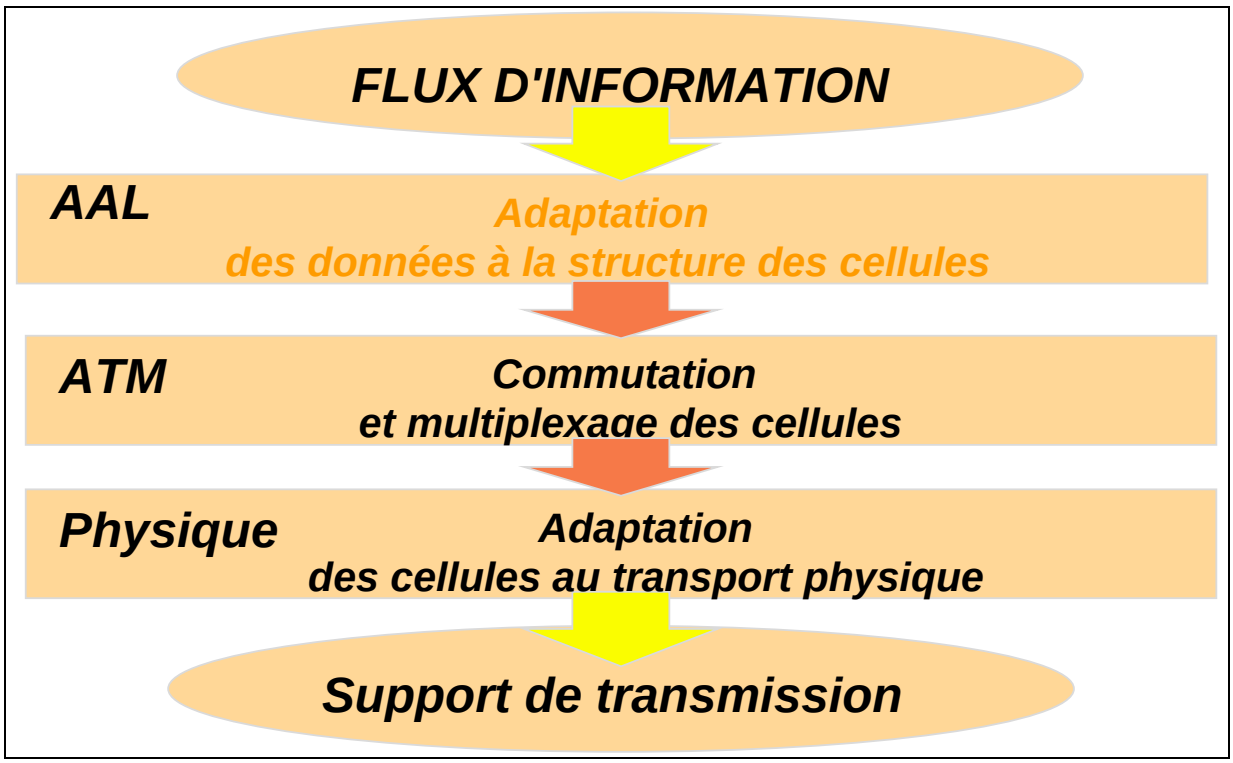
ATM permet de créer des réseaux virtuels (groupe d'utilisateurs) au départ de noeud séparés géographiquement.

Les connexions entre commutateurs ATM sont appelées NNI (Network-to-Network Interface). Un commutateur qui est uniquement relié à d'autres concentrateurs est appelé " brasseur "

Les connexions entre commutateurs ATM et noeud sont appelées UNI (User-to-Network Interface)

2. Structuration de ATM

a) Généralités



ATM est subdivisé en 3 couches. Voyons maintenant ces 3 couches en détail.

b) Couche Physique

Il faut remarquer que comme les commutateurs supportent différents débits, il est possible de prévoir des liaisons FO entre commutateurs mais aussi de prévoir des connexions moins coûteuses entre une station ATM et le commutateur si les débits sont moins élevés (une paire torsadée non blindée)

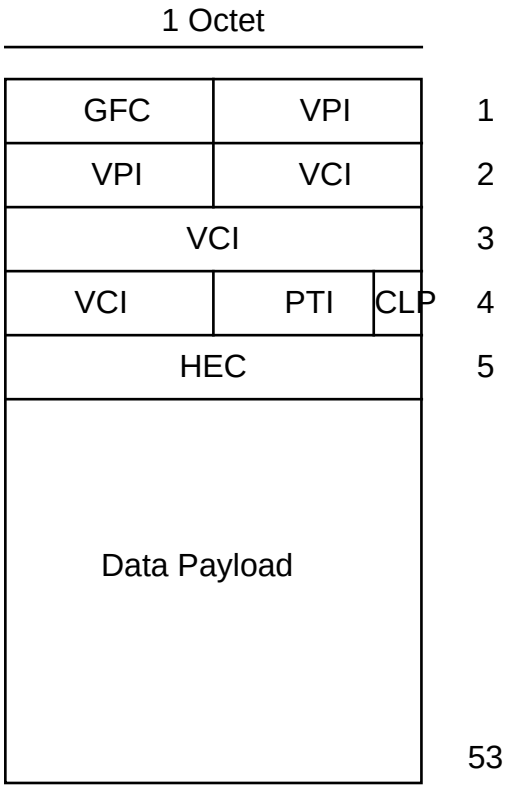
En cours d'étude actuellement :
25-51 Mbit/s sur UTP niveau 3
100 Mbit/s sur FO multimode
155 Mbit/s sur FO monomode et STP
622 Mbit/s sur FO monomode

ATM utilise les systèmes de transport SDH (Synchronous Digital Hierarchy), SONET (Synchronous Optical Network) , FDDI II.

c) Couche ATM

(1) La cellule ATM

Sur une connexion UNI, on trouve des cellules de 53 octets.



Une cellule ATM se compose donc des champs qui suivent:
- Le champ GFC (Generic Flow Control) (4 bits) : permet de réguler le débit de transmission dans le réseau ATM
- Le champ VPI (Virtual Path Identifier) (8 bits) : ce champ fait partie de

l'identificateur de connexion ATM. Il identifie un groupe de circuits virtuels qui utilise la même route.
- Le champ VCI (Virtual Channel Identifier) (16 bits) : ce champ fait aussi partie de l'identificateur de la connexion ATM. Ce champ identifie une connexion entre deux stations ATM.
- Le champ PTI (Payload Type) (3 bits) : ce champ indique si le champ de données (Payload) contient des données ou des informations de gestion du réseau.
- Le champ CLP (Cell Loss Priority) (1 bit) : ce champ indique si la cellule peut être effacée en cas de surcharge du réseau.
- Le champ HEC (Header Error Control) (8 bit): cet octet permet de détecter les erreurs qui se trouvent dans les 4 premiers octets de la cellule.
- Le champ Data Payload (48 octets).

On remarque donc qu'il n'y a pas de détection d'erreurs sur les données.

Dans le cas des cellules qui sont transmises sur les liaisons NNI, les cellules sont légèrement différentes car elles n'ont pas besoin du champ GFC puisque cette régulation du flot de cellules se fait sur les liaisons UNI.

(2) Capacité du réseau

Les commutateurs peuvent supporter beaucoup de liaisons à différentes vitesses. La capacité totale est la somme des différentes liaisons dans le réseau. ATM fournit des liaisons point-à-point (pas un bus) donc la sécurité des données est assurée. Les commutateurs ATM envoient les données uniquement au destinataire.

(3) ATM et routage

Comme les informations de routage sont très réduites (24 bits), le routage peut être intégré dans les composants électroniques.

ATM cherche d'abord la route pour atteindre le correspondant puis envoie les données. ATM crée donc des liaisons Virtuelles. Lors de l'établissement du circuit virtuel, il faut indiquer le débit d'information qui doit être supporté (Mbit/s ou cellule/s), le type de données se trouvant dans le champ payload (débit constant ou variable) et la priorité des données (haute ou basse).

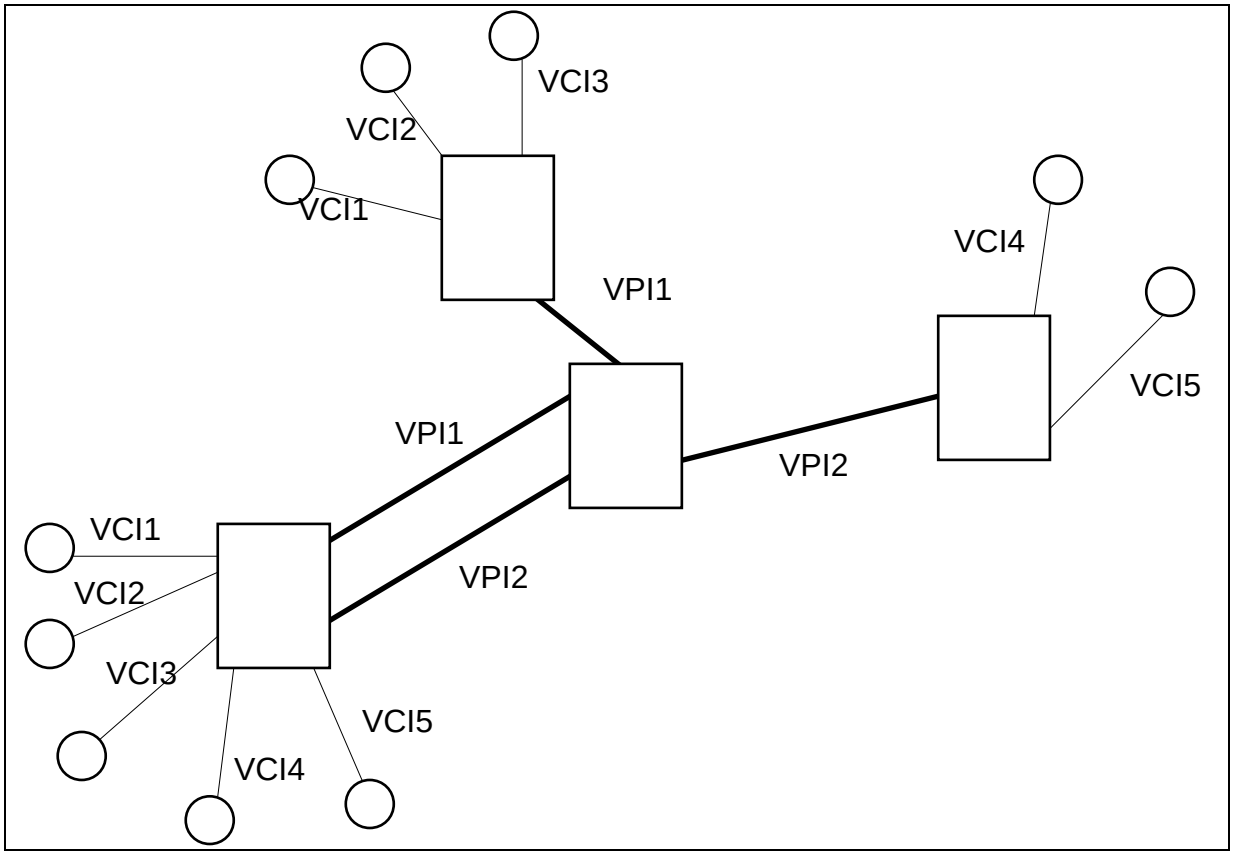
ATM permet d'obtenir deux types de connexions virtuelles :

♦ Connexions Virtuelles Permanente (PVC) : mises en place lors de la configuration du réseau par le gestionnaire

♦ Connexions Virtuelles commutées (SVC) : ces connexions sont dynamiques et peuvent être mises en place par les stations.

Les PVC réservent une partie de la bande passante de manière permanente alors que, dans le cas des SVC, on alloue de la bande passante de manière dynamique.

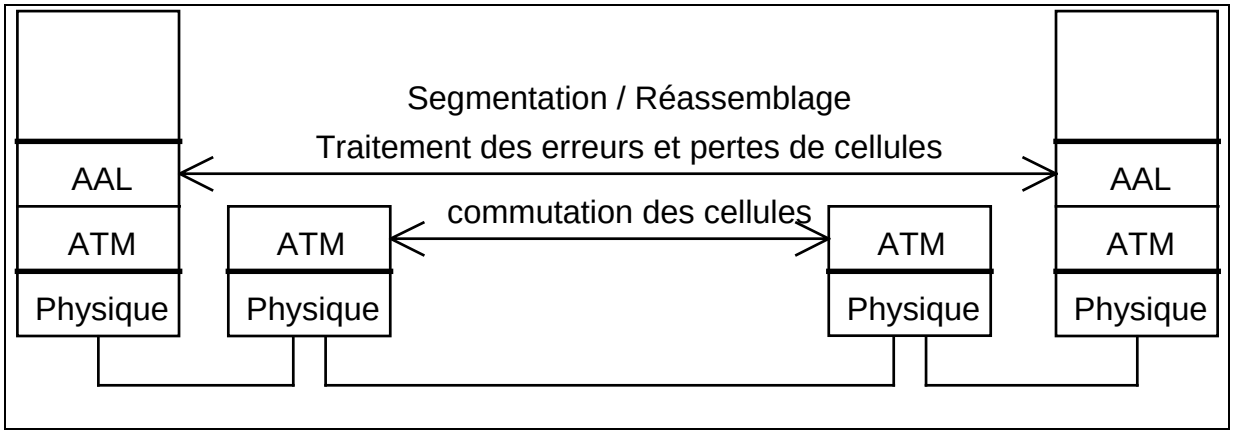
Les données transportées par une connexion virtuelle sont identifiées au travers de 4 classes de données (A à D). Ces classes se différencient par le délai de transfert, un débit constant ou variable, la perte de cellule qu'elles peuvent tolérer ou non, le mode connexion.



d) La couche AAL

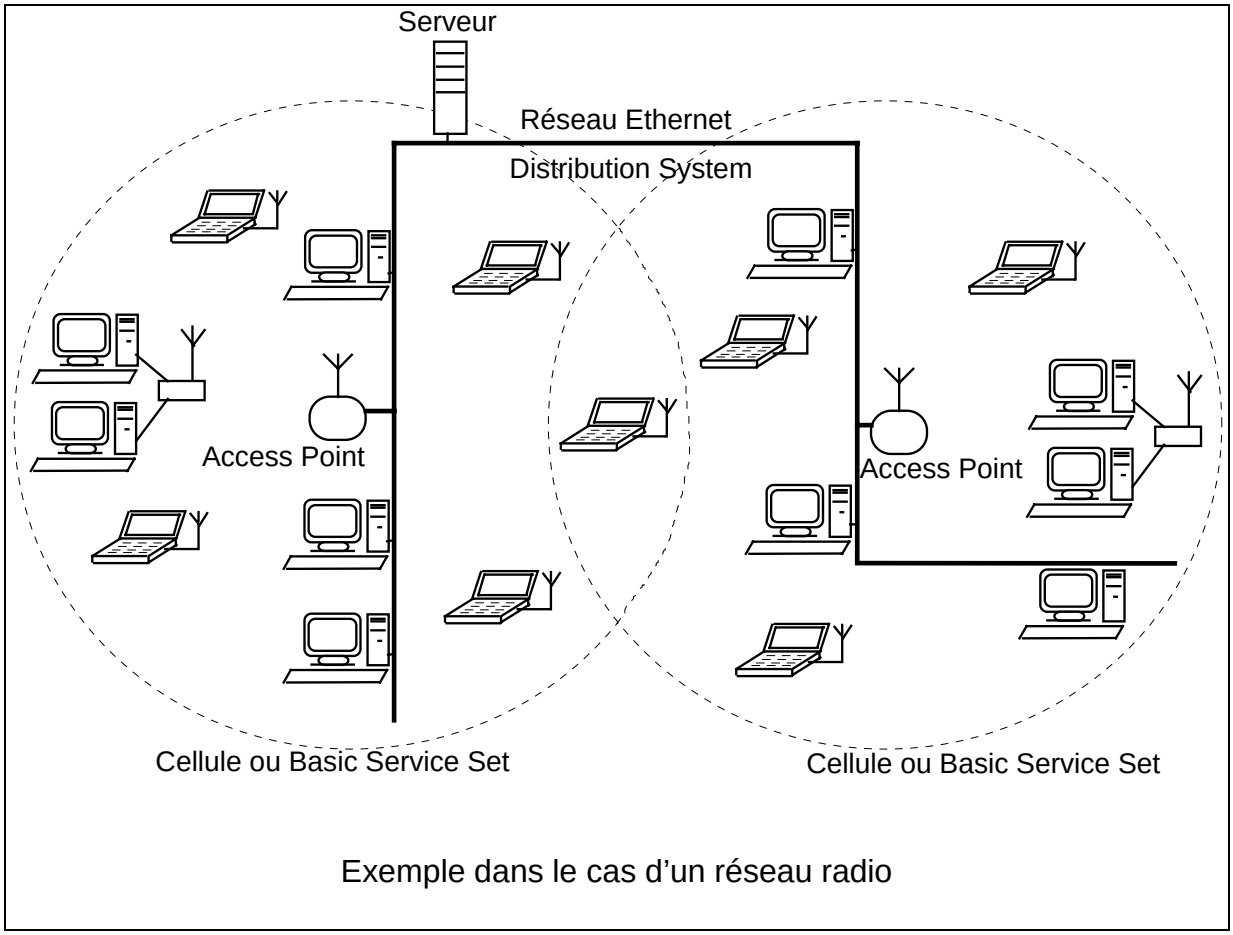
Cette couche offre 5 classes de services :

AAL-1 : Trafic à débit constant. Voix/vidéo
AAL-2 : Trafic à débit variable. Voix & vidéo comprimées
AAL-3 : Trafic par rafale. Mode connecté avec correctioon d'erreur. Multiplexage de cellules
AAL-4 : Trafic par rafale. Mode non connecté avec correction d'erreur . Multiplexage de cellules
AAL-5 : Trafic par rafale. Réseaux locaux.
La couche AAL permet par exemple de transférer des paquets IP sur ATM, d'emuler un réseau LAN.



G. RESEAU SANS FIL (WIRELESSLAN) : IEEE802.11 et IEEE802.11b HR

1. Architecture



Exemple dans le cas d'un réseau radio

Un réseau IEEE 802.11 se compose de :

stations (souvent portables) équipées d'une interface 802.11
de concentrateurs équipés d'une interface IEEE 802.11
points d'accès (Access Point) connectés au réseau cablé (souvent Ethernet) :
l'ensemble des AP et du réseau cablé est appelé Distribution System
de cellules (Basic Service Set) gérées chacune par un point d'accès

2. Les supports physiques de IEEE 802.11

a) Généralités

IEEE 802.11 prévoit une seule couche MAC qui peut utlliser trois couches physiques :

Infrarouge diffus (PPM pulse-position modulation)
FHSS ou Frequency Hopping Spread Spectrum
DSSS ou Direct Sequence Spread Spectrum

b) Infrarouge diffus (PPM pulse-position modulation)

Il offre une portée de 50 à 60 mètres, un débit élévé et n'est pas soumis aux réglementations concernant les ondes radio puisqu'il utilise le spectre visible. Par contre, cette technique a les défauts suivants :

forte consommation de l'interface réseau => utilisable uniquement sur des stations de travail
environnement de bureaux ouverts
pas de roaming possible (passage d'une cellule à une autre sans coupure.

c) FHSS ou Frequency Hopping Spread Spectrum

Il s'agit d'émettre en bande étroite (1 MHz) en changeant de fréquence porteuse à intervalles réguliers dans un canal de spectre large. Les moments et la séquence des sauts de fréquence sont connus de toutes les interfaces réseaux. Il est moins sensible aux brouillages thermiques ou industriels puisqu'il est possible passer à une autre fréquence si la fréquence choisie est brouillée. Elle utilise la bande des 2,4 GHz.

d) DSSS ou Direct Sequence Spread Spectrum

Avec cette technologie, le spectre d'émission d'une station est étalé sur toute la largeur de bande de fréquence disponible dans un canal. Ici, le débit peut être très élevé (11 Mbit/s avec IEEE802.11b High Rate) mais, par contre, l'immunité aux interférences est plus faible et la distance séparant les stations communiquantes joue aussi un rôle important. Elle utilise la bande des 2,4 GHz.

3. La couche MAC de IEEE802.11

a) La méthode d'accès : CSMA/CA

La méthode d'accès de IEEE 802.11 est le CSMA/CA ou Carrier Sense Multiple Access with Collision Avoidance. C'est la même méthode que celle de IEEE802.3 mais ici on essaie d'éviter les collisions. Dans un environnement sans fil, on utilise pas la variante CSMA/CD car :

pour détecter une collision, il faudrait une liaison radio full duplex, capable de recevoir et transmettre immédiatement, ce qui augmenterait fort le prix des interfaces réseaux
dans ce type de réseau, on n'est pas certain que toutes les stations s'entendent entre elles (principe de base de CMSA/CD) et donc le fait que le support soit libre pour l'émetteur n'est pas une garantie qu'il l'est pour le récepteur

Dans IEEE 802.11, on veut donc éviter les collisions et de plus, la station réceptrice doit envoyer un accusé de réception positif. Avant de transmettre la station écoute le support, s'il est occupé, l'envoi est différé. Si le support est libre pendant un temps spécifique alors la station peut transmettre. La station réceptrice vérifie s'il n'y a pas eu d'erreur de transmission via le CRC du fragment et renvoie un accusé de réception positif. Lorsque l'émetteur reçoit l'accusé de réception, il sait qu'aucune collision n'a eu lieu.

b) Fragmentation/ réassemblage

Dans le cas des résaux locaux filaires, les trames peuvent avoir des tailles de plusieurs centaines d'octets. Pour les réseaux locaux sans fil, le taux d'erreur des liaisons radio est beaucoup élevé de sorte que le nombre d'octets envoyés en une fois a été réduit. Aussi la couche MAC implémente un mécanisme de fragmentation/réassemblage qui divise une trame en plusieurs fragments et inversément. La trame peut donc être de type Ethernet ou Token-RING.

c) Mécanisme d'entrée dans une cellule d'une station

Pour accéder à un Basic Service Set (cellule), la station doit d'abord obtenir les informations de synchronisation de la part de l'Access Point. Pour cela, elle peut attendre que l'AP émette une trame balise (Beacon Frame) ou bien elle peut

transmettre une trame de demande d'enquête (Probe request Frame) et attendre la réponse de l'AP. Une fois que l'AP est trouvé, le processus d'authentification est lancé : l'AP et la station échangent des informations qui permettent à l'un et l'autre de prouver leur identité par la connaissance d'un mot de passe. Ensuite commence le processus d'association qui consiste à échanger des informations sur les différentes stations et les capacités de la cellule. A partir de ce moment, la station peut échanger des trames de données.

d) Roaming

IEEE802.11 prévoit d'une station doit pouvoir passer d'une cellule à l'autre sans perte de connexion.

e) WEP (Wired Equivalent Privacy)

Pour éviter une écoute clandestine, la couche MAC crypte les données avec une clé de codage de 40 bits.

f) Economie d'énergie

Dans le cas des portables, la possibilité de passer en mode économie d'énergie est très importante, aussi IEEE802.11 a prévu de faire conserver, par l'Access Point, les trames destinées aux stations en veille jusqu'à ce que ces dernières les demandent. De plus, les AP indiquent dans la trame balise (qu'ils émettent périodiquement) quelles stations ont des trames en attente.

Possibilité de liaison point à point entre deux stations sans la présence d'un Access Point.

IV. LES SERVEURS

A. Généralités

Dans un réseau, on trouve trois manières de partager les ressources :
- soit un réseau avec des partages **poste à poste** (Peer to Peer)
- soit un réseau comportant des **serveurs «centraux»** et des clients
- soit un mélange des deux types de partage

Les Serveurs sont les systèmes qui mettent des ressources à disposition des clients, utilisateurs du réseau.

Les Clients sont les machines qui utilisent les ressources partagées.

Différentes ressources sont partageables au travers d'un réseau. En fonction du type de ressource, on parlera de :

Serveurs de fichiers : ils partagent leurs espaces disques
Serveurs d'imprimantes : ils partagent les imprimantes qui y sont connectées
Serveurs de communications : ils partagent les modems/ fax qui y sont connectés
Serveurs d'applications : ils partagent l'application qu'ils exécutent (ex : Système Gestion Base de Données Relationnelle)

B. Le réseau poste à poste

Dans ce type de réseau, chaque poste peut être client et serveur en même temps. La taille des réseaux poste à poste est limitée (une dizaine de machines).

Ce type de réseau nécessite un investissement relativement faible par machine.

La gestion de la sécurité se fait au niveau de chaque machine où chaque utilisateur indique les ressources partagées ainsi que les restrictions d'accès. Il est possible de fournir un accès complet (RW) ou en lecture seule : un mot de passe est associé à chaque type d'accès. La gestion n'est pas centralisée.

Pour accéder aux ressources partagées, les clients doivent fournir un mot de passe : en fonction du mot de passe, les clients auront un accès complet ou en lecture seule.

C. Les serveurs de fichiers

1. Généralités

Les serveurs de fichiers permettent d'étendre virtuellement la capacité de stockage des stations connectées au travers du réseau au serveur et de partager des données, les logiciels. Ils permettent le partage des fichiers et des ressources entre les différentes stations du réseau. Ils permettent en outre le partage de ressources comme les imprimantes par exemple, des modems, des faxs. Ces fonctions de partage de périphériques peuvent aussi se faire sur des postes de travail.

Les serveurs de fichiers sont gérés par un système d'exploitation réseau (Netware (Novell), Windows NT Server (Microsoft), Lan Server (IBM), Lan Manager (Microsoft), UNIX)

Les serveurs peuvent être des systèmes dédiés (Serveur Novell) ou non (UNIX, Lan Server, Windows NT,...).

Nous détaillerons ici les concepts utilisés au niveau des serveurs Novell. Il seront facilement transférables vers d'autres systèmes de serveurs de fichiers.

2. Connexion au serveur de fichiers

La connexion au serveur de fichiers se fait toujours par une phase d'identification de l'utilisateur au niveau du serveur de fichiers avec envoi d'un mot de passe optionnel (mais fortement recommandé). C'est la phase de "Login".

Une fois que cette opération est réussie. Le logiciel réseau exécuté sur la station Cliente permet d'associer un identificateur d'unité logique (e:, f:, g:, ... dans le monde MS-DOS) à un répertoire de fichiers situé dans le serveur. Pour les utilisateurs, les accès aux fichiers du serveur se font de la même manière que s'il

s'agissait de fichiers sur un disque local. C'est le logiciel de réseau qui intercepte les demandes de lectures/écritures vers les fichiers du serveur, qui veillent à transformer ces requêtes compréhensibles par le serveur et qui traduit les réponses du serveur en informations compréhensibles par la station Cliente.

3. Exemple dans le cas du logiciel réseau Netware 3.11

Netware 3.11 est le système d'exploitation du serveur de fichiers. Ce système d'exploitation est multitâches, supporte différentes configurations hardware de réseau (Ethernet, Tokenring, ...).

Chaque serveur est en plus routeur pour le protocole réseau IPX, utilisé par les serveurs Novell Netware et les stations Clientes pour acheminer les données au travers du réseau. IPX (Internetwork Packet Exchange) contient les informations de routage et de type de protocole de couche supérieur utilisé.

Dans les couches Transport et supérieures, les serveurs et les stations utilisent le Netware Core Protocol (NCP) qui permet de se connecter aux serveurs et SPX (Sequenced Packet Exchange) qui assure la transmission fiable de données (utilisé pour les impressions).

a) Les gestionnaires réseau des stations Clientes (Novell)

Dans la station client, il faut une des deux séries de logiciels suivants pour pouvoir se connecter au serveur Novell :

IPX : ce programme permet d'échanger des paquets IPX sur le réseau. Il est propre à chaque marque de carte réseau puisqu'il contient le logiciel nécessaire pour commander l'électronique de la carte réseau. Ce logiciel doit être configuré au niveau des interruptions, de la mémoire partagée et des ports d'entrées/sorties utilisés. Si un des éléments est mal configuré, le programme ne parviendra pas à fonctionner. Et, **NETX** : ce programme transforme les requêtes MS-DOS d'accès aux fichiers en ordre compréhensible par le serveur et renvoie les résultats sous format MS-DOS au Client. Par l'utilisation des logiciels ipx et netx, on se limite à l'accès au serveur Novell uniquement (un seul protocole possible).

LSL (Link Support Layer), **driver carte** (MLID ou Multiple Link Interface Drivers), **IPXODI**, **NETX**. Cette solution permet d'utiliser plusieurs protocoles en même temps sur la carte. On peut donc se connecter à un serveur Netware mais aussi faire des transferts sous TCP-IP simultanément

Une fois que les programmes ont été lancés dans l'ordre. Le répertoire f:\login (si LASTDRIVE =E dans config.sys) apparaît dans les unités logiques disponibles. A ce moment, on pourra effectuer le "login"

b) Les informations utilisées par le Serveur

Dans tout système serveur de fichiers, il existe toujours une base de données reprenant l'identification d'un utilisateur au sein du système ainsi que son mot de passe.

Cette Base de données contiendra aussi les informations sur

- les privilèges d'accès des différents utilisateurs
- la date de dernière connexion des utilisateurs
- le possesseur des fichiers (chaque fichier du serveur appartient à un utilisateur)
- les groupes d'utilisateurs définis ainsi que les droits associés à ces groupes
- le type d'accès permis au niveau de chaque fichier
- La date et heure de création, de dernier accès, de dernière modification des fichiers

Au niveau des utilisateurs, le système de gestion du serveur de fichiers permet souvent :

- de fournir les informations sur les ressources disques employées par chaque utilisateur et de les "facturer" le cas échéant
- de permettre l'accès à certains utilisateurs au départ de certaines stations et à certains moment du jour uniquement

D. Sécurité de fonctionnement /Sauvegarde

Afin de se prémunir contre une panne au niveau du serveur , il existe différentes possibilités :

Disk mirroring
2 disques et un contrôleur

Disk Duplexing
2 disques et deux contrôleurs

Server Mirroring
2 Serveurs synchronisés au travers d'un réseau haut débit

Dans les environnements de micro-ordinateurs, la sauvegarde se fait principalement sur :

- bande DAT (lecteur cher mais bandes bon marché par rapport à la capacité)
- WORM
- disques magnéto-optiques (accès très rapide plus cher capacité limitée)
- data cartrigde (lecteur pas cher mais bande relativement chère)

Au niveau de la sauvegarde des données, on peut utiliser cette manière de faire :

quotidienne (incrémentale) : uniquement les fichiers qui ont l'attribut Archive sont sauvés.

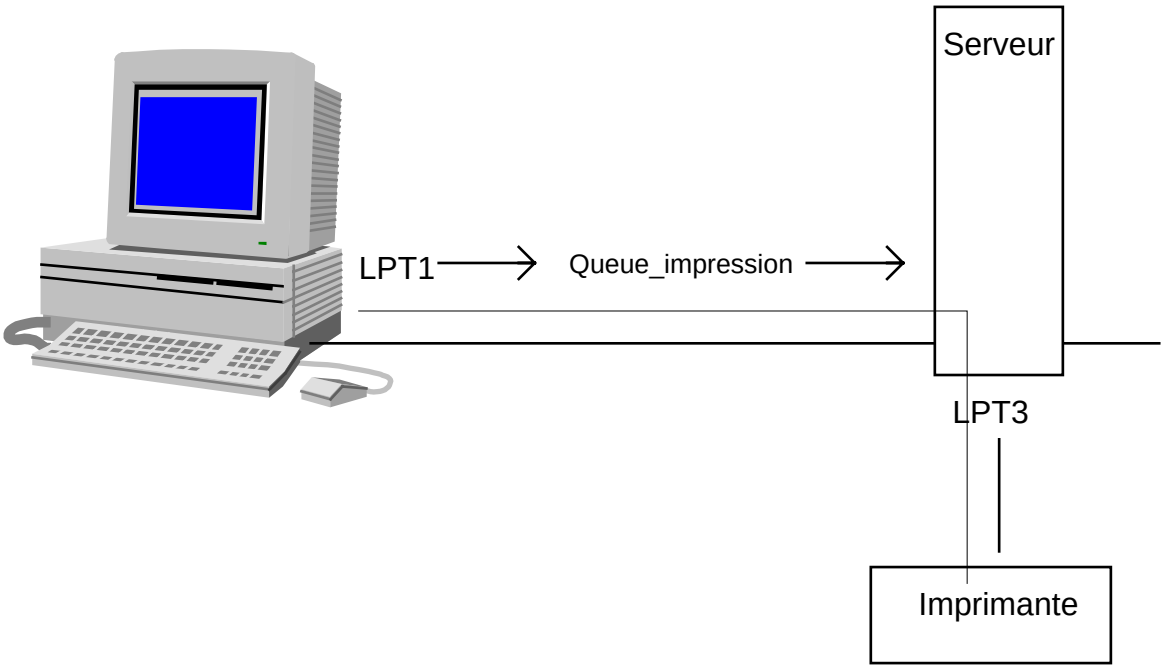
hebdomadaire (complète)

mensuelle (complète) : cette copie des données est mise en lieu sûr.

La politique de sauvegarde des données est très importante; le type de sauvegarde et la fréquence dépenderont du temps dont on dispose pour restaurer un système en cas de panne.

E. Les serveurs d'impressions

Au niveau des serveurs Novell, il faut définir des serveurs d'imprimantes permettant de commander les ports de sorties parallèles et série pour faire les impressions. Les serveurs d'impressions impriment les données qui proviennent des **queues d'impression.** Les queues d'impression reçoivent les données provenant de la redirection des ports d'imprimantes des stations Clientes. Les queues d'impression consistent en une pile FIFO de jobs d'impressions : si deux stations clientes impriment en même temps dans la même queue d'impression, il n'y aura pas mélange des deux impressions.



On remarquera que le serveur d'imprimante peut être confondu avec le serveur de fichiers. Dans ce cas, les ports d'imprimantes du serveur de fichiers sont raccordés aux imprimantes.

Il faut remarquer qu'une queue d'impression peut desservir plusieurs imprimantes physiques mais aussi qu'une imprimante peut desservir plusieurs queues d'impression.

V. LE MONDE TCP-IP

A. Historique

Aux Etats Unis, vers la moitié des années 70, DARPA (Defense Advanced Research Projects Agency) a fourni des budgets de recherche destinés à permettre de relier les différents centres de recherche de l'armée américaine. Jusqu'à ce moment, les systèmes étaient interconnectés par le réseau ARPANET, réseau à commutation de paquets, basé sur des connexions point-à-point sur lignes louées. Les protocoles TCP-IP ont pris leur forme actuelle entre 1977 et 1979. Le réseau Internet est apparu vers 1980 lorsque DARPA commença à convertir les ordinateurs de ses centres de recherche en noeuds utilisant TCP-IP : le réseau ARPANET devint la dorsale pour le réseau Internet. En janvier 1983, la transition vers la technologie TCP-IP devint complète lorsque OSD (Office of the Secretary of Defense) décida que les systèmes connectés à des réseaux longues distances devaient utiliser TCP-IP. A ce moment, DCA (Defense Communication Agency) sépara ARPANET en deux parties distinctes : ARPANET (relie les centres de recherche) et MILNET (destiné aux communications militaires).

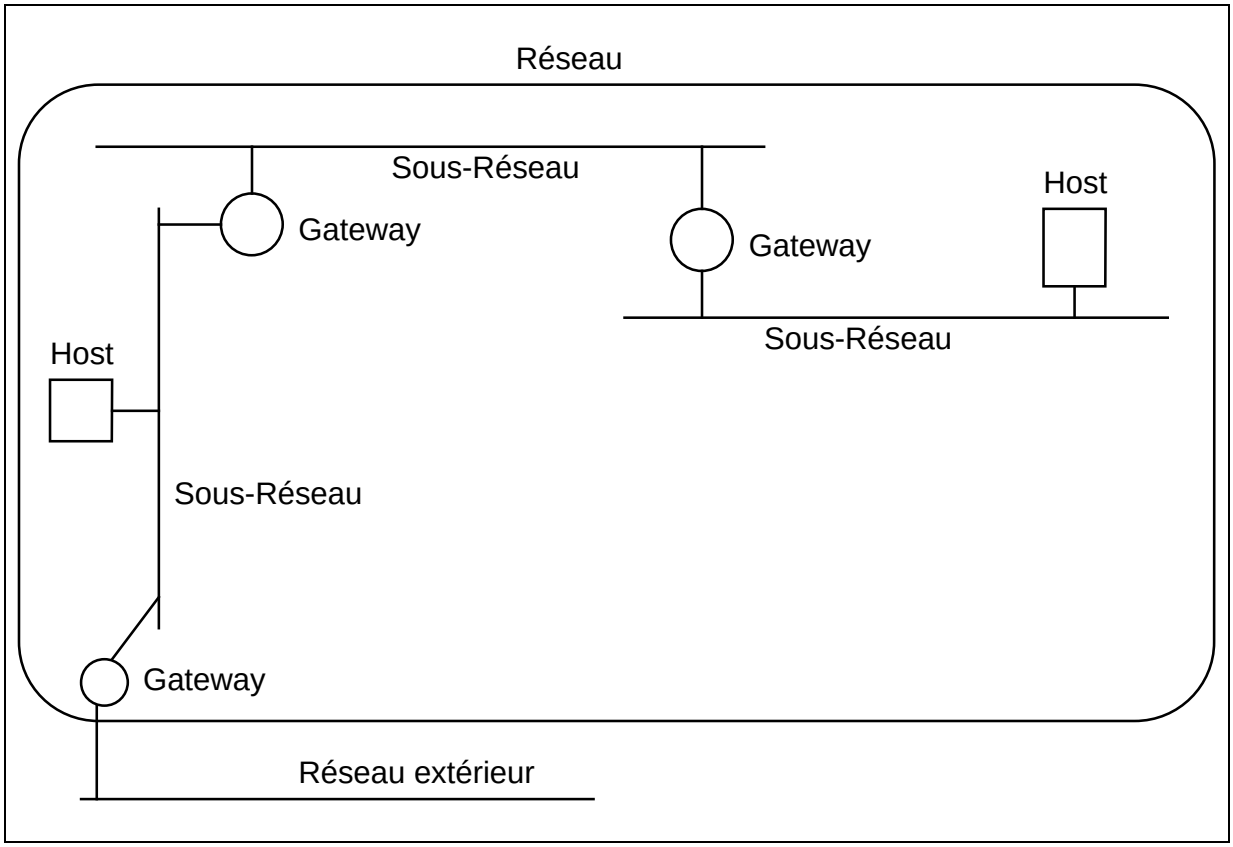
Afin de promouvoir l'utilisation de TCP-IP parmi les chercheurs universitaires, DARPA fournit une implémentation à faible coût de TCP-IP. TCP-IP fut intégré dans BSD Unix (Berkeley Software Distribution) de l'université de Californie. En 1986, NSF (National Science Foundation) créa une nouvelle dorsale longue distance (NSFNET) pour relier un maximum de scientifiques.

Depuis lors le réseau Internet est devenu mondial et compte pas moins de 100 millions de machines. TCP-IP a été adopté par beaucoup d'entreprises de différents secteurs pour leurs communications internes et ces dernières sont aussi souvent connectées au réseau Internet.

Comme on le voit, TCP-IP n'appartient pas à une entreprise et est adapté à beaucoup de hardwares. Les RFC (Request For Comment) détaillent les différents protocoles de TCP-IP, fournissent des rapports techniques et des propositions de nouveaux protocoles.

B. Les composantes d'un réseau local sous TCP-IP

Un réseau typique comporte des stations (***Host***) et des routeurs (***Gateway***) permettant d'interconnecter les différents sous-réseaux composant le réseau ou de connecter le réseau avec un réseau extérieur. Le gateway peut être un système dédié mais aussi simplement une station de travail qui comporte plusieurs cartes réseau étant connectées sur différents réseaux. Cette station de travail doit bien sûr disposer du logiciel permettant le routage.

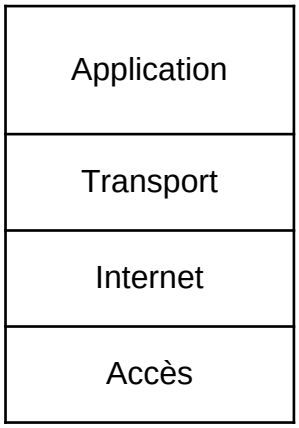


TCP-IP prévoit différents types de supports physiques. Dans notre cas, nous l'analyserons pour Ethernet mais TCP-IP peut aussi utiliser des transmissions radio, des transmissions sur ligne téléphonique, etc.

Les stations échangent des **paquets IP** pour communiquer. Les Routeurs veillent à faire parvenir les paquets IP chez le destinataire sur base de l'adresse du réseau ou sous-réseau.

Le réseau **Internet** correspond à un réseau virtuel composé de réseaux physiques (sous-réseaux) interconnectés par les Routeurs (Gateway).

C. Architecture de TCPIP



Architecture de TCP-IP

TCP-IP a prévu une structure en 4 couches du système de communication. TCP-IP est antérieur au modèle OSI de ISO. Quand TCP-IP a été défini, on avait donc prévu moins de couches.

La couche Accès correspond à peu près aux couches Physique et Liaison de donnée. La couche Internet correspond à la couche réseau du modèle OSI. La couche Application est l'équivalent des couches Session, Présentation et application de OSI.

D. La couche Réseau

1. L'adressage IP des stations

a) Les adresses IP

Toutes les interfaces réseaux des stations connectées à un réseau sous TCP-IP ont chacune une adresse IP différente.

Cette adresse est codée sur 4 octets ou 32 bits. Habituellement, on note les adresses de la manière suivante : 125.132.95.250

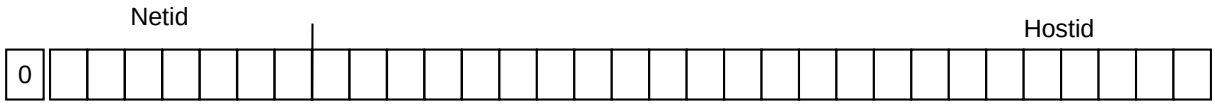
Cette adresse comporte une partie identifiant le réseau et une partie identifiant la station sur le réseau.

Adresse = (netid,hostid)

En fonction du nombre de stations à connecter dans le réseau, on définit différentes classes de réseau TCP-IP.

(1) Réseau de Classe A

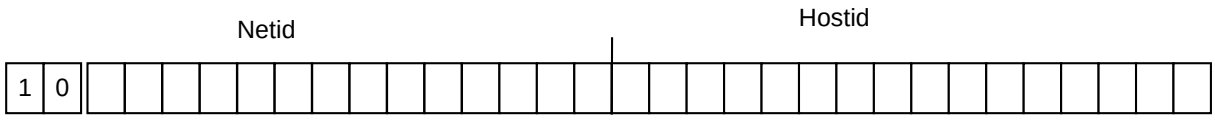
Les adresses de Classe A comportent 1 octet pour la partie **netid** et 3 octets pour la partie **hostid**.



Ces adresses seront caractérisées par le premier octet ayant une valeur comprise entre 0 et 126.

(2) Réseau de Classe B

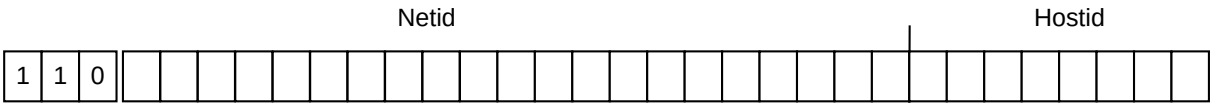
Les adresses de Classe B comportent 2 octets pour la partie **netid** et 2 octets pour la partie **hostid**.



Ces adresses seront caractérisées par le premier octet ayant une valeur comprise entre 128 et 191.

(3) Réseau de Classe C

Les adresses de Classe C comportent 3 octets pour la partie **netid** et 1 octet pour la partie **hostid**.



Ces adresses seront caractérisées par le premier octet ayant une valeur comprise entre 192 et 223.

Les adresses commençant par **127** sont destinées au loopback.

Classe D (224 à 239) adressage multicast

Classe E (240 à 247) extensions futures

Une adresse correspondra à une adresse de réseau si les bits de la partie hostid sont à 0.

Une adresse broadcast sur un sous-réseau déterminé aura les bits de la partie hostid à 1.

Broadcast sur le sous-réseau local : 255.255.255.255

(4) Les sous-réseaux (Subnet)

Un réseau d'une certaine classe peut être ou doit être subdivisé en plusieurs sous-réseaux. Par exemple un réseau de classe B, s'il ne comporte que des stations sur Ethernet, devra être composé de plusieurs sous-réseaux puisque un réseau Ethernet peut comporter un maximum de 1024 stations.

Dans le cas de sous-réseaux, une partie des bits réservés pour la partie **hostid** des adresses est utilisée par la partie netid. On utilise le masque de sous-réseau pour indiquer quels bits de la partie hostid de l'adresse est utilisée pour identifier le sous-réseau (Subnet Mask).

Attention, l'adresse du sous-réseau (bits de la partie hostid à 0) et l'adresse broadcast du sous-réseau (bits de la partie hostid à 1) ne sont pas

utilisables en tant que adresse de host. De plus, les adresses IP du sous-réseau **0** ne sont pas utilisées ainsi que celle du dernier sous-réseau.

Supposons que nous sommes en présence d'un réseau de classe C (192.6.250) et que ce réseau comporte 3 sous-réseaux reliés par deux gateways. Dans ce cas, il faut prévoir un masque de sous-réseau permettant de coder 3 adresses de sous-réseau. Il faut donc prendre 3 bits (8-2=6 possibilités) de la partie hostid. Le masque de sous-réseau sera donc

255.255.255.224 (ou en binaire 11111111.11111111.11111111.11100000)

Par l'intermédiaire de ce masque, on sait que les 5 derniers bits de la partie hostid sont réellement affectés à l'adresse de la station sur le sous-réseau.

Il faut remarquer qu'une station ne peut pas être déconnectée d'un réseau et reconnectée à un autre réseau sans modifier la configuration de la station. On ne peut donc pas simplement déplacer une station et la connecter au nouveau réseau présent à son nouvel emplacement (à moins que ce ne soit le même réseau (et sous-réseau)).

b) DNS Domain Name Service (application)

Quand le service de noms de domaines n'est pas utilisé, le fichier /etc/hosts de chaque station contient les informations qui permettent de faire correspondre une adresse IP à un nom de station et ses alias. Dans le cas de petits réseaux, il est possible de gérer les fichiers /etc/hosts de chaque station. Par contre dans le cas de réseaux plus importants en taille, il est nécessaire d'utiliser les services de noms de domaines car le fichier /etc/hosts devient trop important et trop difficile à gérer.

Les noms de domaines représentent une organisation imposée pour des raisons administratives. Le système des noms de domaines utilise une structure hiérarchique.

Nom canonique de station : ns.gramme.hemes.be.

Il existe un domaine racine qui est **. (point)**

En dessous du domaine racine, on trouve les domaines principaux. Dans notre exemple, il s'agit de **be** (domaine géographique). **hemes** est un sous-domaine de **be**. Et ainsi de suite. **ns** représente le nom de la station.

Cette structure hiérarchique permet de distribuer la gestion des adresses et des noms de stations.

Le service DNS est assuré par un serveur primaire de domaine, un serveur secondaire de domaine et les serveurs cache. Les serveurs primaire et secondaire contiennent les informations relatives à la partie locale de l'arbre DN qu'ils gèrent. Les informations contenues dans ces serveurs doivent être gérées de manière très stricte. En cas de demande d'adresse d'une station en dehors de cette partie locale de l'arbre, le serveur primaire envoie une requête vers le serveur de la racine du domaine. Ce dernier indique le serveur de domaine à contacter pour obtenir l'adresse IP de la station en question.

Exemples de TLD ou Top Level Domain (voir http://www.icann.org)

.com, .net, .org : commercial
.edu : éducation USA
.gov : gouvernement USA
.be, .nl, .ch, ... : région géographique

De nouveaux TLD ont été créés récemment (.aero, .biz, .coop, .info, .museum, .name, .pro)

2. ARP ou Address Resolution Protocol (RFC 826) - RARP Reverse Address Resolution Protocol (RFC 903) (couche accès)

Comme on aura pu le remarquer, une adresse IP ne correspond pas à une adresse Ethernet. En effet, les longueurs de ces différentes adresses ne sont pas identiques. Aussi doit -il exister un système qui permet de tout de même utiliser les adresses IP et les adresses Ethernet.

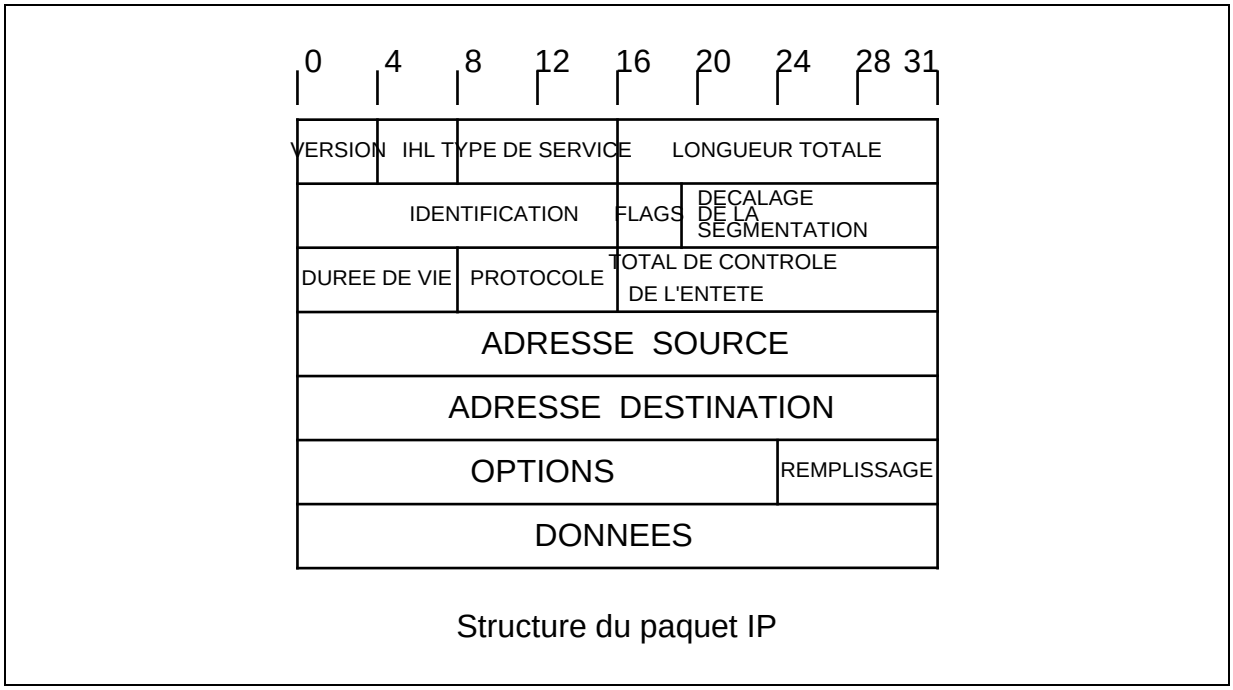
ARP permet d'associer l'adresse de l'interface Ethernet à l'adresse IP de l'interface. Chaque station dispose d'une table qui met en correspondance l'adresse IP d'une station et l'adresse Ethernet vers laquelle il faut envoyer le paquet IP pour adresser cette station.

RARP permet d'obtenir l'adresse IP d'une station au départ de l'adresse de l'interface Ethernet de la station. Pour cela, il faut prévoir des serveurs pour le protocole RARP. Ce serveur doit disposer d'une table qui fait la correspondance entre l'adresse d'une carte Ethernet et l'adresse IP.

Ces protocoles font partie de la couche Accès de TCP-IP.

3. IP ou Internet Protocol (RFC 791)

IP (Internet Connectionless Packet Delivery Service) fournit un service réseau non connecté, non fiable. Il correspond à la couche Réseau du modèle OSI.



Structure du paquet IP

Un paquet IP comporte un en-tête et une partie de données.

Les principales informations codées dans l'entête sont décrites ci-après.

Le champ **VERSION** indique la version du protocole IP utilisée (4). Le champ **IHL** indique la longueur de l'entête en multiple de 4 octets (IHL min=5 (20 octets) et IHLmax= 15 (60 octets). Le champ **TYPE DE SERVICE** permet d'indiquer le type de service demandé (priorité du paquet, la qualité de service demandée).

Un paquet IP (datagramme) peut avoir une longueur maximale de 65535 octets (champ **LONGUEUR**). Chaque Datagramme est identifié de manière unique par un entier (Champ **IDENTIFICATION** de 2 octets). Il comporte bien sûr l'adresse IP source et destination.

Comme les supports de transmissions ne supportent pas tous les mêmes longueurs de trames, un Datagramme peut être fragmenté afin de pouvoir être envoyé sur le support physique. On parlera du MTU (Maximum Transfer Unit) de chaque support de transmission. Le MTU correspond au nombre de données maximum qu'une trame peut véhiculer (Ethernet MTU=1500, IEEE802.3 MTU=1492) . Chaque datagramme comporte donc des flags (champ **FLAGS**) indiquant qu'il s'agit d'un fragment ou du dernier fragment ainsi que l'offset du fragment dans le datagramme complet (Champ **DECALAGE DE LA SEGMENTATION**).

Les fragments d'un même datagramme ont tous la même identification et la même structure qu'un datagramme IP normal. Ils seront réassemblés par le destinataire.

Le datagramme IP comporte en plus des champs (**OPTIONS**) permettant d'indiquer le chemin que le paquet doit suivre (les différents gateway par lesquels il doit passer) (Source Routing) mais aussi des champs permettant de voir par où le Datagramme est passé et à quel moment.

Pour éviter qu'un datagramme ne circule indéfiniment sur le réseau, le datagramme comporte aussi un champ **DUREE DE VIE** (Time-To-Live) qui est décrémenté lors du passage par un gateway. Lorsque ce champ est à zéro et qu'il n'a pas encore atteint son destinataire, le gateway qui observe cette valeur 0 du champ efface le datagramme. Le champ **PROTOCOLE** indique le protocole

transport utilisant le paquet IP c'est à dire qu'il indique à quel protocole transport utilise la partie donnée du paquet.

L'entête est protégée par un Checksum.

4. ICMP ou Internet Control Message Protocol (RFC 792)

Ce protocole permet :

- de contrôler le débit d'un émetteur de paquets IP
- d'indiquer qu'un réseau ou une station est inaccessible
- d'indiquer à une station de mettre sa table de routage à jour
- de demander un écho (ceci permet de vérifier l'accessibilité d'une station)

5. Les protocoles de routages

a) Généralités

Le routage se fait sur base de la partie de l'adresse IP identifiant le réseau ainsi que le sous-réseau (Subnet). Les gateways disposent d'une table qui associe à une adresse de réseau une adresse de gateway (sauf s'il s'agit des réseaux auxquels il est connecté). Ils connaissent donc l'adresse des autres gateways vers lesquels il faut envoyer le paquet IP pour qu'il atteigne le destinateur.

b) Routage interne

(1) RIP (Routing Information Protocol)

Il assure la mise à jour des tables de routage de gateway. Un réseau peut être à un distance maximale de 15 sauts. Un saut correspond à un passage de gateway. RIP utilise le chemin comportant le nombre minimum de sauts pour atteindre la destination ce n'est pas toujours la route la plus rapide. Ce protocole ne peut pas être utilisé dans le cas de réseau WAN.

(2) OSPF (Open Shortest Path First)

OSPF permet d'associer un coût à chaque route. Ainsi, OSPF sait accomplir du "Load Balancing" et donc un acheminement multivoies des paquets en

fonction de la charge de chaque route. Du point de vue échange des informations de routage, OSPF nécessite une petite bande passante puisqu'il échange uniquement les modifications des routes et seulement quand cela est nécessaire.

c) Routage externe

(1) BGP : Border Gateway Protocol

(2) EGP (Exterior Gateway Protocol)

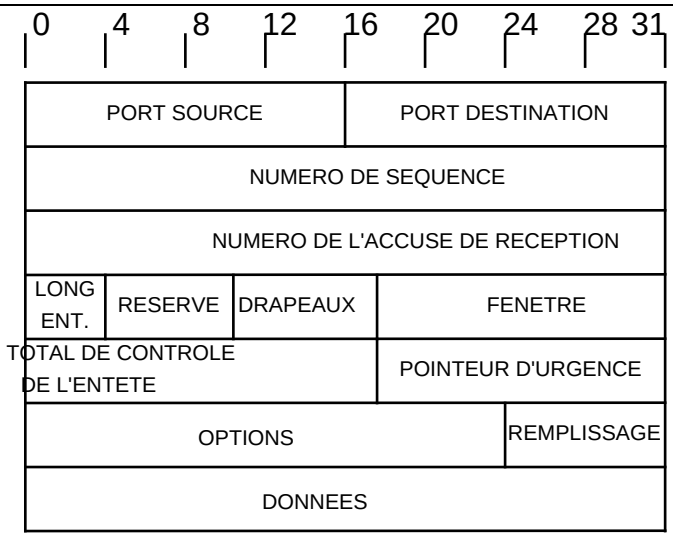
Ce protocole permet aux gateway qui sont connectés sur une dorsale de vérifier qu'une destination peut être atteinte.

E. La couche Transport

1. TCP ou Transmission Control Protocol (RFC 793)

Ce protocole assure une transmission fiable de bout en bout des données entre le port de la station émettrice et le port de la station réceptrice. Ce service est totalement orienté connexion. Il dispose de différentes phases (établissement de la connexion, échange de données et maintien de la connexion, fermeture de la connexion). Comme la connexion se fait au travers de ports, une station peut supporter plusieurs connexions TCP simultanément.

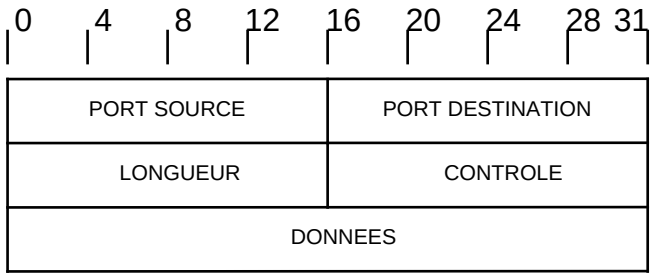
Le fichier **services** contient la correspondance entre le port et l'application.



Structure du segment TCP

2. UDP ou User Datagram Protocol (RFC 768)

Ce protocole assure un service sans connexion, non fiable. Il permet d'envoyer des messages d'une station à une autre. Aucune détection de perte de données n'est assurée, ni de détection d'erreur. L'application qui utilise UDP doit s'en charger. Le fichier **services** contient la correspondance entre le port et l'application.



Structure du message UDP

F. La couche Application

1. Telnet (RFC854/855)

Cette application est un émulateur de terminal. Elle permet d'ouvrir un terminal ASCII sur une machine distante. Cette application se compose d'une

application Cliente (Telnet) et d'une application Serveur (Telnet Daemon ou telnetd). Attention le nom de login et le mot de passe sont véhiculés en clair sur le réseau.

telnet newton

2. FTP File Transfer Protocol (RFC 959)

Cette application permet de transférer des fichiers entre systèmes. Les fichiers peuvent avoir un contenu ASCII ou binaire. Elle permet aussi de se déplacer dans l'arborescence de répertoires de la station distante.

Une vérification de l'utilisateur qui essaie de se connecter est effectuée avant le transfert.

Cette application se compose d'une application Cliente (FTP) et d'une application Serveur (FTP Daemon).

FTP se base sur TCP et utilise deux connexions TCP : une pour le transfert de données des fichiers et l'autre pour l'échange des commandes.

Attention le nom de login et le mot de passe sont véhiculés en clair sur le réseau.

ftp newton

3. TFTP Trivial File Transfer Protocol

Transfert de fichiers sans protection d'accès par mot de passe. TFTP n'utilise pas 2 connexions TCP et est donc moins complexe et moins lourd

4. BOOTP Bootstrap Protocol

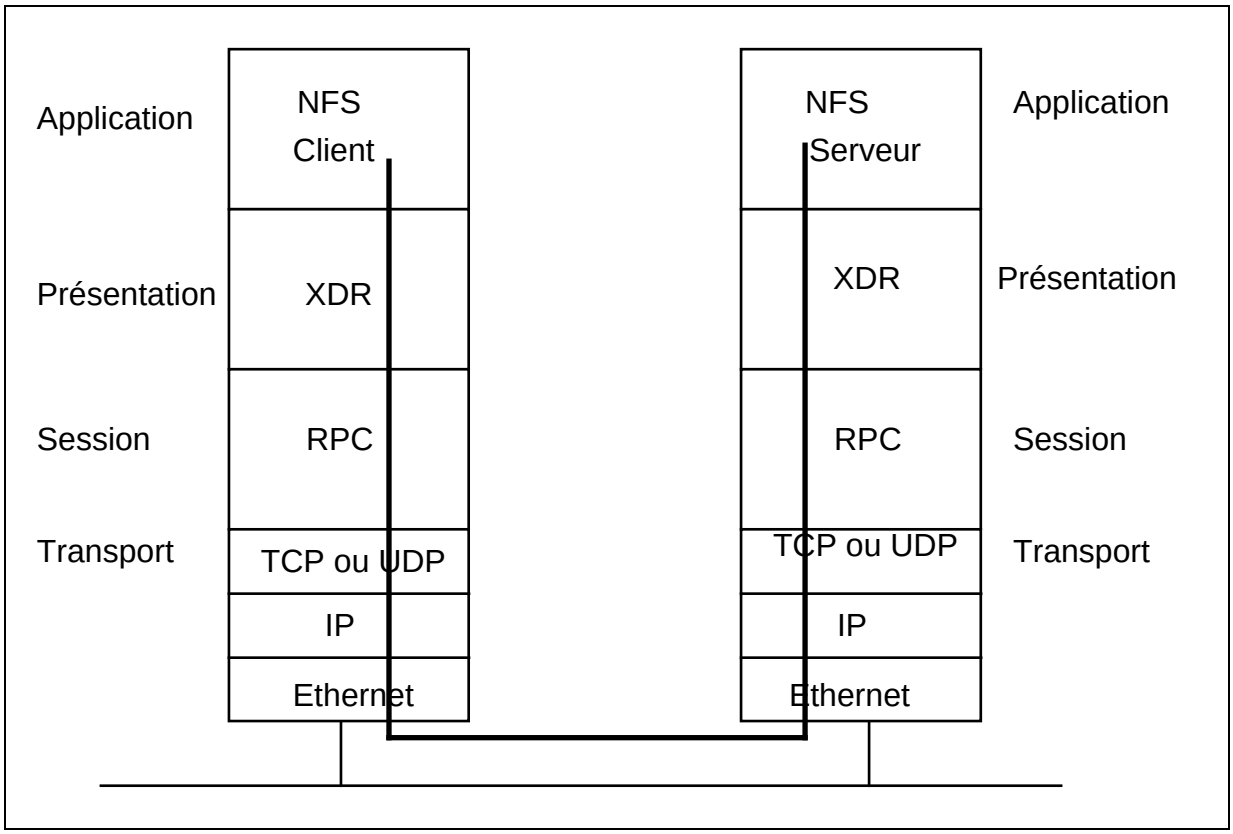
Ce protocole permet à une station Cliente de connaître son adresse IP, l'adresse de la station serveur TFTP et de charger le code de démarrage de la station par TFTP.

5. SMTP Simple Mail Transfer Protocol (RFC 821/822/…)

Un service de routage de messages sur le réseau. Utilise une connexion TCP pour recevoir des messages et une autre pour les envoyer.

Les adresses des boites aux lettres sont de la forme suivante : *nom_utilisateur@nom_officiel_ de_station*

6. NFS Network File System



Ce service permet de monter des répertoires de disques distants sur des répertoires locaux.

Les répertoires qui pourront être "exportés" sont contenus dans le fichier **/etc/exports**. Dans ce fichier, on indiquera les noms des stations (contenus dans /etc/hosts) qui peuvent accéder au répertoire ainsi que le type d'accès autorisé (lecture,écriture).

RPC Remote Procedure Call : RPC permet à des clients d'exécuter de manière transparente des procédures sur des systèmes éloignés. Le client envoie une requête contenant une demande de service et son authentification. Le serveur renvoie la réponse.

XDR External Data Representation : cette couche logicielle permet de transformer les différents systèmes de représentation des données vers une représentation commune au réseau et inversement.

7. X-Windows (norme X11)

Cette application permet de gérer des interfaces utilisateurs types X-Windows.

Cette application comporte deux parties. Le Client X et le Serveur X. Attention, le serveur doit fonctionner dans la station sur laquelle vont apparaître les fenêtres X-Windows tandis que le Client X est l'application qui utilise les fenêtres X-windows pour s'exécuter.

Le fichier /etc/X0.hosts contient les noms des stations qui peuvent accéder au serveur X local.

8. Les services r

Les services r sont dédiés aux systèmes UNIX. Il se basent sur l'équivalence entre utilisateurs de différentes stations.

Les services r peuvent être utilisé si la station distante est configurée d'une des deux manières suivantes :

- Soit :
  - ♦ l'utilisateur qui désire les commandes r dispose d'un compte avec le même nom sur la machine distante
  - ♦ et le nom de la station locale doit se trouver dans le fichier /etc/hosts.equiv distant
- Soit :
  - ♦ l'utilisateur qui désire les commandes r dispose d'un compte sur la machine distante

♦ et le nom de la station locale et le nom d'utilisateur local doivent se trouver dans le fichier .rhosts situé dans son home directory sur la machine distante.

Le fichier /etc/hosts.equiv contient les stations et les utilisateurs fiables pour la totalité du système.

[+|-][nom station fiable] [nom utilisateur]

Quand /etc/hosts.equiv d'une station X ne contient que le nom des stations fiables, les utilisateurs de ces stations fiables ont accès à leur compte sur la station X (si même identificateur) sans devoir utiliser de mot de passe. Si le nom de la station fiable est suivi d'un nom d'utilisateur, cet utilisateur aura un accès total à la station X sans devoir préciser de mot de passe. Les signes + et - permettent d'accorder ou d'interdire l'accès sans mot de passe.

Le fichier .rhosts définit les stations et les utilisateurs fiables pour un compte individuel sur la station X. Ce fichier doit se trouver dans le home directory de l'utilisateur de la station X.

[+|-][nom station fiable] [nom utilisateur]

a) rcp : remote copy

Ce service permet de copier des fichiers, des répertoires et ses sous-répertoires entre différentes stations du réseau.

rcp ruser@rhostname:/pathname/filename file

b) rlogin : remote login

Ce service permet d'ouvrir un terminal sur la station distante.

c) rsh : remote shell

Ce service permet d'exécuter des commandes sur la station distante. La sortie standard et les erreurs apparaissent sur la machine locale.

9. FTP Search

Cette application permet d'accéder à une banque de données distribuées contenant les noms de fichiers de la plupart des archives de logiciels.

10. World Wide Web

Cette application est basée sur hypertext. Ainsi un mot dans un document renvoie vers un autre document. Elle associe textes, sons, images, vidéos.

WWW utilise le protocole HTTP (HyperText Transfer Protocol) et les liens hypertextes. Les pages hypertextes sont codées en utilisant le langage HTML (Hypertext Markup Language)

Un URL ou Uniform Resource Locator permet d'identifier des sites et des fichiers disponible dans le WWW.

11. Messagerie électronique (E-Mail)

Ce service permet l'échange de messages auxquels peuvent être rattachés des fichiers.

La messagerie repose sur des serveurs et des clients. Le serveur est la machine qui reçoit les messages en attendant qu'ils soient consultés par les Clients. La consultation des messages peut se faire directement sur le serveur ou bien les clients se connectent au serveur pour récupérer leurs messages. Les serveurs utilisent le protocole SMTP pour échanger les messages entre eux. Les Clients utilisent, principalement, le protocole POP-3 (Post Office Protocole version 3 - RFC 1939) pour se connecter au serveur et récupérer les messages et le protocole SMTP pour envoyer les messages.

Attention, certains serveurs de mail n'acceptent pas les accentués et de plus la taille des fichiers attachés peut être limitée.

L'information contenue dans les messages n'est pas cryptée et peut donc être lue par des yeux indiscrets.

Face à ce problème de confidentialité, il est possible (pas toujours légal) d'utiliser des logiciels de cryptage à clé publique (PKI). PGP (Pretty Good Privacy) est un de ces algorithmes de cryptage à clé publique.

Dans les systèmes de cryptage à clé publique, chaque correspondant dispose d'une clé privée (à laquelle le correspondant est le seul à avoir accès) et d'une clé publique associée (qui peut être diffusée et qui sera utilisée par vos correspondants). Le correspondant AAA cryptera un message qui est destiné au correspondant BBB au moyen de la clé publique de BBB. BBB décodera le message au moyen de sa clé privée. La taille de la clé peut être configurée.

Le système de cryptage à clé publique (PKI) permet en outre de signer électroniquement le message et donc d'authentifier le contenu ainsi que l'expéditeur du message.

Format d'une adresse E-mail :
nom@nom_serveur_mail

12. Network News

Network News est une masse d'informations subdivisée en catégories de sujets appelées **newsgroups**. Les articles qui forment le newsgroup sont écrits par les personnes intéressées par le sujet. Les articles sont envoyés au newsgroup ainsi d'autres personnes pourront lire, commenter ou répondre à ces articles.

La plupart des newsgroups se trouvent sur le réseau USENET. Network News utilise le Protocole NNTP (Network News Transfer Protocol) pour transférer des news sur Internet.

Les logiciels clients permettent de lire, de répondre, de créer de nouveaux articles.

13. Mailing Lists

Ce service permet de partager de l'information via E-mail avec beaucoup de personnes et ce concernant un sujet bien déterminé.

Il est possible de s'inscrire dans une mailing list se rapportant à un certain sujet.

Fonctionnement des mailing lists : les messages concernant un sujet bien précis sont envoyés vers une adresse e-mail (adresse de la mailing list) et ce message est alors redistribué vers tous les membres de la mailing list.

Les mailing lists peuvent être modérées, restreintes, totalement libres.

G. Intranet

On parlera d'Intranet quand on utilise les services d'Internet dans une entreprise sans pour autant être d'office connecté à Internet. En effet, la suite de protocoles TCP-IP est par essence prévue multiplateformes, environnements, matériels et constitue une solution standardisée.

H. Connexion du réseau local à Internet

1. Réseaux Privés (RFC 1918)

Les réseaux qui ont des adresses dans les plages 192.168.x.y,172.16.x.y -> 172.31.x.y, 10.x.y.z sont réservés à des utilisations privées. Les routeurs dans Internet ne doivent pas router les datagrammes destinés à ces réseaux.

2. Firewall

Le Firewall est un système par lequel tout le trafic Internet passe avec la possibilité de filtrage sur base des adresses IP, des protocoles et du contenu.

3. Proxi

Le proxi est un système intermédiaire auquel les clients envoient les requêtes; le système envoie les requêtes vers les différents serveurs s'il ne dispose pas déjà en "cache" de la réponse à ces requêtes. => réduction du trafic sur la liaison vers INTERNET. Du point de vue des serveurs, toutes les requêtes émanent du proxi qui lui-même redirige vers les clients.

Les clients doivent être configurés pour utiliser le proxi.

4. Routeur avec translation d'adresses (NAT)

Ce système intermédiaire (simple) par lequel passe tout le trafic Internet des machines qu'il dissimule vis-à-vis de l'extérieur (Internet). Du point de vue des machines extérieures, toutes les requêtes sont lancées par le routeur et ce dernier renvoie les réponses aux machines dissimulées.

Contrairement au proxi pour les stations du réseau local, il n'y a rien de particulier à configurer.

I. Utilitaires

winipcfg (win 9x) / ipconfig (WIN-NT) / ifconfig (UNIX) : utilitaire permettant d'obtenir la configuration des différentes interfaces réseau de la machine.

ping *adresse* : utilitaire permettant de vérifier l'accessibilité d'une machine distante *adresse* par l 'envoi d'un paquet ICMP demandant un ECHO.

tracert *adresse*: utilitaire permettant d'obtenir la liste des routeurs par lesquels un datagramme IP passe pour atteindre une machine distante *adresse*

netstat : utilitaire permettant d'obtenir l'état des connexions TCP et la liste des ports ouverts.

arp : utilitaire permettant d'afficher et de modifier la table de correspondance des adresses MAC et IP.

route : utilitaire permettant d'afficher et de modifier la table de routage.

VI. SYSTEME DE CABLAGE

Lors du câblage de nouveaux bâtiments, les règles de précâblage suivantes prévoient :

2 prises 230V, une prise téléphonique (RJ-45) ainsi que une prise réseau (RJ-45) par 8 m²,
un local technique à chaque niveau : la distance maximale entre ce dernier et les prises est de 90 mètres.

Dans le local technique, on trouvera un patch panel au départ duquel on pourra connecter chaque prise soit sur le central téléphonique ou sur le réseau. Dans ce local, on trouvera donc les concentrateurs et autres éléments de réseaux. 2 x 5 mètres de câble sont prévus pour relier le PC à la prise et le patch panel à l'élément réseau : on arrive ainsi a la longueur maximale de 100 mètres que l'on retrouve dans différentes normes réseaux.

Le câblage est effectué en paire torsadée catégorie 5 FTP.

Cette manière de faire permet d'utiliser indifféremment le câblage pour le réseau et pour le téléphone.

VII. ADMINISTRATION DE RESEAU ET ANALYSEURS

A. Les analyseurs

Il est possible de surveiller le fonctionnement du réseau :

par des analyseurs
par des sondes
en consultant les statistiques des cartes de communications qui disposent d'un agent SNMP

Ces systèmes permettent de donner :

le % d'utilisation
le nombre de trames par seconde
le nombre d'erreurs par seconde
le nombre de koctets/s en entrée et en sortie par station
le nombre total de koctets émis et reçus par station
les types de protocoles utilisés par station

Ces informations instantanées sont mémorisées pour donner les tendances pour chaque paramètre.

Dans un premier temps, il faut déterminer un fonctionnement normal du réseau (sur un mois par exemple). Ensuite, on peut définir des seuils d'alarme pour ces différents paramètres.

Voici, par exemple, les paramètres utilisés dans le cas d'un réseau Ethernet.

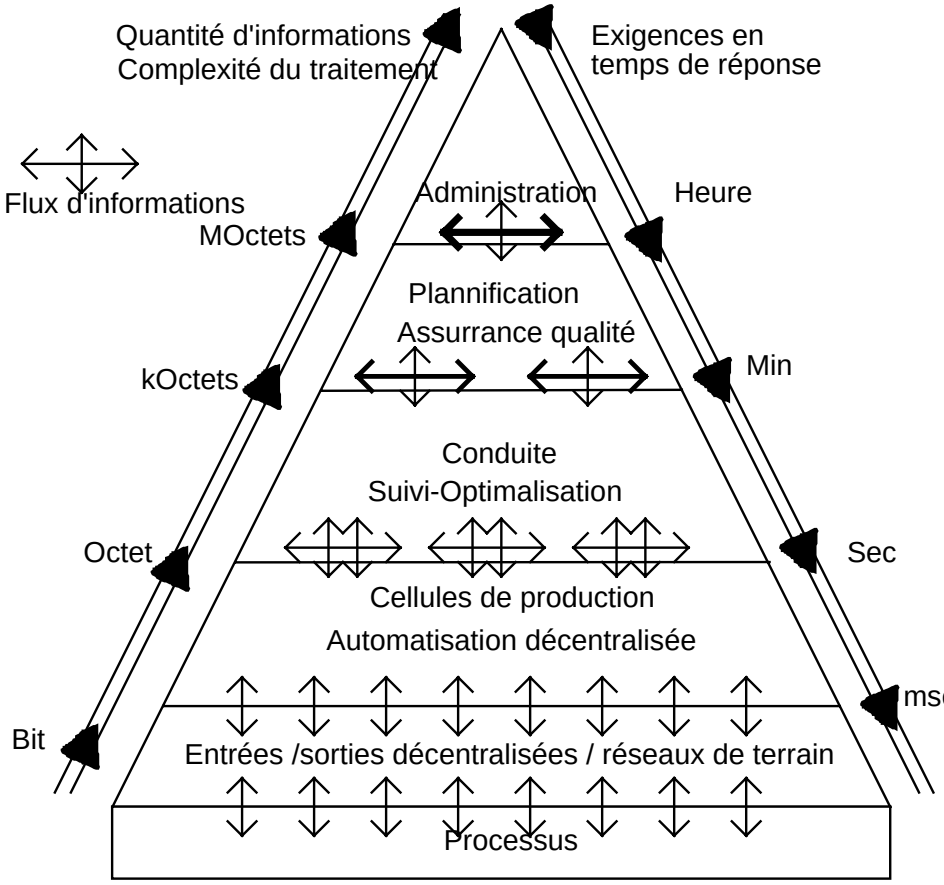
trame trop courte (=> driver bogué)
trame trop longue (=> driver bogué)
erreur de CRC (=> mauvais support)
erreur de fragmentation (=> collision)
erreur de "jabber" (=> transceiver défectueux)

B. Administration du réseau

Lorsque la taille et la complexité d'un réseau deviennent importantes, il est intéressant de pouvoir surveiller et configurer les équipements du réseau à distance. Les outils d'administration ont pour objet de détecter le plus rapidement possible les défauts qui apparaissent au sien des composants du réseau.

L'administration du réseau se base sur les éléments suivants :
la **MIB** (Management Information Base) : chaque matériel stocke localement des informations sur son état, sa configuration sous une forme standard. Cette MIB peut être implémentée dans les concentrateurs, ponts, routeurs, serveurs et postes de travail.
une **plate-forme d'administration** qui utilise des protocoles d'interrogation de la MIB des composants du réseau : **SNMP** (Simple Network Management Protocol) (standard) ou **CMIB** (Common Management Information Protocol) (la norme ISO peu répandue)
des **agents** SNMP ou CMIB dans chaque équipement du réseau qui permettent d'accéder à des champs de la MIB (variables) et de les envoyer vers le demandeur ou de signaler des alarmes (TRAP SNMP) vers la plate-forme d'administration.
des **sondes RMON** (Remote Monitoring): équipements qui permettent de surveiller des segments de réseau avec une MIB adaptée (taux d'utilisation, nombre d'erreurs, de collisions, …). Ceci est intéressant si le réseau global comporte des bridges et routeurs puisque, ainsi, il est possible de surveiller les réseaux locaux à distance.
des **proxy-Agent** : équipement permettant de collecter les données de différents équipements qui utilisent SNMP ou des codages et protocoles propriétaires (fonction de passerelle) et, de ce fait, permet de décharger la plate-forme d'administration et les liens qui la relient avec le réseau local distant. Elle peut aussi surveiller les différents paramètres du réseau et signaler des alarmes (TRAP SNMP) vers la plate-forme d'administration

VIII. LA PYRAMIDE C.I.M.



Niveau 6
Gestion administrative
Gestion du personnel
Finances, Comptabilité
Achats
Statistiques

Niveau 5
Plannification de la Productio
Gestion de Stocks
Contrôle Qualité
Entretien, Bureau Etudes
Méthodes

Niveau 4
Supervision d'atelier
Suivi et Optimalisation

Niveau 3
Controle/commande
Conduite décentralisée
Régulation

Niveau 2
Mise en forme des signaux
d'entrées/sorties

Niveau 1
Capteurs/actionneurs
Instruments

IX. MAP ET MMS

A. MAP et MMS : Qu'est-ce que c'est ?

1. Introduction

Pour pouvoir être performante et compétitive, une entreprise industrielle doit disposer de systèmes intégrés de production. En effet, ainsi, un produit pourra être suivi constamment depuis la demande d'offre du client jusqu'à la facturation. Le système intégré de production permet de connaître en temps réel l'état de l'entreprise (stock matières premières, pièces en cours de fabrication, ...) et intègre dans un seul ensemble les départements commercial, bureau d'études, planification, bureau des méthodes, la fabrication. Tous ces éléments permettent d'augmenter la réactivité d'une entreprise.

Les systèmes intégrés de production reposent sur :

l'utilisation de l'informatique dans l'ensemble de ces départements (GPAO, CAO, CFAO, CNC, API, Contrôleur de robot, ...),

l'utilisation de réseaux reliant les différents départements et permettant l'échange des informations entre ceux-ci.

En 1980, face à l'absence de systèmes de communication permettant cette intégration (les systèmes de communications représentaient 50% du coût de l'automatisation), General Motors mit sur pied un comité de travail (MAP Task Force) qui devait développer un protocole de communication permettant la communication entre matériels hétérogènes (G.M. utilisait des matériels de différents fournisseurs). Ce comité de travail qui rassemblait des représentants de ses divisions et de ses fournisseurs s'est intéressé aux communications dans l'environnement de production. **MAP** ou **M**anufacturing **A**utomation **P**rotocol et **MMS** ou **M**anufacturing **M**essage **S**pecification allaient naître.

Au même moment à peu près, la réalisation d'un protocole universel pour les environnements administratif et de développement fut pris en charge par Boeing qui lança aussi un comité de travail. Ce comité de travail allait développer **TOP** ou **T**echnical and **O**ffice **P**rotocol.

Les comités de travail de MAP et TOP ont choisi d'utiliser, aux niveaux des protocoles, les normes internationales ISO définies dans le cadre du modèle OSI (Open System Interconnection). C'est ainsi qu'ils resteront fortement liés et compatibles. L'utilisation de normes internationales et du modèle OSI assure l'ouverture des systèmes et permet à tout fabricant de créer son logiciel de communication MAP ou TOP.

Voyons ci-après les grandes étapes dans la naissance de MAP.

2. Les étapes importantes de MAP

1980

* General Motors forme le "MAP Task Force"
* Le modèle OSI de ISO est spécifié

1984

* MAP version 1.0
* Première démonstration à NCC de Las Végas

1985

* MAP version 2.0/2.1 / TOP version 1.0
* Démonstration MAP/TOP à AUTOFACT'85 à Détroit
* Formation du MAP/TOP Users Group nord américain
* Premières installations MAP

1986

* MAP verison 2.2
* Création du MAP Users Group européen (EMUG)
* Première démonstration européenne de MAP - CIMAP, GB
* Création de la fédération mondiale MAP/TOP Amérique du Nord, Europe, Japon, Australie

1988

* MAP/TOP version 3.0
* Démonstration EMUG MAP à SYSTEC 88 à Munich

1990

* Ensemble de base de produits MAP 3.0 disponible
* Premières installations MAP 3.0 fonctionnent
* Création du groupe d'intérêt MAP/TOP Europe de l'Est
* Démonstration EMUG MAP à SYSTEC 90 à Munich

1991

* MAP/TOP 3.0 - 1991 Supplement

1992

* Démonstration EMUG MAP à SYSTECH 92 à Munich

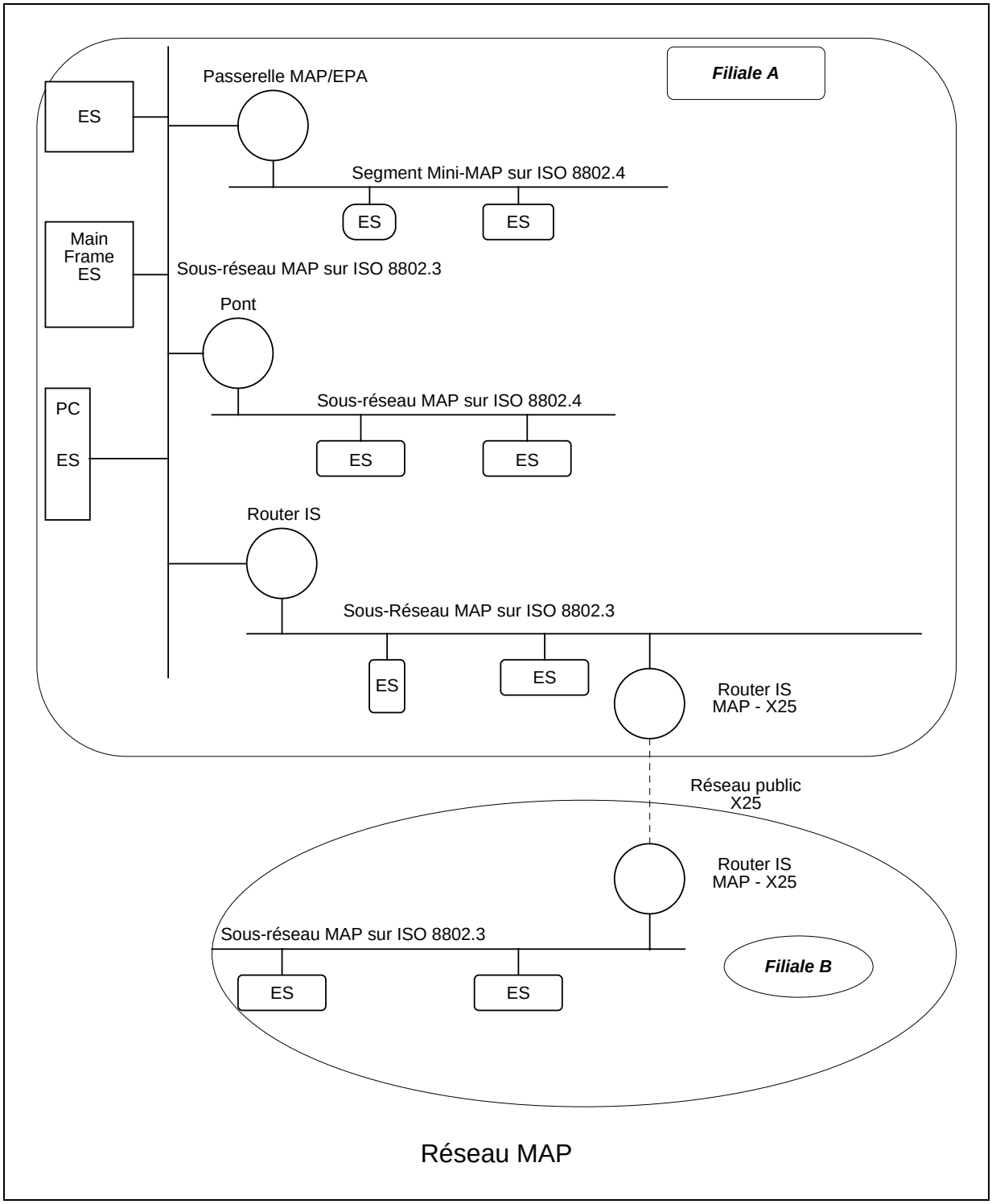
1993

* MAP/ TOP 3.0 - 1993 Supplement (MAP sur Ethernet et Mini-MAP amélorié)

3. Les avantages de MAP 3.0

Utilisateurs	Fabricants
• Choix de l'équipement approprié • Disparition des coûts élevés de développement de hardware et software de communication • 1 seule formation (MAP/MMS) • Coûts d'intégration et de support moindres • Flexibilité pour les extensions et les changements	• Disparition des duplications des coûts de développement de hardware et software de communication • Possibilité pour le vendeur de se concentrer sur l'amélioration du produit • Marketing moins compliqué • Accès à des systèmes et réseaux multivendeurs • Accès plus aisé au marché mondial

B. Les composants d'un réseau MAP typique



Réseau MAP

Le réseau MAP est composé d'un certain nombre de sous-réseaux qui peuvent comporter un certain nombre de segments. Un réseau MAP peut

s'étendre sur le monde entier. Il est clair qu'un réseau MAP peut se limiter à un seul sous-réseau comportant un seul segment.

De plus, comme on peut le voir sur la figure, des sous-réseaux peuvent être reliés par des réseaux non-OSI (réseau X.25) pour constituer un réseau très étendu.

Les segments peuvent être de deux types :

♦ **MINIMAP**,
♦ **FULL MAP**.

L'architecture de **MINIMAP** ne prévoit pas l'ensemble des 7 couches de la pile de protocoles pour des **raisons de temps de réponse**. Ces noeuds sont plus simples. En effet, dans MINIMAP, les couches 3 à 6 ne sont pas prévues. Ceci entraine une forte réduction des fonctionnalités et rend l'intégration de systèmes hétérogènes impossible. Aussi, cette achitecture est fortement controversée actuellement puisque MAP a pour but de permettre cette intégration.

La connexion d'un segment MINIMAP au réseau MAP se fait au travers d'une passerelle qui est un noeud **MAP/EPA** . Celui-ci transforme les fonctionnalités MAP en fonctionnalités MINIMAP. Ce noeud dispose donc des deux piles de protocoles (MINIMAP et FULLMAP).

L'architecture **FULLMAP** comporte l'ensemble des 7 couches de la pile de protocoles MAP et permet la création d'un réseau MAP étendu.

Un réseau MAP peut comporter 4 types de noeuds différents :

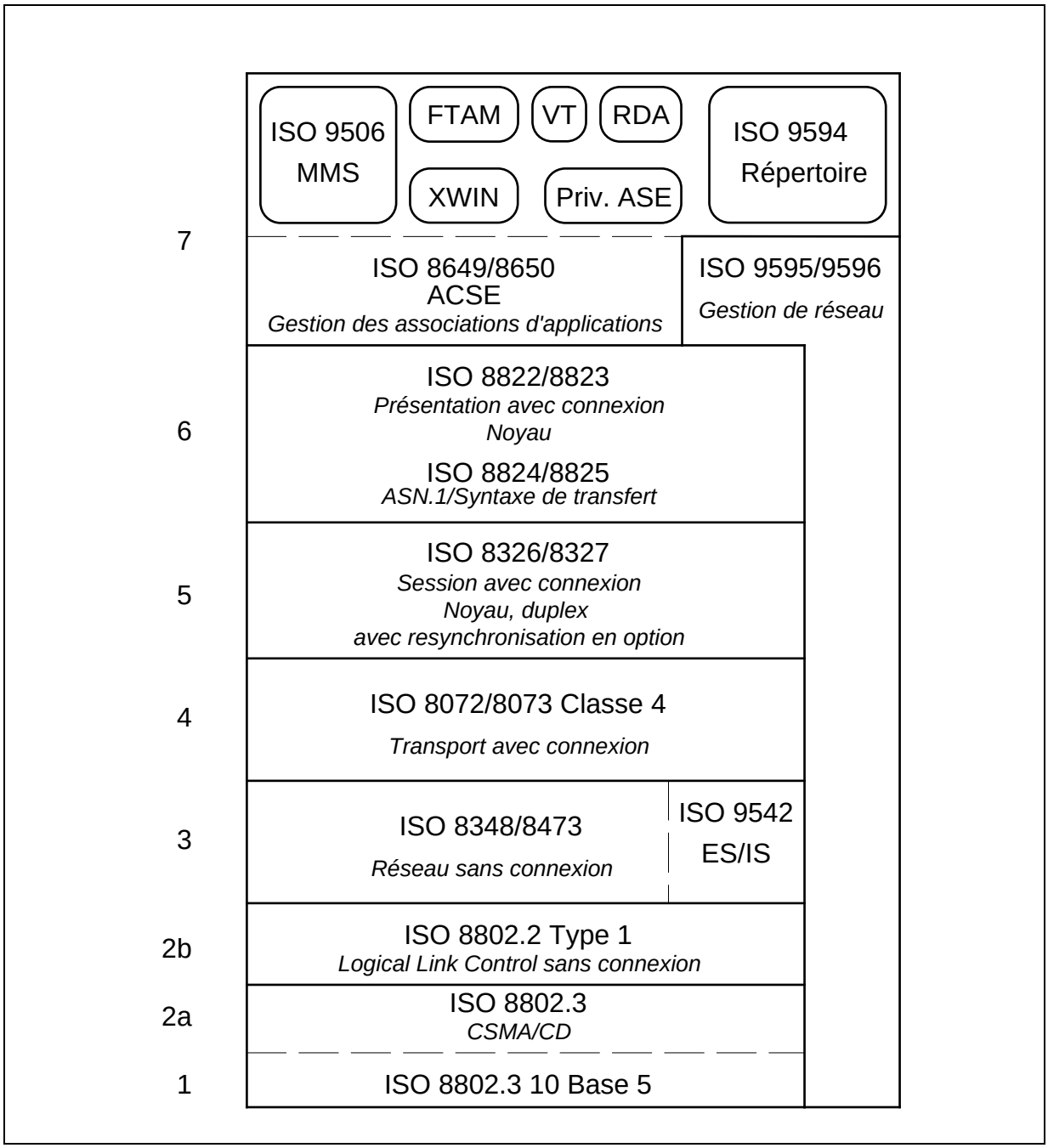
♦ **Routers** ou Intermediate System (IS) sont chargés de relayer les paquets réseau entre les différents sous-réseaux composant le réseau MAP global.
♦ Les **noeuds** ou End System (ES) de type **MINIMAP**
♦ Les **noeuds** ou End System (ES) de type **FULLMAP**
♦ Les **noeuds MAP-EPA** qui font la conversion des services MAP en services MINIMAP et permettent ainsi la communication entre noeuds MINIMAP et FULL MAP.

Il est clair que, dans les sous-réseaux, des **Répéteurs** ou **Amplificateurs** et des **Ponts** peuvent être présents.

Dans la suite, nous décrirons principalement l'architecture FULL MAP que nous appellerons MAP.

C. La Pile de Protocole MAP

1. La pile de protocole MAP 3.0 sur "Ethernet"



2. La couche physique et MAC

La pile de protocoles MAP 3.0 permet différents supports physiques et méthodes d'accès au support.

Initialement, MAP prévoyait un support en câble coaxial 75 ohms et un accès de type bus à jeton (ISO 8802.4 Carrier Band ou Broad Band).

Sous l'impulsion de EMUG (European MAP User Group), MAP 3.0 dans sa dernière version supporte aussi le câblage "Ethernet" (ISO 8802.3 10Base5, 10Base2 ou FOIRL). En effet, en Europe, les réseaux "Ethernet" sont bien implantés dans le monde industriel.

On remarquera que l'emploi de l'un ou l'autre support physique n'entraine aucune différence pour les couches supérieures.

3. La couche LLC ou Logical Link Control

La couche LLC est composée de la norme ISO 8802.2 de classe 1 ou de classe 3.

Dans les noeuds MAP, on utilise un échange de trames en mode non connecté (ISO 8802.2 type 1) : c'est la couche Transport qui se charge de veiller à l'acquittement des informations transmises.

On peut aussi utiliser l'échange de trames en mode non connecté avec un acquittement simple (ISO 8802.2 type 3). Ce type d'échanges est utilisé dans le cas des noeuds MINIMAP.

Comme le support physique est un réseau local (support relativement fiable) et que l'on désire un transfert rapide des messages, le protocole LLC sans connexion (avec des échanges de type datagramme) est utilisé.

Unité de donnée de protocole LLC ou LPDU

DLSAP	SLSAP	Champ de Contrôle	Données Utilisateur (couche réseau)

Cette LPDU est composée de quatre parties :

♦ le **DLSAP** (1 octet) indique l'adresse du LSAP de destination,

♦ le **SLSAP** (1 octet) indique l'adresse du LSAP émetteur,

♦ le **champ de contrôle** (1 ou 2 octets) indique le type de LPDU

♦ le **champ données utilisateur** comporte les informations provenant de ou destinée à la couche supérieure (Service Data Unit).

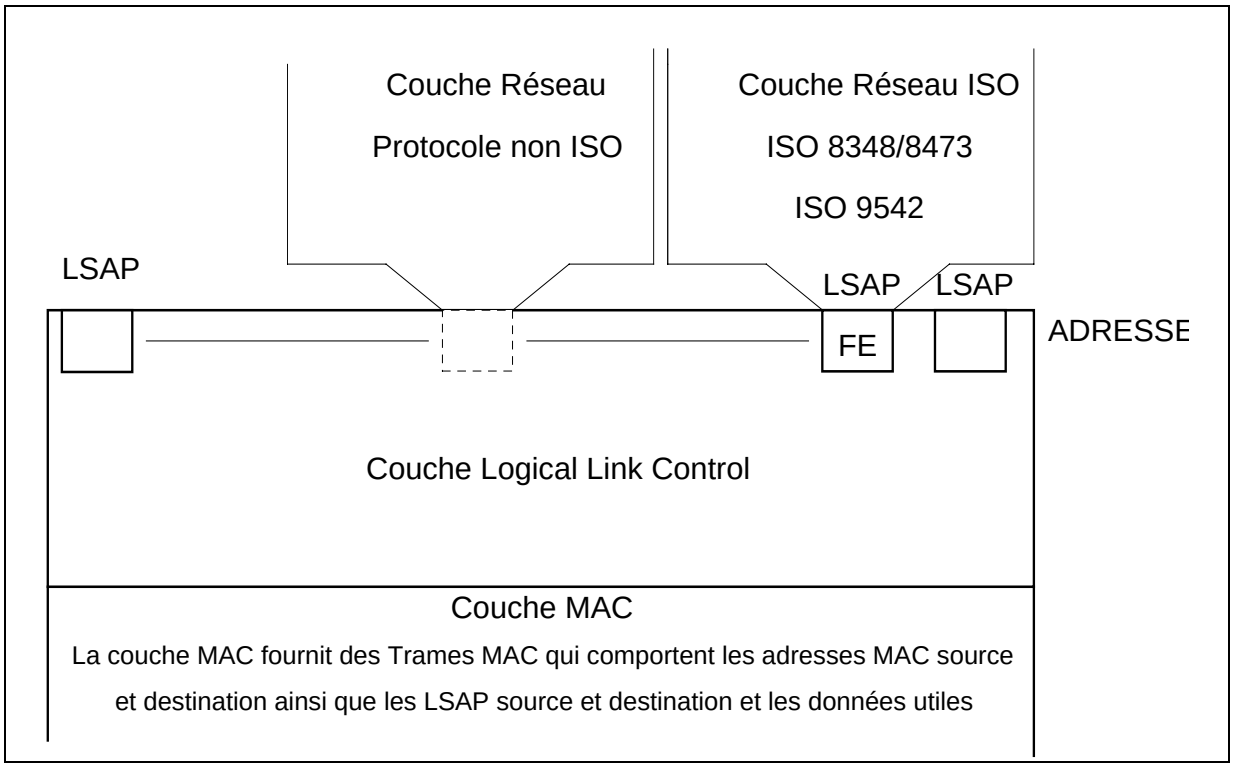
La couche réseau accède aux services offerts par la couche LLC par l'intermédiaire d'un **LSAP** (**L**ogical link control **S**ervice **A**cces **P**oint). A chaque point d'accès correspond une adresse bien précise ayant une valeur comprise entre 0 et 255 (00h à FFh).

Dans le cas de MAP, cette adresse est **FE**h qui correspond bien sûr au protocole réseau ISO.

D'autres protocoles qui utilisent les services de la couche LLC y accèdent par une autre adresse LSAP.

Par exemple, l'adresse **F0**h est utilisée par le protocole Netbios.

On remarque donc que l'adresse LSAP est le sélecteur du protocole de la couche supérieure avec lequel on désire communiquer.



4. La couche Réseau

a) Généralités

La couche Réseau comporte deux protocoles :

- ISO 8473 sans connexion CLNP (Connection Less Network Protocol)
- ISO 9542 ES/IS

b) Le protocole ISO 8473

Le protocole ISO 8473 CLNP est en fait un protocole de la sous-couche SNICP. Dans le cas de MAP, les sous-couches SNACP et SNDCP ne sont pas nécessaires car la couche SNICP ISO 8473 CNLP utilise des services très proches de ceux fournis par la couche LLC ISO 8802.2 type 1. Comme indiqué précédemment, il utilise le LSAP d'adresse FEh.

Ce protocole assure :

- la composition/décomposition du paquet,
- l'analyse du format d'entête,
- le contrôle de la durée de vie du paquet (permet d'éviter qu'un paquet ne boucle dans le réseau),
- le routage du paquet,
- l'expédition du paquet,

♦ la segmentation/réassemblage du paquet,
♦ la purge du paquet,
♦ la détection des erreurs dans l'en-tête.

MAP utilise la variante Full Protocol de ISO 8473 CNLP qui précise que la segmentation est permise.

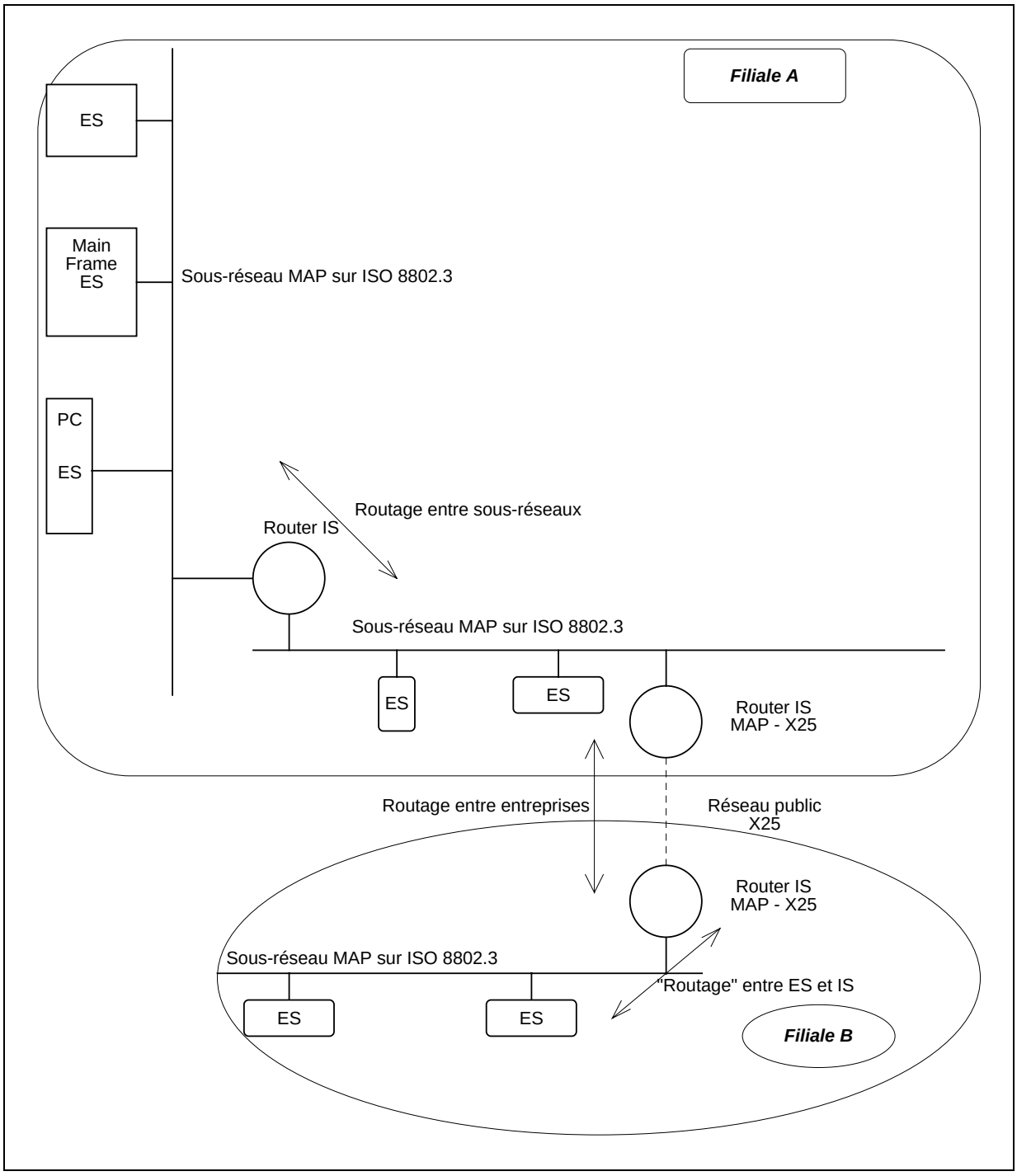
c) Le protocole ISO 9542

Le protocole ISO 9542 ES/IS est utilisé pour mettre à jour dynamiquement les informations de routage. Il fournit des informations sur la configuration du réseau et sur les différents chemins de routage possibles pour accéder à des correspondants. Ces opérations sont totalement automatiques.

Le protocole ISO 9542 ES/IS utilise le même LSAP que le protocole réseau ISO 8473 CLNP. Les paquets ont une entête différente de celle de ISO 8473 CLNP.

d) Routage et Adressage des noeuds pour MAP

L'adressage des noeuds est organisé selon 3 niveaux :
♦ routage entre entreprises
♦ routage entre sous-réseaux
♦ routage ES/IS entre systèmes d'extrémité et systèmes intermédiaires adjacents.



MAP et TCP-IP

Actuellement, MAP utilise aussi la pile de protocoles TCP-IP (jusqu'en couche 4). Ceci montre bien l'intérêt de la structuration en couches. En effet, les couches 5-6-7 (orientées information) n'ont pas été modifiées . Les normes ISO des couches 1-2-3-4 ont été remplacée par TCP-IP.

5. La Couche Transport ou ISO 8073 Classe 4

En couche 4, la norme ISO 8073 Classe 4 est mise en oeuvre. Elle prévoit des liaisons en mode connecté. C'est cette couche qui veille à ce que l'échange de données se fasse sans erreur (répétition de l'envoi en cas d'erreur, ...).

Ainsi, elle prévoit entre autres :

♦ la reprise d'erreur et la détection d'erreur pour détecter la perte, la corruption, la duplication, le mélange des TPDUs,

♦ la possibilité de multiplexage/démultiplexage pour partager une connexion réseau unique entre deux ou plusieurs connnexions transport,

♦ la possibilité de segmentation et réassemblage (un TSDU (Transport Service Data Unit) est transmis en utilisant plusieurs TPDU),

♦ la possibilité d'envoyer des données urgentes,

♦ le contrôle du flux des TPDUs, ...

L'accès aux services de la couche 4 se fait par l'intermédiaire des **TSAP** (**T**ransport **S**ervice **A**ccess **P**oint).

Lors de l'établissement d'une connexion Transport, les utilisateurs des services de la couche Transport peuvent échanger des Transport Selector (TSEL). Le TSEL est un mot de passe par l'intermédiaire duquel les utilisateurs des services Transport peuvent vérifier que le partenaire appelant peut établir une connexion.

Si la demande de connexion reçoit une réponse positive, le partenaire fournit une adresse de TSAP au travers de laquelle ils vont communiquer.

6. La Couche Session ou ISO 8326/8327

Au niveau de la couche Session, MAP ne prévoit que le noyau et le transfert de données en mode duplex.

L'accès aux services de la couche Session se fait au travers des **SSAP** (**S**ession **S**ervice **A**ccess **P**oint).

Le **S**-Selector est un paramètre (mot de passe) échangé lors de l'établissement de la connexion. Il permet aux utilisateurs des services de la couche Session de vérifier si le partenaire éloigné a la droit d'établir la connexion (sécurité d'accès).

Une connexion Session correspond à une et une seule connexion Transport.

7. La Couche Présentation ou ISO 8822/8823

La couche Présentation doit comporter les fonctions du noyau (établissement/libération de connexion et transfert de données). Elle négocie aussi les contextes d'application supportés (les protocoles d'application supportés).

L'accès aux services de la couche Session se fait au travers des **PSAP** (**P**resentation **S**ervice **A**ccess **P**oint). Le **P**-Selector est un paramètre (mot de passe) échangé lors de l'établissement de la connexion. Il permet aux utilisateurs des services de la couche Présentation de vérifier si le partenaire distant a le droit d'établir la connexion (sécurité d'accès).

Une connexion Présentation correspond à une et une seule connexion Session.

Dans le cas de MAP, la syntaxe de transfert ISO 8824/8825 (X.209) doit toujours être supportée mais il est permis d'en utiliser d'autres.

8. La Couche Application

La couche application comporte différents éléments ou ASE (Application Service Element).

MAP prévoit **obligatoirement** la présence de :

♦ ISO ACSE (**A**ssociation **C**ontrol **S**ervice **E**lement) ou service de contrôle d'associations d'application (connexion entre applications),

♦ ISO MMS (**M**anufacturing **M**essage **S**pecification) ou messagerie industrielle.

La présence des éléments repris ci-dessous sont **fortement recommandés** par MAP :

♦ OSI NM (**N**etwork **M**anagement) ou gestion de réseau,

♦ OSI DS (**D**irectory **S**ervices) ou service de répertoire (destiné à fournir les adresses des différentes applications partenaires sur le réseau),

Les élements dans la liste qui suit, sont optionnels :

♦ OSI FTAM (**F**ile **T**ransfer and **A**ccess **M**anagement) ou Transfert de fichiers,

♦ OSI VT (**V**irtual **T**erminal) ou Terminal Virtuel,

♦ OSI XWIN (**X**-**W**indows) ou Interface X-Windows,

♦ OSI RDA (**R**emote **D**atabase **A**ccess) ou accès à des bases de données distantes,

♦ **P**rivate **A**pplication **S**ervice **E**lement ou application privée.

Nous ne décrirons dans ces notes que ACSE et MMS.

9. ACSE ou Association Control Service Element (ISO 8649/8650)

Cette partie de la couche Application du modèle OSI est destinée à gérer les associations d'applications ou connexion en couche 7.

Les autres protocoles de la couche Application utilise ACSE pour gérer les associations d'application. C'est grâce à ACSE que les processus d'application ont accès aux ressources réseau OSI.

ACSE gère l'établissement des associations d'application, la terminaison (normale ou rupture) des associations d'application ainsi que la négociation et la définition des contextes d'application (MMS, FTAM, ...).

D. MMS ou Manufacturing Message Specification

1. Généralités

MMS (Manufacturing Message Specification) est destiné à permettre de gérer l'ensemble des équipements automatisés de production et cela par l'intermédiaire d'une série de services normalisés. Ces services généraux permettent de rencontrer les fonctionnalités les plus couramment utilisées, nécessaires pour commander les équipements dans le milieu industriel.

On remarque que toutes les couches inférieures à MMS (couche physique jusqu'à ACSE) assurent l'interconnexion des équipements hétérogènes tandis que MMS fournit l'interopérabilité entre systèmes hétérogènes.

MMS définit différents éléments :

le VMD ou **V**irtual **M**anufacturing **D**evice ou Equipement Virtuel de Production : MMS fournit une représentation normalisée des composants d'automatisation du point de vue du réseau,

les objets qui composent le VMD : ces objets permettent d'accéder aux ressources des machines réelles sous-jacentes au VMD.

un ensemble de services qui permettent de gérer le VMD et les objets associés.

le modèle Client-Serveur dans lequel le Client envoie une requête au Serveur pour lui demander l'exécution d'un service. Le Serveur exécute le service demandé et peut renvoyer une Réponse au Client.

2. MMS norme internationale

MMS est une norme internationale. Cette norme comporte plusieurs parties.

Les services et le protocole de base MMS (MMS Core Services and Procotol), ISO 9506/1 & 2, sont normalisés en tant que norme internationale (**IS**). MMS Core fournit 86 services différents qui permettent de rencontrer la majorité des fonctionnalités courantes utiles dans la gestion des systèmes de production programmables.

Il existe des normes d'accompagnements (Companion Standard) qui définissent des services et objets plus appropriés à certaines applications et donc types de machines (robots, commande numérique, automate programmable, processus continu, ...) : ISO 9506/3->6. Ces normes ne sont pas encore toutes des

normes matures (stade IS). Il faut bien remarquer que les services de base sont toujours suffisants pour faire toutes les opérations.

De plus, MAP a prévu des classes de mise en oeuvre (MAP 0 -> MAP 7) qui définissent des sous-ensembles des 86 Services MMS disponibles. En effet, la mise en oeuvre de l'ensemble des services n'est pas souvent intéressantes.

Dans le cadre de cette étude, nous étudierons les services MMS de base (ISO 9506/1) utilisables dans un environnement d'automates programmables : l'ensemble des 86 services MMS ne sera donc pas étudié complètement.

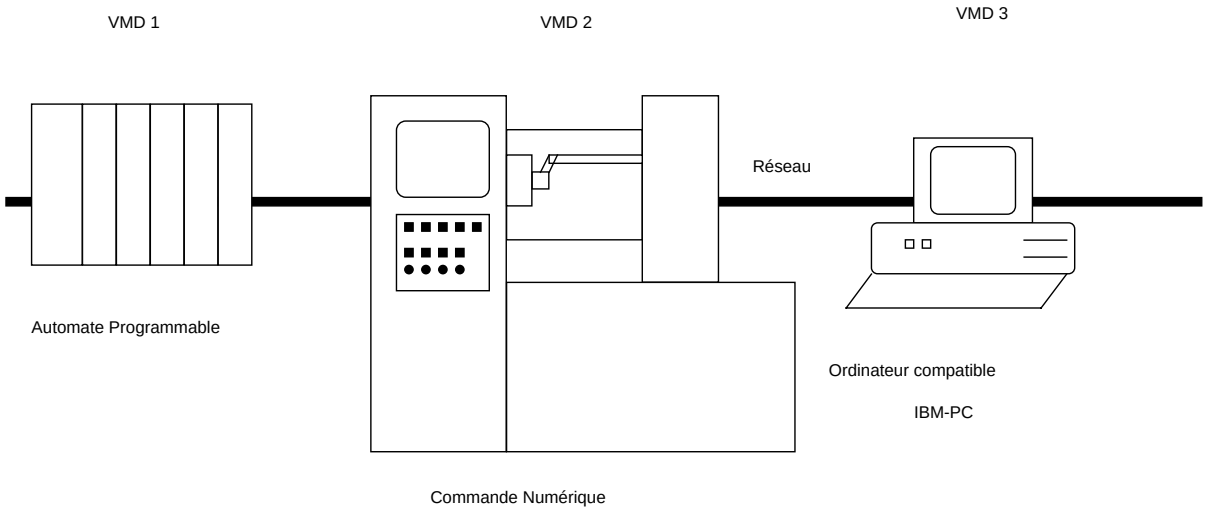
3. Le V.M.D. ou Virtual Manufacturing Device

Pour permettre la communication entre systèmes hétérogènes, MMS de MAP 3.0 rend, transparentes, les différences qui existent entre les différents matériels qui utilisent MAP 3.0. Vu du réseau, un automate programmable aura le même aspect qu'une commande numérique, qu'un ordinateur compatible IBM PC ou que tout autre système qui utilise MMS et MAP 3.0.

Pour ce faire, MMS fournit une représentation normalisée des composants d'automatisation du point de vue du réseau.

Cette représentation normalisée est le V.M.D. (**V**irtual **M**anufacturing **D**evice) ou équipement de production virtuel.

Au départ du réseau, l'accès à un VMD est le même quel que soit le matériel associé à ce VMD.

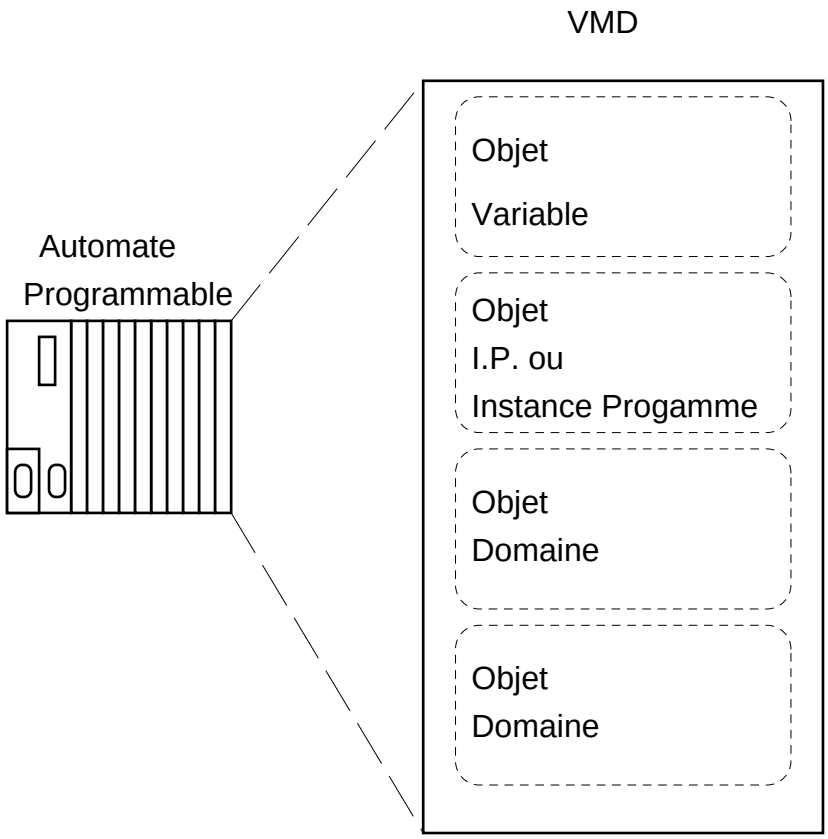


Les VMD rendent, transparentes, les différences entre matériels hétérogènes en fournissant une représentation normalisée du matériel au point de vue du réseau.

Le VMD modélise le comportement extérieur visible du Serveur (voir modèle Client-Serveur) et rend disponible les ressources et fonctionnalités de la machine réelle.

MMS veille donc à transformer les informations destinées au VMD en informations compréhensibles par le système réel.

Le VMD renferme des "objets" qui sont des Variables, des Domaines, des Instances de Programmes (Program Invocation), ... exploitables par les services MMS.



Le VMD comporte différents objets que les services MMS peuvent manipuler.

4. Le modèle Client-Serveur

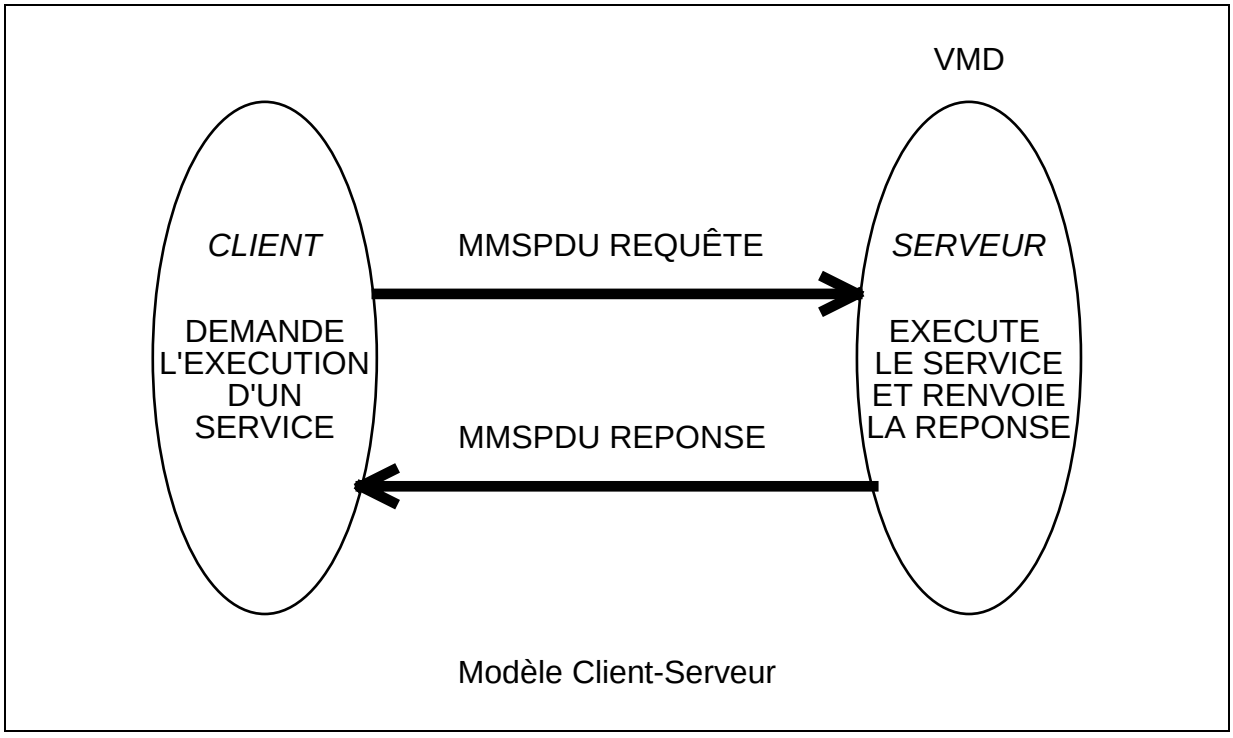
Dans les protocoles orientés informations, le principe de la relation Client-Serveur est utilisé.

Le Client envoie une Requête vers le Serveur pour demander l'exécution d'un service.

Le Serveur reçoit l'information du Client, exécute le service demandé et renvoie une Réponse au Client.

L'échange des informations entre Client et Serveur se fait via l'envoi d'APDUs (Application Protocol Data Units) puisque nous sommes en présence d'une association d'Applications. En fait, il s'agit de MMSPDU. Ce sont des MMSPDU

puisque le Client et le Serveur emploient les services MMS qui sont des services de la couche Application.

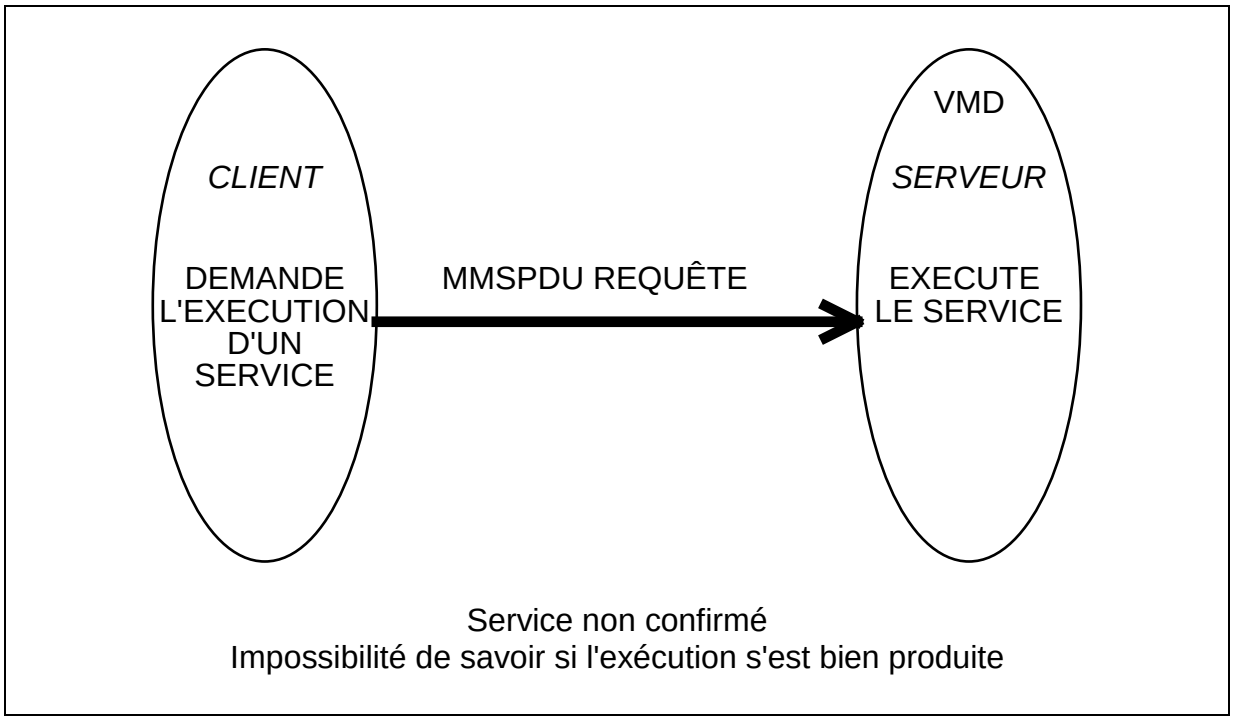


Modèle Client-Serveur

Les services qu'un Client demande à un serveur peuvent être **confirmés ou non**.

Dans la figure ci-dessus, nous sommes en présence d'un service confirmé. L'intérêt du service confirmé est que l'on sait si la requête demandée a été exécutée correctement ou non.

Un service non confirmé ne fournit pas la possibilité d'obtenir un acquittement après l'exécution du service. Dans le cas de MMS, le nombre de services non confirmés est faible.



Service non confirmé
Impossibilité de savoir si l'exécution s'est bien produite

On remarquera que le Client ne doit pas nécessairement avoir un VMD qui lui est associé.

Par exemple dans le cas d'un PC de supervision, il n'est pas nécessaire de prévoir un VMD associé au PC puisque le PC ne joue pas le rôle de Serveur vis-à-vis des autres matériels.

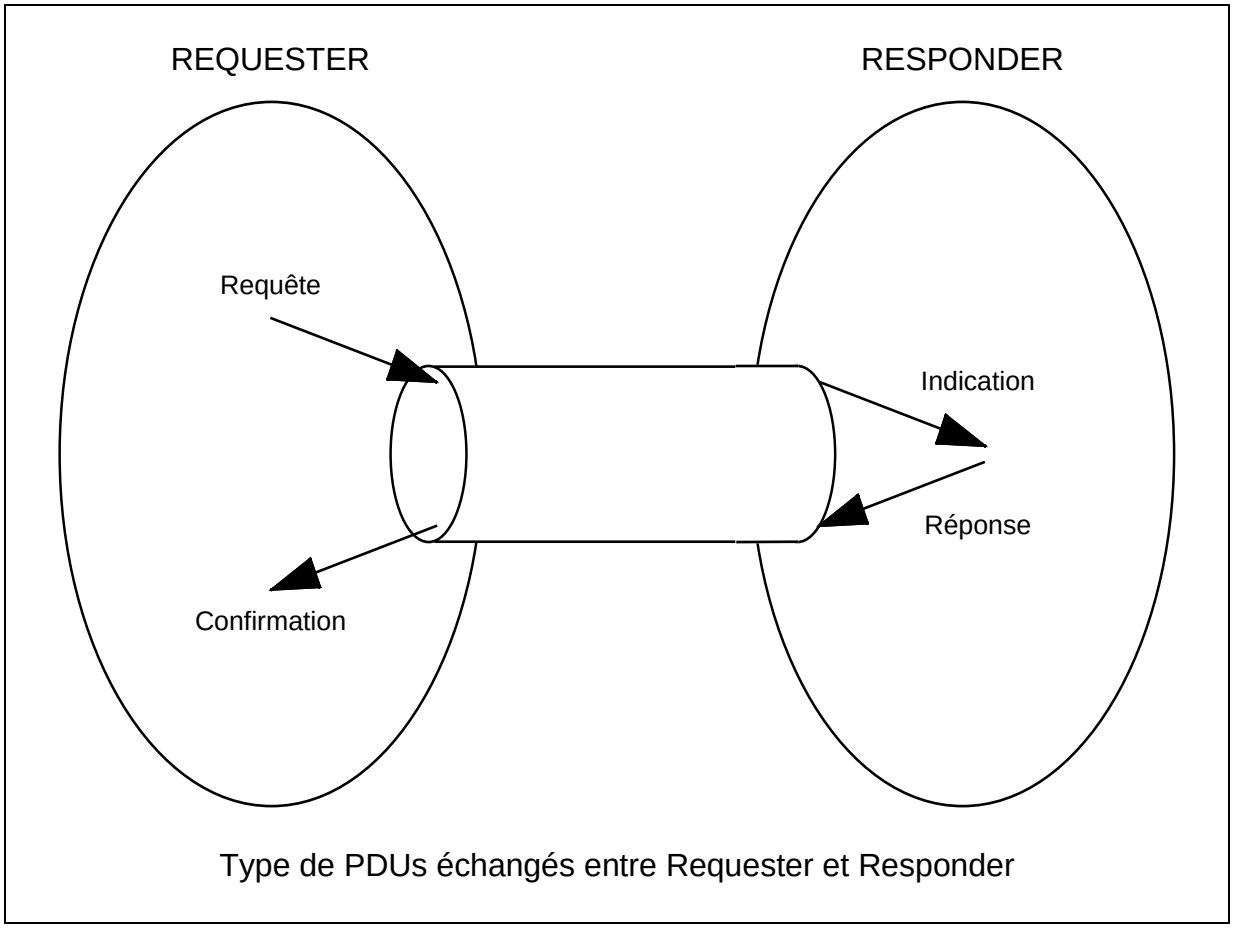
5. Le Requester et le Responder

Comme nous l'avons remarqué dans le modèle Client-Serveur, les deux partenaires n'ont pas le même rôle dans la communication. L'un envoie des requêtes et l'autre répond aux requêtes.

L'émetteur d'une requête est le **Requester.** Celui qui renvoie la réponse à la requête est le **Responder.** Habituellement, c'est le Client qui envoie les requêtes et le Serveur, les réponses.

Lorsqu'un Client utilise les services de Domaine (voir services de domaines), il peut aussi recevoir des requêtes du Serveur. Le Client est alors **Responder**.

Il faut bien retenir que Client n'est pas synonyme de Requester et Serveur n'est pas équivalent à Responder uniquement.



Type de PDUs échangés entre Requester et Responder

Lorsqu'un Requester envoie une demande vers un Responder, le message est appelé **Requête** (Request). Le Requester envoie donc des Requêtes.

Lorsque cette requête arrive chez le Responder, ce message est appelé **Indication** (Indication). Le Responder reçoit des Indications.

Lorsque le Responder renvoie un message en réponse à l'Indication, ce message est appelé **Réponse** (Response). Le Responder renvoie donc des Réponses.

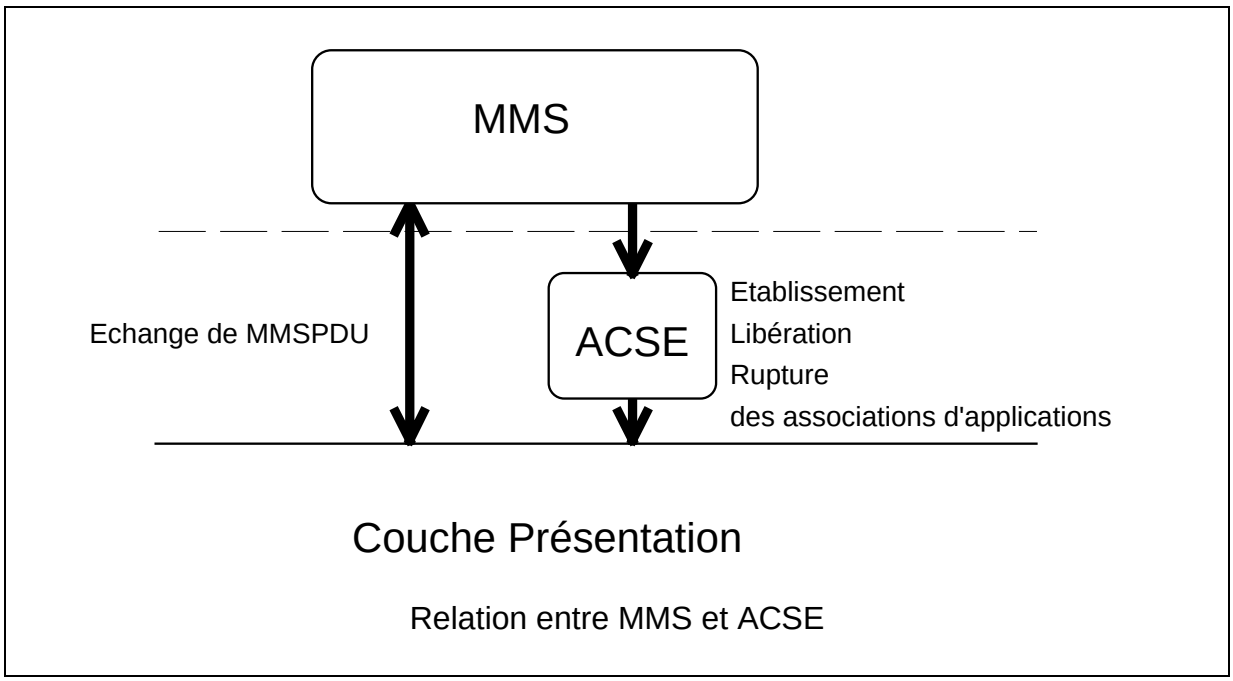
Lorsque la Réponse arrive chez le Requester, ce message est appelé **Confirmation** (Confirmation). Il est à noter qu'une confirmation peut aussi bien être négative que positive. Le Requester reçoit donc des Confirmations.

Dans le cas d'un service non confirmé, l'Indication arrivant chez un Responder ne demande pas une Réponse.

On observera qu'un système peut en même temps être *Requester* et *Responder*.

6. Accès aux couches inférieures

MMS utilise ACSE pour établir, libérer ou interrompre les associations d'application et accède ensuite directement à l'interface avec la couche Présentation pour échanger ses MMSPDUS.



Relation entre MMS et ACSE

E. Les objets MMS

1. Généralités

MMS prévoit 12 classes d'objets nommés :

♦ *Le domaine (Domain)* : cet objet contient les données utilisateurs représentant un

programme ou une partie de programmes ou les données d'un programme.

♦ *L'instance de programme (Program Invocation)* : cet objet permet de commander l'exécution d'un programme créé au départ d'un ou de plusieurs domaines.

♦ *La station opérateur (Operator Station)* : cet objet permet de faire apparaître ou demander des informations à un opérateur.

♦ *La variable nommée (Named Variable)* : cet objet correspond à une donnée à laquelle on accède par l'intermédiaire de son nom.

♦ *L'accès dispersé (Scattered Access)* : cet objet permet de rassembler dans un ensemble nommé plusieurs variables nommées ou anonymes ou bien d'autres variables à accès dispersé. Il offre donc une possibilité de définition récursive.

♦ *La liste nommée de variables (Named Variable List)* : cet objet correspond à une liste de variables (nommées ou anonymes ou à accès dispersé). Elle ne permet pas d'avoir une définition qui comporte d'autres listes nommées de variables. L'accès à cette liste se fait par l'intermédiaire de son nom.

♦ *Le type nommé (Named Type)* : cet objet définit un type de variable.

♦ *Le sémaphore (Semaphore)* : cet objet comporte un certain nombre de jetons qui représentent la possibilité d'utiliser une ressource partagée ou bien c'est un objet qui permet de synchroniser des applications. Il permet donc de contrôler et de coordonner l'utilisation de ressources uniques partagées d'un serveur MMS.

♦ *La condition événementielle (Event Condition)* : cet objet contient la condition définissant un événement.

♦ *L'action événementielle (Event Action)* : cet objet définit l'action entreprise (service MMS confirmé qui doit être exécuté) lorsque la condition événementielle associée est validée. Cet action est exécutée avant la notification de l'événement à un Client MMS.

♦ *L'enregistrement d'événement (Event Enrollment)* : cet objet contient les informations indiquant le Client qui doit être informé de l'occurrence de l'événement ou le service MMS devant être exécuté.

♦ *Le journal (Journal)* : cet objet permet d'enregistrer les informations significatives pour le système c'est à dire les événements, les noms ainsi que les valeurs de variables ou des commentaires.

Un objet est déterminé par son **domaine de validité** ou **portée** et par son identificateur ou nom.

La portée d'un objet définit le domaine de visibilité de l'objet dans le VMD. Un objet peut être spécifique :

♦ au VMD (l'objet est visible dans tout le VMD),

♦ à un domaine (l'objet est rattaché à un domaine),
♦ à une association d'applications (l'objet n'est visible qu'au départ de l'association d'applications à laquelle il est rattaché).

L'identificateur doit être unique dans son domaine de validité.

Ces différents objets peuvent être **statiques** ou **dynamiques** c'est à dire qu'ils sont prédéfinis ou qu'ils peuvent être le résultat d'un service MMS.

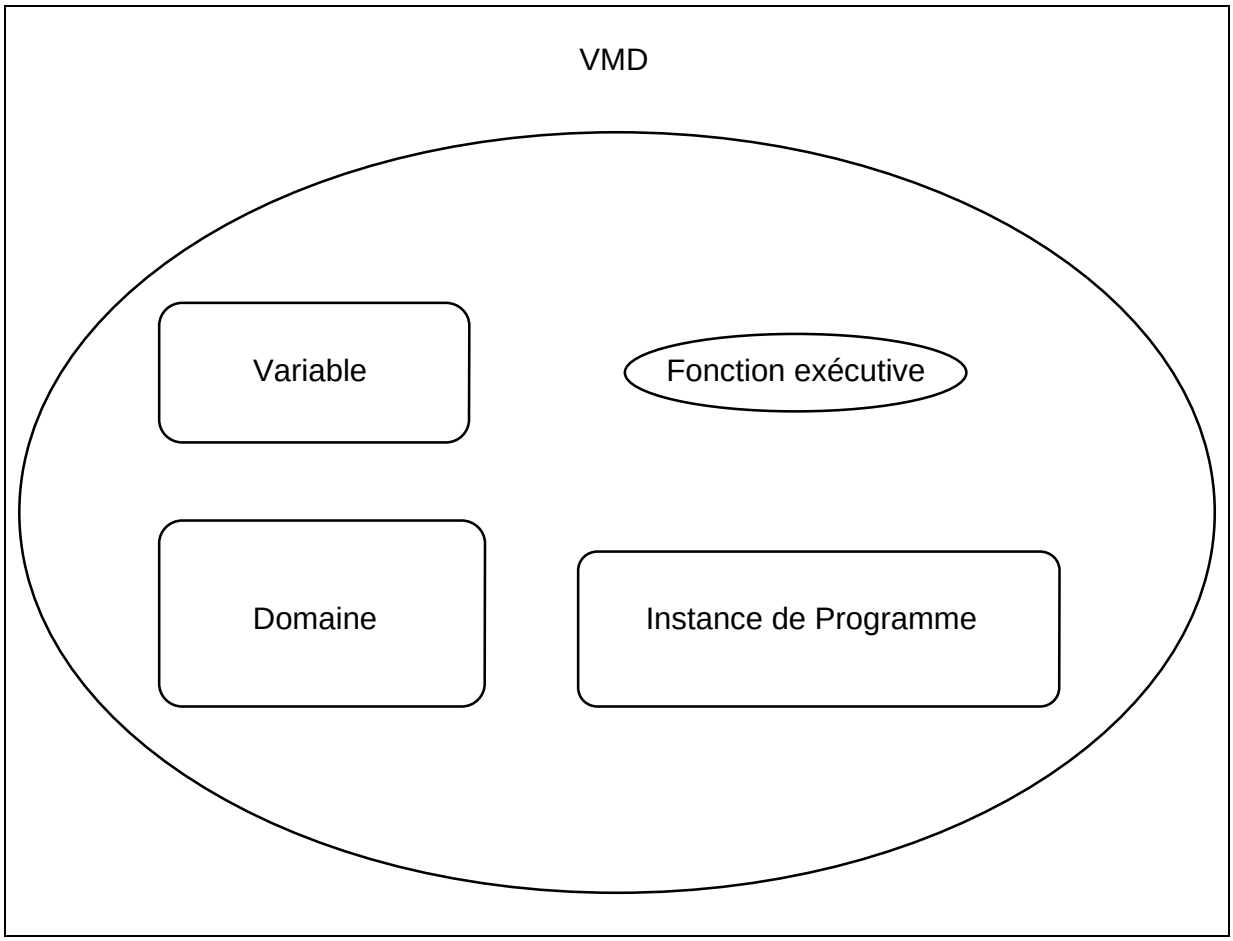
Ils disposent d'un attribut qui indique si l'objet est **effaçable** ou non par l'intermédiaire des services MMS.

Chaque classe d'objets comporte un ensemble d'attributs.

Dans la cadre de cette étude, nous n'envisagerons que les objets disponibles actuellement dans les VMD des automates (Variables nommées, l'accès dispersé, liste nommée de variables, Domaine, Instance de Programme, l'automate de sauvegarde).

2. Le VMD

Le VMD (**V**irtual **M**anufacturing **D**evice) est un équipement virtuel de production. C'est cet équipement (Serveur MMS) que les services MMS vont gérer, commander. Vu du réseau, tous les équipements virtuels ont le même aspect et sont donc gérés de la même manière par les services MMS. Sous-jacente au VMD se trouve la machine réelle.



Dans le VMD, on retrouve les différents objets énumérés précédemment et la fonction exécutive (Executive Function).

C'est la fonction exécutive qui gère les ressources (Capabilities) du VMD et qui veille à faire correspondre aux requêtes MMS des actions au niveau de la machine réelle.

Par exemple, lors de la lecture d'une variable par un Client, le Serveur doit aller chercher la valeur de la variable dans la machine réelle puis renvoyer celle-ci au Client qui la demandait.

Les attributs de l'objet VMD sont :

- ♦ fonction exécutive (Executive Function)
- ♦ nom du constructeur (Vendor Name)
- ♦ nom du modèle (Model Name)

♦ numéro de version (Revision)
♦ liste des syntaxes abstraites supportées (List Of Abstract Syntaxes Supported) : MMS core doit toujours être présent.
♦ état logique (Logical Status)
♦ liste de ressources (List Of Capabilities)
♦ état physique (Physical Status)
♦ liste des instances de programmes (List Of Program Invocations)
♦ liste des domaines (List Of Domains)
♦ liste des objets transactions (List Of Transaction Objects)
♦ liste des automates de sauvegarde(ULSM) (List Of Upload State Machines)
♦ liste d'autres objets spécifiques au VMD (List Of Other VMD-Specific Objects)
♦ détails supplémentaires (Additional Detail).

L'**Etat logique** et l'**Etat physique** du VMD sont deux attributs du VMD. L'Etat logique du VMD détermine les services MMS auxquels le VMD est à même de répondre. L'état physique du VMD indique l'état opérationnel de l'équipement réel sous-jacent.

a) Etat logique

Changements d'état autorisés (**STATE-CHANGES-ALLOWED**)	Tous les services MMS prévus dans le VMD sont exploitables.
Changements d'état interdits (**NO-STATE-CHANGES-ALLOWED**)	Seuls les services appartenant à une liste de 28 services sont utilisables.
Services limités autorisés (**LIMITED-SERVICES-PERMITTED**)	Seuls les services Abort, Conclude, Status et Identify sont permis.
Services installés autorisés (**SUPPORT-SERVICES-ALLOWED**)	Tous les services MMS prévus dans le VMD sont utilisables sauf les services sur les instances de programme (Start,Stop, Reset, Resume et Kill)

Liste des services supportés dans l'état logique **NO-STATE-CHANGES-ALLOWED**

Ces services sont au nombre de 28. **Dans notre cas, nous nous limiterons, parmi ces 28 services, à ceux explicités dans la suite des notes** :

- ♦ Abort,
- ♦Conclude,
- ♦ Cancel,
- ♦ GetCapabilityList,
- ♦ GetDomainAttributes,
- ♦ GetNameList,
- ♦ GetProgramInvocationAttributes,
- ♦ GetVariableAccessAttributes,
- ♦ Identify,
- ♦ Initiate,
- ♦ Read,
- ♦ Status.

b) Etat physique

Opérationnel (**OPERATIONNAL**)	L'équipement réel est entièrement opérationnel. Aucune déficience détectée.
Partiellement opérationnel (**PARTIALLY-OPERATIONAL**)	Présence de problèmes matériels qui entraînent l'inopérabilité de certaines fonctions.
Hors-service (**INOPERABLE**)	Fonctionnement impossible suite à des problèmes matériels graves.
Réparation requise (**NEEDS-COMMISSIONING**)	L'équipement doit être réparé par une intervention locale.

La correspondance entre cet état physique du VMD et l'état de l'équipement est déterminée par le constructeur. Cette information doit être fournie par le constructeur.

3. Les Domaines

De manière simple, on peut écrire que le domaine est un objet MMS qui contient une partie ou la totalité d'un programme utilisateur et/ou une partie ou l'ensemble des données de programmes. Il peut être chargé dans le VMD, sauvegardé sur un serveur de fichiers, effacé. Par exemple, dans le cas des automates, le contenu des domaines correspond à des programmes ou des données uniquement.

L'identification d'un domaine se fait par l'intermédiaire d'un nom (32 caractères ASCII au maximum) (*Attribut Domain Name*).

Un domaine regroupe une partie des ressources (capacités) du VMD. L'attribut liste des capacités indique les ressources utilisées (*Attribut List Of Capabilities*). Dans le cas d'un automate multiprocesseurs, les ressources sont les différents CPUs qui composent l'automate. Cet attribut permet alors d'indiquer dans quel CPU doit être chargé le domaine.

Les domaines peuvent être :
- **chargeables** : ils proviennent d'une communication et peuvent être effacés,
- **statiques** : ils sont prédéfinis dans le VMD et ne peuvent être effacés. Ceci correspond à l'attribut effaçable via MMS (*Attribut MMS Deletable*)

Le domaine dispose de l'attribut contenu de domaine dans lequel se trouve le contenu du domaine (*Attribut Domain Content*).

A un domaine peuvent être associés plusieurs objets (variables, ...) (*Attribut List Of Subordinate Objects*).

Un domaine peut être partageable ou non (*Attribut Sharable*). Ceci est intéressant dans le cas de systèmes multitâches puisque le domaine ne doit être chargé qu'une seule fois et peut être utilisé par plusieurs Instances de Programmes.

Chaque domaine dispose d'un attribut indiquant les instances de programmes qui l'utilisent (*Attribut List Of Program Invocation References*).

Comme un domaine peut être sauvegardé, il existe aussi un attribut qui indique que le domaine est en cours de sauvegarde (*Attribut Upload In Progress*).

L'Objet domaine est caractérisé par son état (*Attribut State*).

Voic le diagramme d'état de l'objet domaine. Il faut savoir que le chargement d'un domaine dans un VMD se fait par le chargement successif de segments qui composent l'ensemble du domaine.

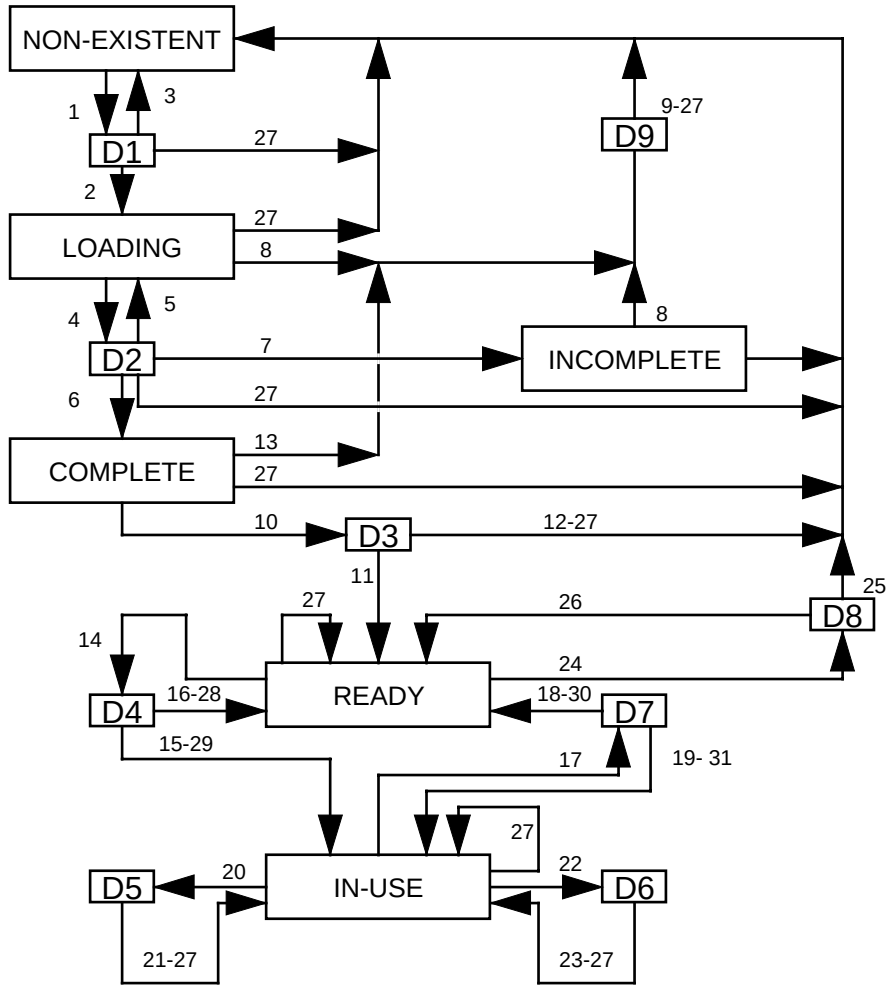


Diagramme d'état d'un domaine dans un VMD lors d'un chargement

Signification des transitions entre états :

1	Indication de début de séquence de chargement (Un Client a envoyé cette requête au VMD).
2	Réponse positive à la demande de début de séquence de chargement (le VMD envoie cette réponse au Client).
3	Réponse négative à la demande de début de séquence de chargement (le VMD envoie cette réponse au Client).
4	Requête de chargement de segment (le VMD envoie cette requête au Client).
5	Confirmation positive du chargement de segment. D'autres segments suivent (le VMD a reçu le segment du Client).
6	Confirmation positive du chargement de segment. Dernier segment (le VMD a reçu le dernier segment du Client).
7	Confirmation négative du chargement de segment (Client -> VMD).
8	Requête de demande de terminaison de la séquence de chargement avec présence d'effacement (VMD -> Client).
9	Confirmation positive ou négative de la demande de terminaison de la séquence de chargement (Client -> VMD).
10	Requête de demande de terminaison de la séquence de chargement sans présence d'effacement (VMD -> Client).
11	Confirmation positive de la demande de terminaison de la séquence de chargement (Client -> VMD).
12	Confirmation négative de la demande de terminaison de la séquence de chargement (Client -> VMD).
13	Requête de demande de terminaison de la séquence de chargement avec présence d'effacement (VMD -> Client).
14	Indication de création d'une instance de programme. Nombre d'instances de programme = 0 (Client -> VMD).

15	Réponse positive à la création d'une instance de programme (VMD -> Client).
16	Réponse négative à la création d'une instance de programme (VMD -> Client).
17	Indication de l'effacement d'une instance de programme. Nombre d'instances de programme = 1 (Client -> VMD).
18	Réponse positive à l'effacement d'une instance de programme (VMD -> Client).
19	Réponse négative à l'effacement d'une instance de programme (VMD -> Client).
20	Indication de création d'une instance de programme. Nombre d'instances de programme > 0 (Client -> VMD).
21	Réponse positive ou négative à la création d'une instance de programme (VMD -> Client).
22	Indication de l'effacement d'une instance de programme. Nombre d'instance de programme > 1(Client -> VMD).
23	Réponse positive ou négative à l'effacement d'une instance de programme (VMD -> Client).
24	Indication de l'effacement d'un domaine (Client -> VMD).
25	Réponse positive à l'effacement d'un domaine (VMD -> Client).
26	Réponse négative à l'effacement d'un domaine (VMD -> Client).
27	Indication d'avortement (Client -> VMD).
28	Indication d'avortement, la création d'une instance de programme a échoué.
29	Indication d'avortement, la création d'une instance de programme est réussie.

30	Indication d'avortement, l'effacement d'une instance de programme est réussi.
31	Indication d'avortement, l'effacement d'une instance de programme a échoué.

Le nombre de domaines qu'un VMD peut contenir dépend du Serveur MMS (donc du constructeur).

4. Les Instances de Programme ou IP (Program Invocation)

L'objet Instance de Programme permet de créer au départ d'un ou de plusieurs domaines un programme exécutable. L'Instance de Programme permet de contrôler le déroulement du programme.

L'identification de l'Instance de Programme se fait sur base d'un nom comportant 32 caractères ASCII au maximum (*Attribut Program Invocation Name*).

Une Instance de Programme peut être réutilisable ou non (*Attribut Reusable*) : ceci est intéressant dans le cas de programmes qui se font sur un cycle et qui peuvent être relancé (usinage de pièces par exemple).

Une Instance de Programme peut être créée au départ de plusieurs domaines. Il existe une liste qui contient les noms des domaines utilisés par l'Instance de Programme (*Attribut List Of Domain References*).

Une Instance de Programme peut être créée, effacée. Elle peut être démarrée, arrêtée, redémarrée, avortée, réinitialisée.

Le nombre d'Instances de Programme que supporte un VMD dépend de l'équipement sous-jacent.

L'Instance de Programme est caractérisée par son état (*Attribut State*).

Voici le diagramme d'état de l'objet Instance de Programme

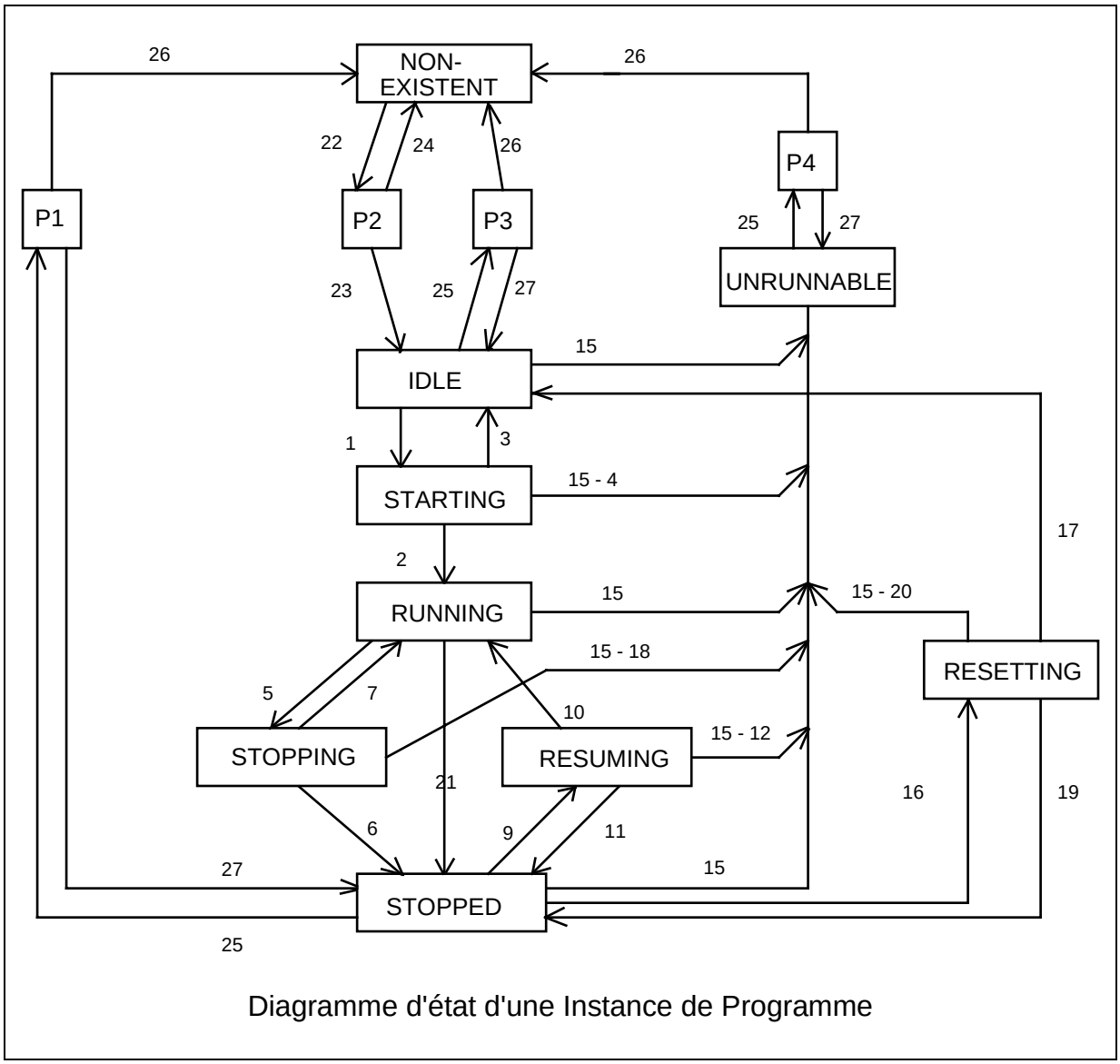


Diagramme d'état d'une Instance de Programme

Signification des transitions entre états :

1	Démarrage du programme
2	Acquittement positif du démarrage de programme
3	Acquittement négatif, non-destructif du démarrage de programme
4	Acquittement négatif, destructif du démarrage de programme
5	Arrêt du programme
6	Acquittement positif de l'arrêt de programme
7	Acquittement négatif, non destructif de l'arrêt de programme

8 Acquittement négatif, destructif de l'arrêt de programme
9 Reprise du programme
10 Acquittement positif de la reprise du programme
11 Acquittement négatif, non destructif de la reprise du programme
12 Acquittement négatif, destructif de l'arrêt de programme
15 Acquittement positif de l'avortement du programme
16 Réinitialisation du programme
17 Acquittement positif de la réinitialisation du programme
19 Acquittement négatif, non destructif de la réinitialisation du programme
20 Acquittement négatif, destructif de la réinitialisation du programme
21 Arrêt de l'Instance de Programme (local : généré par API lui-même)
22 Générer l'Instance de Programme
23 Acquittement positif de la création de l'Instance de Programme
24 Acquittement négatif de la création de l'Instance de Programme
25 Effacement de l'Instance de Programme
26 Acquittement positif de l'effacement de l'Instance de Programme
27 Acquittement négatif de l'effacement de l'Instance de Programme

5. Les Variables

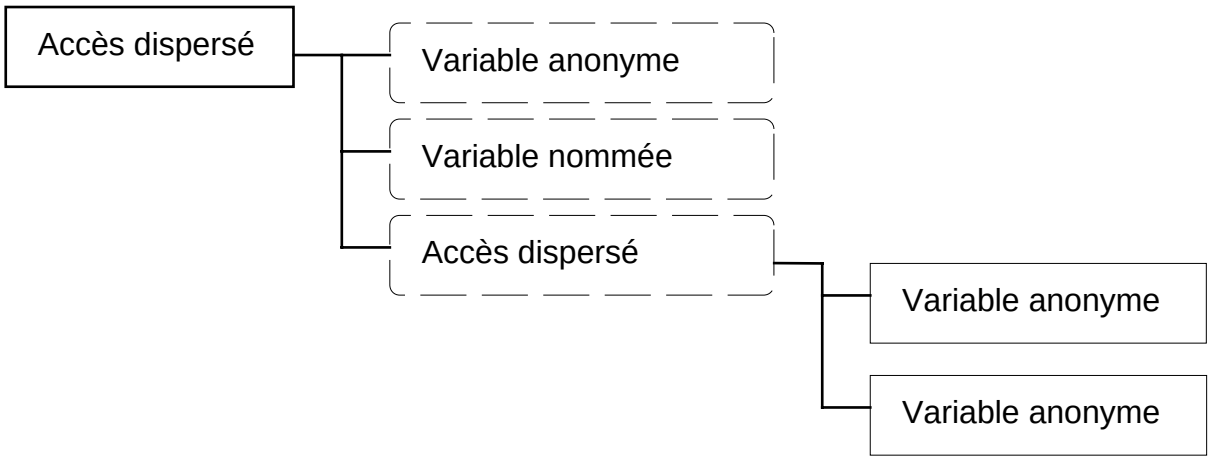
Les Variables représentent les données d'un programme utilisateur, d'une application. On peut lire le contenu d'une variable ou écrire dans une variable.

Il existe cinq objets d'accès aux Variables :

♦ **l'objet variable anonyme** (*Unnamed Variable*) : la variable anonyme permet d'accéder au contenu d'une variable en donnant une adresse qui correspond à son emplacement mémoire dans le VMD. L'utilisation de ce genre de variables est à éviter puisqu'il réduit la portabilité et la transparence de cette variable.

♦ **l'objet variable nommée** (*Named Variable*) : l'accès au contenu des variables nommées se fait par l'intermédiaire du nom des variables. La variable nommée est à préférer à la précédente puisqu'elle permet beaucoup plus de clarté et de flexibilité.

♦ **l'objet variable à accès dispersé** (*Scattered Access Variable*) : cet objet nommé définit un ensemble composé de variables indépendantes du VMD auxquelles cet objet permet d'accéder en une seule fois. Cet ensemble peut comporter des variables anonymes ou nommées ou un autre objet accès dispersé (définition récursive possible). **Il est à noter qu'en cas d'erreur d'accès, un succès partiel n'existe pas**.



Objet Variable à Accès dispersé

l'objet liste nommée (*Named List*) : cet objet nommé définit un ensemble composé de variables indépendantes du VMD auxquelles cet objet permet d'accéder en une seule fois. Cette ensemble peut comporter des variables anonymes ou nommées ou un autre objet accès dispersé. **Il est à noter qu'en cas d'erreur d'accès, un succès partiel est possible**. Aussi, la liste nommée est à préférer à la variable à accès dispersé.

♦ **l'objet type nommé** (*Named Type*) : les variables sont caractérisées par un type. Il existe différents types qui peuvent être simples ou complexes. L'objet Named Type permet d'associer un nom à une description de type complexe.

a) Les Variables nommées

Une variable nommée est définie par son nom, par une description de son type.

NOM
TYPE
DOMAINE DE VALIDITE

Définition d'une variable au point de vue du réseau

Le **nom** de la variable comporte 32 caractères ASCII au maximum.

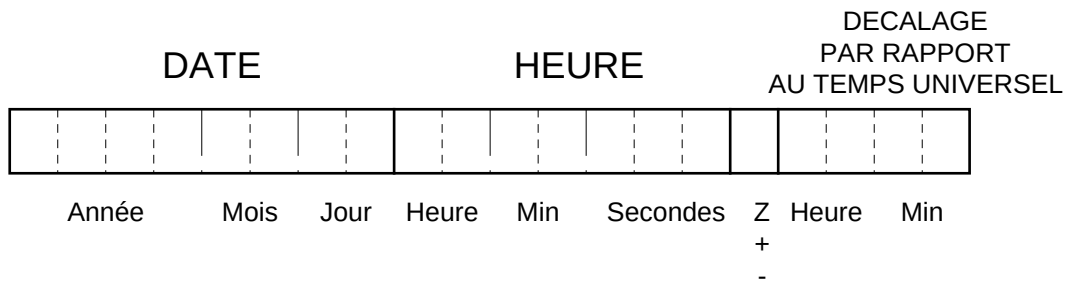
La **type** de la variable peut être simple ou complexe.

Une variable peut être constituée :

♦ d'un des **types de données de base** suivant (principaux types) :

BOOLEAN ou Booléen : ce type comporte deux états VRAI (0001H) ou FAUX (0000H).

- **INTEGER ou Entier (8, 16 ou 32 bits)** : ce type permet de coder un entier signé sur 8, 16 ou 32 bits.
- **UNSIGNED ou Entier non-signé (8, 16 ou 32 bits)** : ce type permet de coder un entier non-signé sur 8, 16 ou 32 bits.
- **FLOATING POINT ou Virgule Flottante (32 bits)** : ce type permet de coder des nombres à virgule flottante.
- **BIT STRING ou Chaîne de bits (n bits)**
- **OCTET STRING ou Chaîne d'octets (n octets)**
- **VISIBLE STRING ou Chaîne de caractères affichables (n caractères)**
- **GENERALIZED TIME** : date et heure (chaîne de caractères affichables).



Codage du temps dans le cas du Generalized Time
(le suffixe **Z** indique que cette heure est l'heure Universelle)

BINARY TIME : date et heure en représentation compactée en binaire (longueur : 4 octets (heure) ou 6 octets (heure et date)).

- **BCD** : valeur décimale codé en binaire.
- **REAL** : valeur réelle au format ISO 8824 (Mantisse x Base $^{\text{Exposant}}$).

d'une **structure d'éléments du type de données de base**

♦ d'un **vecteur d'éléments du type de données de base**

Le degré d'imbrication des données des variables dépend du Serveur MMS (VMD).

```
STRUCTURE _COMPLEXE_NIVEAU_2{
struct1{
        entier1     : INTEGER 16,
        chaîne1     : OCTET STRING 50
        },
struct2{
        réel2       : FLOATING-POINT 32,
        booleen     : BOOLEAN
        },
vecteur [5]: chaîne OCTET STRING 32,
réel            : FLOATING-POINT 32
}
```

Exemple de type complexe

Il existe trois domaines de validité pour les variables. Si une variable a son **domaine de validité** défini pour :

♦ le **VMD**. La variable est accessible par tout Client qui accède au VMD,

♦ un **Domaine**. La variable sera accessible par tout Client si le domaine est chargé dans le VMD,

♦ une **Association d'application**. La variable sera uniquement accessible par le client qui utilise cette association d'application.

Au niveau du réseau, les programmeurs ne doivent connaître que le nom de la variable, la description de son type et son domaine de validité.

Les variables peuvent être :

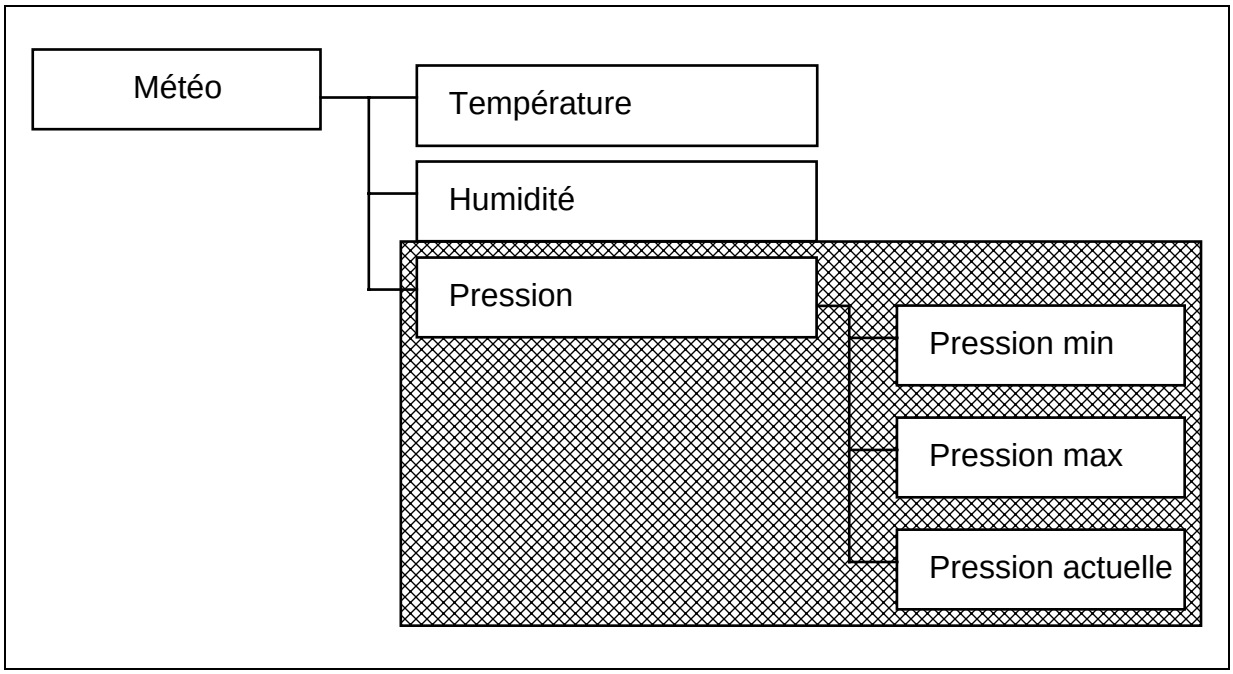
♦ **statiques** (définition globale au niveau du VMD),

♦ **dynamiques** : elles sont générées par la communication et sont définies au niveau d'un domaine ou d'une association d'applications (elles sont donc effaçables). Elles sont uniquement accessibles lorsque le domaine pour lequel elles sont définies est chargé dans le VMD.

b) L'Alternate Access

Il est possible de spécifier un **Alternate Access**. L'Alternate Access permet de modifier, de manière temporaire, le type d'une variable complexe pour n'accéder qu'à une partie de la variable.

Dans l'Alternate Access, il faut indiquer à quelle partie de la variable complexe on désire accéder. Cette partie peut être un élément de tableau ou de structure ou une série d'éléments de tableau.



Dans cet exemple, l'Alternate Access permet d'accéder uniquement à la sous-structure *Pression* de la variable *Météo.*

F. Les Services MMS

L'ensemble de services MMS peut être regroupés en différents sous-ensembles portant sur une même fonctionnalité ou un même type d'objets.

Il existe 10 sous-ensembles:

♦ *La gestion générale de l'environnement MMS (Environment and General Management)* : ces services permettent d'établir, de terminer ou d'interrompre les

associations d'application et d'annuler des demandes de services.

♦ *La gestion du VMD (VMD Support)* : ces services permettent à un Client de demander des informations sur un Serveur (obtention de l'état du VMD, identification du VMD, obtention des noms des objets du VMD, obtention des capacités de la VMD, modification de noms d'objets du VMD (pas utilisé avec les automates), signalisation spontanée de l'état du VMD).

♦ *La gestion des domaines (Domain Management)* : ces services permettent de charger, sauvegarder, effacer les domaines ainsi que d'obtenir les attributs des domaines.

♦ *La gestion des instances de programme (Program Invocation Management)* : ces services permettent de créer, effacer, lancer, arrêter, faire reprendre, réinitialiser ou d'interrompre les instances de programme ou d'obtenir les attributs des instances de programme.

♦ *L'accès aux variables (Variable Access)* : ces services permettent d'écrire dans des variables, de lire le contenu des variables, de signaler le contenu d'une variable, d'obtenir les attributs de variables et de type nommé, de créer et de détruire des variables et des types nommés.

♦ *La gestion des Sémaphores (Semaphore Management)* : ces services permettent de contrôler un service MMS confirmé par un sémaphore, de prendre le contrôle et de libérer un sémaphore distant, de créer et de détruire un sémaphore, de lire l'état d'un sémaphore, ...

♦ *La communication avec l'opérateur (Operator Communication)* : ces services permettent d'envoyer des informations vers une interface opérateur simple et de récupérer des informations provenant de cette interface opérateur.

♦ *La gestion des événements (Event Management)* : ces services permettent de notifier, d'acquitter un événement, de contrôler par un événement un service MMS confirmé, de définir, d'effacer les conditions pour qu'un événement ait lieu, ...

♦ *La gestion de journal (Journal Management)* : ces services permettent de créer, détruire, initialiser, d'écrire et lire une rubrique d'un journal, d'obtenir le nombre de rubriques d'un journal.

♦ *La gestion des fichiers (File Management)* : ces services permettent d'ouvrir, de lire, de fermer, d'effacer, de renommer un fichier, de consulter le répertoire des fichiers.

Ne seront décrits ici que les services MMS qui sont utilisables actuellement avec des VMD d'automates :

♦ la gestion générale de l'environnement MMS

♦ la gestion des domaines,

♦ la gestion des Instances de programmes
♦ l'accès aux variables,
♦ la gestion du VMD.

1. La gestion générale de l'environnement MMS (Environment and General Management)

Ces services permettent d'établir, de terminer ou d'interrompre les associations d'application et d'annuler des demandes de services.

a) Le service Initiate

Ce service **confirmé** permet d'établir les associations d'application avec les entités d'application distantes. Lors de l'établissement de l'association d'application, les entités MMS partenaires échangent des informations permettant de négocier les possibilités de l'association d'application.

b) Le service Conclude

Ce service **confirmé** permet de terminer les associations d'application de manière normale. Ceci implique que :
♦ l'émetteur n'a plus de requêtes à transmettre,
♦ le récepteur peut continuer à renvoyer des réponses liés à des indications reçues qui sont en attente,
♦ **le récepteur peut refuser de terminer l'association d'application**.

De plus, il faut négocier qui, des deux partenaires communicants, a le droit d'utiliser ce service. Sinon, il risque d'y avoir blocage si les deux partenaires envoient en même temps une requête CONCLUDE.

c) Le service Abort

Ce service **non confirmé** permet d'interrompre les associations d'application (fin anormale) que le récepteur l'accepte ou non. Il peut être généré par l'utilisateur ou par le fournisseur de service OSI (l'indication provient alors de source locale). **Il risque d'y avoir perte d'informations**.

d) Le service Cancel

Ce service **confirmé** permet d'annuler un service confirmé en cours. Il ne peut pas annuler le service CANCEL.

2. L'accès aux variables (Variable Access)

a) Le service Lire Variables (Read)

Ce service **confirmé** permet au client de demander le contenu d'une ou plusieurs variables à un serveur. Le Serveur renvoie une réponse qui, lorsqu'elle est positive, comportera le contenu de la variable demandée sinon il renvoie le code d'erreur.

b) Le service Ecrire Variables (Write)

Ce service **confirmé** permet au client d'écrire dans une ou plusieurs variables du Serveur. Le Serveur renvoie une réponse qui indique si l'opération s'est bien déroulée ou non.

c) Le service Signaler Variables (InformationReport)

Le serveur peut ainsi envoyer une variable ou plusieurs variables vers un client sans que ce dernier n'envoie de requête de lecture. Le serveur ne reçoit pas de confirmation de la part du Client pour ce service.

d) Le service Obtention des attributs des variables (GetVariableAccessAttributes)

Ce service **confirmé** permet d'obtenir les attributs des variables nommées, anonyme ou à accès dispersé.

3. La gestion des domaines (Domain Management)

a) Généralités

Au niveau des services de domaines, les fonctionnalités prévues sont le chargement, la sauvegarde, l'effacement et l'obtention des attributs des objets domaine.

Plusieurs Clients peuvent charger simultanément des domaines différents dans un même VMD. C'est le nom du domaine qui permet de différencier, au niveau du VMD, les différents chargements en cours.

Un domaine qui est chargé dans un VMD ne peut pas être rechargé : il faut d'abord l'effacer puis le recharger. Aussi, un même domaine ne peut pas être chargé en même temps par plusieurs Clients. Seule la demande de chargement d'un seul des Clients sera acceptée, les autres demandes seront refusées.

Il est aussi possible de sauvegarder le contenu d'un même domaine sur plusieurs Clients en même temps. L'objet Automate de sauvegarde (Upload State Machine) permet de gérer chaque sauvegarde. Ainsi, le nombre d'Automate de sauvegarde prévus dans le VMD limite le nombre de sauvegarde qui peuvent se dérouler simultanément.

b) Les fonctions de Chargement de domaines dans un VMD

(1) La fonctionnalité de chargement de domaine lancé par un Client

Cette fonctionnalité permet à un Client de charger un domaine dans un Serveur (VMD). En réalité, elle utilise trois services différents:

- le service **InitiateDownloadSequence**,
- le service **DownloadSegment**,
- le service **TerminateDownloadSequence**.

Voici le déroulement du chargement dans le cas de cette fonctionnalité :

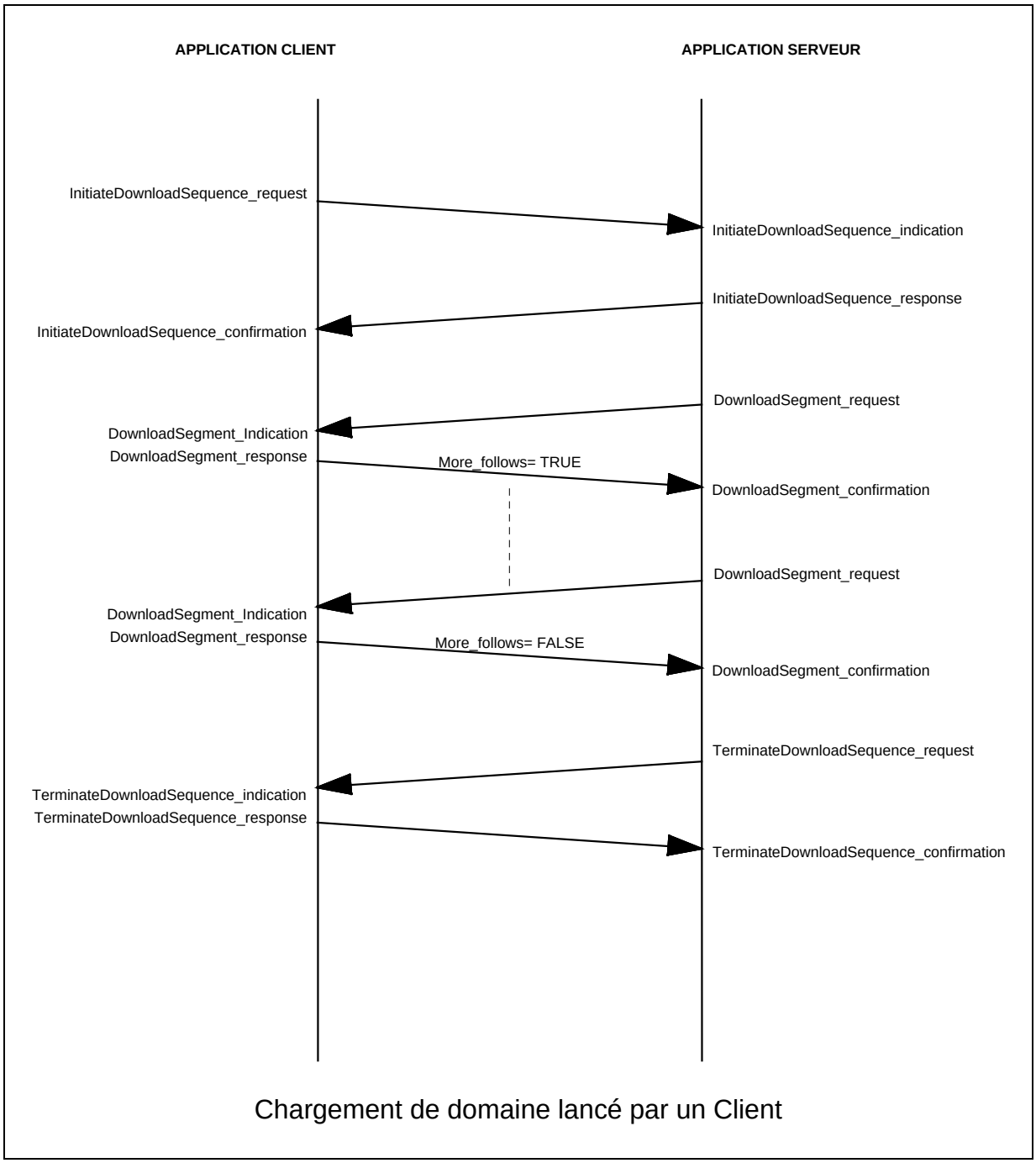
1° Le Client envoie une requête **InitiateDownloadSequence** au Serveur dans laquelle le Client indique le nom du domaine à charger, la liste des ressources utilisée par le domaine et si le domaine est partageable ou non.

2° Le Serveur envoie plusieurs requêtes **DownloadSegment** au Client pour obtenir la totalité du contenu du domaine. Le Client indique dans ses réponses si le segment qu'il envoie est le dernier ou non (MORE_FOLLOWS = TRUE ou FALSE).

3° Le Serveur envoie une requête **TerminateDownloadSequence** qui indique si le chargement s'est bien terminé ou non.

On remarquera que le contenu du domaine se trouve chez le Client.

De plus, il apparaît que c'est le VMD (Serveur) qui régule la transmission des segments. En effet, les VMD peuvent être des équipements relativement lents par rapport au taux de transfert de données possible au travers du réseau.



Chargement de domaine lancé par un Client

(2) La fonctionnalité de chargement de domaine lancé par un Serveur

Cette fonctionnalité permet à un Serveur de charger un domaine au départ d'un Client (serveur de fichiers). Ceci correspond donc à un **autochargement**. Elle repose sur les quatre services suivants:

- le service **RequestDomainDownload**,
- le service **InitiateDownloadSequence**,
- le service **DownloadSegment**,
- le service **TerminateDownloadSequence**.

Voici le déroulement du chargement dans le cas de cette fonctionnalité :

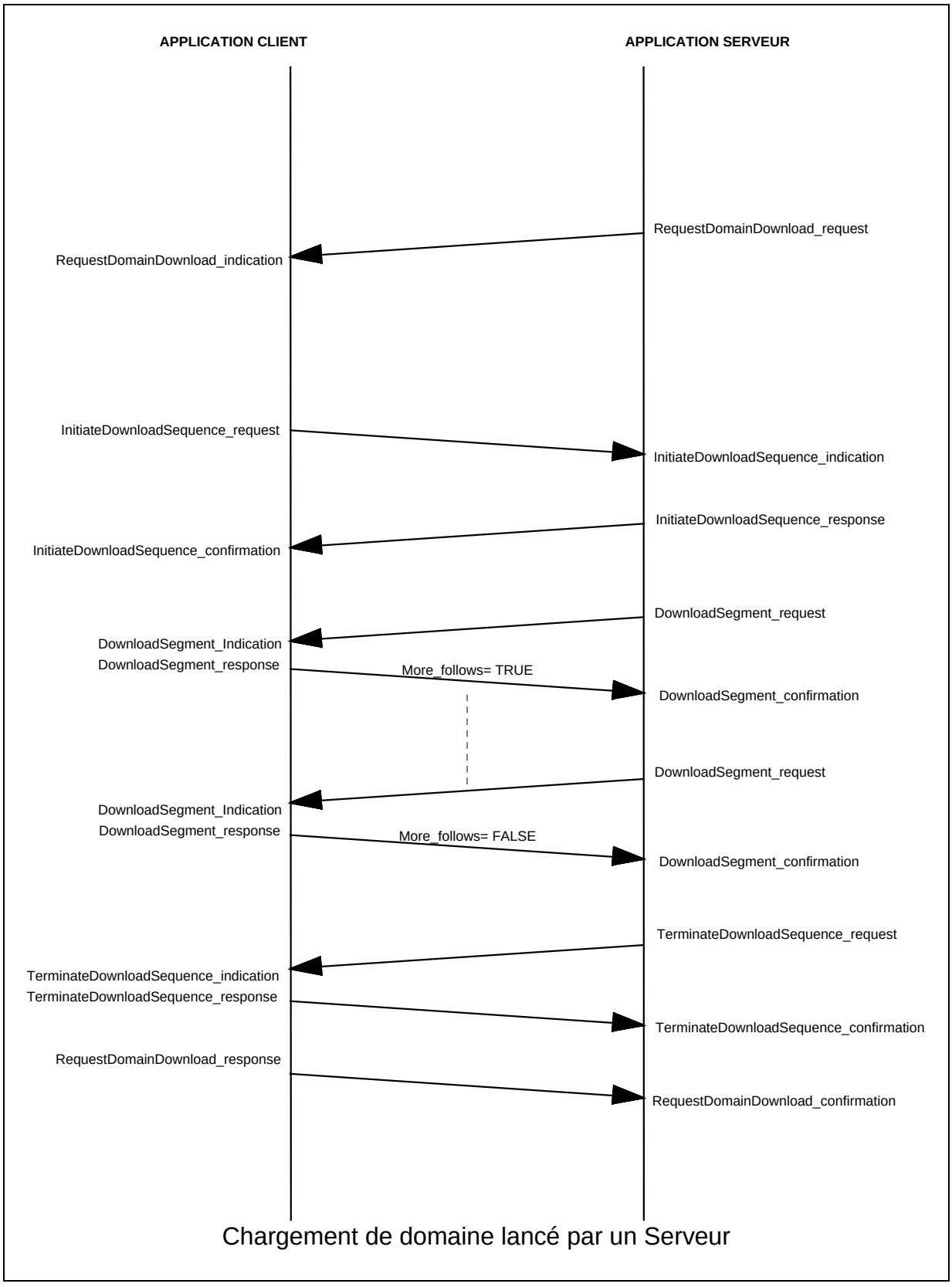
1° Le Serveur envoie une requête **RequestDomainDownload** au Client dans laquelle le Serveur indique le nom du domaine à charger, la liste des ressources, si le domaine est partageable ou non et le nom du fichier (chez le Client) où se trouve le contenu du domaine.

2° Un chargement normal est opéré

3° Le Client envoie la réponse à la requête **RequestDomainDownload** (1°).

On voit donc qu'en fait, le Serveur demande à un Client que ce dernier demande le chargement d'un domaine dans le Serveur.

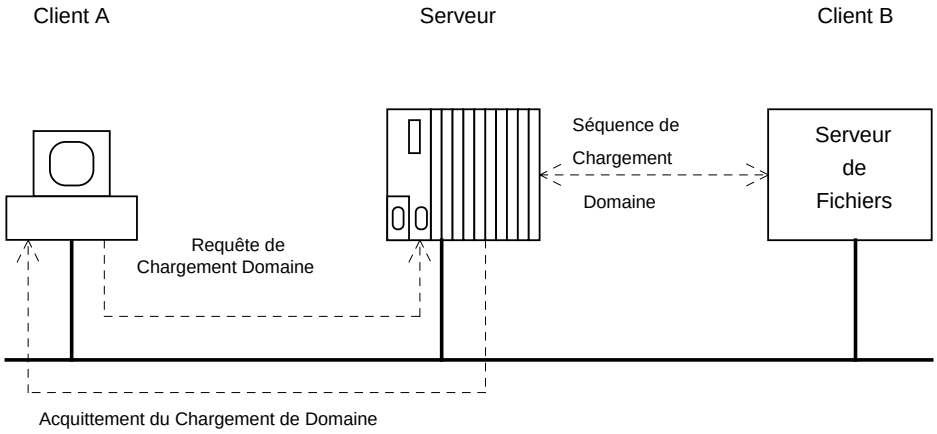
De plus, il apparaît que c'est le VMD (Serveur) qui régule la transmission des segments. En effet, les VMD peuvent être des équipements relativement lents par rapport au taux de transfert de données possible au travers du réseau.



Chargement de domaine lancé par un Serveur

(3) La fonctionnalité de chargement de domaine lancé par un Client avec une tierce partie.

Cette fonctionnalité permet à un Client de charger un domaine dans un Serveur (VMD). **Le contenu du domaine se trouve chez un autre Client (serveur de fichiers).**



Chargement de domaine avec tierce partie (Serveur de fichiers)

Elle emploie les cinq services suivants:

- le service **LoadDomainContent**,
- le service **RequestDomainDownload**,
- le service **InitiateDownloadSequence**,
- le service **DownloadSegment**,
- le service **TerminateDownloadSequence**.

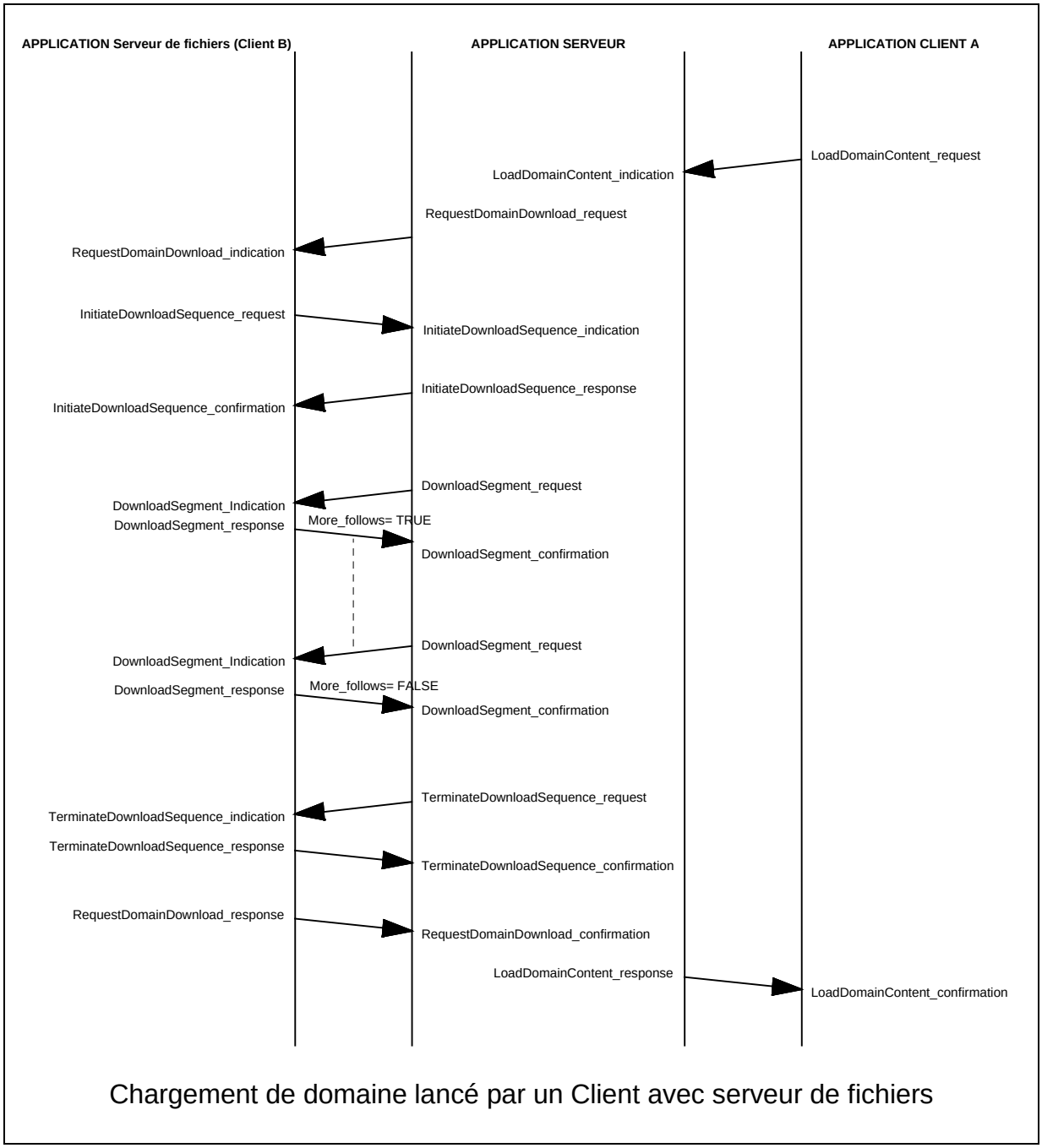
Voici le déroulement du chargement dans le cas de cette fonctionnalité :

1° Le Client (Client A) envoie une requête **LoadDomainContent** au Serveur dans laquelle le Client indique le nom du domaine à charger, la liste des ressources, si le domaine est partageable ou non, la référence vers la tierce partie (Client B - serveur de fichiers) et le nom du fichier (dans le serveur de fichiers) dans lequel le contenu du domaine se trouve.

2° Le VMD opère un autochargement

3° Le Serveur (VMD) envoie la réponse à la requête **LoadDomainContent** du Client A.

On voit donc qu'en fait, ici le Client A demande un autochargement au Serveur; cet autochargement doit se faire au départ d'un autre Client (Client B - Serveur de fichiers). **Pour que ce service soit supporté par le Serveur (VMD), ce dernier doit supporter le CCB Third party et ces cinq services de domaines.**



Chargement de domaine lancé par un Client avec serveur de fichiers

c) Les fonctions de Sauvegarde de domaines d'un VMD

(1) La fonctionnalité de sauvegarde de domaine lancée par un Client

Cette fonctionnalité permet à un Client de sauvegarder le contenu d'un domaine qui se trouve sur le Serveur (VMD). **Le contenu du domaine est stocké chez le Client qui a demandé la sauvegarde**.

Cette fonctionnalité repose sur l'utilisation des 3 services suivants:

- le service **InitiateUploadSequence**,
- le service **UploadSegment**,
- le service **TerminateUploadSequence**.

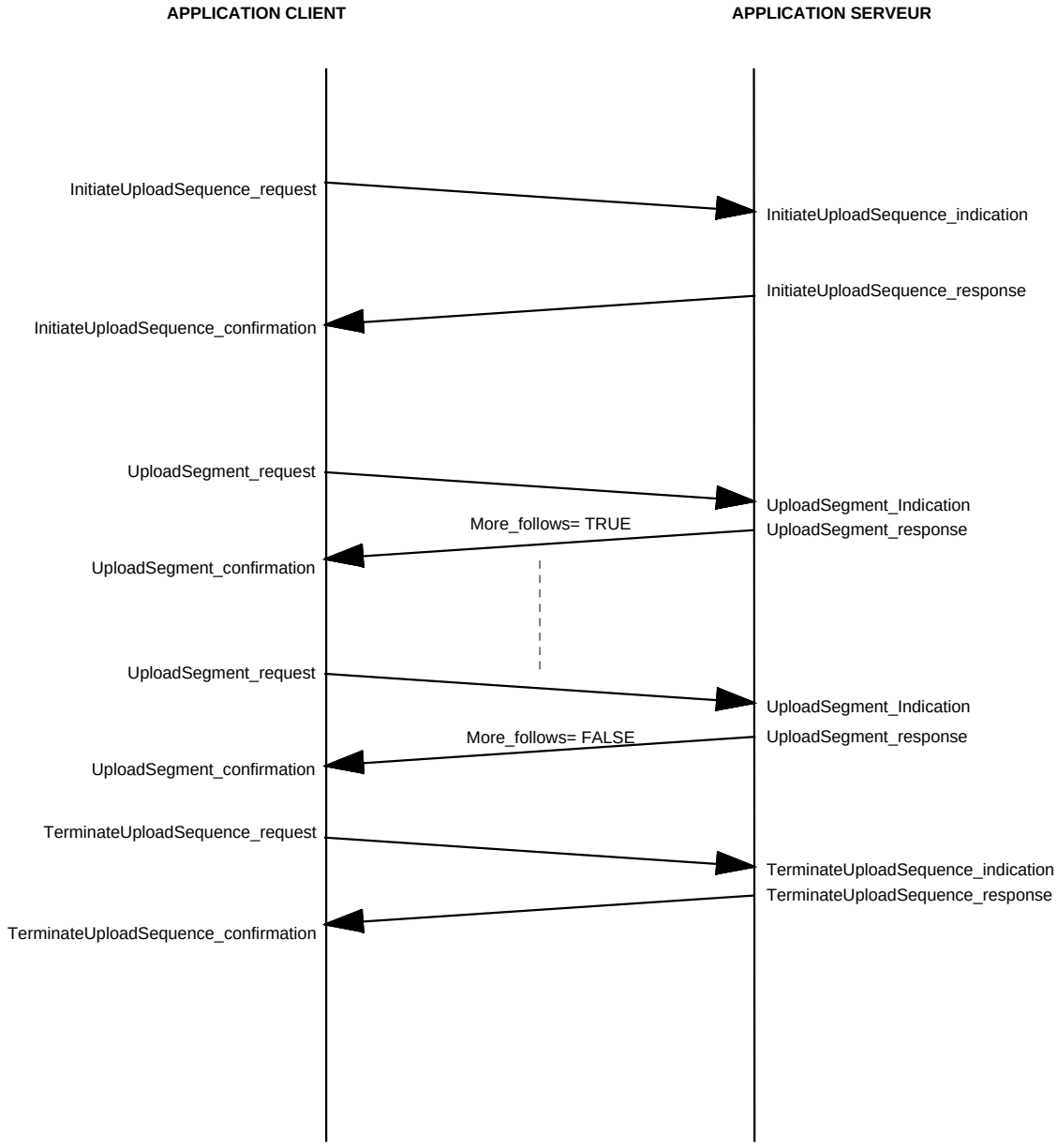
Voici le déroulement du chargement dans le cas de cette fonctionnalité :

1° Le Client envoie une requête **InitiateUploadSequence** au Serveur. Dans celle-ci, le Client indique le nom du domaine à charger. Le Serveur renvoie en réponse la liste des ressources utilisées par le domaine ainsi qu'une référence à l'automate de sauvegarde qui est utilisé pour gérer la sauvegarde.

2° Le Client envoie plusieurs requêtes **UploadSegment** au Serveur pour obtenir la totalité du contenu du domaine (ces requêtes contiennent la référence de l'automate de sauvegarde). Le Serveur indique dans ses réponses si le segment qu'il envoie est le dernier ou non (MORE_FOLLOWS = TRUE ou FALSE).

3° Le Client envoie, au Serveur, une requête **TerminateUploadSequence** qui indique si la sauvegarde s'est bien terminée ou non.

On remarquera que c'est le Client qui régule la vitesse d'échange des segments de données. En effet, c'est ce dernier qui doit avoir le temps "d'ingurgiter" les données qu'il reçoit.



Sauvegarde de domaine lancée par un Client

(2) La fonctionnalité de sauvegarde de domaine lancée par le Serveur

Cette fonctionnalité permet à un VMD (Serveur) d'opérer une sauvegarde du contenu de l'un de ses domaines chez un Client. Il s'agit ici donc d'une autosauvegarde.

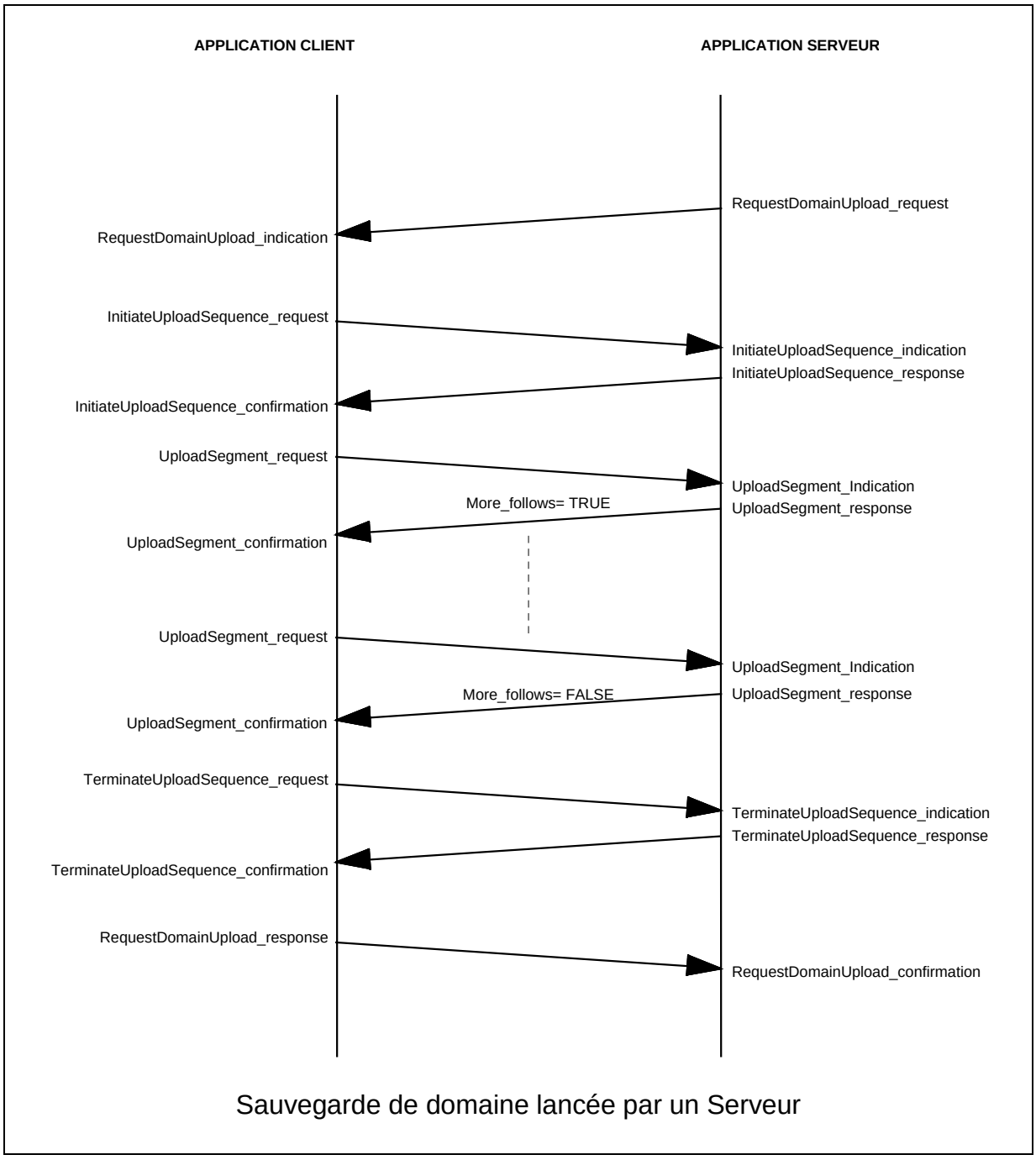
Cette fonctionnalité repose sur l'utilisation des 4 services suivants:

- le service **RequestDomainUpload**,
- le service **InitiateUploadSequence**,

♦ le service **UploadSegment**,
♦ le service **TerminateUploadSequence**.

Voici le déroulement d'une autosauvegarde:

1° Le Serveur envoie une requête **RequestDomainUpload** à un Client en y indiquant le nom du domaine à sauvegarder ainsi que le nom du fichier dans lequel devra être stocké le contenu chez le Client.
2° Le Client opère une sauvegarde du domaine du VMD.
3° Le Client envoie la réponse à la requête initiale **RequestDomainUpload** en indiquant si l'opération s'est déroulée sans erreur.

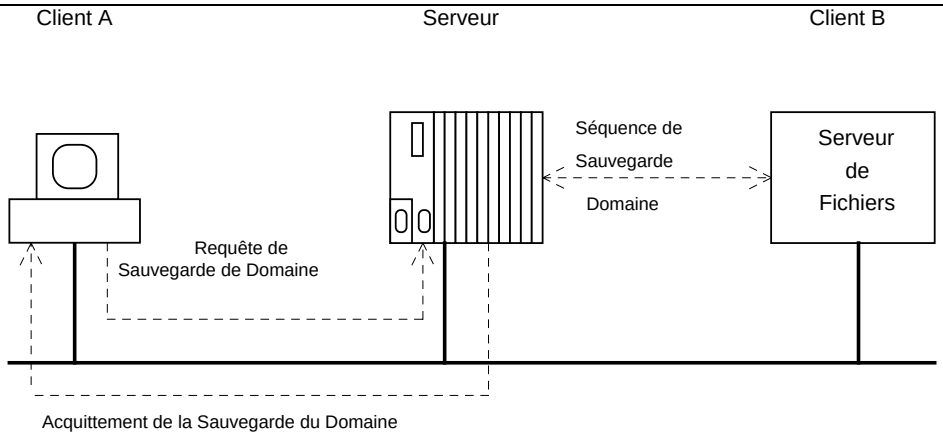


Sauvegarde de domaine lancée par un Serveur

On remarquera que c'est le Client qui régule la vitesse d'échange des segments de données. En effet, c'est ce dernier qui doit avoir le temps "d'ingurgiter" les données qu'il reçoit.

(3) La fonctionnalité de sauvegarde de domaine lancée par un Client avec une tierce partie

Cette fonctionnalité permet à un Client de commander la sauvegarde d'un domaine d'un VMD sur un serveur de fichiers (Client - Tierce partie). Le contenu du domaine est donc sauvegardé dans un fichier du serveur de fichiers.



Sauvegarde de domaine avec liaison tripartite (Serveur de fichiers)

Cette fonctionnalité repose sur l'utilisation des 5 services suivants:

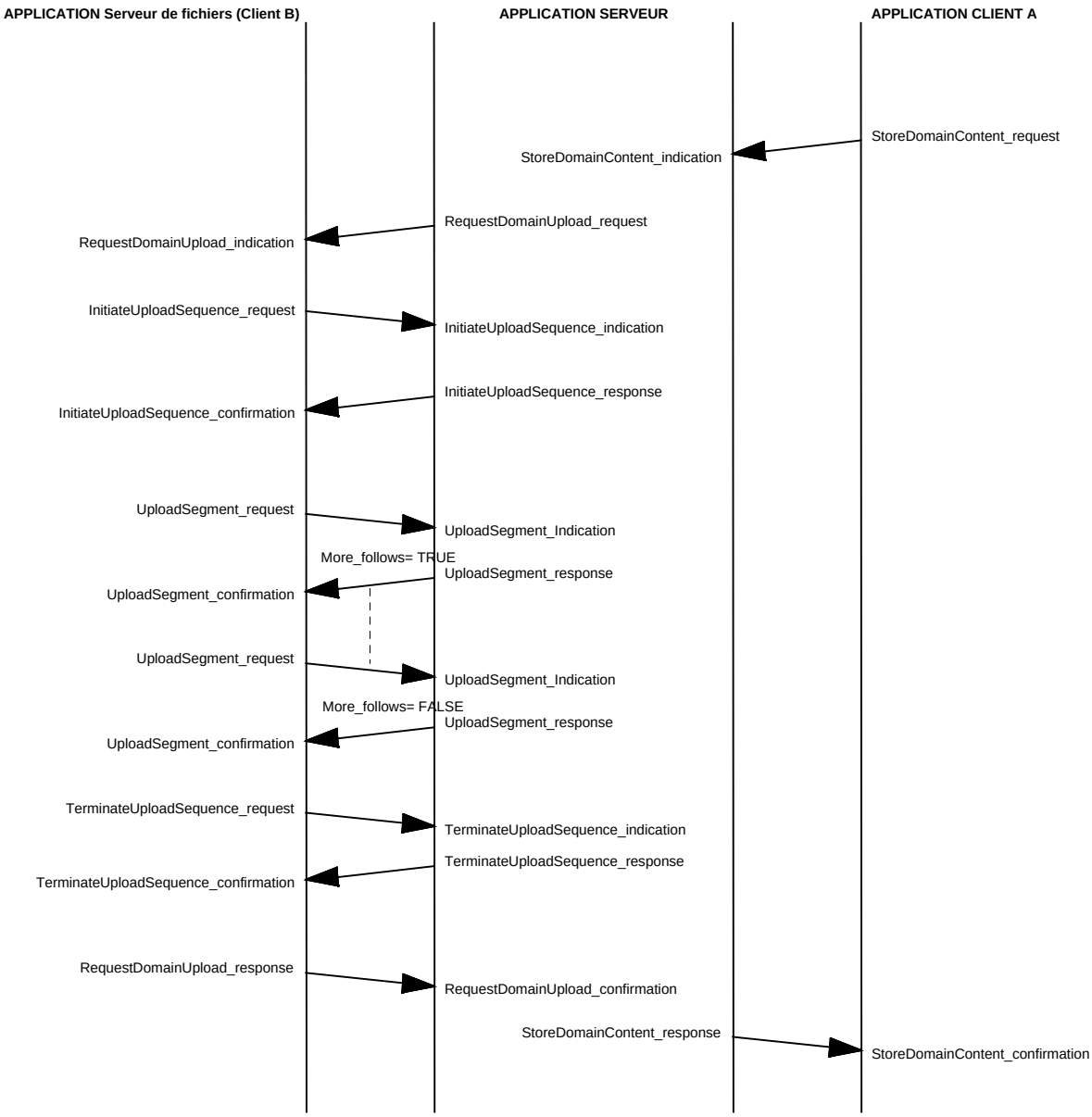
- le service **StoreDomainContent**,
- le service **RequestDomainUpload**,
- le service **InitiateUploadSequence**,
- le service **UploadSegment**,
- le service **TerminateUploadSequence**.

Voici le déroulement de la sauvegarde dans le cas de cette fonctionnalité:

1° Le Client A envoie une requête **StoreDomainContent** au VMD (Serveur) dont un domaine doit être sauvegardé en y indiquant le nom du domaine à sauvegarder, le nom du fichier dans lequel le contenu devra être stocké ainsi qu'une référence indiquant la partie tierce (Client B - serveur de fichiers) où le contenu du domaine doit être stocké.

2° Le VMD opère une autosauvegarde vers le serveur de fichiers

3° Le Serveur envoie, au Client A, la réponse à la requête initiale **StoreDomainContent** en indiquant si l'opération s'est bien déroulée. (1°)



Chargement de domaine lancé par un Client avec serveur de fichiers

d) Le service Effacement de domaine (DeleteDomain)

Ce service **confirmé** permet d'effacer un domaine dans un VMD. Il suffit d'indiquer le nom du domaine. Cependant, il est nécessaire que :

- ♦ l'attribut effaçable par services MMS soit VRAI,
- ♦ l'état du Domaine soit READY,
- ♦ aucune sauvegarde de ce domaine ne soit en cours.

e) Le service Obtention des attributs d'un domaine (GetDomainAttributes)

Ce service **confirmé** permet d'obtenir les attributs liés au domaine dont le nom a été indiqué dans la requête. Ce service permet donc d'obtenir les renseignements suivants:

- ♦ Liste des ressources (List Of Capabilities),
- ♦ Etat (State),
- ♦ Effaçable via services MMS (MMS Deletable),
- ♦ Partageable (Sharable),
- ♦ Liste des Instances de Programmes (List Of Program Invocations).

4. La gestion des instances de programme (Program Invocation Management)

Les services sur les Instances de Programme permettent au Client de gérer l'exécution de programmes dans un Serveur (VMD).

a) Le service Création d'Instance de programme (CreateProgramInvocation)

Ce service **confirmé** permet à un Client de créer une Instance de Programme dans un Serveur au départ de domaines chargés dans le Serveur.

Ce service demande de fournir le nom de l'Instance de Programme à créer, les domaines utilisés par l'Instance de Programme et si l'Instance de Programme est réutilisable.

b) Le service Effacement d'Instance de programme (DeleteProgramInvocation)

Ce service **confirmé** permet à un Client d'effacer, dans un Serveur, l'Instance de Programme dont le nom est spécifié dans la requête.

Pour que ce service puisse s'exécuter sans erreur, l'Instance de Programme doit être effaçable par les services MMS et dans l'état **IDLE** ou **UNRUNNABLE**.

c) Le service Démarrage d'Instance de programme (StartProgramInvocation)

Ce service **confirmé** permet à un Client de démarrer l'Instance de Programme dont le nom est spécifié dans la requête.

Ce service ne peut être utilisé que sur une Instance de Programme se trouvant à l'état **IDLE** sinon une erreur sera renvoyée.

L'Instance de Programme passera de l'état **IDLE** à l'état **RUNNING** en passant par l'état transitoire **STARTING**.

d) Le service Arrêt d'Instance de programme (StopProgramInvocation)

Ce service **confirmé** permet à un Client d'arrêter une Instance de Programme dont le nom est spécifié dans la requête.

Ce service ne peut être utilisé que sur une Instance de Programme se trouvant à l'état **RUNNING** sinon une erreur sera renvoyée.

L'Instance de Programme passera de l'état **RUNNING** à l'état **STOPPED** en passant par l'état transitoire **STOPPING**.

e) Le service Réinitialisation d'Instance de programme (ResetProgramInvocation)

Ce service **confirmé** permet à un Client de réinitialiser l'Instance de Programme dont le nom est spécifié dans la requête.

Ce service ne peut être utilisé que sur une Instance de Programme se trouvant à l'état **STOPPED** sinon une erreur sera renvoyée.

L'Instance de Programme passera de l'état **STOPPED** à l'état **IDLE** en passant par l'état transitoire **RESETTING**.

f) Le service Reprise d'Instance de programme (ResumeProgramInvocation)

Ce service **confirmé** permet à un Client de demander la reprise de l'exécution de l'Instance de Programme dont le nom est spécifié dans la requête.

Ce service ne peut être utilisé que sur une Instance de Programme se trouvant à l'état **STOPPED** sinon une erreur sera renvoyée.

L'Instance de Programme passera de l'état **STOPPED** à l'état **RUNNING** en passant par l'état transitoire **RESUMING**.

g) Le service Avortement d'Instance de programme (AbortProgramInvocation)

Ce service **confirmé** permet à un Client d'avorter l'Instance de Programme dont le nom est spécifié dans la requête.

Ce service peut être utilisé que sur une Instance de Programme se trouvant dans un état quelconque.

L'Instance de Programme passera de l'état **UNRUNNABLE**.

h) Le service Obtention des attributs d'Instance de programme (GetProgramInvocationAttributes)

Ce service **confirmé** permet à un Client d'obtenir les attributs de l'Instance de Programme dont le nom est spécifié dans la requête.

Le VMD répond en renvoyant :

- l'état de l'Instance de Programme,
- la liste des domaines utilisés par l'Instance de Programme,
- l'attribut indiquant si l'Instance de Programme est effaçable via les services MMS,
- l'argument d'exécution,
- l'attribut indiquant s'il est réutilisable.

5. La gestion du VMD (VMD Support)

Ces services permettent à un Client de demander des informations sur un Serveur (obtention de l'état du VMD, identification du VMD, obtention des noms des objets du VMD, obtention des capacités du VMD, modification de noms d'objets du VMD (pas utilisé avec les automates), signalisation spontanée de l'état du VMD.

a) Le service Obtention de l'état du VMD (Status)

Ce service **confirmé** permet à un Client de demander l'état d'un VMD. Le VMD renvoie son état physique et logique ainsi qu'une liste de détails locaux (contenu non défini par la norme ISO 9506 si ce n'est que la longueur du champ d'informations doit être inférieure à 128 bits).

b) Le service Identification du VMD (Identify)

Ce service **confirmé** permet à un Client d'obtenir l'identification du VMD. Le VMD renvoie :

- le nom du fabricant (64 caractères max.),
- le nom du modèle (64 caractères max.),
- le numéro de révision,
- la liste des syntaxes abstraites supportées (optionnelle).

c) Le service Obtention des noms des objets du VMD (GetNameList)

Ce service **confirmé** permet à un Client d'obtenir la liste des objets contenus dans un VMD. Il suffit d'indiquer la classe des objets dont on désire obtenir la liste et leur zone de validité (Object Scope). Si la zone de validité est un domaine, le nom du domaine est bien sûr nécessaire. De plus, on peut indiquer à partir de quel nom on désire obtenir la liste. En effet, le Serveur va renvoyer la liste des noms objets correspondant à la classe et à la zone de validité choisies mais il se peut que la liste soit trop longue pour être envoyée en une fois. Le Serveur indique alors qu'il existe encore plus d'éléments dans la liste (More Follows). Le Client doit relancer la même requête en indiquant de commencer la liste par le dernier nom reçu dans la confirmation de la requête précédente.

d) Le service Obtention de la liste des ressources du VMD (GetCapabilityList)

Ce service **confirmé** permet à un Client d'obtenir la liste des ressources disponibles dans le VMD.

e) Le service Modification de noms d'Objets (Rename)

Ce service **confirmé** permet à un Client de modifier le nom d'un objet d'un VMD (mais pas sa zone de validité ou portée).

f) Le service Signalisation de l'état du VMD (UnsollicitedStatus)

Ce service **non-confirmé** permet à un VMD d'envoyer son état logique et physique et des détails locaux à un Client.

G. CLASSES DE MISE EN OEUVRE

Les classes de mise en oeuvre définissent des sous-ensembles des services MMS qui doivent être mis en oeuvre. Elles sont au nombre de 8.

Elles portent les noms MAP 0, MAP 1, MAP 2, MAP 3, MAP 4, MAP 5, MAP 6, MAP 7.

X. LE RESEAU DE TERRAIN PROFIBUS

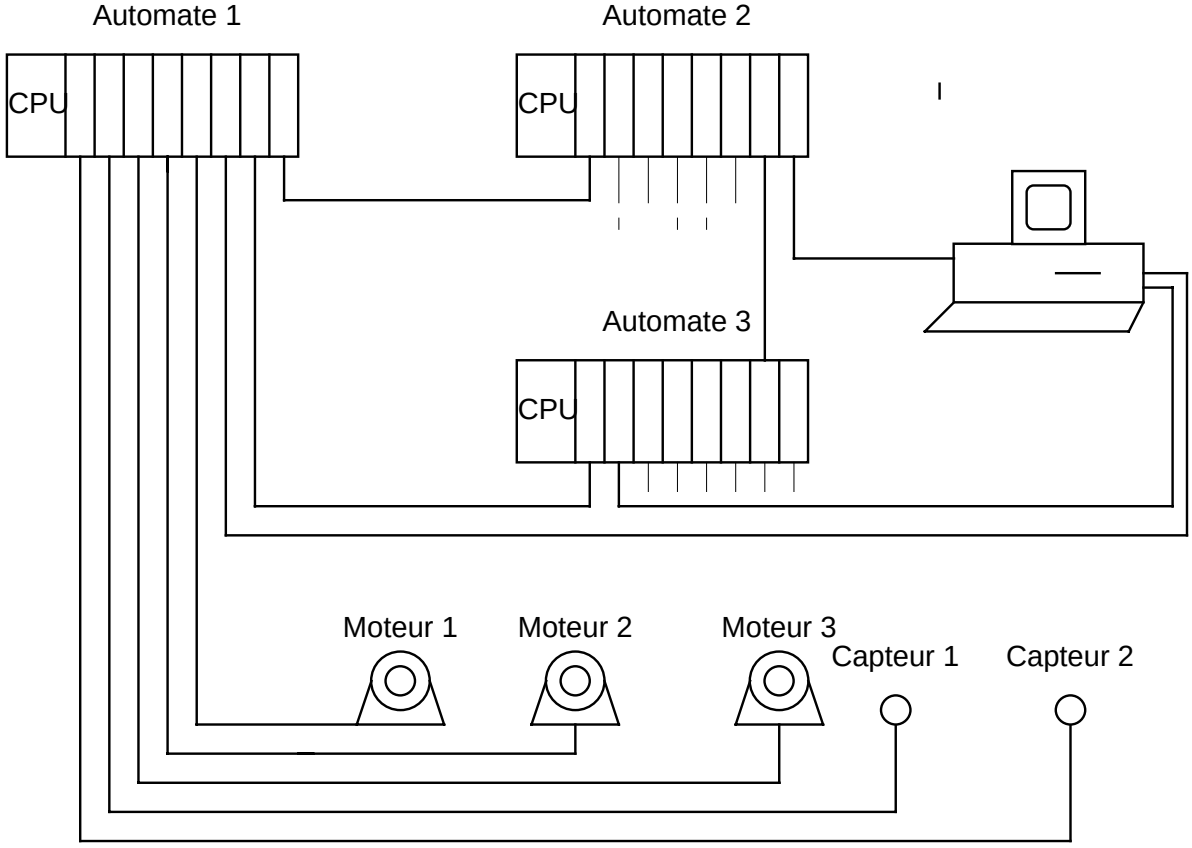
A. Introduction

Jusque récemment, le raccordement des capteurs et des commandes de processus se faisait par la pose de conducteurs électriques entre le capteur ou l'actionneur et les modules d'entrées/sorties du système qui gérait le processus (automates programmables, ...). La communication entre les systèmes gérant le processus (automates programmables, ordinateurs) se faisait aussi par l'intermédiaire de liaisons point à point.

Cette technique de câblage entraînait, entre autres, :

- des longueurs de câbles énormes,

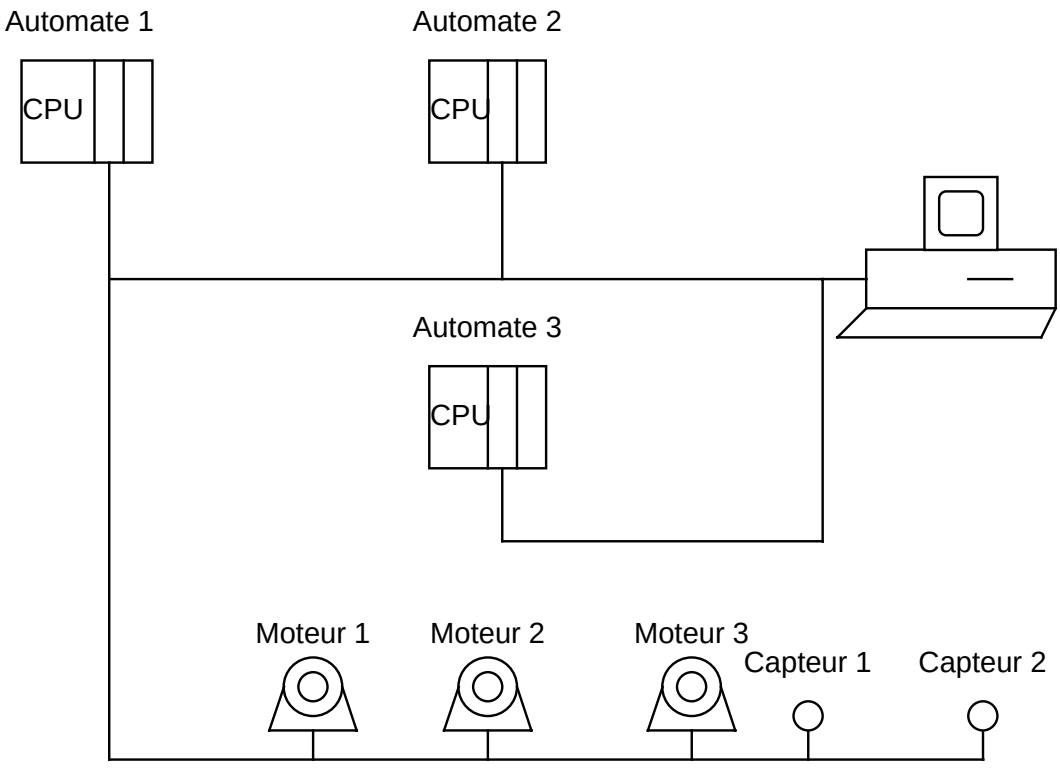
❑ des coûts élevés de pose des câbles,
❑ des problèmes de bruit dans le cas où les capteurs sont éloignés des cartes d'entrées/sorties analogiques,
❑ des problèmes de fiabilité et de maintenance.



Exemple de câblage point à point

L'évolution technologique offre maintenant une solution à ces problèmes. Ainsi, il existe des bus sur lesquels on peut raccorder des capteurs et actionneurs intelligents, des modules d'entrées/sorties déportés, les systèmes de commande (automates programmables, ordinateurs, ...). Le même "câble" est utilisé par l'ensemble des partenaires en communication.

Cette solution conduit, entre autres, à :
❑ une réduction importante des coûts de câbles et de pose (le capteur ou l'actionneur peut être beaucoup plus proche de l'élément à contrôler),
❑ une maintenance simplifiée,
❑ une fiabilité accrue et des fonctionnalités supplémentaires par l'emploi de capteurs et actionneurs "intelligents".



Exemple d'utilisation d'un réseau

Au début de l'apparition des bus/réseaux de terrain, chaque constructeur a développé sa solution propriétaire et donc souvent fermée (support de communication propre et protocole propre à chaque constructeur). Chacun de ces bus de terrain répond très bien à une application spécifique mais est limité à ce type d'application. Or, les utilisateurs n'aspirent eux qu'à obtenir des solutions polyvalentes, non propriétaires et donc ouvertes. Ceci ne peut se faire qu'au travers d'une normalisation. C'est ainsi que sont apparues les deux principales normes pour les réseaux de terrain de type polyvalent :

- Profibus
- FIP.

Profibus est une norme allemande (DIN 19245) alors que FIP est une norme française.

Dans le cadre de ces notes, nous étudierons le réseau de terrain Profibus.

Le réseau Profibus a vu le jour grâce à l'association des utilisateurs de Profibus (PNO - Profibus Nutzer Organisation). PNO comporte les constructeurs (plus de 150), les utilisateurs (bureau d'études, utilisateurs finaux) de matériels de réseau de terrain.

B. Généralités

Actuellement, Profibus propose 2 profils de communication ou piles de protocoles normalisés :

- Profibus FMS (premier profil à être normalisé),
- Profibus DP.

Ces profils de communication utilisent les mêmes protocoles au niveau de l'accès au support de transmission et au niveau de la couche Liaison de données (Data Link Layer) du modèle OSI. Les couches supérieures du modèle OSI ont des contenus différents dans le cas des deux profils.

Profibus FMS est utilisé pour la communication entre matériels hétérogènes "intelligents" au travers des services FMS (Field Message Specification). Ces communications se situent au niveau supérieur (niveau cellule) mais aussi au niveau du terrain. Les services utilisés sont un sous-ensemble des services MMS de MAP 3.0 (optimisés pour l'utilisation dans le cas de réseaux de terrain) et quelques services propres aux réseaux de terrain. Ce profil est défini par DIN 19245 Partie 1 et Partie 2.

Profibus DP est une version optimisée du point de vue performance de la première version de Profibus. Elle sera utilisée pour les communications temps critique entre systèmes d'automatisation et périphériques décentralisés. Elle utilise un sous-ensemble de la norme DIN 19245 Partie 1 et est définie par la norme DIN 19245 Partie 3 pour les couches supérieures à la couche 2.

Comme indiqué précédemment, ces notes étudierons uniquement Profibus FMS.

C. Profibus FMS dans le modèle OSI (Open System Interconnection) de ISO

<table>
<tr><td>Gestion
=
Fielbus
Management
FMA 7</td><td>Couche application
=
Fieldbus Message Specification (FMS)
&
Lower Layer Interface(LLI)</td><td>DIN
19245
Partie 2</td></tr>
<tr><td></td><td colspan="2">Les couches 3 à 6 ne sont pas apparentes de manière spécifiques</td></tr>
<tr><td rowspan="2">Gestion
=
Fieldbus
Management
FMA 1 & 2</td><td>Couche liaison de données
=
Couche Fieldbus Data Link (FDL)</td><td rowspan="2">DIN
19245
Partie 1</td></tr>
<tr><td>Couche Physique</td></tr>
</table>

Le profil de protocole Profibus FMS

Le profil de communication de Profibus suit la structuration en couches du modèle OSI de ISO. On remarque que le Profil distingue les couches 1, 2 et 7 du modèle OSI. Les couches 3 à 6 ne sont pas apparentes en tant que telles car elles sont intégrées dans la couche application (7).

Dans Profibus, la couche Liaison de données est appelée Fieldbus Data Link ou FDL. La couche comporte deux sous-couches :

- ♦ la couche Lower Layer Interface ou LLI,
- ♦ la couche Fieldbus Message Specification ou FMS.

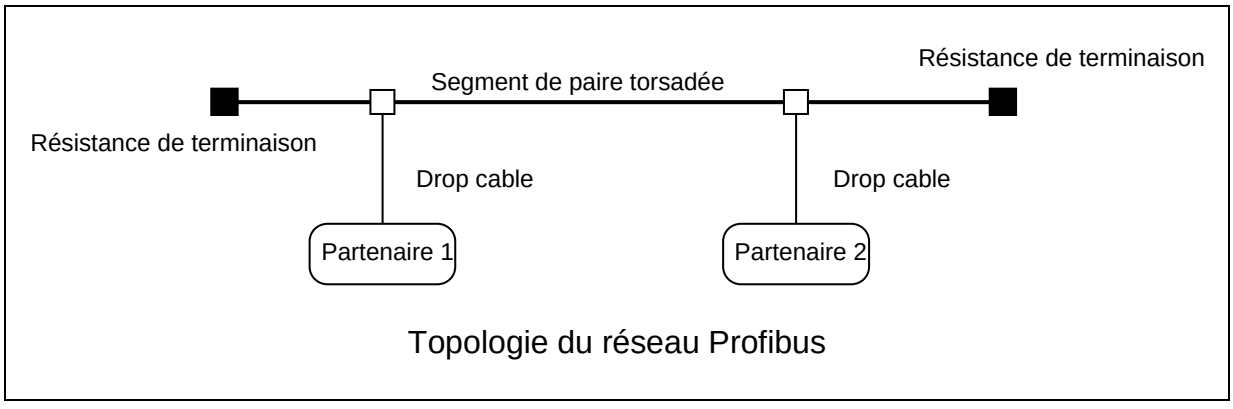
Maintenant, nous allons passer en revue les différentes couches énumérées précédemment en commençant par la couche physique (couche 1).

D. La couche physique de Profibus

Au niveau de la couche physique, les partenaires d'un réseau Profibus peuvent utiliser deux types de supports qui sont des paires torsadées ou de la fibre optique mais emploie toujours a la même technique d'accès au support de transmission. Le choix du type de support physique sera fonction de l'environnement d'utilisation, de l'étendue à couvrir par le réseau et de la vitesse de transmission à assurer.

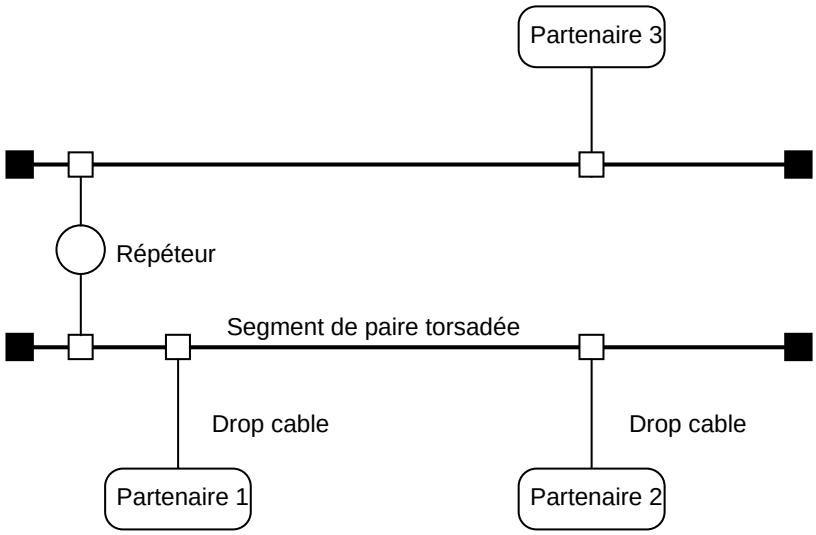
Le **support physique** consistant en des **paires torsadées** suit la norme américaine EIA **RS 485**. Ces paires torsadées peuvent être blindées (**STP** ou **S**hielded **T**wisted **P**air) ou non (**UTP** ou **U**nshielded **T**wisted **P**air). Leur impédance caractéristique doit se situer entre 100 et 130 Ohms. Il existe deux types de câbles (Câble A et Câble B) qui se différencient par la longueur maximale du réseau.

La **topologie du réseau** est de type **Bus Linéaire.** Les partenaires se connectent donc tous en parallèle sur la paire torsadée par l'intermédiaire de d'un câble de liaison. Un réseau complet comporte plusieurs **segments** de paires torsadées. Chaque segment doit disposer de **résistances de terminaison** à chacune de ses extrémités.



Topologie du réseau Profibus

La liaison entre deux segments se fait par l'intermédiaire d'un **répéteur** (amplificateur de ligne bidirectionnel). Deux partenaires quelconques du réseau doivent être séparés par au maximum 3 répéteurs.



Réseau à plusieurs segments

Le nombre **maximum de participants (répéteurs inclus) par segment** est de **32**. Le nombre **maximum de participants (répéteurs inclus) par réseau** est limité à **127**.

Le **codage** du signal se fait par la **technique NRZ**.

Profibus prévoit les **vitesses de transmission** suivantes :
9,6 - **19,2** - **93,75** - **187,5** - **500** - **1500** - **12000** kbit/s.

La **longueur maximale des segments** dépend de la vitesse de transmission choisie et du type de câble (A ou B) :

- Câble A : **200 m à 1500 kbit/s** jusqu'à **1,2 km à 93,75 kbit/s**.
- Câble B : **200 m à 500 kbit/s** jusqu'à **1,2 km à 93,75 kbit/s.**

Elle ne peut dépasser 1,2 km quelle que soit la vitesse.

La distance maximale séparant deux partenaires quelconques du réseau vaut 4 fois la longueur maximale des segments (à 93,75 kbit/s cette distance vaut 4,8 km). En effet, un maximum de 3 répéteurs peuvent se trouver entre deux participants quelconques. Les extrémités de segments sont terminées par des bouchons.

E. Couche Liaison de données de Profibus ou Fieldbus Data Link (FDL)

1. Généralités

Dans le cas des réseaux de terrains (à l'image des réseaux locaux), la couche Liaison de Données est subdivisée en deux sous-couches :

- ♦ la sous-couche MAC ou Medium Access Control,
- ♦ la sous-couche LLC ou Logical Link Control.

Comme son nom l'indique la couche MAC est responsable de l'accès au support de transmission. La couche LLC veille à l'échange de trames entre les différents partenaires. Les trames permettent la transmission de données utilisateurs. Dans le cas de Profibus, ces deux sous-couches sont regroupées dans le couche Fieldbus Data Link.

2. La couche MAC de Profibus

a) Généralités

Les stations s'échangent des trames. Ces trames sont composés de caractères comportant :

- ♦ un bit de départ (0),
- ♦ huit bits de données,
- ♦ un bit de parité paire,
- ♦ un stop bit (1).

La transmission se fait donc en mode sériel et asynchrone, half-duplex puisque nous sommes en présence d'un bus unique.

Toute trame d'appel doit être précédée d'un état de repos durant 33 bits au moins (SYN-TIME). Par contre, il ne peut exister de temps de repos entre les caractères d'un même télégramme ou trame.

Les trames sont protégées au niveau des bits par une distance de Hamming de 4. On sait détecter de manière sûre jusqu'à 3 bits qui changent dans une trame.

Voyons maintenant la technique d'accès utilisée.

La sous-couche MAC (Medium Access Control) utilise une technique hybride d'accès au support de transmission :

♦ le principe de gestion **Token-Passing** pour les stations **actives**,

♦ le principe **maître-esclave** pour la communication entre les stations **actives** et **passives**.

On remarquera que le réseau Profibus comporte deux types de partenaires :

♦ les stations **actives** ou **maîtres** qui sont des composants d'automatisation d'une **complexité certaine** (Automates programmables, Ordinateurs industriels), elles peuvent prendre l'initiative d'échanger des données avec d'autres partenaires;

♦ les stations **passives** ou **esclaves** qui sont des périphériques **plus simples** (Capteurs, actuateurs, commande de moteurs, modules périphériques intelligents). Elles accèdent au réseau uniquement lorsqu'un maître leur a envoyé une demande.

Chaque partenaire dispose d'une adresse unique sur le réseau comprise entre 0 et 126. **Dans le cas d'application en temps critique, le nombre maximum de stations actives limité à 32** sinon le réseau peut comporter 127 stations actives. Dans le cas d'un réseau ne comportant qu'un seul maître, le nombre maximum de stations passives est de 126.

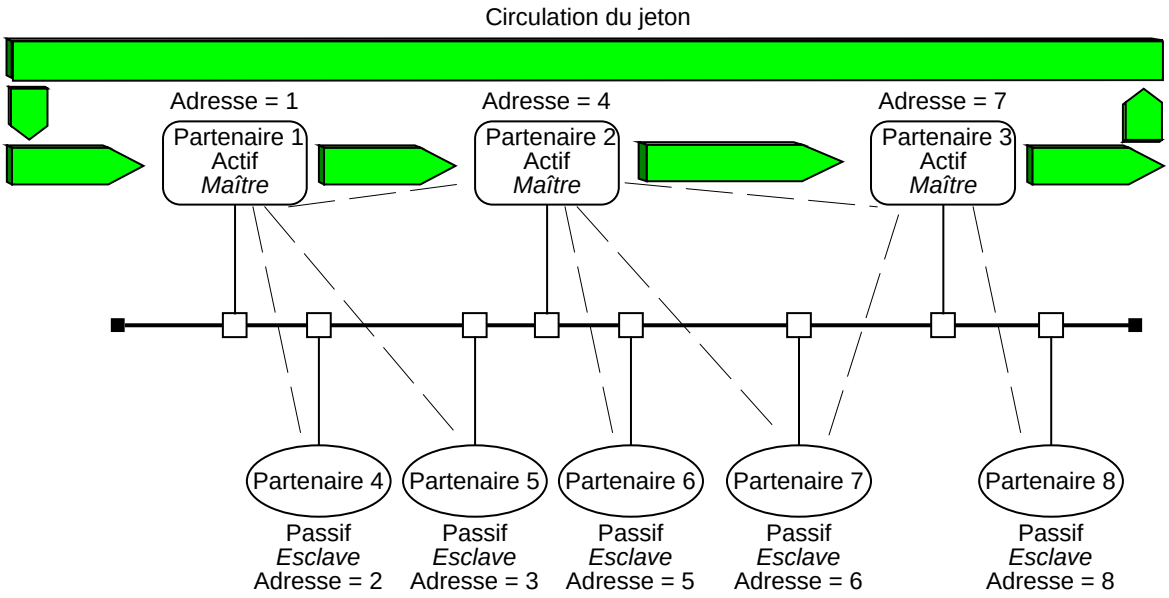
Les maîtres s'échangent le jeton suivant un anneau logique (principe du Token Bus). Lorsqu'un maître dispose du jeton, il a droit à communiquer avec d'autres maîtres ou avec des esclaves. On est donc face à une communication de type :

♦ Maître - Maître

♦ Maître - Esclave

Lorsque le maître a terminé de communiquer avec ses partenaires ou que son temps maximum de rétention du jeton est écoulé, il le passe au maître qui suit dans l'anneau logique.

On retiendra aussi que tous les partenaires sont en parallèle sur le réseau et entendent tous toutes les trames qui passent sur le réseau.



Les Maîtres, les esclaves et le jeton

b) La gestion Token-Passing de Profibus

Le jeton est une trame spéciale qui circule entre les stations actives du réseau. Elle donne le droit à la parole sur le réseau. Elle comporte les **adresses** de :

♦ l'**émetteur** : celui qui envoie le jeton à la prochaine station active dans l'anneau logique,

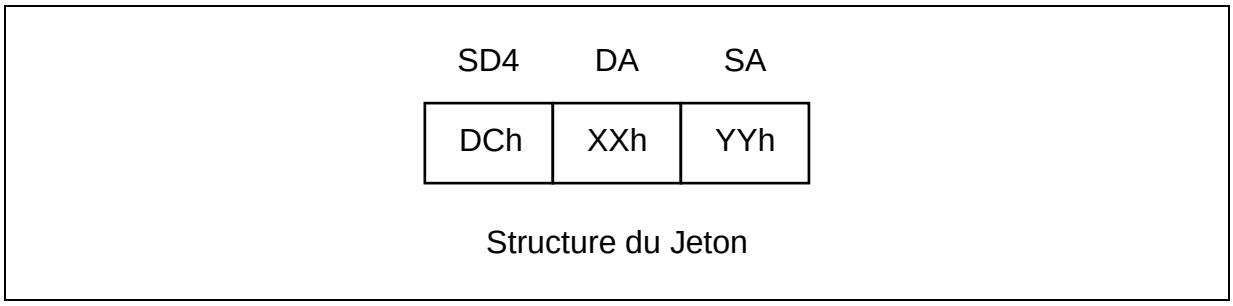
♦ et du **destinataire** : la station active à laquelle est destiné le jeton et qui va donc pouvoir prendre la parole si elle le désire.

Elle comporte trois octets :

♦ le premier octet (Start Delimiter 4) vaut DCh pour indiquer qu'il s'agit d'une trame représentant un jeton,

♦ le second octet (Destination Address) contient l'adresse du destinataire,

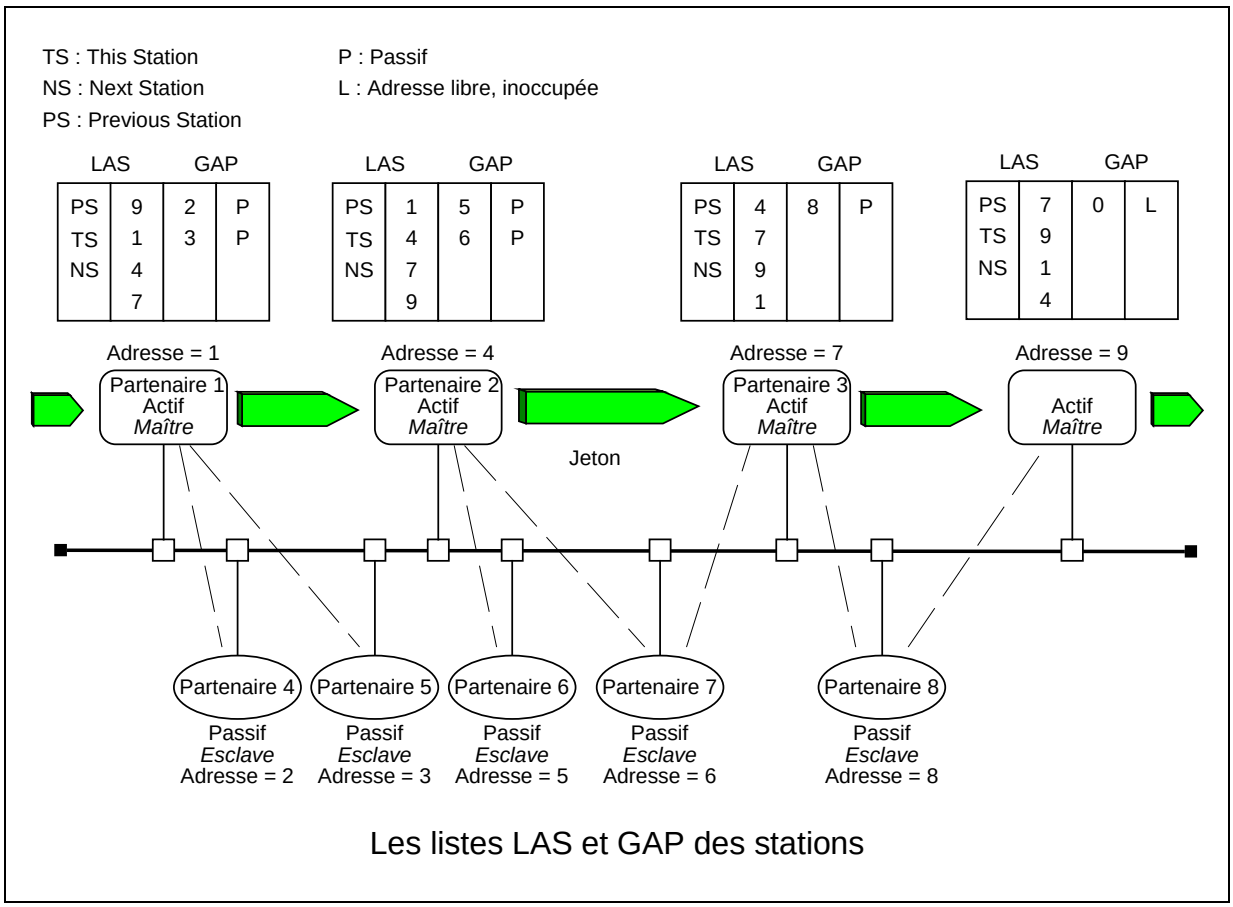
♦ le troisième octet (Source Address) contient l'adresse de l'émetteur.



Structure du Jeton

Comme le jeton comporte les deux octets d'adresse, chaque station connaît la station qui le précède (PS ou Previous Station) et qui le suit (NS ou Next Station) dans l'anneau logique. De plus, chaque station active dispose de deux listes :

- la liste des stations actives (LAS ou List of Active Stations),
- la liste GAP reprend la liste des adresses et l'état des stations associées (Passive, Non prête ou absence de réponse) qui se trouvent entre l'adresse de la station et l'adresse NS.



Les listes LAS et GAP des stations

c) Initialisation de l'anneau logique

Au départ d'un nombre minimum de paramètres de fonctionnement, les partenaires du réseau parviennent à initialiser le réseau.

Nous savons que chaque station dispose d'une adresse qu'elle connaît (TS ou This Station). De plus, les stations actives connaissent l'adresse la plus élevée utilisée dans le réseau (paramètre HSA ou Highest Station Address). Lors de l'initialisation de l'anneau logique (mise sous tension des partenaires du réseau), dans un premier temps, il n'y aura pas d'activité sur le bus. Lors de la mise sous tension des partenaires actifs, dans chacun d'entre eux une temporisation d'inactivité est démarrée (T_{TO} ou Time Out). La durée de cette temporisation est directement proportionnelle à l'adresse de la station. Ainsi, chez la station qui a l'adresse la plus basse, cette temporisation sera écoulée en premier lieu. Si toute activité est toujours absente sur le bus, cette station va initialiser l'anneau logique (état *Claim-Token*). Pour ce faire, elle va :

♦ **envoyer le jeton deux fois à elle-même** : les autres stations actives (dont la temporisation n'est pas encore écoulée) voient ainsi que l'on est dans la **phase d'écoute (initialisation de l'anneau)**. Toutes les stations continuent donc à écouter tous les messages passant sur le bus (état Listen-Token) afin d'établir leur liste de stations actives (LAS ou List of Active Stations).

♦ Elle envoie ensuite une requête de demande d'état de la FDL (FDL-status Request) à la station dont l'adresse suit directement la sienne. Deux cas se présentent :

- Si celle-ci répond par "*Pas Prête*", "*Station Passive*" ou ne répond pas, cette adresse ainsi que son état sont inscrites dans la liste GAP (intervalle d'adresse séparant TS et NS) de l'émetteur du jeton. L'émetteur du jeton envoie une requête d'état de la FDL à la prochaine station et réagit de la manière décrite précédemment.
- Si celle-ci répond par "*Prête pour le jeton*", cette dernière prend possession du jeton (les autres stations actives mettent leur LAS à jour, la station qui a envoyé la requête FDL -Status indique dans sa LAS que la station qui a pris le jeton est sa Next Station NS et ferme sa liste GAP tandis que la station qui a pris le jeton indique dans la LAS que la station qui lui a donné le jeton est sa Previous Station PS). La station qui détient le jeton maintenant, envoie une requête de demande d'état de la FDL à la station dont l'adresse suit directement la sienne et réagit de la même manière que la première station qui a envoyé le jeton.

Comme les stations disposent du paramètre HSA (adresse la plus élevée utilisée

dans le réseau), la dernière station active sait quand elle doit renvoyer le jeton à la station d'adresse la plus basse (0). De plus, les LAS des stations actives sont considérées comme bonnes lorsque la LAS n'a plus changé pendant deux tours de jetons

d) Ajout de stations

Comme indiqué dans la partie concernant l'initialisation de l'anneau logique, chaque station active du réseau dispose d'une liste GAP qui contient les adresses des stations qui occupent les adresses comprises entre elle-même (TS) et sa prochaine station active (NS). Chaque station vérifie après un intervalle de temps défini T_{GUD} ou GAP Update Time si sa GAP list est toujours la même ou si une nouvelle station active s'est insérée dans l'intervalle d'adresse qui le sépare de sa station NS. Pour ce faire, il envoie des requêtes FDL-Status. Si une adresse passe à l'état actif, la LAS de tous les participants est mise à jour et le jeton est passé à la nouvelle station active qui dispose déjà de sa LAS et de sa liste GAP ainsi que de l'adresse de sa NS.

e) Suppression ou défaillance de stations

Si une station active est supprimée ou est défaillante, sa station PS essaie de lui envoyer son jeton. Cette dernière remarque que le bus reste inactif, Après une certaine durée (T_{Slot} ou Slot-Time), la PS renvoie le jeton une deuxième fois. Si le bus reste toujours inactif pendant une durée T_{Slot}, la PS renvoie le jeton une troisième fois et si le bus reste toujours inactif pendant une durée T_{Slot} , PS envoie le jeton deux fois vers la prochaine station active de sa LAS, actualise sa LAS et sa liste GAP. En recevant deux fois le jeton de PS, la prochaine station active sait qu'elle doit actualiser sa LAS et y indiquer PS comme la station qui la précède dans l'anneau.

f) Passage de jeton

La station qui dispose du jeton envoie ce dernier à sa station NS lorsqu'il a terminé ses communications ou que son temps de communication est écoulé. Pendant la durée $T_{Sl,}$ la station qui a envoyé le jeton écoute le bus pour identifier une trame correcte émanent de sa station NS. Si c'est le cas, elle suppose que sa station NS a bien reçu le jeton

g) Aspect temporel de la gestion du réseau Profibus

Comme indiqué précédemment, le jeton circule selon un anneau logique entre les différentes stations du réseau : la station qui dispose du jeton à un moment donné a le droit d'effectuer ses échanges de données avec les autres stations. Cependant les stations ne peuvent pas conserver le jeton une durée illimitée privant ainsi les autres stations de leurs échanges de données.

Voyons d'abord la définition d'une série de temps utilisée par la couche Fieldbus Data Link. **Les temps *T* dont la liste va suivre sont mesurés en Bits. Ainsi un temps *t* (exprimé en seconde) doit être divisé par la durée d'un bit t_{BIT}.**

Durée d'un bit **t_{BIT}** ou **Bit Time** : t_{BIT} représente la durée utilisée pour émettre un bit. C'est donc ici l'inverse de la vitesse de transmission.

$$t_{BIT} = 1 / \text{vitesse de transmission (en bit/s)}$$

Station Delay Time **T_{SDx}** : ce temps représente la durée de traitement du protocole. Il consiste en l'intervalle de temps s'écoulant entre l'envoi ou la réception du dernier bit d'une trame et la réception ou l'envoi du premier bit de la trame suivante. Trois Station Delays sont définis :

- Station Delay Time de l'Initiateur **T_{SDI}** (la station qui envoie son jeton ou une requête)

$$T_{SDI} = t_{SDI} / t_{BIT}$$

- Minimum Station Delay Time du Répondeur **min T_{SDR}** : durée minimale s'écoulant avant qu'une station envoie un acquittement ou une réponse.

$$\min T_{SDR} = \min t_{SDR} / t_{BIT}$$

- Maximum Station Delay du Répondeur **max T_{SDR}** : durée maximale s'écoulant avant qu'une station envoie un acquittement ou une réponse.

$$\max T_{SDR} = \max t_{SDR} / t_{BIT}$$

• GAP Update Time : ce temps représente la durée après laquelle les stations actives actualisent leur liste GAP. Après l'écoulement de cette durée, la liste sera actualisé dès que la station disposera du jeton et qu'il lui reste du temps de communication après ses échanges de données. Si ce n'est pas le cas, on tentera l'opération lors du prochain passage du jeton dans la station.

$$T_{GUD} = G \cdot T_{TR} \quad \text{avec } 1 \leq G \leq 100$$

• Target RotationTime T_{TR} : Profibus définit un temps de rotation maximum du jeton T_{TR} (Target Rotation Time) qui doit permettre d'assurer l'aspect temps critique du réseau. Ce système permet d'assurer que chaque station a au moins pu échanger des données une fois endéans le temps T_{TR}.

Les stations disposent d'une temporisation qui leur indique le temps réel de rotation du jeton T_{RR} (Real Rotation Time).

Lors de la prise de possession du jeton, la station enclenche une temporisation qui s'incrémente jusqu'à la prochaine prise de possession du jeton (cette temporisation indique donc le T_{RR}) et elle calcule la différence entre le T_{TR} et le T_{RR}. Cette différence représente la durée de parole maximale à laquelle la station a droit pour effectuer ses échanges de données avec les autres stations. Si TRR est TTR lors de la prise de possession du jeton alors seule une requête de haute priorité peut être exécutée. Si TRR dépasse encours d'exécution d'une requête, celle-ci est totalement menée à bien : on attend la trame de réponse ou d'acquittement même si on dépasse TTR. Lors du prochain passage du jeton, la station aura une durée de droit de parole réduite en conséquence.

Pour maintenir T_{TR} à une valeur faible, on veillera à utiliser les cycles de messages à haute priorité uniquement pour des événements qui surviennent rarement ou qui sont très importants. De plus, leur longueur sera fortement limitée (données ≤ 20 octets).

3. La couche LLC ou Logical Link Control de FDL

La sous-couche LLC de la couche FDL propose les services suivants:

♦ Services cycliques
- Emission de données cycliques avec demande de données et renvoi de données (CSRD ou Cyclic Send and Request Data with Reply)

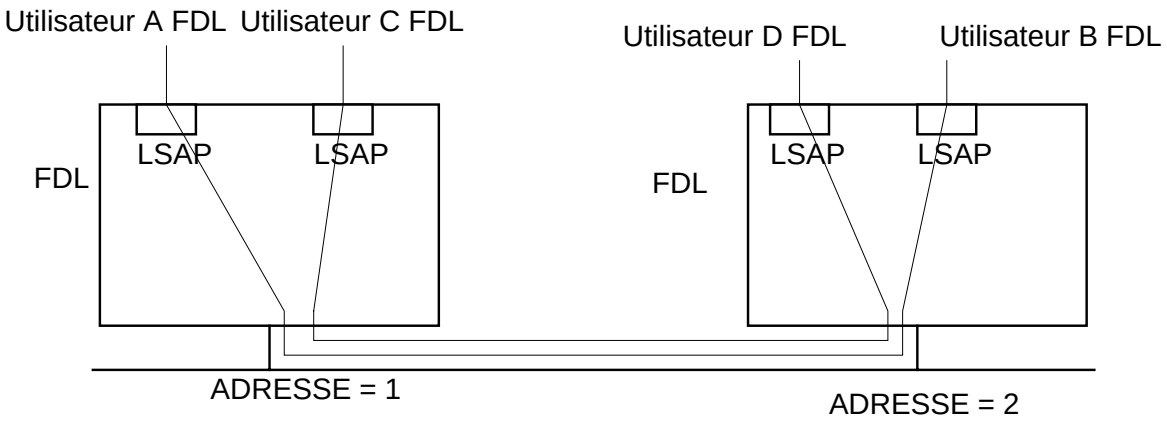
♦ Services acycliques
- Emission de données sans accusé de réception (SDN ou Send Data with No Acknowledgement)
- Emission de données avec accusé de réception (SDA ou Send Data with Acknowledgement)
- Emission de données avec demande de données et renvoi de données (SRD ou Send and Request Data with Reply)

Cette couche offre la possibilité de faire du multicast (envoi d'une trame requête non confirmée à un groupe de stations) et du broadcast (envoi d'une trame requête non confirmée à toutes les stations).

Les utilisateurs de la couche FDL y accèdent par un LSAP (Link Service Access Point). Pour pouvoir communiquer, les utilisateurs de la couche FDL doivent connaître leurs adresses sur le bus et en plus leurs adresses LSAP.

Lorsque les trames de l'expéditeur ne comportent pas l'adresse du LSAP destination, les données sont envoyées vers un LSAP par défaut (**Default-LSAP**). Dans des situations où le temps de réponse est primordial, les trames sont réduites à leur longueur minimale (les LSAP sont omis) et les données contenues dans celles-ci sont à ce moment transmises à l'utilisateur FDL par l'intermédiaire du LSAP par défaut.

Dans le cas du service cyclique CSRD, la Poll-liste contient la liste des adresses et des LSAP des partenaires vers lesquels il faut échanger des données de manière cyclique. Cette liste est rattachée à un LSAP local (Poll-list-LSAP).



Les utilisateurs de la couche FDL communiquent au travers de LSAP (Link Service Access Point)

Dans notre cas, l'utilisateur de la couche FDL est la couche LLI et FMS.

La couche FDL prévoit quatre formats différents de trame :

♦ Trame avec champ d'information fixe sans données

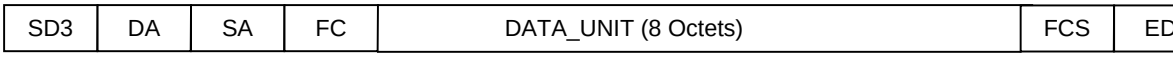
Requête et Réponse

SD1	DA	SA	FC	FCS	ED

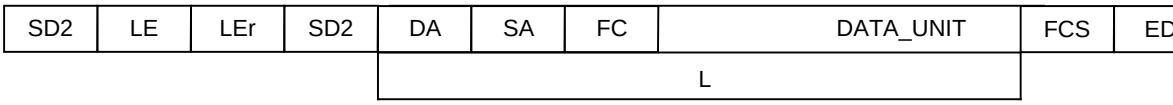
Acquittement court

SC

Trame avec champ d'information fixe comportant des données (Requête et Réponse)



Trame avec champ d'information de longueur variable (Requête et Réponse)



Trame jeton

Toutes les trames en dehors de la trame d'acquittement court (1 octet) comportent :

- ♦ un octet indiquant le format de trame (SD1 à SD4),
- ♦ un octet comportant l'adresse destination (DA) (de 0 à 127(broadcast)),
- ♦ un octet comportant l'adresse source (SA) (de 0 à 126)

Chaque adresse peut comporter jusqu'à deux octets d'extension (si bit 8 de SA/DA est à un). Ces octets d'extension, quand ils existent, sont les premiers octets de la partie DATA_UNIT.

Le premier octet d'extension comporte :

- ♦ une adresse LSAP si bit 7 de l'octet à 0 (SLSAP : 0 à 62, DSLSAP : 0 à 63)
- ♦ une adresse Région/Segment pour bridge si bit 7 de l'octet à 1

De plus, si le bit 8 de cet octet est à 1, il existe un second octet d'extension qui suit directement le premier octet d'extension et contient l'adresse LSAP (les bits 8 et 7 de ce dernier octet sont à 0).

Les trames autres que le jeton comporte en plus :

- ♦ un octet indiquant le **type de service** demandé ou le type de réponse (**FC**)
- ♦ un **octet de contrôle** (**FCS**): il s'agit de la somme arithmétique des octets de la

trame en dehors du premier octet SD et du dernier octet ED.

♦ Un **champ de données** (**DATA_UNIT**) dont la longueur peut être comprise entre 1 et 246 octets si on n'utilise pas d'extension d'adresse. Par contre, si les extensions d'adresse sont utilisées la longueur peut varier entre 1 et 242.

♦ Cette longueur est données par les deux octets **LE** et **LEr** (copie de LE).

Les couches 7 des partenaires Profibus vont donc utiliser les LSAP de la couche 2 pour communiquer.

F. La couche 7 de Profibus ou LLI et FMS

1. Généralités

Comme indiqué précédemment, la couche 7 de Profibus se compose de deux sous-couches :

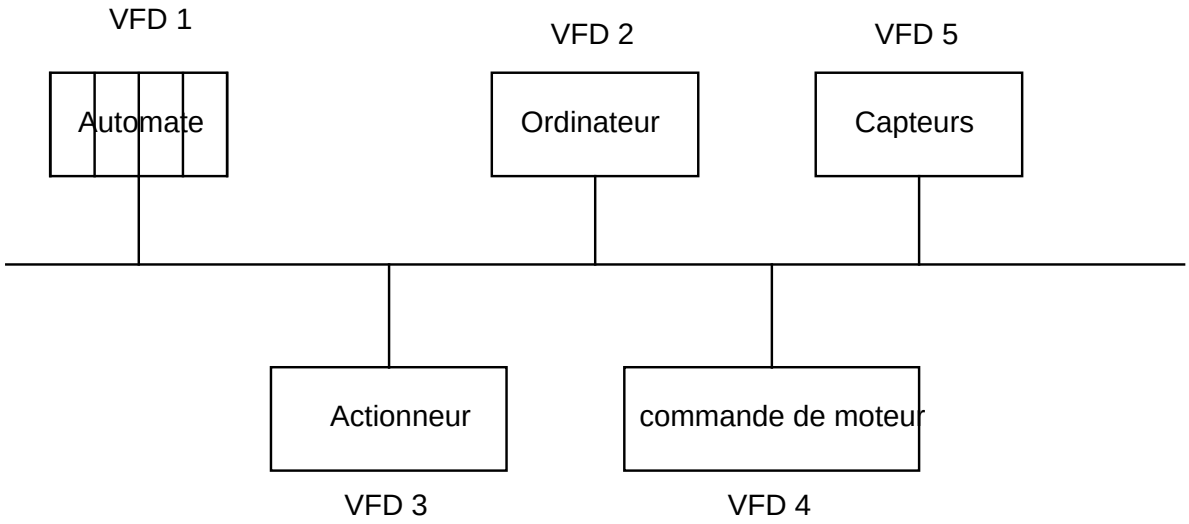
♦ la sous-couche FMS ou Fieldbus Message Specification qui définit les objets de communication, les services disponibles pour gérer ces objets et les modèles utilisés pour la communication (vu du réseau).

♦ la sous-couche LLI ou Lower Layer Interface qui permet d'adapter les fonctions de la couche Application aux caractéristiques de la couche 2.

2. La couche FMS

a) Le VFD

La couche FMS permet de transformer l'équipement réel de réseau de terrain en un **équipement virtuel de réseau de terrain** ou **VFD** (Virtual Field Device).



Vu du réseau, tous les équipements sont des VFD dont les caractéristiques sont définies par FMS

Vu du réseau, tous les équipements ont le même aspect, celui d'un VFD qui comporte différents objets (correspondant à des éléments réels de l'équipement de réseau de terrain). La couche FMS fournit un ensemble de services qui permettent de gérer la communication avec les VFD et modifier les VFD et leurs objets associés. Habituellement, on compte un VFD par système réel mais il pourrait en comporter plusieurs.

La communication entre les différents VFD se fait au travers des relations de communication ("connexion"). L'ensemble des relations de communications définies pour le système réel se trouvent dans la FMS-KBL.

La liste des objets présents dans chaque VFD à un moment donné est contenue dans son dictionnaire local des Objets (OD ou Object Dictionary). Ce dictionnaire peut être prédéfini dans le cas d'équipements simples. Les équipements plus complexes permettent de laisser modifier le contenu du dictionnaire par l'intermédiaire des services FMS (création de variables, d'instances de programme).

Le dictionnaire d'objets se compose de :

- Une entête qui contient des informations sur la structure du dictionnaire,
- La liste des types statiques (type de données : entier, virgule flottante, ...),
- Le dictionnaire des Objets Statiques,
- La liste dynamiques des liste de variables,
- La liste dynamique des Instances de Programme.

Sont mise en oeuvre uniquement les parties de l'OD que l'équipement supporte.

Les objets statiques sont définis par le constructeur ou lors de la configuration du réseau. Profibus prévoit les objets suivants :

- les variables simples (*Simple Variable*),
- les tableaux (*Array*),
- les structures de variables simples de types différents (*Record*),
- les domaines (*Domain*),
- les événements (*Event*).

Les objets dynamiques peuvent être prédéfinis lors de la configuration du réseau ou être créé, effacé, modifié dynamiquement au travers des services FMS. Profibus prévoit les objets dynamiques suivants :

- les Instances de Programme (*Program Instances*)
- les Variables Listes (*Variable List*) : séquence de variables simples, de structures ou de tableaux.

Les informations décrivant ces différents Objets du VFD sont reprises dans le dictionnaire de Objets (OD) du VFD.

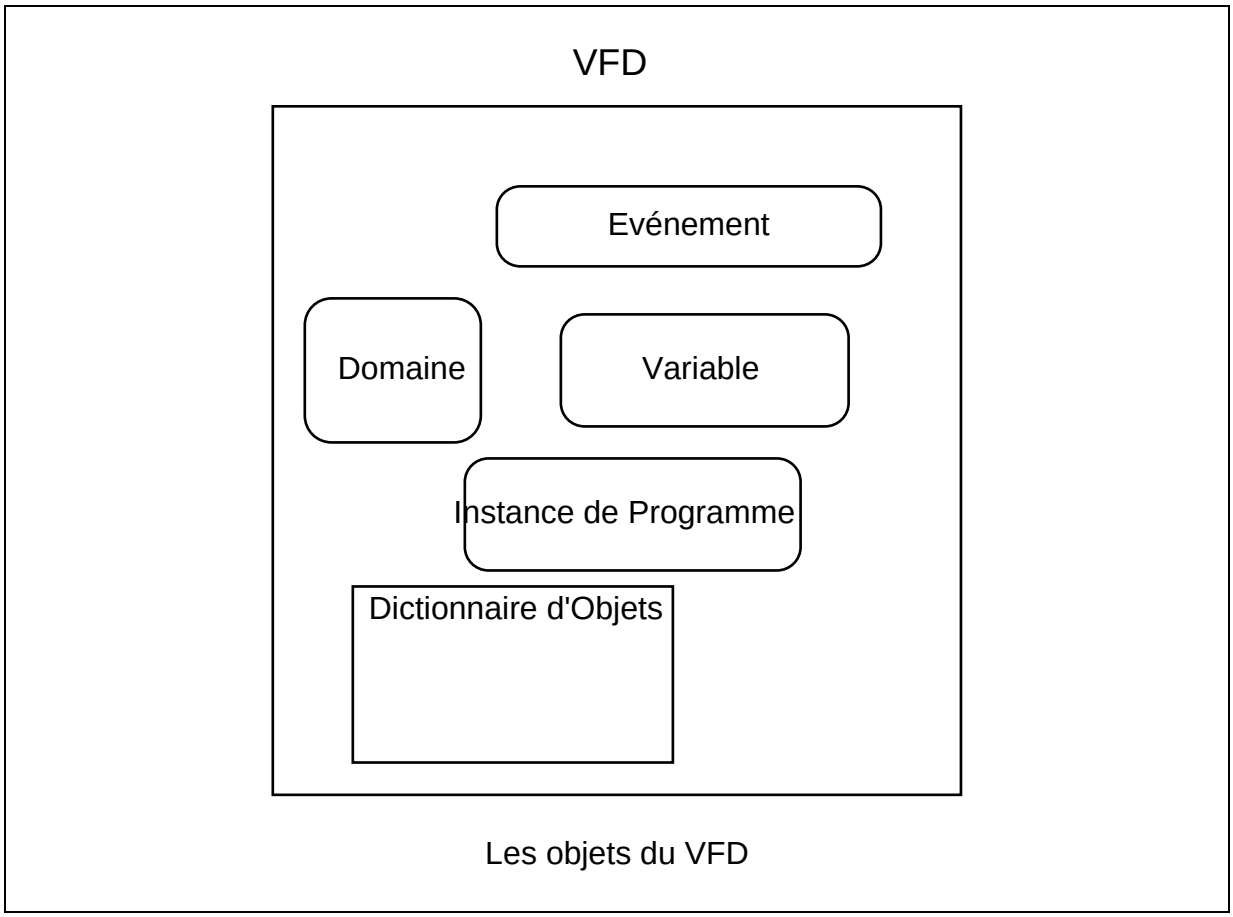
FMS prévoit la possibilité de protéger l'accès aux différents objets. Seules les utilisateurs autorisés auront accès aux objets. L'utilisation de la protection d'accès ou non est indiqué dans l'entête de la FMS-KBL (FMS-KBL-Header). La protection d'accès prévoit des droits d'accès pour les utilisateurs quelconques, des droits d'accès pour les utilisateurs appartenant à un groupe d'accès, des droits d'accès pour les utilisateurs disposant d'un mot de passe. Huit groupes d'accès sont prévus.

Au niveau des objets, on peut prévoir un mot de passe (Attribut : Password), les groupes qui peuvent accéder à l'objet (Attribut : Access-Groups) et définir les droits dont jouit l'utilisateur qui dispose du mot de passe, qui fait partie d'un groupe d'accès, qui est un utilisateur quelconque (Attribut:Access-Rights). Les droits disponibles sont du type lecture, écriture, droit d'exécution, effacement.

Objet	Droits
Domaine	Upload, Download, création IP
Instance de Programme	Start, Stop, Effacement
Simple-Variable	lecture, écriture
Array	lecture, écriture
Record	lecture, écriture
Variable-List	lecture, écriture, effacement
Event	alter, acknowledge

Les paramètres Access-Protection-Supported, Password, Access-Groups sont échangées entre les deux partenaires en communication lors l'établissement de la communication (service Initiate).

Si la protection d'accès n'est pas gérée, les attributs décrits ci-dessus n'existent pas au niveau de chaque objet.



(1) Contenu du dictionnaire d'objets

Data Type Dictionary

Index	**Object-Code**	**Meaning**
1	Data Type	Integer8
2	Data Type	Integer16
....	...	...
6	Data Type	Floating Point

Static Object Dictionary

Index	**Object-Code**	**Data Type**	**Internal Address**	**Symbol**

20	VAR	1	1234H	Pression
21	VAR	6	2000H	Température
22	VAR	2	500H	Niveau

La correspondance des objets du VFD avec les ressources, les éléments de l'équipement réel de réseau de terrain est assurée par la couche ALI (Application Layer Interface). Dans l'OD, on trouve pour chaque objet une adresse logique (Index), un code reprenant le type d'objet, un code indiquant un attribut de l'objet, l'adresse interne de l'objet (correspondance avec l'objet réel), son nom (symbol)

b) Le Modèle Client-Serveur

Comme nous sommes au niveau de la couche 7 du modèle, les utilisateurs de la couche FMS (partenaires de la communication) sont appelés des Processus Applications.

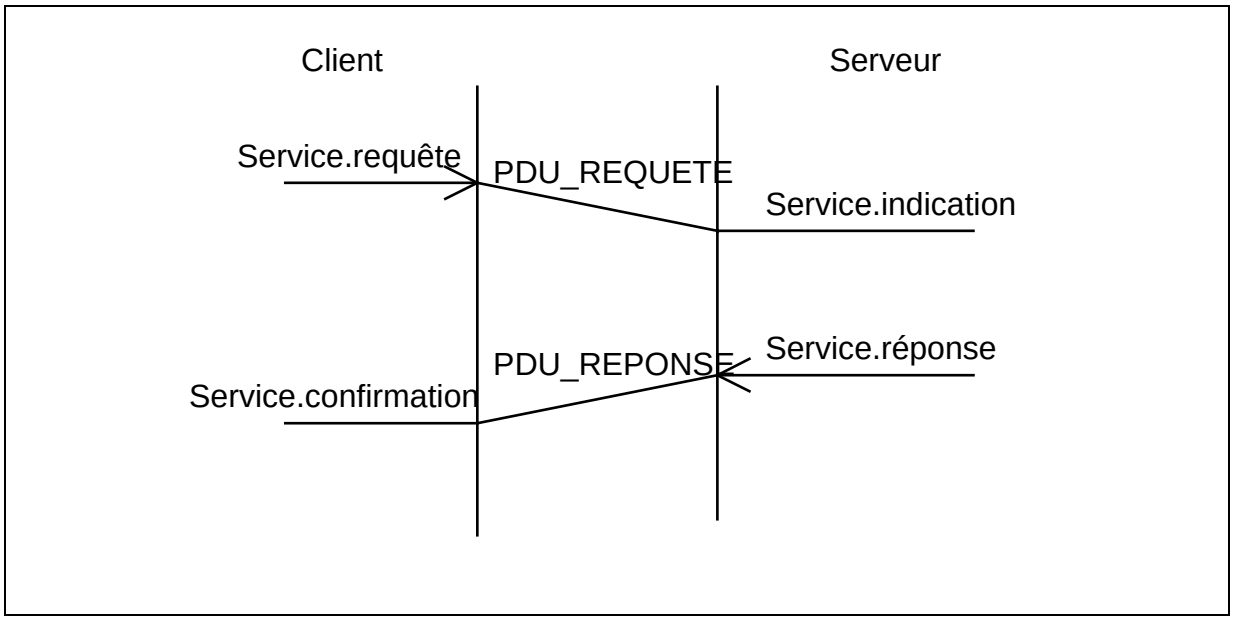
FMS repose sur des communications de type Client -Serveur.
Le **Client** est le Processus Application qui utilise les fonctionnalités d'un partenaire (Processus Application - ici VFD) distant au travers des services FMS.
Le **Serveur** est le partenaire qui met les fonctionnalités de son VFD à la disposition d'un Client.

Il est clair qu'un Processus Application peut bien sûr être Client et Serveur en même temps

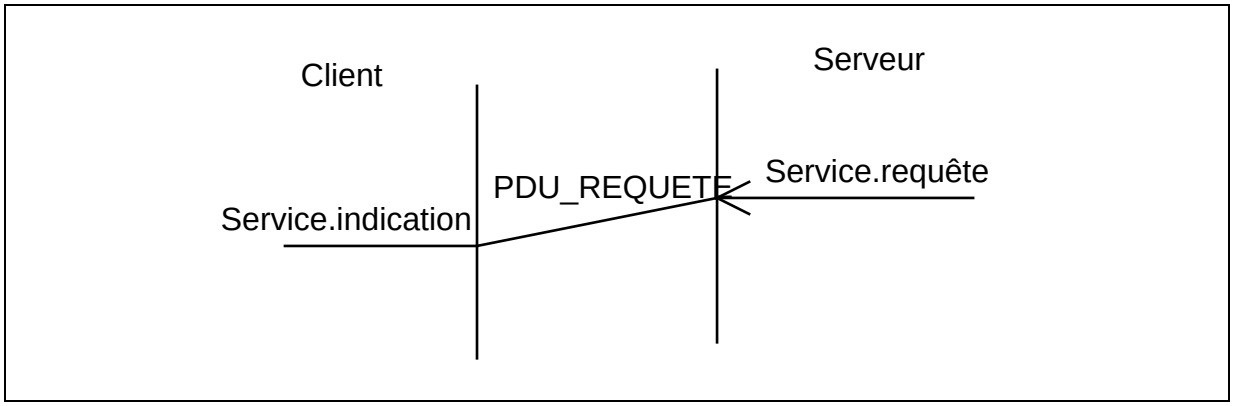
c) Relation entre le type de services FMS et le modèle Client -Serveur

Les services FMS peuvent être de 2 types :

♦ Confirmé : le Client envoie une requête que le Serveur doit confirmer.



Non Confirmé : le Serveur envoie une requête que le Client ne doit pas confirmer.



d) Les Objets prévus par FMS

(1) L'Objet VFD

Celui-ci comporte 4 attributs:

Vendor-Name : nom du constructeur

- **Model Name** : nom du modèle
- **Revision** : numéro de version
- **Profile-Number** : chaîne de deux octets contenant l'identification du Profile

- **Logical Status** : état logique. Le VFD peut prendre 4 états logiques différents:

- *Prêt à communiquer* (valeur 0). Tous les services sont disponibles.
- *Services limités* (valeur 2). Seuls, les services Initiate, Abort, Reject, Identify,

Status, Get-OV sont supportés par le Serveur.
• *OV-Loading-Non-Interacting* (Valeur 4). Dans cet état, le service Initiate-Put-OV n'est pas permis.
• *OV-Loading-Interacting* (Valeur 5). Toutes les connexions sont bloquées en dehors de la connexion par laquelle est arrivé l'indication du service Initiate-Put-OV. Les nouvelles demandes de connexions sont rejetées. Seuls, les services Initiate, Abort, Reject, Identify, Status, Phys-Read, Phys-Write, Initiate-Put-OV, Put-OV, Terminate-Put-OV, Get-OV sont supportés par le Serveur.

♦**Physical Status** : état physique . Le VFD peut avoir 4 états physique différents :
• *Opérationnel* (valeur 0)
• *Partiellement Opérationnel* (Valeur 1)
• *Inopérationnel* (Valeur 2)
• *Demande maintenance* (valeur 3)

♦ **List Of VFD-Specific Objects** : liste des objets spécifiques au VFD.

(2) L'objet Dictionnaire d'Objets (OD ou OV en allemand)

Comme indiqué précédemment l'OD comporte la description des objets statiques et dynamiques voyons le maintenant plus en détail.

Index	Contenu
0	**Description du dictionnaire d'Objets lui-même**
1 ... c	**Liste des Types de données statiques** *Objets type de données définis par Profibus* (commence toujours à l'index 1)
d ... i	*Objets Types de données et description de la structure de types de données* (suit toujours directement la liste des types définis par Profibus)
k ... n	**Liste des Objets statiques**

p **Liste des variables listes dynamiques**
...
t

v **Liste des Instances de Programme dynamiques**
... *Instances de Programme prédéfinies*
w
x
... *Instances de Programme dynamiques*
z

Une entrée dans le dictionnaire des objets a toujours la structure suivante :

- une adresse logique (Index),
- un code reprenant le type d'objet,
- d'autres attributs de l'objet,
- l'adresse interne de l'objet,
- son nom (optionnel),
- une extension.

L'index 0 correspond à la description de l'objet dictionnaire des objets (OD).

(3) L'Objet FMS-KBL

Cet objet contient la liste des relations de communication (Connexion) (KBL=**K**ommunikations**b**eziehungs**l**iste). Cette liste contient la description spécifique à FMS de toutes les relations de communication (connexion) d'un système, indépendamment de l'instant d'utilisation.

La KBL est composée de lignes. Chaque ligne peut être adressée par l'intermédiaire de l'index Référence de communication (KR ou Kommunikationsreferenz).Chaque ligne comporte une partie de statique et une partie dynamique. Elle contient donc toutes les informations spécifiques à FMS pour la connexion en question.

Dans le FMS-KBL-Header (la première ligne (KR=0)) se trouvent les informations concernant la construction des entrées (lignes) de la KBL.

FMS-KBL-Header

Cet entête de la FMS KBL contient quatre champs :

- **Key** vaut 0 car entête de FMS KBL
- **Nombre d'entrées dans la FMS KBL**
- **Symbol-Length** indique la longueur de symboles repris dans la KBL (0 à 32). Quand zéro est indiqué, on utilise pas de symboles.
- **VFD-Pointer-Supported** indique si dans la KBL plusieurs VFD sont supportés.

Key FMS-KBL-Header

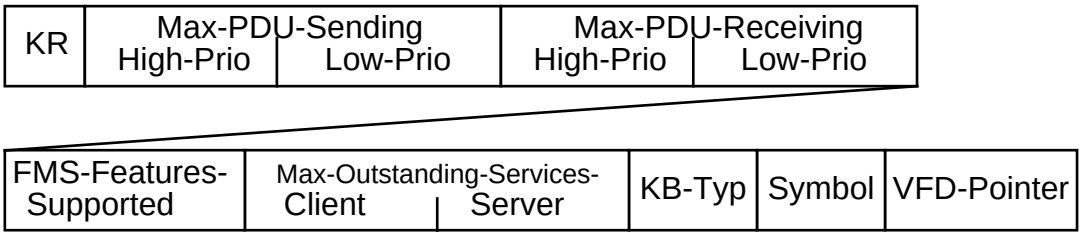
0	Nombre d'entrées dans la FMS KBL	Symbol-Length	VFD-Pointer-Supported

Partie Statique d'une entrée de la KBL

La partie statique des entrées de la KBL comportent 11 champs :

- **KR** : contient la référence de la connexion. Elle doit être unique au sein de la KBL.
- **Max-PDU-Sending High-Prio** : indique la taille maximale en octets des FMS-PDU de haute priorité qui peuvent être envoyés et traités sur cette connexion.
- **Max-PDU-Sending Low-Prio** : indique la taille maximale en octets des FMS-PDU de priorité basse qui peuvent être envoyés et traités sur cette connexion.
- **Max-PDU-Receiving High-Prio** : indique la taille maximale en octets des FMS-PDU de haute priorité qui peuvent être reçus et traités sur cette connexion.
- **Max-PDU-Receiving Low-Prio** : indique la taille maximale en octets des FMS-PDU de priorité basse qui peuvent être reçus et traités sur cette connexion.
- **FMS-Features-Supported** : indique les services FMS supportés en tant que Client et Serveur sur cette connexion.
- **Max-Outstanding-Services-Client** : indique le nombre de services confirmés peuvent rester en attente.
- **Max-Outstanding-Services-Server** : indique le nombre de services confirmés en attente qui peuvent être traité par le Serveur sur cette connexion.
- **KB-Typ** : contient le type de communication. Si true alors communication orientée connexion sinon communication connectionless.
- **Symbol** : Nom symbolique pour la référence de la connexion. Son existence et sa longueur sont définies dans le FMS-KBL-Header.

♦ **VFD-Pointer** : ce pointeur renvoie vers le VFD auquel est associée cette connexion.



Partie dynamique d'une entrée de la KBL

La partie dynamique des entrées de la FMS-KBL comprend 3 champs :

♦ **Outstanding-Services-Counter-Client** : indique le nombre de services confirmés (en tant que Client) en attente sur cette connexion (OSCC)

♦ **Outstanding-Services-Counter-Server** : indique le nombre de services confirmés (en tant que Serveur) en cours de traitement sur cette connexion (OSCS)

♦ **KB-State** : indique l'état actuel de la connexion. 6 états sont possibles :

pour les communications orientées connexion

- CONNECTION-NOT-ESTABLISHED
- CONNECTION-ESTABLISHING (CALLING)
- CONNECTION-ESTABLISHING (CALLED)
- CONNECTION-ESTABLISHED

Pour les communications en mode connectionless

- CONNECTIONLESS-CLIENT
- CONNECTIONLESS-SERVER

(4) L'Objet Domaine

L'objet domaine représente une zone de mémoire qui peut contenir des données ou du code de programme.

Cet Objet comporte les attributs suivants :

Index : c'est l'index associé au domaine dans l'OD.

♦ **Domaine-Name** : le nom du domaine. Son existence et sa longueur maximale sont définies par l'attribut Name-Lentgh de l'Objet OD.

♦ **Max-Octets** : nombre maximum d'octets disponible pour le domaine.

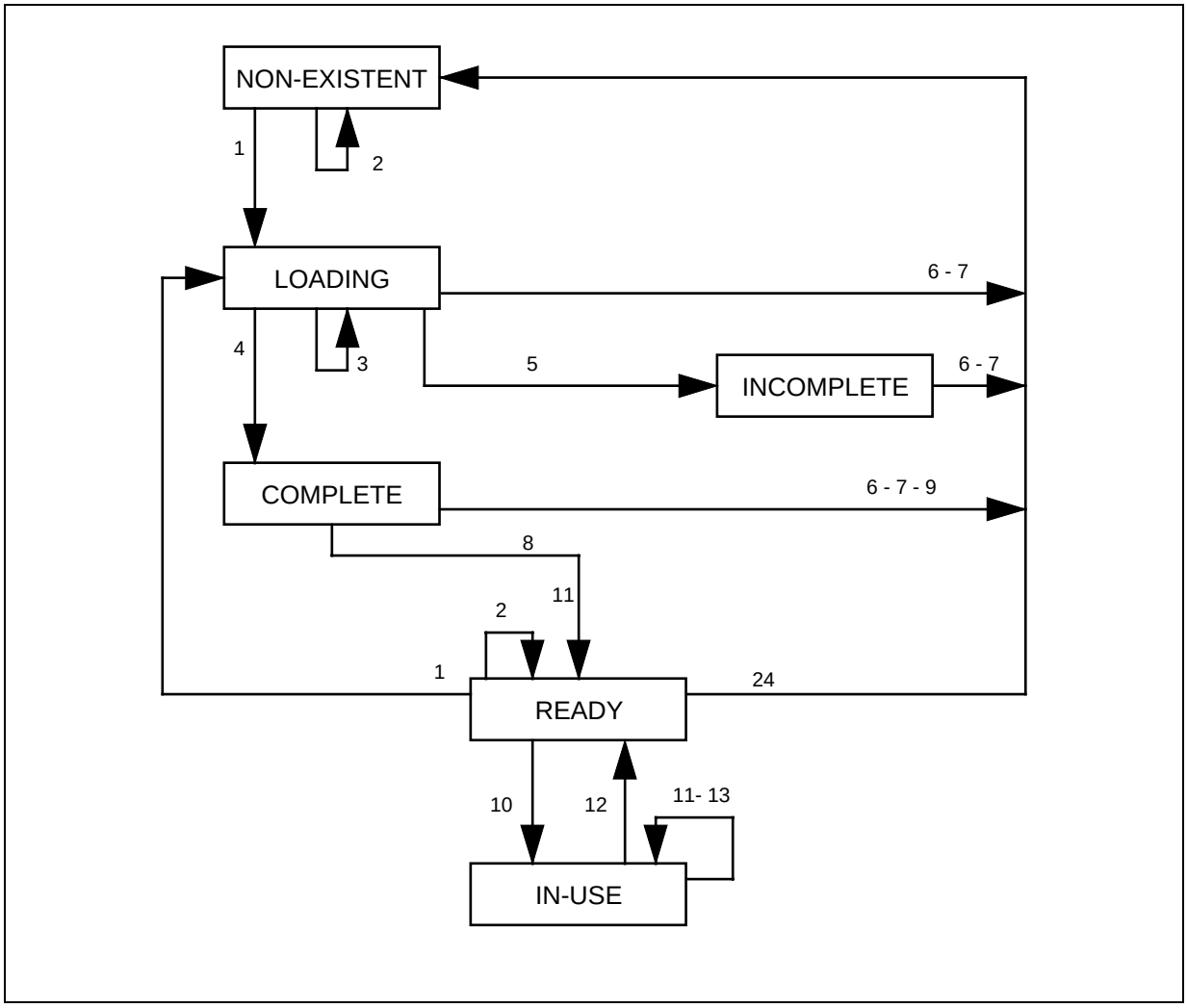
♦ **Password** : il contient le mot de passe pour protéger l'accès au domaine.

♦ **Access-Groups** : indique les groupes pouvant accéder au domaine (1 à 8)

♦ **Access-Rights** : indique les droits d'accès au domaine (droit de lecture, droit d'écriture, droit de création d'Instance de Programme au départ de ce domaine) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie d'un groupe d'accès, pour les utilisateurs quelconques.

♦ **Local-Address** : adresse interne renvoyant à l'objet réel.

♦ **Domain-State** : le domaine peut être dans l'un des états donné par ce diagramme d'états :



1	Indication de début de séquence de chargement (Un Client a envoyé cette requête au VFD). Réponse positive à la demande de début de séquence de chargement (le VFD envoie cette réponse au Client).
2	Indication de début de séquence de chargement (Un Client a envoyé cette requête au VFD). Réponse négative à la demande de début de séquence de chargement (le VFD envoie cette réponse au Client).
3	Requête de chargement de segment (le VFD envoie cette requête au Client). Confirmation positive du chargement de segment. D'autres segments suivent (le VFD a reçu le segment du Client).
4	Requête de chargement de segment (le VFD envoie cette requête au Client).Confirmation positive du chargement de segment. Dernier segment (le VFD a reçu le dernier segment du Client).
5	Requête de chargement de segment (le VFD envoie cette requête au Client). Confirmation négative du chargement de segment (Client -> VFD).
6	Requête de demande de terminaison de la séquence de chargement avec présence d'effacement (VFD -> Client). Confirmation positive ou négative de la demande de terminaison de la séquence de chargement (Client -> VFD)
7	Abort.
8	Requête de demande de terminaison de la séquence de chargement sans présence d'effacement (VMD -> Client). Confirmation positive de la demande de terminaison de la séquence de chargement (Client -> VMD).
9	Requête de demande de terminaison de la séquence de chargement sans présence d'effacement (VFD -> Client). Confirmation négative de la demande de terminaison de la séquence de chargement (Client -> VFD).
10	L'attribut Counter est incrémenté et il vaut 1 (Start/Resume)
11	L'attribut Counter est incrémenté de 1. (Start/Resume).
12	L'attribut Counter est décrémenté de 1 et il vaut 0 (Stop/ Kill/ Fin de programme/ Arrêt de Programme)

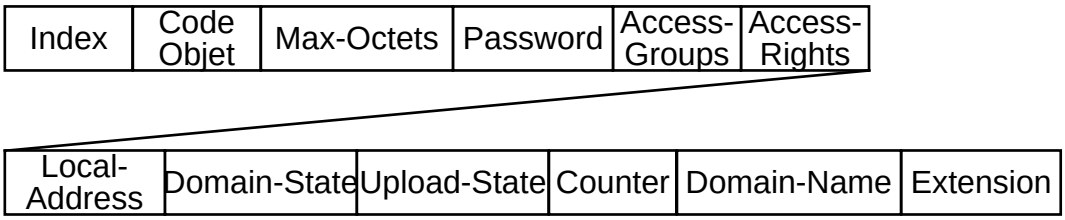
13	L'attribut Counter est décrémenté de 1 (Stop/ Kill/ Fin de programme/ Arrêt de Programme)

Upload-State : cet attribut indique l'état de l'UPLOAD STATE MACHINE (NON-EXISTENT, UPLOADING, UPLOADED).

♦ **Counter** : contient le nombre d'Instances de Programme qui utilisent le domaine.

♦ **Extension** : contient des données spécifiques à un profil.

Contenu de la description d'un domaine dans le dictionnaire des Objets



Il est clair que le code de l'objet est Domain.

(5) L'Objet Simple-Variable

L'objet Variable-Simple correspond à une variable caractérisée par un type de données dont le contenu se trouve dans une variable simple du système. Un Objet Simple-Variable ne peut être caractérisé que par un seul type de données.

Voici les attributs de l'Objet Simple-Variable :

Index : c'est l'index associé à la variable simple dans l'OD.

♦ **Variable-Name** : le nom de la variable simple. Son existence et sa longueur maximale sont définies par l'attribut Name-Length de l'Objet OD.

♦ **Data-Type-Index** : contient l'index du type de données associé à la variable

♦ **Length** : longueur en octets pour les données

♦ **Password** : il contient le mot de passe pour destiné au droit d'accès.

♦ **Access-Groups** : indique les groupes pouvant accéder à la variable (1 à 8)

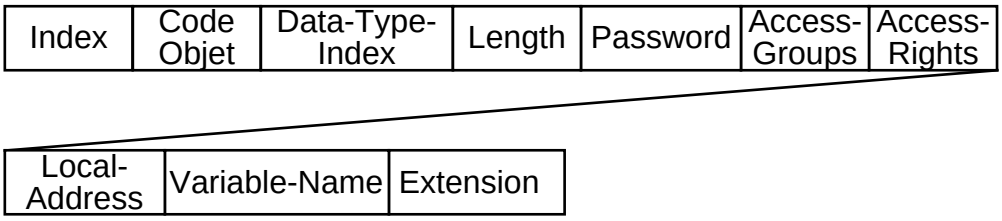
♦ **Access-Rights** : indique les droits d'accès à la variable (droit de lecture, droit d'écriture) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie

d'un groupe d'accès, pour les utilisateurs quelconques.

♦ **Local-Address** : adresse interne renvoyant à l'objet réel.

♦ **Extension** : contient des données propres aux profils.

Contenu de la description d'une Simple-Variable dans le dictionnaire des Objets.



Il est clair que le code de l'objet vaut VAR

(6) L'Objet Array

L'objet Array correspond à une séquence de variables simples toutes caractérisées par un même type de données dont le contenu se trouve dans un tableau du système. Un Objet Array ne peut être caractérisé que par un seul type de données pour tous les éléments du tableau.

Voici les attributs de l'Objet Array :

Index : c'est l'index associé au tableau dans l'OD.

♦ **Variable-Name** : le nom du tableau. Son existence et sa longueur maximale sont définies par l'attribut Name-Length de l'Objet OD.

♦ **Data-Type-Index** : contient l'index du type de données associé au tableau.

♦ **Length** : longueur en octets pour un élément du tableau.

♦ **Number-Of-Elements** : indique le nombre d'éléments que comporte le tableau.

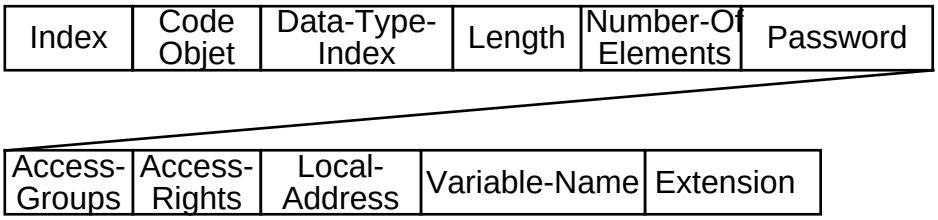
♦ **Password** : il contient le mot de passe pour destiné au droit d'accès.

♦ **Access-Groups** : indique les groupes pouvant accéder au tableau (1 à 8).

♦ **Access-Rights** : indique les droits d'accès au tableau (droit de lecture, droit d'écriture) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie d'un groupe d'accès, pour les utilisateurs quelconques.

♦ **Local-Address** : adresse interne renvoyant à l'objet réel.
♦ **Extension** : contient des données propres aux profils.

Contenu de la description d'un Array dans le dictionnaire des Objets.



Il est clair que le code de l'objet vaut Array.

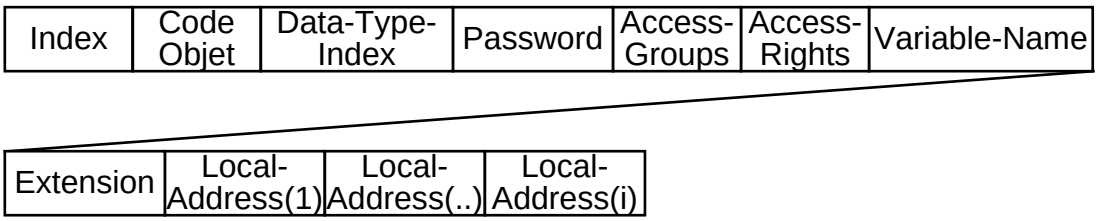
(7) L'Objet Record

L'objet Record correspond à une séquence de variables simples de type de données différents dont le contenu se trouve dans une structure interne au système.

Voici les attributs de l'Objet Record :

Index : c'est l'index associé à la structure dans l'OD.
♦ **Variable-Name** : le nom de la structure. Son existence et sa longueur maximale sont définies par l'attribut Name-Length de l'Objet OD.
♦ **Data-Type-Index** : contient l'index du type de données associé à la structure.
♦ **Password** : il contient le mot de passe destiné au droit d'accès.
♦ **Access-Groups** : indique les groupes pouvant accéder au tableau (1 à 8).
♦ **Access-Rights** : indique les droits d'accès au tableau (droit de lecture, droit d'écriture) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie d'un groupe d'accès, pour les utilisateurs quelconques.
♦ **Extension** : contient des données propres aux profils.
♦ **List-Of-Local-Address** : contient les adresses internes des éléments de la structure dans le système.

Contenu de la description d'un Record dans le dictionnaire des Objets.



Il est clair que le code de l'objet vaut Record.

(8) L'Objet Variable-List

L'objet Variable-List correspond à une séquence de variables simples, de tableaux et de structures dont le contenu se trouve dans des variables, des tableaux et des structures internes au système. Il contient la liste des index des objets qui la composent. Il faut retenir que les Variable-List sont toujours effacées lorsque l'on recharge l'OD du serveur.

Voici les attributs de l'Objet Variable-List :

Index : c'est l'index associé à la Variable-List dans l'OD.

♦ **Variable-List-Name** : le nom de la Variable-List. Son existence et sa longueur maximale sont définies par l'attribut Name-Length de l'Objet OD.

♦ **Number-Of-Elements** : contient le nombre d'éléments que comporte la Variable-List.

♦ **Password** : il contient le mot de passe destiné au droit d'accès.

♦ **Access-Groups** : indique les groupes pouvant accéder à la Variable-List (1 à 8).

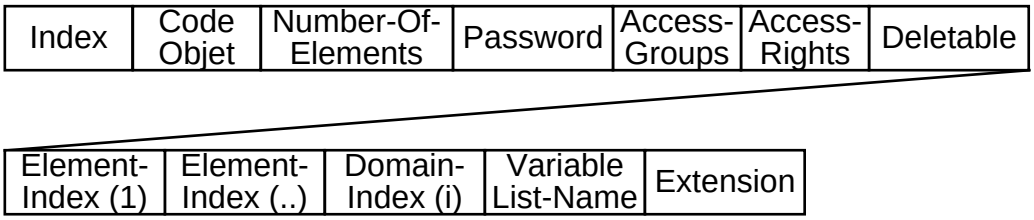
♦ **Access-Rights** : indique les droits d'accès à la Variable-List (droit de lecture, droit d'écriture, droit d'effacement) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie d'un groupe d'accès, pour les utilisateurs quelconques.

♦ **Deletable** : indique si la Variable-List est effaçable ou non.

♦ **List-Of-Element-Index :** contient la liste des index des éléments composant la Variable-List.

♦ **Extension** : contient des données propres aux profils.

Contenu de la description d'une Variable-List dans le dictionnaire des Objets.



Il est clair que le code de l'objet vaut Var-List.

(9) L'Objet Type de données

L'Objet Type de données permet de caractériser une variable en lui associant la liste des valeurs possibles pour la donnée de ce type et les opérations qui peuvent être effectuées sur ce type de données, la syntaxe et la représentation de ce type données au sein du système de communication. Profibus définit une série de type données par défaut mais permet aussi de créer ses propres types de données. Ainsi Profibus prévoit les types suivants :

♦ *Boolean* prend la valeur True ou False
♦ *Integer* (8,16, 32)
♦ *Unsigned* (8,16, 32)
♦ *Floating-Point* (codage 4 octets IEEE 754 DShort Real Number)
♦ *Visible-String* : série de caractères affichables (codage ISO 646)
♦ *Octet-String* : série d'octets (codage binaire)
♦ *Bit-String* : série de bits (la longueur de la Bit-String doit être un multiple positif entier de 8)
♦ *Date* : 7 octets qui donne l'heure jusqu'à la milliseconde (HH:MM:S.xxx), le jour de la semaine, le jour du mois, le mois, l'année.
♦ *Time-Of-Day* : 6 octets qui contiennent le nombre de millisecondes écoulées depuis minuit (4 octets codage binaire) ainsi le nombre de jours écoulés depuis le 1/01/1984 (2 octets codage binaire)
♦ *Time-Difference* : 6 octets qui contiennent une différence temporelle en millisecondes (4 octets codage binaire) ainsi qu'une différence de nombre de jours (2 octets codage binaire)

L'Objet Type de Donnée comporte les attributs suivants :

- **Index** : index de cet Objet dans l'OD
- **Description** : contient la description de l'Objet sous forme verbale
  - **Symbol-Length** : longueur du symbole descriptif du type
  - **Symbol** : contient une Visible-String de 0 à 32 octets.

Contenu de la description d'un type de données dans le dictionnaire des Objets.

Index	Code Objet	Description	
		Symbol-Length	Symbol

Il est clair que le code de l'objet vaut Type de données (DatenTyp).

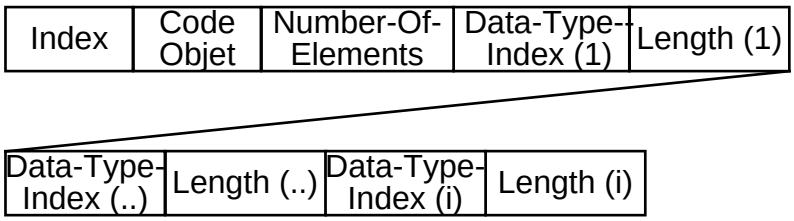
(10) L'Objet Description de structure de types de données

Cet objet permet de définir la taille et la structure d'un Record. Il comporte le type et la longueur des éléments composant le Record.

L'Objet Description de structure de types de Données comporte les attributs suivants :

- **Index** : index de cet Objet dans l'OD.
- **Number-of-Elements** : Nombre d'éléments compris dans cette description.
- **List Of Element** : contient une liste d'éléments composés du type de données et de la longueur.
  - **Data-Type-Index** : index correspondant au type de données de l'élément N
  - **Length** : longueur de l'élément N.

Contenu de la description de l'objet Description de Structure de Type de Données dans le dictionnaire des Objets.



Il est clair que le code de l'objet vaut Description de Structure de Type de données (Ds).

(11) L'Objet Instance de Programme

L'objet Instance de Programme permet de créer au départ d'un ou plusieurs domaines un programme exécutable qui sera gérable par les services sur les Instances de programmes. Les instances de programmes peuvent être prédéfinies ou créées dynamiquement.

L'objet Instance de Programme comporte les attributs suivants :

Index : c'est l'index associé à l'IP dans l'OD

♦ **PI-Name** : le nom de l'Instance de Programme. Son existence et sa longueur maximale sont définies par l'attribut Name-Length de l'Objet OD.

♦ **Number-Of-Domains** : nombre de domaines utilisés par cette IP. Le nombre est limité par le nombre maximum d'octets disponible pour la requête Create-Program-Invocation.

♦ **Password** : il contient le mot de passe pour protéger l'accès à l'IP.

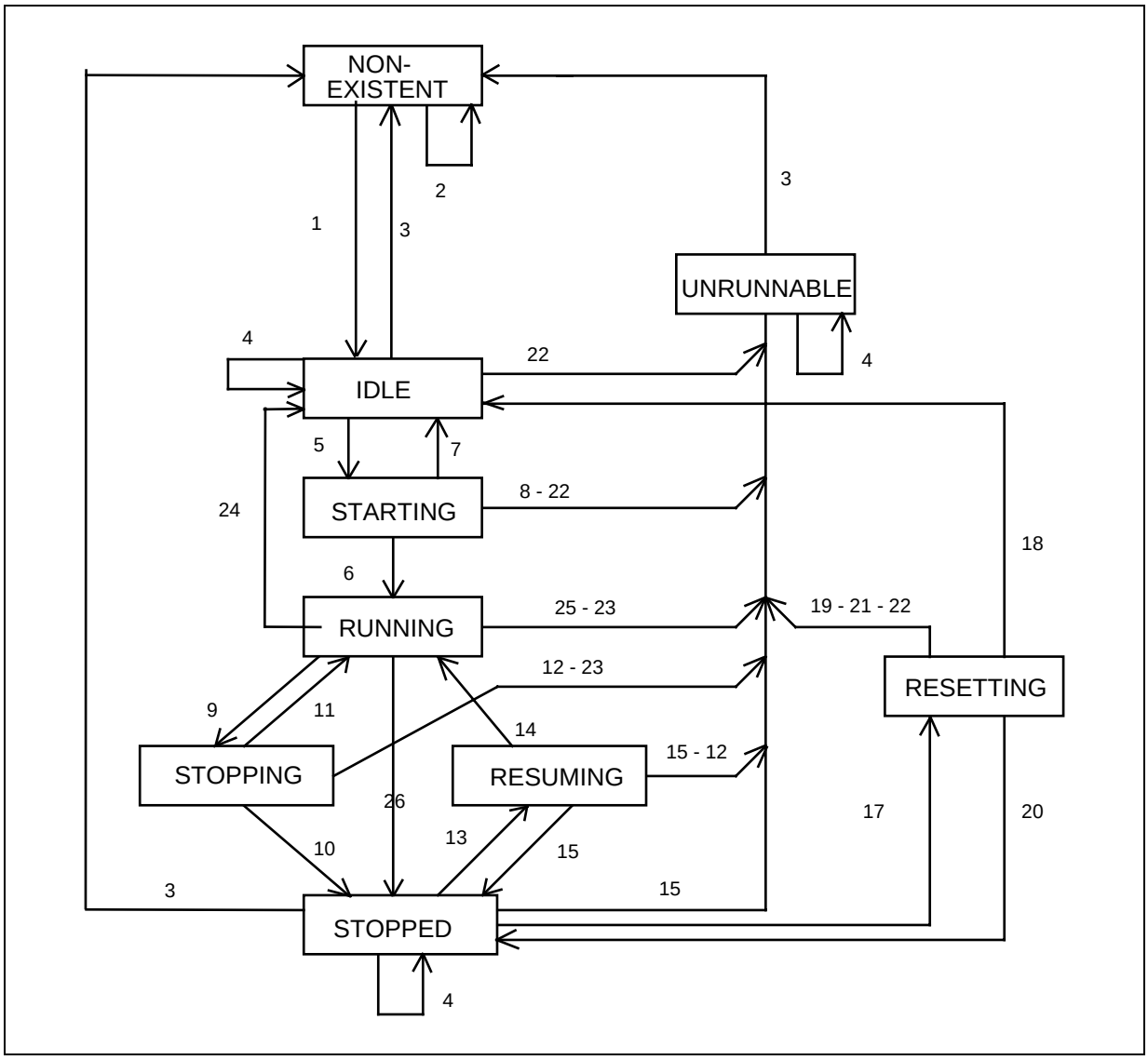
♦ **Access-Groups** : indique les groupes pouvant accéder à l'IP (1 à 8)

♦ **Access-Rights** : indique les droits d'accès au domaine (droit de démarrage (Start, Resume, Reset), droit d'arrêt (Stop), droit d'effacement (Kill, Delete IP) pour l'utilisateur disposant du mot de passe, pour l'utilisateur faisant partie d'un groupe d'accès, pour les utilisateurs quelconques.

♦ **Deletable** : indique si l'IP peut être effacée par le service Delete IP.

♦ **Reusable** : indique si après exécution l'IP passe à l'état IDLE (True) ou UNRUNNABLE (False).

♦ **PI-State** : l'objet Instance de Programme peut prendre un des états suivants donné par le diagramme d'état.



Signification des transitions entre états :

1 Création de l'Instance de Programme (Indication). Acquittement positif de la création de l'IP

2 Création de l'Instance de Programme (Indication). Domaines non présents ou accès interdit. Acquittement négatif.

3 Effacement de l'IP (Ind).Acquittement positif de l'effacement de l'Instance de Programme

4 Effacement de l'Instance de Programme (ind) Acquittement négatif de l'effacement de l'Instance de Programme.

5 Démarrage de l'IP (indication).

6	Démarrage correct. Counter est incrémenté. Acquittement positif du démarrage.
7	Démarrage a échoué, non destructif. Domaine n'était pas dans l'état Ready ou In-Use. Acquittement négatif du démarrage.
8	Démarrage a échoué, destructif. Acquittement négatif du démarrage.
9	Arrêt de l'IP (indication)
10	Arrêt réussi. Counter est décrémenté. Acquittement positif de l'arrêt.
11	Arrêt a échoué, non destructif. Acquittement négatif de l'arrêt.
12	Arrêt a échoué, destructif. Counter est décrémenté. Acquittement négatif de l'arrêt.
13	Reprise (indication)
14	Reprise réussie. Counter est incrémenté. Acquittement positif de la reprise.
15	Reprise a échoué, non destructive. Acquittement négatif de la reprise.
16	Reprise a échoué, destructive. Acquittement négatif de la reprise.
17	Réinitialisation (indication)
18	Réinitialisation réussie. Reusable = True. Acquittement positif de la Réinitialisation.
19	Réinitialisation réussie. Reusable = False. Acquittement positif de la Réinitialisation.
20	Réinitialisation a échoué, non destructive. Acquittement négatif de la Réinitialisation.
21	Réinitialisation a échoué, destructive. Acquittement négatif de la Réinitialisation.
22	Kill (indication). Acquittement positif.
23	Kill (indication). Counter décrémenté. Acquittement positif.

24 Fin de programme. Reusable = True. Counter décrémenté.

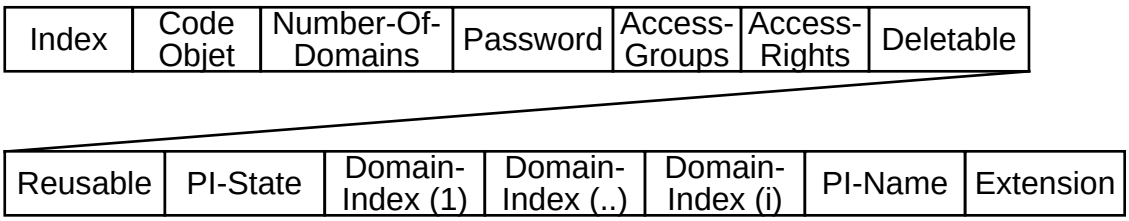
25 Fin de programme. Reusable = False. Counter décrémenté.

26 Arrêt de programme. Counter décrémenté.

List Of Domain-Index : liste des index des domaines utilisés par l'IP. Le premier index de la liste doit être un index de domaine contenant un programme exécutable.

♦ **Extension** : cet attribut contient des données spécifiques aux profiles.

Contenu de la description de l'IP dans le dictionnaire des Objets



Il est clair que le code de l'objet est PI.

e) Les services FMS

Il existe 7 groupes de services:

♦ les services de gestion du contexte (connexion),
♦ les services de gestion du répertoire d'objets (variables, ...),
♦ les services de gestion du VFD,
♦ les services d'accès aux variables,
♦ les services de gestion des Invocations de programmes,
♦ les services de gestion des événements,
♦ les services de gestion des domaines.

Ces services sont basés sur MMS de MAP 3.0 avec des adaptations propres aux fonctionnalités et contraintes des réseaux de terrain (temps de réaction minimum).

(1) Services de gestion de contexte (règles de dialogue)

(a) Le service Initiate

Ce service confirmé permet d'établir une connexion entre deux partenaires. Lors de l'appel de ce service, les deux partenaires s'échangent une série de paramètres qui permettront de déterminer s'ils sont aptes à communiquer et de définir le contexte de la communication (règles). Les partenaires doivent avoir les mêmes paramètres pour pouvoir communiquer : il n'a y pas de négociation.

(b) Le service Abort

Ce service permet de terminer une connexion entre deux partenaires (des règles de dialogue)

(c) Le service Reject

Ce service confirmé permet de rejeter un FMS PDU non reconnu.

(2) Services de gestion du dictionnaire d'objets

(a) Le service Get-OV

Ce service confirmé permet à un Client d'obtenir une ou plusieurs descriptions d'objets reprises dans le dictionnaire des objets (OD en anglais ou OV en allemand) d'un Serveur. Il est possible de demander un objet en particulier, tous les objets d'un même type, tous les objets. Si les descriptions de tous les objets demandés ne savent pas contenir dans un FMS PDU alors il faut relancer ce service en indiquant l'index +1 de la dernière description d'objet reçue. Il se peut qu'il faille relancer le service plusieurs

(b) Le service Initiate-Put-OV.

Ce service confirmé permet à un Client d'informer le Serveur qu'il va lui envoyer des descriptions d'objets qu'il devra intégrer dans son OD. Le Client précise si les modifications qu'il va opérer, ont une incidence ou non sur les autres Clients. Aucune autre indication Initiate-Put-OV ne sera acceptée tant que le Serveur n'aura pas reçu une indication Terminate-Put-OV.

(c) Le service Put-OV

Ce service confirmé permet à un Client d'envoyer à un Serveur des descriptions d'objets que ce dernier intégrera dans son OD.

(d) Le service Terminate-Put-OV

Ce service confirmé permet à un Client d'indiquer au Serveur que l'envoi de description par le service Put-OD est terminé.

(3) Les services de gestion du VFD

Les services de gestion de VFD sont les mêmes que ceux supportés par MMS.

(4) Les services d'accès aux variables

(a) Le service Read

Ce service confirmé permet à un client de lire le contenu d'un objet Simple-Variable, Array, Record, Variable-List d'un VFD distant. Dans le cas des Array ou des Record, il est possible d'accéder à un élément de l'Array ou du Record.

(b) Le service Write

Ce service confirmé permet à un client d'écrire dans un objet Simple-Variable, Array, Record, Variable-List d'un VFD distant. Dans le cas des Array ou des Record, il est possible d'accéder à un élément de l'Array ou du Record.

(c) Le service Read with Type

Ce service confirmé permet à un client de lire le contenu d'un objet Simple-Variable, Array, Record, Variable-List et la description du type de données associé.

(d) Le service Write with Type

Ce service confirmé permet à un client d'écrire dans un objet Simple-Variable, Array, Record, Variable-List d'un VFD distant et d'envoyer la description du type associé à cet objet.

(e) Le service Phys-Read

Ce service confirmé permet à un client de lire le contenu d'une variable existant réellement dans le système. Il faut indiquer l'adresse locale dans le système et le nombre d'octets à lire.

(f) Le service Phys-Write

Ce service confirmé permet à un client d'écrire dans une variable existant réellement dans le système. Il faut indiquer l'adresse locale dans le système et les données à écrire à partir de cette adresse.

(g) Le service Information Report

Ce service non confirmé permet à un Serveur d'envoyer spontanément un objet Simple-Variable, Array, Record, Variable-list à un ou plusieurs Client. Dans le cas des Array ou des Record, il est possible d'accéder à un élément de l'Array ou du Record.

(h) Le service Information Report with Type

Ce service non confirmé permet à un Serveur d'envoyer spontanément un objet Simple-Variable, Array, Record, Variable-list à un Client.

(5) Les services de gestion des Invocations de programmes

Les services d'invocations de programmes sont les mêmes que ceux supportés par MMS.

(6) Les services de gestion des domaines

Les services de domaines sont les mêmes que ceux supportés par MMS.

(7) Les services minimum

Profibus prévoit que tous les équipements doivent au moins supporté les services suivants :

- Le service Initiate,
- Le service Abort,
- Le service Reject,
- Le service Status,
- Le service Identify,
- Le service Get-OV

3. La couche LLI

La couche LLI ou Lower Layer Interface est la couche qui fait le lien entre la couche FMS en la couche FDL. Elle doit permettre de réaliser les services FMS au départ de la couche FDL. En effet, FMS est un utilisateur de la couche LLI.

LLI utilise les services FDL suivants :

- Send Data With No Acknowledge (SDN),
- Send Data With Acknowledge (SDA),
- Send and Request Data With Reply (SRD),
- Cyclic Send and Request Data With Reply (CSRD).

Profibus prévoit des différents types de connexions :

dans le cas de relations de communication entre Maître et Esclave

- Connexion pour échange de données cyclique sans Initiative de l'esclave
- Connexion pour échange de données cyclique avec Initiative de l'esclave
- Connexion pour échange de données acyclique sans Initiative de l'esclave
- Connexion pour échange de données acyclique avec Initiative de l'esclave

- dans le cas de relations de communication entre Maître et Maître
  - Connexion pour échange de données acyclique

Dans le cas des connexions destinées à échanger des données de manière cyclique, seuls les services confirmés Read et Write sont permis. Sur ce genre de connexion, le maître et l'esclave (s'il en a le droit) peuvent tout de même envoyer des services FMS non confirmé.

Dans le cas des connexions destinées à échanger des données de manière acyclique, le maître peut utiliser des services FMS confirmés ou non et l'esclave (s'il en a le droit) peut envoyer des services FMS non confirmés.

Les communications peuvent être orientées connexion. Ce type de communication pourra être utilisé dans le cas de relation de communication de type 1 à 1. Les communications orientées connexion comportent toujours une phase de création de la connexion, une phase d'échange de données et une phase de terminaison de la connexion. Et il est clair que les données ne peuvent être échangées que si la phase de création de la connexion a abouti.

Les communications peuvent aussi se réaliser en mode non connecté (connectionless). Les relations de communication de type un à plusieurs ou un à tous ne peuvent être que du mode non connecté.

Pour accéder au service de la couche LLI, celle-ci propose deux LLI-SAP. Le LLI-SAP 0 est utilisé par FMS tandis que le LLI-SAP 1 est utilisé par FMA qui ne fait pas partie de ces notes.

Au niveau de la LLI, il existe une LLI-KBL qui comporte les paramètres propres à la LLI de la FMS-KBL. C'est donc la LLI-KBL qui contient la liste des relations de communication qui peuvent être établies. Chaque entrée comprend différents paramètre permettant d'établir la connexion avec le partenaire distant (adresse du partenaire, LSAP destinataire..). Chaque entrée de cette liste est référencée par KR (Kommunikationsreferenz).

XI. FIP OU FIELDBUS INSTRUMENTATION PROTOCOL

A. Généralités

Le réseau FIP est destiné à relier les systèmes se situant dans les niveaux 1(capteurs/actionneurs, ...) et 2 (API, régulateurs, ...) de la pyramide C.I.M.

C'est une base de données réparties en temps critique. Aussi est-on certain de disposer des données rafraîchies à une cadence déterminée fonction du processus à commander.

FIP offre en outre la possibilité d'échange de messages MMS.

B. La couche Physique

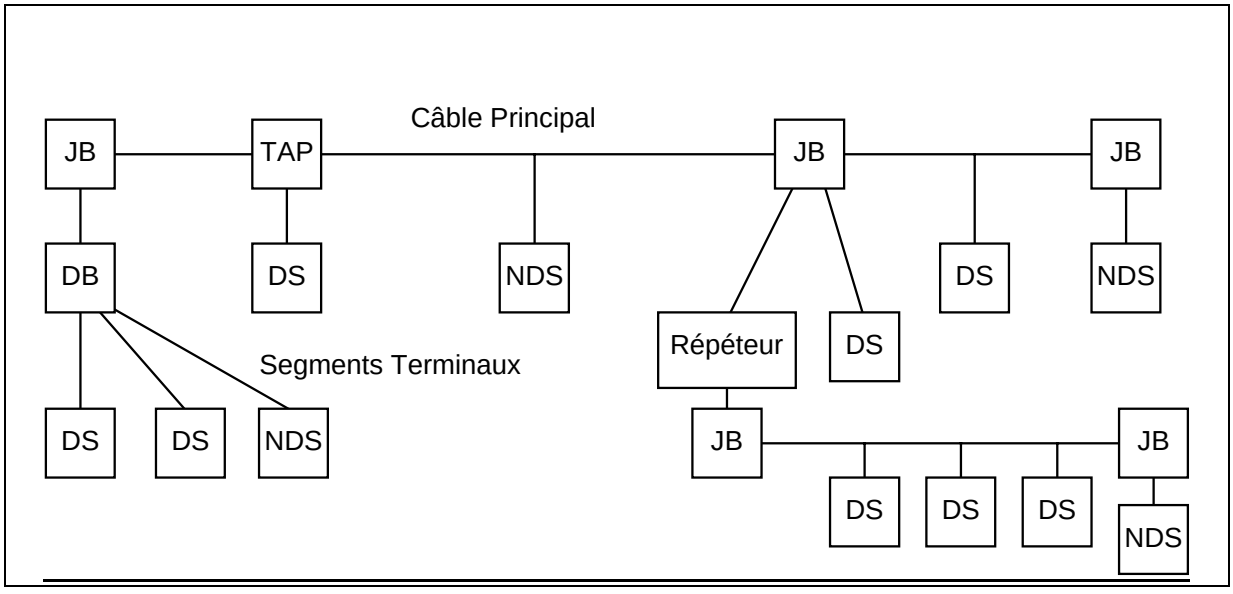
1. Support physique

Paire torsadée blindée (STP)

♦ Fibre optique

2. Topologie

Le réseau se compose de câbles principaux , de répéteurs et de segments terminaux.



JB : Boîte de jonction = dérivateur multiple passif avec deux accès débrochables au moins
TAP : Dérivateur = système fournissant un accès sur le câble principal
Répéteur : permet d'interconnecter les câbles principaux du réseau de terrain
DB : boîte de diffusion = permet de réunir plusieurs segments terminaux sur un câble principal
DS : station déconnectable localement
NDS : station non déconnectable localement

3. Vitesse de transmission

Sur STP :

- S1 : 31,25kb/s
- S2 : 1Mb/s (Vitesse standard)
- S3 : 2,5 Mb/s

Sur Fibre optique, on dispose en plus de la vitesse 5Mb/s.

4. Technique de codage des signaux

Manchester avec violation de code

5. Trame

La trame se compose de :

un préambule de 8 symboles à 1 (PRE)
- un délimiteur de début de trame (FSD) : il contient des violations de codage Manchester qui le rendent reconnaissable. (6 symboles)
- le champ de contrôle et de données (CAD) : il contient les informations provenant de la couche Liaison de données
- un délimiteur de fin de trame (FED) : il contient des violations de codage Manchester qui le rendent reconnaissable. (7 symboles)

On remarquera donc que la couche physique ajoute 21 symboles à toute trame émise. A 2,5 Mb/s, la durée de transmission de ces symboles inutiles est de 8,4 µs.

C. La couche Liaison de données

1. Généralités

Celle-ci prévoit l'échange de variables identifiées et le transfert de messages soit :

♦ cycliquement : les noms des objets échangés cycliquement ainsi que la période de mise à jour sont fixées lors de la configuration du système. Ces échanges ont lieu sans intervention de l'utilisateur.

♦ sur demandes explicites des utilisateurs

Le Producteur et les Consommateurs d'une variable correspondent respectivement à la station qui doit fournir au réseau la valeur de cette variable et aux stations qui désirent obtenir le contenu de la variable. Chaque variable ne peut avoir qu'un et un seul producteur.

2. Adressage des objets

FIP prévoit deux espaces d'adressage différents:

♦ Adressage des variables :

A chaque variable du système répartie est associé un identifieur qui caractérise la variable de manière unique. C'est un adressage global. L'identifieur est codé sur 16 bits. Théoriquement, le système pourrait comporter 65536 variables. On remarquera que les utilisateurs qui échangent des variables n'utilisent pas un adressage physique des stations mais plutôt font référence aux identifieurs des variables à échanger.

♦ Adressage de messages

L'échange de message se fait de point-à-point sur le réseau ou en multicast sur un même segment. L'adresse des stations est codées sur 24 bits (adresse du segment dans le réseau et adresse de la station dans le segment)

3. Les buffers de la couche Liaison de données

La couche liaison de données comporte un ensemble de buffers de variables produites ou consommées. La couche Application accède à ces buffers pour envoyer des variables sur le réseau (producteur) et pour lire des variables provenant du réseau (consommateur).

4. Mécanisme d'allocation du support physique

Au niveau de l'accès au support de communication, le réseau comporte des stations possédant deux types de fonctionnalités :

- ♦ l'arbitre de bus : gère l'accès au support de communication
- ♦ les stations productrices et consommatrices.

Toute station FIP peut disposer des deux fonctionnalités, mais un instant donné, une seule station assure la fonction d'arbitre de bus.

L'arbitre de bus dispose d'une table de scrutation de variables contenant la liste des identifieurs à faire circuler sur le bus ainsi que des périodes de mise-à-jour de chaque variable. Tout ceci est défini lors de la configuration du réseau.

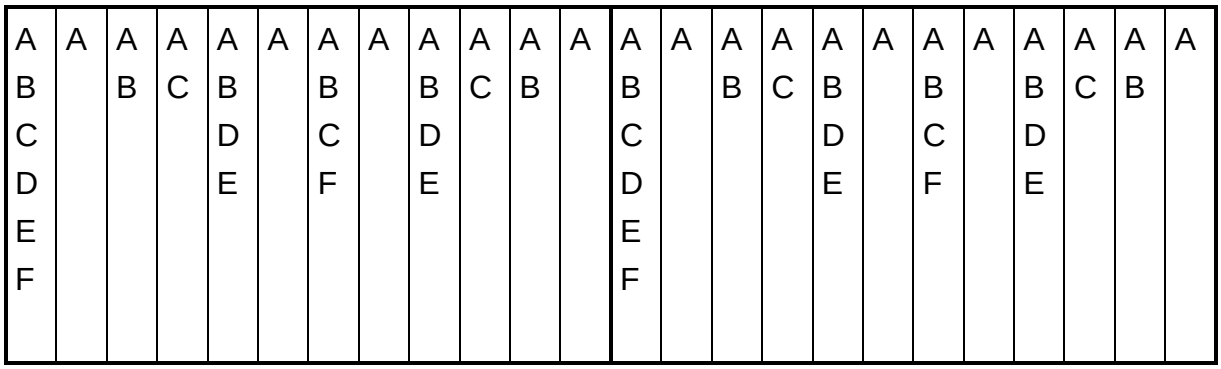
L'arbitre de bus est chargé de diffuser sur le bus une trame question (ID_DAT) et le nom de l'identifieur (X). Cette trame (ID_DAT_X)est reçue par l'ensemble des stations :

- ♦ Le producteur de l'identifieur sait ainsi qu'il va devoir émettre le contenu de cet identifieur
- ♦Les consommateurs de cet identifieur savent que la trame de réponse à cette question contient la valeur de l'identifieur.

Le producteur renvoie une trame réponse avec le contenu de X (RP_DAT_X). Toutes les stations consommatrices de X décodent la trame RP_DAT_X et en extraient la valeur de X.

La mise à jour des buffers des variables consommées et l'envoi des buffers des variables produites se font donc à l'initiative de l'arbitre de bus.

Au niveau de l'arbitre de bus, on définit un cycle élémentaire synchrone. Pendant ce cycle élémentaire, l'arbitre de bus doit demander les valeurs des différents identifieurs (variables). Comme les variables n'ont pas toutes les mêmes périodes de mise-à-jour, le nombre de demandes d'identifieurs variera d'un cycle élémentaire à l'autre. Cette variation du nombre de demandes sera cyclique et définit un macrocycle.



Au niveau des cycles élémentaires, il reste donc du temps disponible après la demande des variables périodiques. Un cycle élémentaire synchrone comporte donc en réalité une fenêtre temporelle pour le trafic cyclique d'identifieurs mais aussi une fenêtre pour le trafic acyclique d'identifieurs, une fenêtre pour le trafic acyclique de messages et une fenêtre pour la synchronisation (identifieurs de bourrage). L'arbitre de bus peut aussi utiliser un cycle élémentaire asynchrone.

Le trafic acyclique de variables

L'arbitre de bus envoie une trame ID_DAT_ X et le producteur de X désire obtenir un variable au travers du trafic acyclique. Le producteur de X répond par une trame RP_DAT_X_RQ (bit de requête apériodique positionné dans le champ de contrôle de la réponse). L'arbitre de bus (BA) met dans la file d'attente de demandes de transfert apériodiques l'identifieur X (il existe une file d'attente à faible priorité et une autre à haute priorité). Dans la fenêtre temporelle réservée au trafic apériodique de variables, le BA envoie une requête ID_RQ_X. Le producteur de X envoie la réponse RP_RQ avec la liste des identifieurs (jusqu'à 64 identifieurs). Le BA stocke dans une autre file d'attente la liste des identifieurs fournie par le producteur de X. Quand le BA disposera de temps, il enverra des trames ID_DAT correspondant à la liste des identifieurs demandés.

On remarquera qu'il faut être producteur d'une variable pour pouvoir utiliser les échanges apériodiques de variables.

5. Le trafic acyclique de messages sans acquittement

L'arbitre de bus envoie une trame ID_DAT_ X et le producteur de X désire envoyer un message. Le producteur de X répond par une trame RP_DAT_X_MSG (bit de requête apériodique de message positionné dans le champ de contrôle de la réponse). L'arbitre de bus (BA) met dans la file d'attente de demandes d'envoi de message l'identifieur X . Durant la fenêtre réservée au trafic apériodique de message, le BA envoie la trame ID_MSG_X et le producteur de X envoie la trame de message sans acquittement RP_MSG_NOACK (contient l'adresse du destinataire et du producteur X). Le producteur termine en envoyant la trame RP_FIN pour rendre le contrôle du bus au BA (BA dispose d'une temporisation pour reprendre la main si le RP_FIN se fait trop attendre)

6. Le trafic acyclique de messages avec acquittement

L'arbitre de bus envoie une trame ID_DAT_ X et le producteur de X désire envoyer un message. Le producteur de X répond par une trame RP_DAT_X_MSG (bit de requête apériodique de message positionné dans le champ de contrôle de la réponse). L'arbitre de bus (BA) met dans la file d'attente de demandes d'envoi de message l'identifieur X . Durant la fenêtre réservée au trafic apériodique de message, le BA envoie la trame ID_MSG_X et le producteur de X envoie la trame de message RP_MSG_ACK (contient l'adresse du destinataire et du producteur X). Le destinataire renvoie une trame d'acquittement RP_ACK. Le producteur termine en envoyant la trame RP_FIN pour rendre le contrôle du bus au BA. (BA dispose d'une temporisation pour reprendre la main si le RP_FIN se fait trop attendre).

D. Couche Application - MPS

MPS correspond aux services périodiques/apériodiques sur les variables.

La couche application comporte en outre subMMS (sous-ensemble de la messagerie MMS) et ABAS (services application d'arbitrage de bus). Nous ne traiterons ici que de MPS.

MPS fournit les services suivants :
lecture/écriture locales

lecture/écriture distantes
indications d'émission/réception de variables
informations de fraîcheur des informations consommées
informations de cohérence spatiales et temporelles de données

XII. TABLE DES MATIERES

I. Généralités 1

A. A quoi cela sert-il ? 1

B. Topologie 1

1. Bus 1

2. Anneau 2

3. Etoile 2

4. Hybrides 3

C. Mode de transmission 4

1. Codage du signal 4

2. Type signal 5

D. Support de transmission 6

E. Méthode d'Accès 6

1. Maître-Esclaves 6

2. Jeton 7

3. Accès Aléatoire 7

II. Le modèle OSI de ISO (ISO 7498) 7

A. Quelques définitions 9

B. Les 7 couches du modèle OSI 12

C. Le modèle OSI et les réseaux locaux 17

D. Les systèmes d'interconnexion de réseaux 18

1. Les Répéteurs 18

2. Les Ponts 18

3. Les Routeurs 20

4. Les Passerelles 21

III. Différents Types de Réseaux 22

A. Réseau Ethernet ou IEEE 802.3 22

1. Principe du CSMA/CD 22

2. Constitution de la trame 23

3. Supports physiques. 26
4. Pont IEEE 802.3 Ethernet 32
5. Les swicth ou commutateurs Ethernet 33
B. IEEE 802.5 Token Passing Ring 34
1. Principe d'accès : Passage de jeton 37
2. Stations fonctionnelles 40
3. Phases de fonctionnement d'un Token Ring 42
4. Les différentes trames 45
5. Pont Token-Ring 49
6. Remarques 50
C. IEEE 802.4 Token-Bus 50
1. Généralités 50
2. Trame 51
D. FDDI (ISO 9314) 51
1. Généralités 51
2. Topologie 52
3. Trames 55
4. Allocation de largeur de bande 56
5. CDDI = Copper Distributed Data Interface 57
6. FDDI-II 57
E. FAST ETHERNET ou IEEE 802.3u 57
1. Généralités 57
2. Supports de transmission 58
3. MII ou Media-Independant Interface 60
4. Les éléments actifs : concentrateur ou switch 60
F. ATM Asynchronous Transfer Mode 61
1. Généralités 61
2. Structuration de ATM 62
G. RESEAU SANS FIL (WIRELESSLAN) : IEEE802.11 et IEEE802.11b HR 67

1. Architecture 67
2. Les supports physiques de IEEE 802.11 67
3. La couche MAC de IEEE802.11 69
IV. Les serveurs 71
A. Généralités 71
B. Le réseau poste à poste 71
C. Les serveurs de fichiers 72
1. Généralités 72
2. Connexion au serveur de fichiers 72
3. Exemple dans le cas du logiciel réseau Netware 3.11 73
D. Sécurité de fonctionnement /Sauvegarde 74
E. Les serveurs d'impressions 76
V. Le monde TCP-IP 77
A. Historique 77
B. Les composantes d'un réseau local sous TCP-IP 78
C. Architecture de TCPIP 79
D. La couche Réseau 79
1. L'adressage IP des stations 79
2. ARP ou Address Resolution Protocol (RFC 826) - RARP Reverse Address Resolution Protocol (RFC 903) (couche accès) 83
3. IP ou Internet Protocol (RFC 791) 84
4. ICMP ou Internet Control Message Protocol (RFC 792) 86
5. Les protocoles de routages 86
E. La couche Transport 87
1. TCP ou Transmission Control Protocol (RFC 793) 87
2. UDP ou User Datagram Protocol (RFC 768) 88
F. La couche Application 88
1. Telnet (RFC854/855) 88
2. FTP File Transfer Protocol (RFC 959) 89
3. TFTP Trivial File Transfer Protocol 89

4. BOOTP Bootstrap Protocol 89
5. SMTP Simple Mail Transfer Protocol (RFC 821/822/…) 90
6. NFS Network File System 90
7. X-Windows (norme X11) 91
8. Les services r 91
9. FTP Search 93
10. World Wide Web 93
11. Messagerie électronique (E-Mail) 93
12. Network News 94
13. Mailing Lists 95
G. Intranet 95
H. Connexion du réseau local à Internet 95
1. Réseaux Privés (RFC 1918) 95
2. Firewall 96
3. Proxi 96
4. Routeur avec translation d'adresses (NAT) 96
I. Utilitaires 96
VI. Système de câblage 97
VII. Administration de réseau et analyseurs 97
A. Les analyseurs 97
B. Administration du réseau 99
VIII. La pyramide C.I.M. 100
IX. MAP et MMS 100
A. MAP et MMS : Qu'est-ce que c'est ? 100
1. Introduction 100
2. Les étapes importantes de MAP 101
3. Les avantages de MAP 3.0 103
B. Les composants d'un réseau MAP typique 104
C. La Pile de Protocole MAP 106

1. La pile de protocole MAP 3.0 sur "Ethernet" 106
2. La couche physique et MAC 107
3. La couche LLC ou Logical Link Control 107
4. La couche Réseau 109
5. La Couche Transport ou ISO 8073 Classe 4 112
6. La Couche Session ou ISO 8326/8327 112
7. La Couche Présentation ou ISO 8822/8823 113
8. La Couche Application 113
9. ACSE ou Association Control Service Element (ISO 8649/8650) 114

D. MMS ou Manufacturing Message Specification 114

1. Généralités 114
2. MMS norme internationale 115
3. Le V.M.D. ou Virtual Manufacturing Device 116
4. Le modèle Client-Serveur 118
5. Le Requester et le Responder 120
6. Accès aux couches inférieures 122

E. Les objets MMS 122

1. Généralités 122
2. Le VMD 124
3. Les Domaines 128
4. Les Instances de Programme ou IP (Program Invocation) 132
5. Les Variables 134

F. Les Services MMS 138

1. La gestion générale de l'environnement MMS (Environment and General Management) 140
2. L'accès aux variables (Variable Access) 141
3. La gestion des domaines (Domain Management) 141
4. La gestion des instances de programme (Program Invocation Management) 155
5. La gestion du VMD (VMD Support) 158

G. CLASSES DE MISE EN OEUVRE 159

X. Le réseau de terrain PROFIBUS 159
A. Introduction 159
B. Généralités 162
C. Profibus FMS dans le modèle OSI (Open System Interconnection) de ISO 163
D. La couche physique de Profibus 164
E. Couche Liaison de données de Profibus ou Fieldbus Data Link (FDL) 166
1. Généralités 166
2. La couche MAC de Profibus 166
3. La couche LLC ou Logical Link Control de FDL 174
F. La couche 7 de Profibus ou LLI et FMS 177
1. Généralités 177
2. La couche FMS 177
3. La couche LLI 203
XI. FIP ou FieldBUS Instrumentation protocol 205
A. Généralités 205
B. La couche Physique 205
1. Support physique 205
2. Topologie 205
3. Vitesse de transmission 206
4. Technique de codage des signaux 206
5. Trame 206
C. La couche Liaison de données 207
1. Généralités 207
2. Adressage des objets 207
3. Les buffers de la couche Liaison de données 207
4. Mécanisme d'allocation du support physique 208
5. Le trafic acyclique de messages sans acquittement 210
6. Le trafic acyclique de messages avec acquittement 210
D. Couche Application - MPS 210

XII. TABLE DES MATIERES 212