

**Module de
Cours**

**Réseaux Locaux Industriels
et Bus de Terrain**

Eric DECKE

Date : Janvier 2006

Version : A (provisoire, en cours de rédaction)

SOMMAIRE

1 LA COMMUNICATION INDUSTRIELLE	6
1.1 PRE REQUIS.....	6
1.2 PANORAMA DES BESOINS USUELS DU « MONDE INDUSTRIEL »	7
1.2.1 Exemples de domaines d'application.....	7
1.2.2 Les « progiciels d'application »	11
1.2.3 Les « équipements »	11
1.3 LES BESOINS ET LES ECHANGES DE DONNEES.....	12
1.3.1 Exemples de « solutions de communication »	14
1.3.1.1 Protocoles.....	14
1.3.1.2 Bus	14
2 MODÈLE OSI (<i>OPEN SYSTEMS INTERCONNECTION</i>).....	15
2.1 MODELE GENERIQUE EN COUCHES	15
2.2 SERVICES ET PRIMITIVES.....	18
3 NOTIONS DE MODELISATION UTILES A L'ETUDE DES COUCHES PHYSIQUES ET LIAISON : LES AUTOMATES A ETATS FINIS.....	21
3.1 LES MACHINES D'ETATS FINIS	21
3.2 LES ETATS	21
3.3 LES TRANSITIONS ET EVENEMENTS	22
3.4 REPRESENTATION SOUS FORME DE TABLE DE TRANSITIONS	23
4 RLI ET BUS DE COMMUNICATION	24
4.1 RESEAU ET BUS DE TERRAIN	24
4.2 TOPOLOGIES	24
4.3 ELEMENTS D'INTERCONNEXION	25
4.3.1 répéteurs ou amplificateurs - convertisseurs	25
4.3.2 ponts	25
4.3.3 routeurs	26
4.3.4 passerelles	26

4.4 RESTRICTION DU MODÈLE OSI	27
4.5 COUCHE PHYSIQUE.....	28
4.5.1 Liaison parallèle	29
4.5.2 Liaison série	29
4.5.2.1 Principales normes des liaisons séries utilisées dans les RLI.	31
4.5.2.2 Normes V28 (électrique).....	31
4.5.3 Interface Rs232.....	33
4.5.4 interface Rs485	33
4.5.5 Modes de transmission	35
4.5.5.1 Transmission en bande de base.	35
4.5.5.2 Transmission modulée.	36
4.5.6 Supports de transmission.....	38
4.5.6.1 Représentation électrique d'un tronçon de ligne	38
4.5.6.2 Classes et catégories	39
4.6 COUCHE LIAISON	40
4.6.1 Notion de protocole	40
4.6.2 Deux exemples de protocoles, couches OSI, et automates associés.....	41
4.6.2.1 Le protocole MODBUS RTU.....	41
4.6.2.2 Le protocole UNITEL-WAY.....	42
4.7 EXEMPLES DE COUCHES « APPLICATION ».....	47
4.7.1 OPC : « standard de fait » actuel incontournable	47
5 LES RLI ET BUS DANS LE MONDE INDUSTRIEL.....	49
5.1 AVANTAGES.....	49
5.2 HISTORIQUE DE L'EVOLUTION DES CONCEPTS – STANDARDS	50
5.2.1 Historique	50
5.2.2 La bouche 4-20 ma	50
5.3 CRITERES DE COMPARAISON.....	51
5.3.1 Les critères techniques	51
5.3.1.1 Topologiques	51
5.3.1.2 Temporels.....	51
5.3.1.3 Autres	51
5.3.2 Les critères stratégiques	52
5.3.2.1 Standards	52
5.3.2.2 Disponibilité de composants, de logiciels et de prestation de services.....	52
5.3.2.3 Autres	52
5.4 COMPARAISON ENTRE LES PRINCIPAUX BUS	53

5.5 HIERARCHIE DES RLI ET BUS.....	54
5.6 CONSTAT.....	55
5.7 WORLDFIP.....	56
5.7.1 Origine de WORLDFIP.....	56
5.7.2 Fonctionnement de WORLDFIP.....	56
5.7.2.1 Topologie.....	56
5.7.2.2 Médias utilisés.....	57
5.7.2.3 Principe de fonctionnement.....	57
5.7.3 Critères de comparaison.....	61
5.8 PROFIBUS.....	62
5.8.1 Origine de PROFIBUS.....	62
5.8.2 Fonctionnement DE PROFIBUS.....	62
5.8.2.1 Topologie.....	62
5.8.2.2 Les couches du protocole.....	62
5.8.2.3 Médias utilisés.....	62
5.8.2.4 Principes de fonctionnement.....	63
5.9 CAN.....	66
5.9.1 Origine de CAN.....	66
5.9.2 Fonctionnement de CAN.....	66
5.9.3 Critères de comparaison.....	67
5.10 INTERBUS.....	68
5.10.1 Origine d'INTERBUS.....	68
5.10.2 Fonctionnement d'INTERBUS.....	68
5.10.3 Critères de comparaison.....	69
5.11 LON.....	70
5.11.1 Origine de LAN.....	70
5.11.2 Fonctionnement de LON.....	70
5.11.3 Critère de comparaison.....	70
5.12 EXEMPLE DE COMPARATIF DE BUS : WORLDFIP / PROFIBUS.....	71
6 COMMUNICATION INTER-USINE : LES RESEAUX DE TRANSPORT.....	72
6.1 NOTIONS DE BASE.....	72
6.2 LES PRINCIPES DE COMMUTATION.....	73
6.2.1 Commutation de circuits - Exemple : RTC.....	73
6.2.2 la Commutation de messages - Exemple : télex, Email.....	75
6.2.3 Commutation de paquets - Exemple : Transpac (X25), Internet (TCP/IP).....	76

6.2.4 Commutation de cellules - Exemple : ATM	77
6.2.5 MODE NON CONNECTÉ (CNLS)	77
6.2.6 MODE CONNECTÉ (CONS)	78
6.2.7 MODE ORIENTÉ CONNEXION.....	79
6.2.8 Commutation pour le réseau transpac	79
6.2.9 Circuit virtuel	79
6.2.10 Exemple d'établissement d'un circuit virtuel	81
6.2.11 le routage dans les nœuds.....	82
7 RESUME DES RECOMMANDATIONS DU CCITT	84
8 BIBLIOGRAPHIE.....	87

1 LA COMMUNICATION INDUSTRIELLE

1.1 Pré requis

La communication entre les constituants des systèmes industriels (PC, automates, commandes numériques, capteurs/actionneurs..) est une problématique technique de la conception des applications en informatique industrielle.

La mise en œuvre d'une solution de « communication » passe par la sélection et la mise en œuvre de couches logicielles de communication de types : pilotes système (driver), bibliothèques de fonctions de communication (API : « Application Programmable Interface », « handler »), serveurs de données (serveur DDE, OPC DA –HDA-AE...) et de divers matériels (coupleurs internes, ports de communication, interfaces réseaux..).

Sont nécessaires à la compréhension de ce domaine les connaissances relatives :

- aux besoins génériques en terme d'applications industrielles (à quoi sert la communication industrielle et quels sont les composants industriels qui sollicitent des services de communication) ;
- aux 7 couches du modèle OSI : les « composants matériels et logiciels » mis en œuvre pour la communication sont organisés et structurés en terme de leur réalisation et des services qu'ils proposent ;
- aux éléments d'architectures élémentaires de communication (circuit de données : DTE, DCE, modems, jonctions, ligne, composants de réseaux...) ;
- à la modélisation « événementielle » des couches de communication (par automates à états finis), utile aux phases de conception / réalisation (et d'étude / compréhension...) des différents « mécanismes d'échanges de données » ;
- aux techniques de transfert élémentaire de l'information (codage, mécanismes de transfert, modulation...)
- aux topologies de réseau (étoiles, maillés, bus, mixtes...)
- aux notions de protocoles (trames, adressage, checksum, maître, esclave, bus....)
- aux principaux « réseaux locaux industriels » (RLI) et Bus de communication, « standard de fait » (MODBUS, CAN..).

1.2 Panorama des besoins usuels du « monde industriel »

Les usines, les bâtiments, les voitures, les avions sont de plus en plus automatisés pour assurer sécurité, fonctionnalité et confort à l'être humain. Pour ce faire on a besoin de récolter des informations sur l'environnement par l'intermédiaire d'éléments simples tels que des capteurs et de les analyser. On fait traiter ensuite ces données par des outils, automates programmables ou ordinateurs.

Une fois les données analysées et traitées, on agit sur l'environnement par l'intermédiaire d'actionneurs si besoin est. On peut retrouver des éléments récoltant des informations tels que des capteurs de températures, des capteurs de pression, des détecteurs d'incendie, des détecteurs de personnes... et des éléments apportant une action tels que des moteurs pas à pas, des électrovannes, des champs électromagnétiques...

On comprend bien que tous ces éléments aient besoins de communiquer entres eux, dans certains cas comme pour la sécurité, la communication doit se faire de façon simple, rapide et fiable; dans d'autres cas, pour des motifs de coût, sont implémentation doit elle aussi être simple, rapide et fiable.

Lors de l'assemblage d'une voiture par exemple, la connexion des capteurs doit pouvoir se faire par un opérateur sans qu'il ait des connaissances techniques particulières. Ces éléments de par leurs nombres engendrent une multitude de fils nécessaires à leur alimentation et à la transmission de leurs données. Le bus de terrain (FieldBus) prend en compte toutes ces exigences, il a été conçu pour ça. La composition dun bus peut être décrite en deux parties, une partie matérielle composée d'un média physique sur lequel transitent l'alimentation électrique des éléments définis ci-dessus et leurs transferts de données et une partie logicielle composée d'une multitude de composantes logicielles fournissant le protocole de communication, les échanges d'informations, la supervision, la sécurité, les interfaces et bien d'autres fonctions.

1.2.1 EXEMPLES DE DOMAINES D'APPLICATION

Le bus de terrain peut être utilisés pour divers besoins, la sécurité, le confort, l'assistance, l'aide à la décision, la fonctionnalité, etc., l'utilisation de ces besoins peuvent être cumulés suivant les domaines d'application. Les domaines d'application peuvent être un bâtiment, une usine, une voiture, un bateau, un train, une centrale nucléaire, un hôpital.

Domaine d'application : Le bâtiment : pour des besoins de sécurité, les bâtiments haut de gamme possèdent des capteurs permettant de détecter la fumer provenant d'un incendie. L'ordinateur relié aux capteurs analyse les informations provenant des capteurs et envoie un ordre de fermeture à un mécanisme de commande des portes automatiques pour cantonner l'incident et un ordre aux actionneurs de commande des buses pour qu'elle propulse le liquide afin d'éteindre l'incendie. Pour des besoins de confort, les bâtiments climatisés possèdent des capteurs de température et des arrivées d'air régulés dans chaque pièce, si la température d'une pièce venait à changer,

l'automate envoie un ordre au mécanisme régulant l'arrivée d'air pour obtenir la température voulue.

Domaine d'application : la voiture, exemple de CAN :

Pour satisfaire les exigences de plus en plus importantes du client en matière de sécurité et de confort, et pour se conformer aux lois de réduction de la pollution et de la consommation de plus en plus drastiques, l'industrie automobile a développé de nombreux systèmes électroniques tels que le système anti-patinage, le contrôle électronique du moteur, l'air climatisé, la fermeture centralisée des portes, etc...

La complexité de ces systèmes et la nécessité d'échanger des données nécessitent de plus en plus de câbles. En plus du coût très important de ce type de câblage, la place qui lui est nécessaire peut le rendre tout simplement impossible. Enfin, le nombre croissant de connexions et de câbles pose de sérieux problèmes de fiabilité et de réparation.

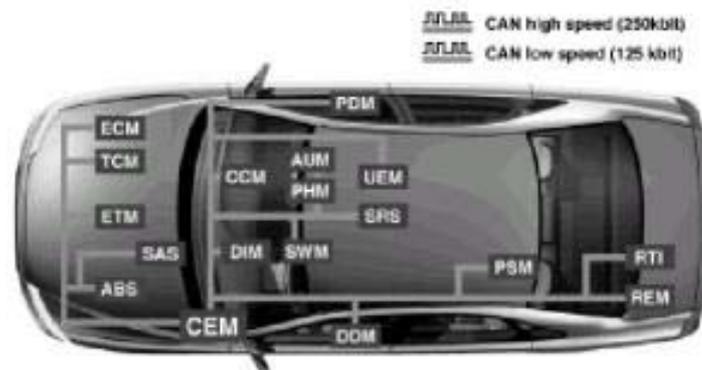
Bosch, un important équipementier automobile, a fourni la solution dans le milieu des années 80 avec le bus CAN. L'entreprise allemande a défini le protocole et a autorisé de nombreux autres fabricants à développer des composants compatibles CAN.

Avec le protocole CAN, les contrôleurs, capteurs et actionneurs communiquent entre eux sur deux câbles à une vitesse pouvant aller jusqu'à 1 Mbits/s.

Contrôle du moteur :

L'architecture CAN a été utilisée pour la première fois dans une automobile chez le constructeur Mercedes Benz qui liait ainsi plusieurs capteurs entre eux (température d'huile moteur, position du vilebrequin, détecteur de cliquetis, injection...). Le procédé a depuis été repris par de nombreux constructeurs tels que Renault, Saab, Audi, Peugeot ou Volkswagen.

Lorsque sont reliés entre eux de tels éléments de contrôle du moteur ou de sécurité (comme les capteurs d'ABS), le débit de transfert des données est généralement de 500 kbit/s.



Éléments de confort :

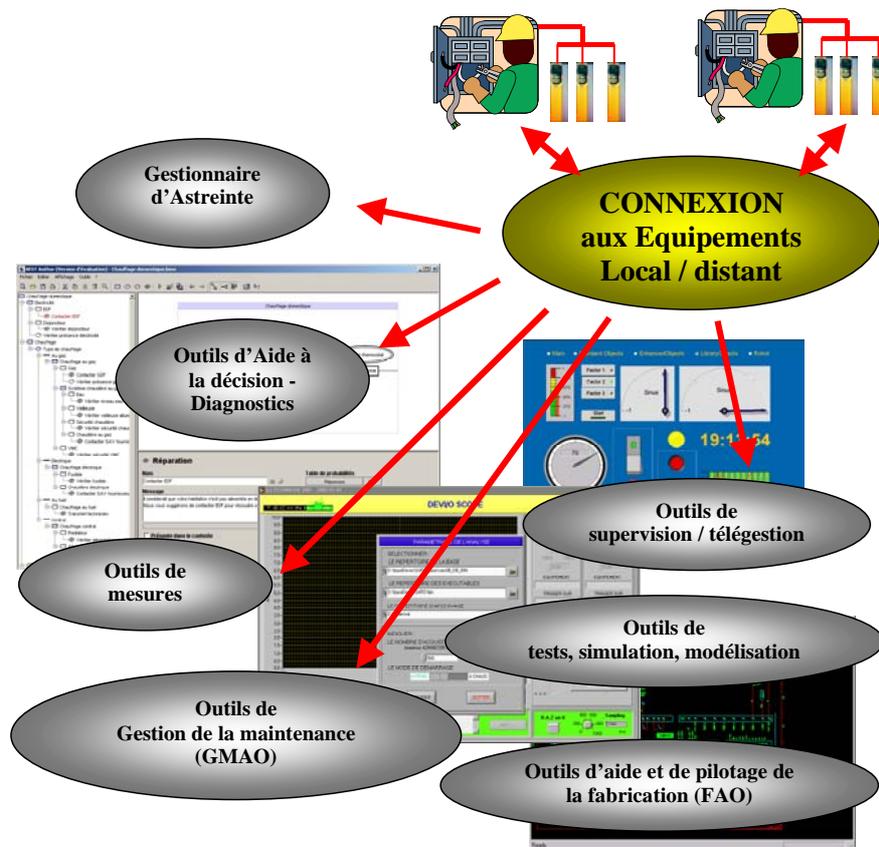
Le CAN utilisé dans une automobile permet de relier facilement plusieurs modules de divertissement et de confort comme la radio, le système de navigation GPS, le téléphone ou encore la climatisation.

Ces éléments peuvent également être interconnectés avec les modules de sécurité (par exemple le déclenchement des Airbags - coussins gonflables - qui provoque un appel téléphonique SOS envoyant ainsi aux secours via le GPS la position de l'automobile accidentée).

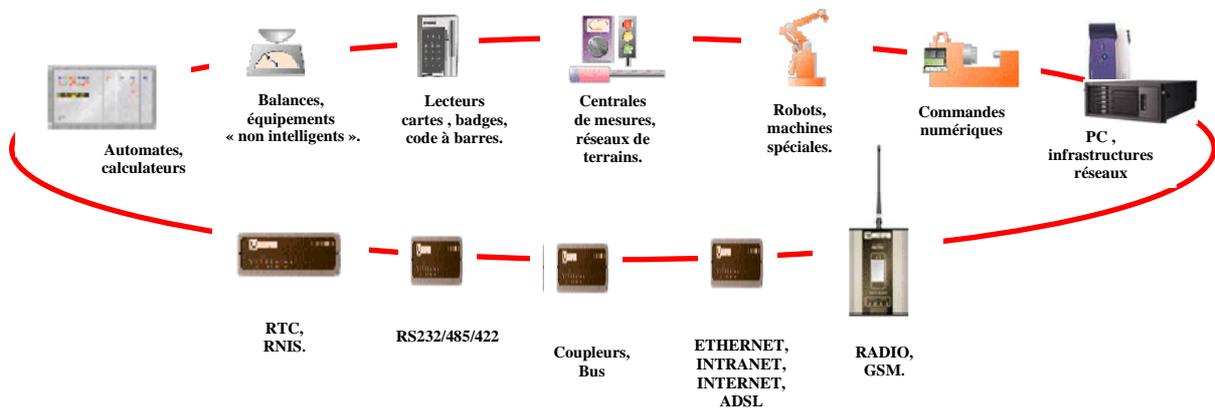


1.2.2

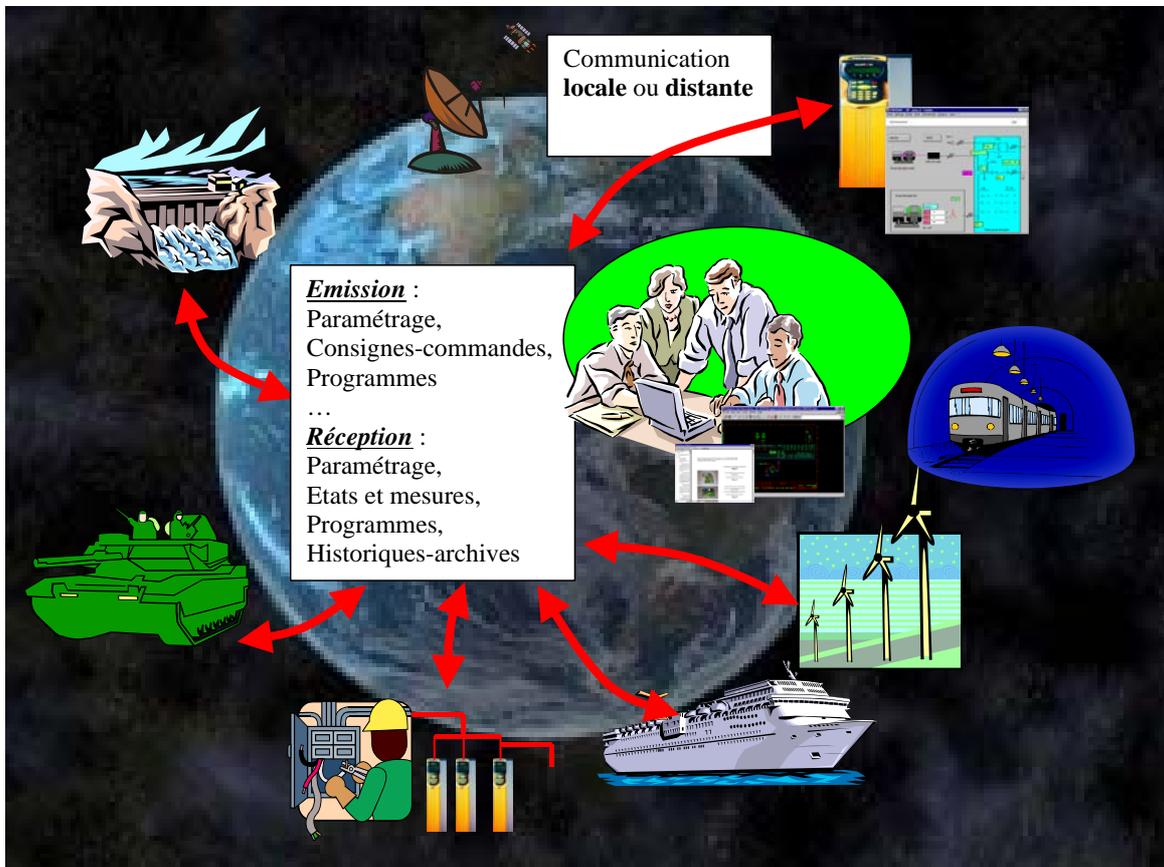
LES « PROGICIELS D'APPLICATION »



1.2.3 LES « EQUIPEMENTS »

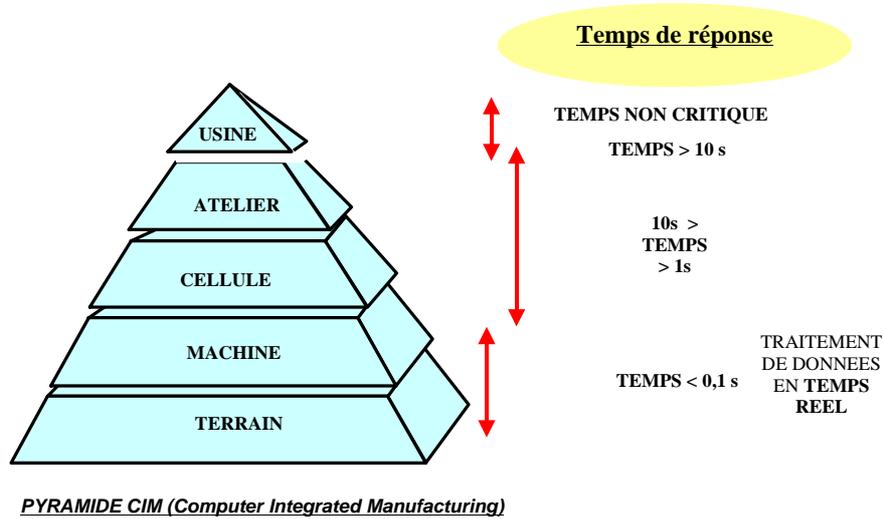
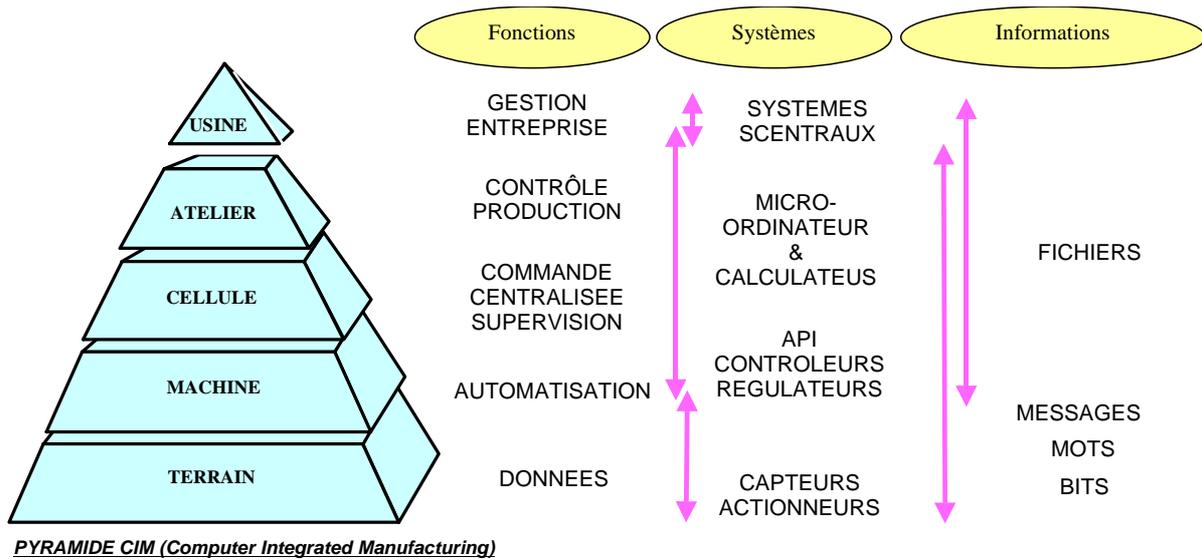


1.3 Les besoins et les échanges de données



Besoins :

- rapidité, fiabilité,
- traitements d'événements, de données cycliques et acycliques, de priorités,
- hétérogénéités des données : valeurs (formats : entier, booléen, réel), fichiers, historiques,
- identification des données, échanges des valeurs mais aussi des indicateurs de pertinence, de datation à la source,
- suivi des statuts de communication,
- communications distantes et locales
- ...



1.3.1

EXEMPLES DE « SOLUTIONS DE COMMUNICATION »

Ces exemples ne sont pas exhaustifs.

1.3.1.1 Protocoles

- Automatisme / télégestion : MODBUS série et TCP/IP, (« standard de fait incontournable : base de nombreux protocoles !! »)
- Automatisme : UNITEL-WAY , SINEC H1- L1 – L2, RK512 AS512, ...
- Télégestion (France) : SOFBUS PL, TRSii, NAPBUS, UTILOG, messageries propriétaires sur GSM-SMS...
- Divers “propriétaires” : STP403, BAC, ADLC, HNZ, MM4, SYSAL 2000 ...
- Réseaux & automatisme : SNMP

1.3.1.2 Bus



2 MODÈLE OSI (OPEN SYSTEMS INTERCONNECTION)

2.1 Modèle générique en couches

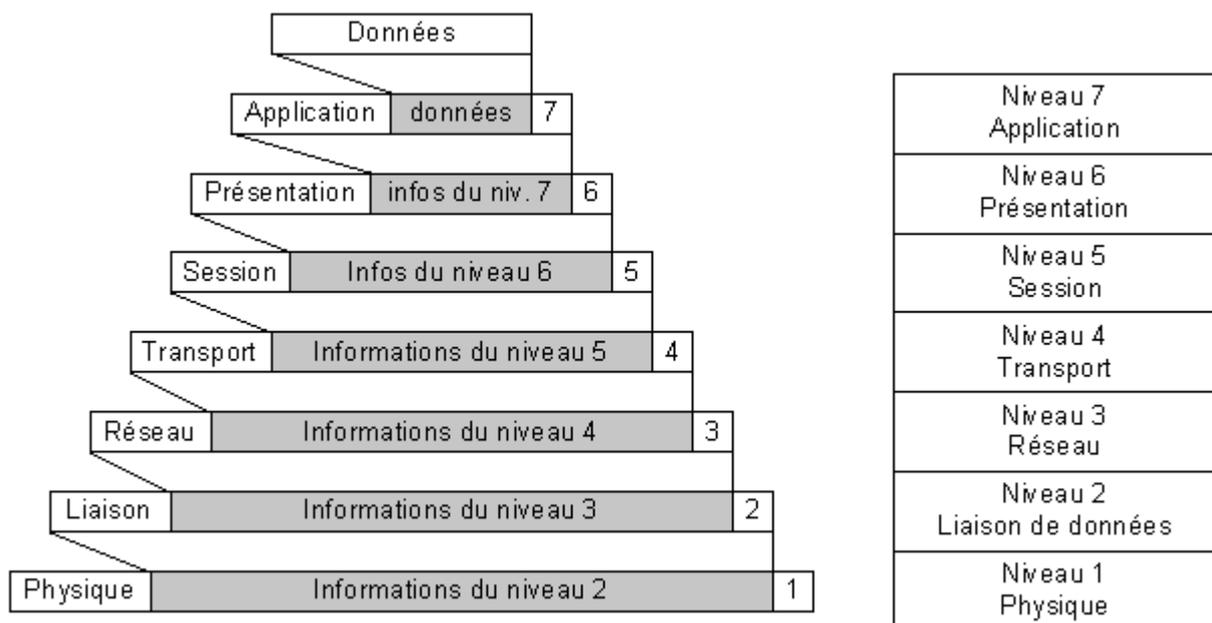
En 1977 l'Organisation Internationale de Normalisation (ISO) a créé pour des besoins de comptabilités entre les différentes machines tout un ensemble de lois de comptabilités baptisé **modèle OSI** (Open System Interconnection model). Celui ci est appelé modèle de référence OSI parce qu'il traite de la connexion entre les systèmes ouverts, c'est à dire des systèmes pouvant s'interfacer avec d'autres systèmes.

Le modèle OSI propose le découpage de la communication en 7 couches, afin de permettre de normaliser les méthodes d'échange entre deux systèmes. Chaque couche a un rôle bien particulier et communique sur requête (sur demande) de la couche supérieure en utilisant des services de la couche inférieure (sauf pour la couche physique 1).

Les données transférées par les services sont des SDU (Service Data Unit) Les échanges de l'information, suivant un protocole, avec les couches distantes de même niveau, sont des PDU (Protocole Data Unit).

Cette décomposition en couches a été créée pour simplifier considérablement la compréhension globale du système et pour faciliter sa mise en œuvre. On doit pouvoir remplacer une couche par une autre couche de même niveau, sans avoir à changer les autres niveaux.

Les différentes couches sont les suivantes :



Le **septième niveau**, la couche **Application**, gère le transfert des informations entre programmes. Cette couche a pour objectif de fournir des services aux utilisateurs d'un réseau. C'est elle qui contient l'application informatique (le programme) qui désire communiquer avec un ordinateur distant. C'est à ce niveau qu'on rencontrera des programmes transfert de fichiers, d'émulation de terminal, de soumission de travaux à distances, d'échange de courrier électronique, etc..

Le **sixième niveau**, la couche **Présentation**, s'occupe de la mise en forme des textes et des conventions d'affichage. Elle permet de fournir une représentation des données, autrement dit une représentation qui ne dépend pas des ordinateurs, systèmes d'exploitation, etc.. et inclus services tels que le cryptage, la compression et le formatage des données.

En effet, il existe de multiples manières de coder les informations en informatique suivant le matériel et les logiciels utilisés. Par exemple:

- Plusieurs codes existent pour coder les caractères (ASCII, BCIDC, etc.).
- Les nombres peuvent être codés sur un nombre d'octets différents.
- Les octets de poids fort et de poids faible peuvent être répartis différemment, autrement dit, un nombre peut être lu de gauche à droite ou de droite à gauche.
- Etc.

Le **cinquième niveau**, la couche **Session**, s'occupe de l'établissement, de la gestion et coordination des communications. Cette couche offre la possibilité d'organiser les échanges en unités indépendantes. Elle offre aussi une structure de contrôle pour la communication entre applications. Elle établit, maintient et clôt les sessions entre les applications. L'un des points forts de cette couche est la sécurité.

Organisation de la communication:

- Droit à la parole (half/full duplex => communication simultanée ou l'un après l'autre).
- Notion d'activité: on peut la démarrer, l'arrêter, l'interrompre et la recommencer.
- Points de synchronisation.
- Etc.

La couche session est aussi la première partie de l'architecture de réseau hors de la communication proprement dite.

Le **quatrième niveau**, la couche **Transport**, gère la remise correcte des informations. Elle est chargée d'établir les connexions, de maintenir la qualité de la connexion et d'interrompre cette dernière de manière ordonnée une fois la conversation terminée. Cette couche transporte des blocs d'octets de longueur quelconque. Elle s'assure que les données sont délivrées sans erreur et dans l'ordre.

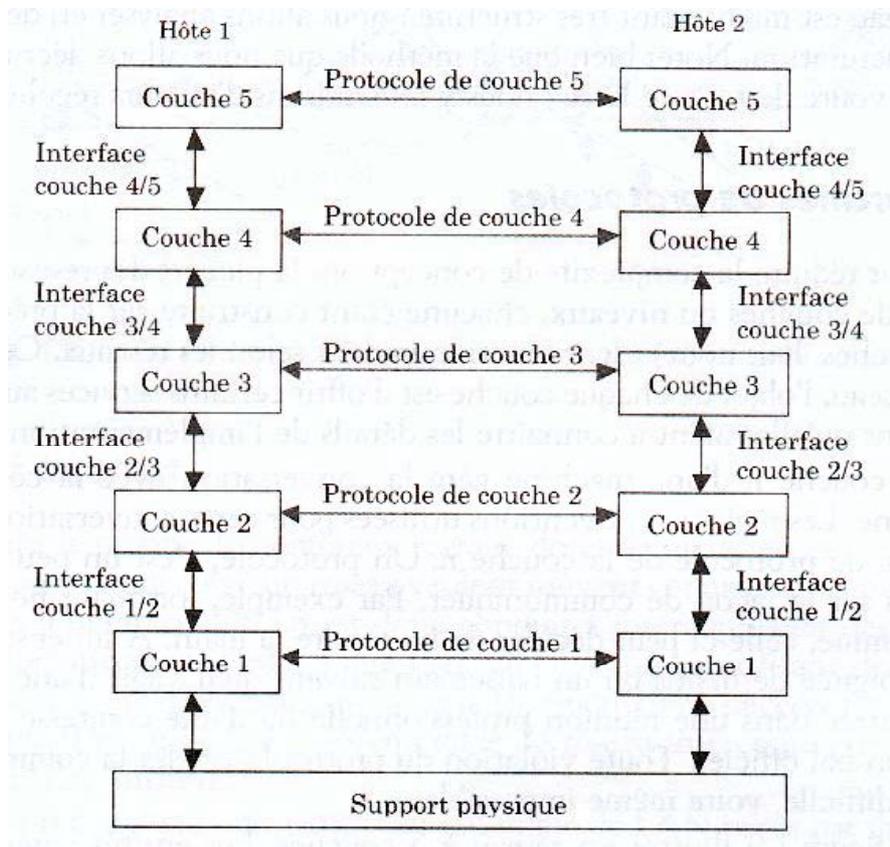
Vient ensuite **le niveau trois**, la couche **Réseau**, qui détermine les routes de transport et qui s'occupe du traitement et du transfert de messages. Cette couche permet aux couches supérieures d'être indépendantes des différents types de liaisons de données ou technologies de transmissions. Elle transporte des blocs d'octets de taille limitée. Elle s'occupe de l'adressage et du routage des paquets à leur destination et a donc besoin d'un plan d'adressage, ainsi que du contrôle de flux. Elle est aussi responsable de l'établissement d'une connexion logique entre source et une destination sur un réseau.

Le **niveau deux**, la couche **Liaison** de données, s'occupe du codage, de l'adressage, et de la transmission des informations. Cette couche fournit les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintien et à la libération des connexions entre entités de réseaux et est chargée d'acheminer sans erreur les données sur chaque liaison du réseau en masquant aux autres couches les différences physique du réseau. Elle assemble les données en blocs, auxquels elle ajoute des informations de contrôle pour constituer une trame de données : l'adresse de destination, la longueur du message, l'information de synchronisation, de détection d'erreur, etc. Les protocoles qui fonctionnent à ce niveau délivrent des données de carte à carte.

Le **premier niveau**, la couche **physique**, gère les connections matérielles. Elle permet de véhiculer l'information et de transformer des séquences de bits (0 ou 1) en séquence de grandeur physique appropriée au médium de communication. Elle fournit aussi les caractéristiques mécaniques (connecteurs), fonctionnelles (activation et désactivation de la connexion physique), ainsi que les signaux (électriques ou optiques). Cette couche est matérialisée par le câble, les connecteurs de niveau « hardware » et les microcontrôleurs et pilotes « système ».

Les quatre premières couches sont les couches réseaux. Elles transportent physiquement les données d'une application vers une autre, sans erreurs. Les trois autres sont chargées de formater les informations et de procurer des voies d'accès multiples à la même application.

2.2 Services et primitives



Les couches d'un même « hôte » dialoguent entre elles par mise en œuvre de « services ». Les couches de même niveau « N » dialoguent entre elles par mise en œuvre d'un « protocole ».

Chaque couche « N » échange des **services** avec sa couche supérieure « N+1 ». La couche N+1 est **utilisateur** des services, la couche N est **fournisseur** des services.

Une couche est découpée en **entités** actives (fonctions logiciel ou matériel). Les entités de deux couches de même niveau localisées sur deux hôtes différents (1 et 2) communiquent entre elles en mettant en œuvre un **protocole**. Le protocole définit l'ensemble des règles qui s'appliquent au format et à la signification des trames, paquets ou messages échangés entre entités paires sur des machines différentes au sein d'une même couche. Les points d'entrée dans un service donné localisé dans une couche sont identifiés par une adresse appelée **SAP** (Service Access Point).

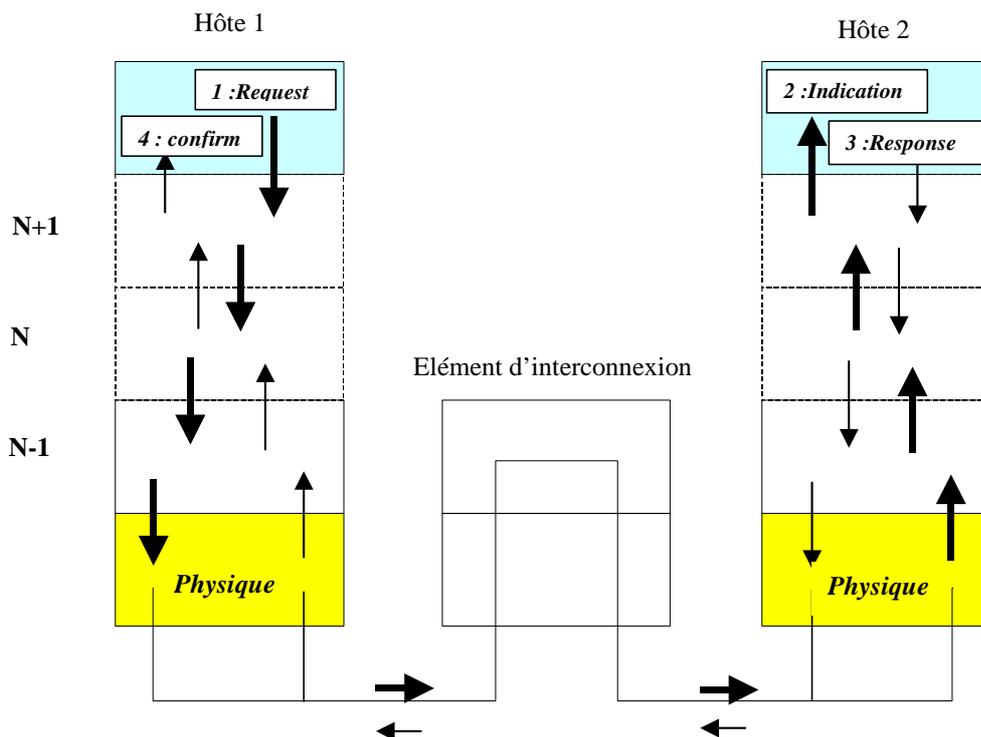
Exemple non exhaustif de services :

- Demande de connexion (**CONNECT**)
- Emission de données (**SEND**)
- Réception de données (**RECEIVE**)
- Demande de déconnexion (**DISCONNECT**)

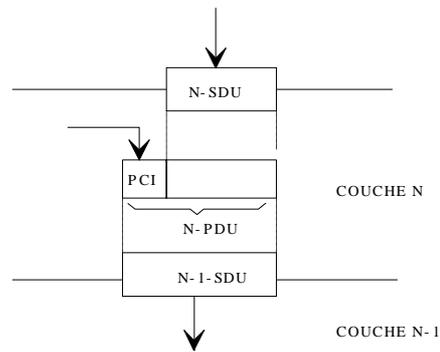
Une demande de service se fait par mise en œuvre de **primitives** :

- Requête (**REQUEST**) : demande de service (couche N sur le hôte 1)
- Indication (**INDICATION**) : indication d'une demande de service (couche N sur le hôte 2)
- Réponse (**RESPONSE**) : réponse à la demande (couche N sur le hôte 2)
- Confirmation (**CONFIRM**) : confirmation du service (couche N sur le hôte 1)

On nomme les primitives en fonction des services auquel elles se rapportent : exemple **CONNECT.REQUEST**, **CONNECT.INDICATION**, **CONNECT.RESPONSE**, **CONNECT.CONFIRM**.



Une primitive de service met en œuvre un transfert d'informations entre les couches. Ces informations sont encodées dans un paquet appelé **SDU** (Service Data Unit). Chaque couche recevant un SDU interprète ce SDU pour créer par suite un **PDU** (protocol Data Unit). Ce PDU contient la transcription du SDU ainsi que des informations propres au protocole de la couche, appelées PCI (protocol control informations). Ce PDU est transmis à la couche de niveau inférieur au travers d'un SDU qui l'encapsule. C'est le principe d'**encapsulation**.



Par convention, on associe une à deux lettres à chaque couche (le nom de la couche est e en anglais...) :

- **A** : Application
- **P** : Presentation
- **S** : Session
- **T** : Transport
- **N** : Net
- **L** : Link

Chaque SDU propre à une couche N est créé par la couche supérieure N+1 et est préfixé de la lettre associée à la couche N.

3 NOTIONS DE MODELISATION UTILES A L'ETUDE DES COUCHES PHYSIQUES ET LIAISON : LES AUTOMATES A ETATS FINIS

3.1 Les machines d'états finis

Une machine d'états finis est un système abstrait que nous pouvons décrire par un graphe orienté étiqueté, composé d'entrées, d'états et parfois de sorties. La machine peut se trouver dans l'un des états qui sont en nombre fini. Les machines à états finis est le formalisme le plus simple et le moins expressif de la théorie des automates.

Les réseaux de télécommunications et particulièrement les protocoles de communication sont un domaine par excellence pour l'utilisation des automates. L'utilisation la plus courante des automates pour les télécommunications concerne les spécifications. Un état de l'art est donné dans l'ouvrage « R.L. Schwartz and P.M. Melliar-Smith. From state machines to temporal logic: Specification methods for protocol standards. volume 30(12), pages 2486-2496. IEEE Transactions on Communications, 1982. »

Les **automates à états finis** permettent de décrire le comportement d'un programme de façon formelle. Ce **comportement** est décrit au moyen **d'états** et **d'événements**. Les programmes suivent le comportement décrit par l'automate, ainsi à un instant donné, on peut dire que le programme se trouve dans un état précis.

Par exemple un fonctionnement de modem peut-être vu comme un automate, il passe successivement des états initialisés, à en attente de connexion, à connecté...

3.2 Les états

Chaque programme est à un moment donné dans un état particulier. Les états se représentent à l'aide de cercles. Les états se caractérisent par la notion de durée et de stabilité. **Un programme est toujours dans un état donné pour un certain temps** et un programme ne peut être dans un état inconnu ou non défini. Un état est toujours la conjonction instantanée des valeurs contenues par les données du programmes, et de la présence ou non de liens, du programme considéré vers d'autres programmes (par exemple une couche OSI peut être en communication avec une autre couche).

Les automates sont déterministes : Un automate à états finis ne doit pas être ambiguë. Cela signifie que l'on doit toujours avoir un état initial pour l'automate : **il y a toujours un état initial**. En revanche, il est possible d'avoir **plusieurs états finaux** qui correspondent chacun à une condition de fin différente. Il est également possible de n'avoir aucun état final, dans ce cas le système ne s'arrête jamais.

Voici la représentation généralement adoptée pour les états initiaux et finaux.



Figure 1 : Représentation des états initiaux et finaux

3.3 Les transitions et événements

Lorsque les conditions dynamiques évoluent, les objets changent d'état en suivant les règles décrites par l'automate. Les automates à états finis sont des graphes orientés. Les états sont reliés par des connexions unidirectionnelles, appelées transitions (ou liens). Le passage d'un état à un autre s'effectue lorsqu'une transition est déclenchée par un événement. Le passage d'un état à un autre est instantané car le système doit toujours être dans un état connu.

Les transitions sont représentées au moyen d'une flèche qui pointe de l'état de départ vers l'état d'arrivée (cf. Figure 1).

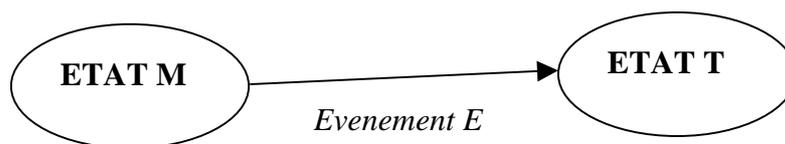
La transition d'un état X à un état Y sur réception d'un événement E s'accompagne de l'exécution d'une action A (fonctions informatique dans notre cas).

3.4 Représentation sous forme de table de transitions

L'automate peut être représenté sous forme d'un tableau où :

- chaque colonne correspond à un état donné
- chaque ligne correspond à un événement donné
- chaque case indique la transition en terme d'action à effectuer et d'état final à atteindre après exécution de l'action.

	ETAT1	ETAT2	ETAT M	ETAT Z
EVENEMENT1				
EVENEMENT E			ActionY() ETAT T	
EVENEMENT X				



↓

**La transition entraîne
le déroulement de
l'ACTION Y**

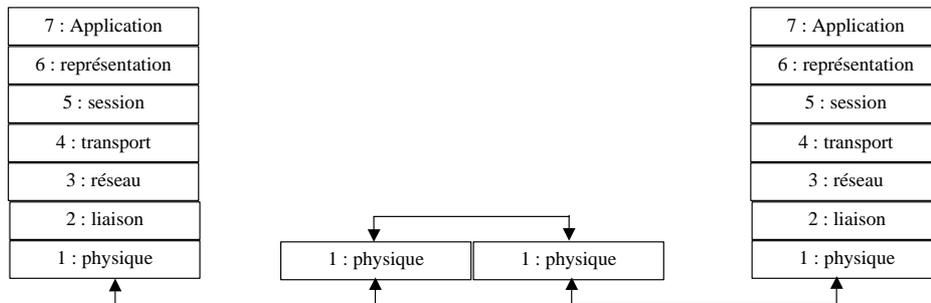
4 RLI ET BUS DE COMMUNICATION

4.1 Réseau et Bus de terrain

- **TERRAIN** = Délimite une zone géographique (Usine, Atelier, Voiture...)
 - **BUS** = Ensemble des fils utilisés pour l'échange de données entre différents circuits électroniques
 - **RESEAU** = Ensemble de lignes de communication qui desservent une même unité géographique
- Topologies**
- anneau
 - étoile
 - bus
 - arbre

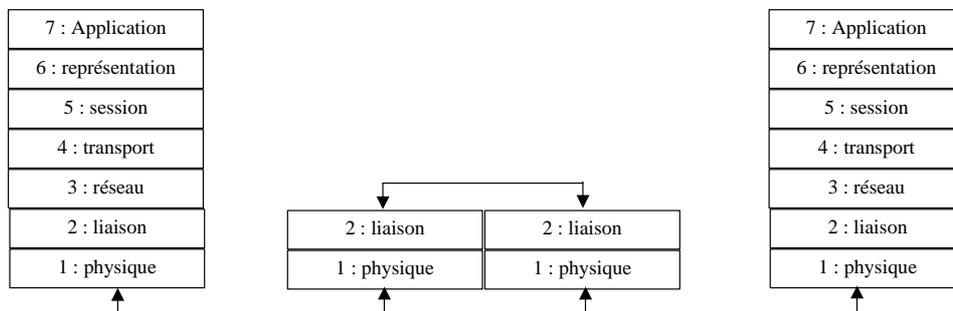
4.3 Éléments d'interconnexion

4.3.1 REPETEURS OU AMPLIFICATEURS - CONVERTISSEURS



- amplification du signal pour augmenter la distance
- conversion de signaux (RS485 vers fibre optique)

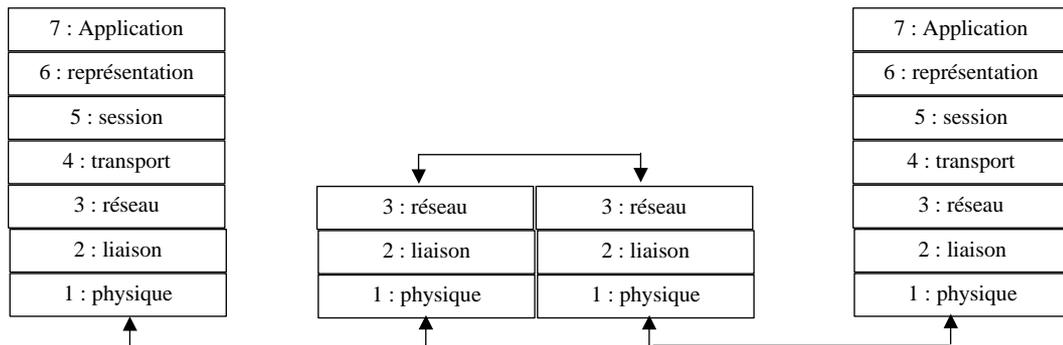
4.3.2 PONTS



- conversion de signaux (couche 1)
- conversion des formats des trames (couche 2)

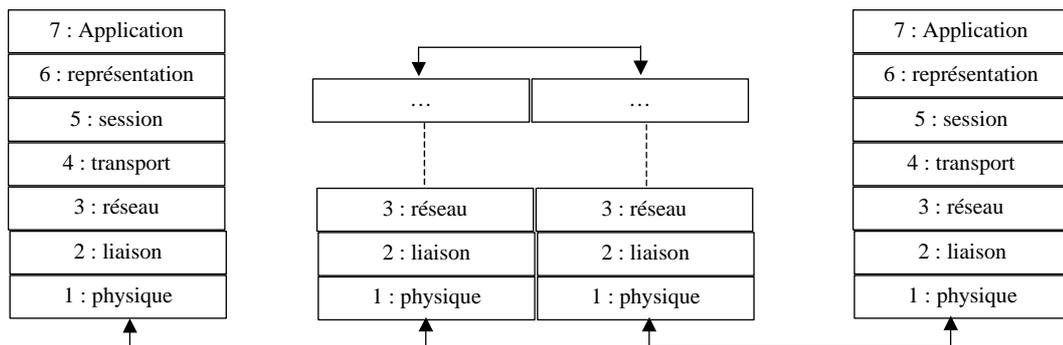
4.3.3

ROUTEURS



- conversion de format des paquets et notamment des adresses
- routage des paquets suivant les adresses entrantes vers des liens prédéfinis (sous-réseaux)
- système intelligent (diminution du débit)

4.3.4 PASSERELLES



- conversion de formats de messages d'une couche des couches supérieures (4 à 7)

4.4 Restriction du modèle OSI

Les « bus de terrain » sont basés sur une restriction du modèle OSI : Seules sont implantées les couches :

- physique
- liaison de données
- application.

Le modèle OSI gère les grands réseaux à commutation de paquets (domaine des télécommunication). Le temps n'a pas été pris en compte. Pour les réseaux locaux, la notion de temps réel est un point très important. La couche physique est indispensable à la communication. La couche liaison de données aussi pour la détection des erreurs. Les couches réseaux et transport ont été définies pour gérer les problèmes des paquets qui transitent par des stations intermédiaires : elles n'ont plus lieu d'être pour les RLI car toutes les stations sont interconnectées. La couche session permet l'échange d'une grande quantité d'information, ce qui n'est pas le cas pour les RLI. La couche présentation peut être figée et non dynamique ce qui la rend transparente. La couche application reste évidemment nécessaire.

Il en résulte **un modèle (IEEE 802) à trois couches**

Le standard international **ISA/SP50** a en plus normalisé la partie applicative et propose en plus :

- une **couche utilisateur** qui définit la « stratégie de contrôle distribué », modélisée sous la forme de blocs fonctionnels et une **base de données distribuée** sur le réseau, pour le contrôle et l'acquisition ;
- une fonction de **supervision** : configuration, monitoring, contrôle des ressources du réseau.

4.5 Couche physique

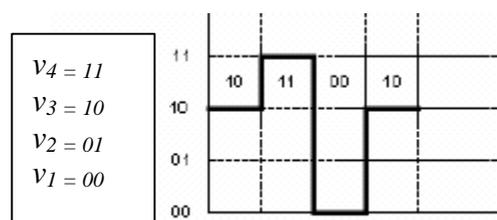
La couche physique traite des **moyens mécaniques, électriques et fonctionnels** (procédures et protocoles de bas niveaux) nécessaires à l'activation, au maintien et à la désactivation des liaisons physiques mises en œuvre pour la transmission de bits **entre deux entités** du circuit de données (donc entre DTE<->DCE, DCE<->DCE ou entre DTE <-> DTE, si aucun DCE n'est utile).

La couche physique gère la transmission des bits de façon brute : et fiable (être sûr qu'un bit à 0 ou à 1 reçu était bien un bit à 0 ou à 1 émis). La couche est régie par des **normes et standards** qui définissent les types de connecteurs (mécaniques) de signaux émis (mécanique et électrique), les caractéristiques (mécaniques et électriques) des supports de type câbles, fibres optiques..., les mécanismes de transmission (fonctionnels) synchronisation, sens des échanges, détection d'erreurs de bas niveau...).

La transmission des bits peut s'effectuer **en série ou en parallèle**. Quel que soit le mode de transmission, l'émission est toujours cadencée par une **horloge** dont la vitesse donne le débit de la ligne exprimé en *bauds*. **Les bauds caractérisent le nombre de tops d'horloge générés une seconde**. Par exemple, un débit de 100 bauds autorise 100 émissions de bits par seconde. **Si à chaque top d'horloge un signal représentant 0 ou 1 est émis, alors, dans ce cas le débit en bits est équivalent au débit en baud**.

On peut, suivant les systèmes électroniques utilisés, imaginer que le signal émis puisse prendre n valeurs distinctes durant un top d'horloge. Si le signal peut prendre 2^n valeurs, on appelle n sa **valence** : si le débit de la ligne est de x bauds il est en fait de $n.x$ bit/s.

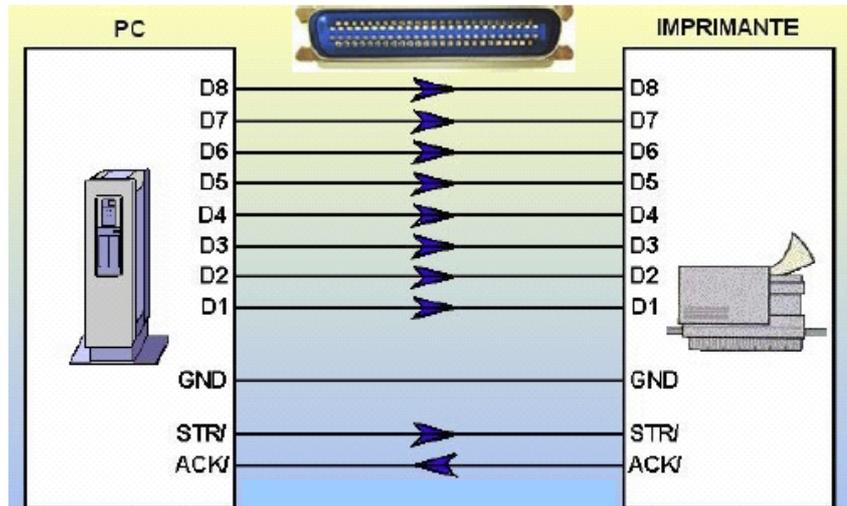
Le signal peut prendre 4 valeurs v_1, v_2, v_3, v_4 à chaque top d'horloge



4.5.1

LIAISON PARALLELE

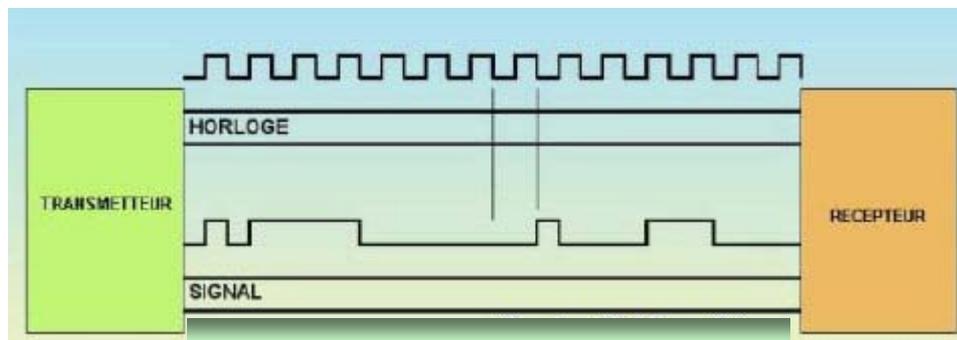
Elle n'est pas utilisée dans les RLI. On peut la rencontrer dans le « domaine de la mesure » où l'on veut du « débit ». Les imprimantes peuvent se connecter en « liaison parallèle », via le connecteur CENTRONICS)



4.5.2 LIAISON SERIE

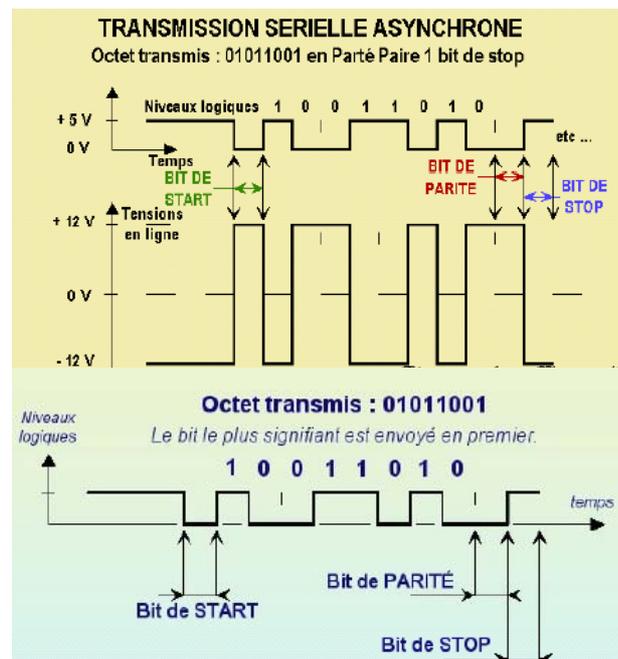
En série, les bits sont envoyés les uns derrière les autres de manière synchrone ou asynchrone.

Dans le **mode synchrone** l'émetteur et le récepteur sont synchronisés sur une même base de temps (un top d'horloge) qui se répète régulièrement durant tout l'échange. L'horloge est par exemple transmise sur un lien physique dédié, entre les deux systèmes (ce qui est trop coûteux ou impossible à faire, surtout sur de grandes distances..). Les signaux de transmission et réception sont échantillonnés suivant cette horloge.



Dans le **mode asynchrone**, les deux systèmes gèrent leur horloge individuellement (leur top). Les horloges sont programmées sur une même valeur théorique.

Chaque information envoyée est précédé d'un **bit de start** (de valeur un top d'horloge) et est suivie d'un **bit de stop** (ou plus précisément d'un **temps de stop**). Le bit de stop peut prendre les valeurs 1, 1.5 ou 2... Le nombre de bits de chaque information est défini pour chaque système et doit être identique (exemple format des données sur 8 ou 7 ... ou anciennement 5 bits). Les *bits de start* et de stop servent à caler l'horloge du récepteur pour qu'il échantillonne le signal reçu. L'état électrique associé au bit de start est différent de l'état électrique associé à la non transmission de données (état de repos de la ligne).



En mode asynchrone, les échantillonnages n'étant pas calés entre les systèmes, des erreurs peuvent survenir (exemple des dérives et précision des horloges) : un bit de contrôle peut être introduit dans la séquence à des fins de détection d'erreurs. Ce bit est appelé **bit de parité**.

La parité est gérée suivant 5 modes possibles, et il va de soit que les systèmes doivent tous adopter un même mode à un instant T :

- **pas de parité** (No parity): aucun de contrôle n'est utilisé.
- **parité paire** (even parity): le nombre total de bits à 1, trouvés entre le bit de start et le bit de stop doit être paire... Si le nombre de bit à 1 des données est paire, le bit de parité est positionné à 0 .
- **parité impaire** (odd parity): le nombre total de bits à 1, trouvés entre le bit de start et le bit de stop doit être impaire... Si le nombre de bit à 1 des données est paire, le bit de parité est positionné à 1
- **parité forcée à 1** (mark parity): le bit de parité vaut toujours 1.
- **parité forcée à 0** (space parity): le bit de parité vaut toujours 0.

Une liaison entre 2 équipements peut être :

- **simplex** (unidirectionnelle). L'un des équipements est toujours l'émetteur et l'autre récepteur. Ce cas est très rare en RLI.
- **half-duplex** (bidirectionnelle à l'alternat). A un instant T, l'un des équipements est émetteur, l'autre récepteur (comme avec des talkies-walkies). Alternativement, ces rôles sont inversés.
- **full-duplex** (bidirectionnelle simultanée). A tout instant, les équipements peuvent émettre (et donc recevoir) simultanément.

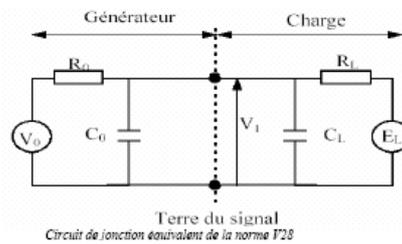
4.5.2.1 Principales normes des liaisons séries utilisées dans les RLI.

Principales normes

Débits	Normes		
	Mécaniques	Electriques	Fonctionnelles
< 20 kbit/s	ISO 2110 DB25	CCITT V28	CCITT V24
	EIA RS232C		
> 20 kbit/s	ISO 2593 DB34	CCITT V11 EIA RS422 - RS485	CCITT V24 RS449
Réseau analogique			
> 20 kbit/s	ISO 4903 DB15	CCITT V11 EIA RS422 - RS485	CCITT X21
Réseau numérique			

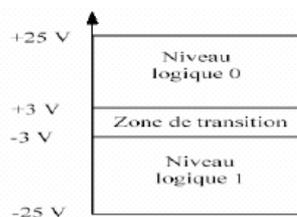
4.5.2.2 Normes V28 (électrique).

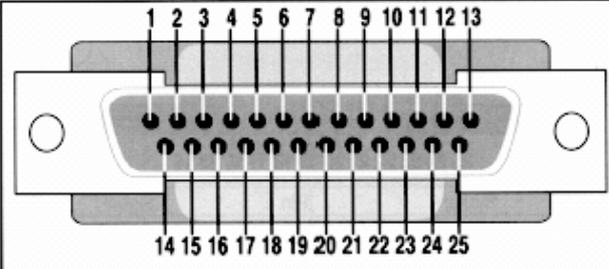
La figure suivante donne les niveaux électriques significatifs de la norme V28.



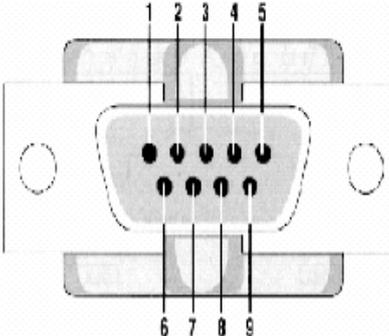
Caractéristiques	Min.	Max.
Tension du générateur en circuit ouvert (V_0)		± 25 V
Courant de sortie en court-circuit (I_0)		0.5 A
Tension de charge en circuit ouvert (E_L)		± 2 V
Tension de jonction sur charge R_L pour $E_L=0$ (V_1)	± 5 V	± 15 V
Résistance de charge (R_L)	3 k Ω	7 k Ω
Capacité parallèle (C_1)		2500 pF
Support	1 fil	
Vitesse	20 kbit/s	
Longueur de câble	50 m	

Principales caractéristiques électriques de la norme V28





Pin	Description	EIA CKT	From DCE	To DCE
1	Frame Ground	AA		
2	Transmitted Data	BA		D (Data)
3	Received Data	BB	D	
4	Request to Send	CA		C (Control)
5	Clear to Send	CB	C	
6	Data Set Ready	CC	C	
7	Signal Gnd/Common Return	AB		
8	Rcvd. Line Signal Detector	CF	C	
11	Undefined			
12	Secondary Rcvd. Line Sig. Detector	SCF	C	
13	Secondary Clear to Send	SCB	C	
14	Secondary Transmitted Data	SBA		D
15	Transmitter Sig. Element Timing	DB	T (Timing)	
16	Secondary Received Data	SBB	D	
17	Receiver Sig. Element Timing	DD	T	
18	Undefined			
19	Secondary Request to Send	SCA		C
20	Data Terminal Ready	CD		C
21	Sig. Quality Detector	CG		C
22	Ring Indicator	CE	C	
23	Data Sig. Rate Selector (DCE)	CI	C	
23	Data Sig. Rate Selector (DTE)	CH		C
24	Transmitter Sig. Element Timing	DA		T
25	Undefined			



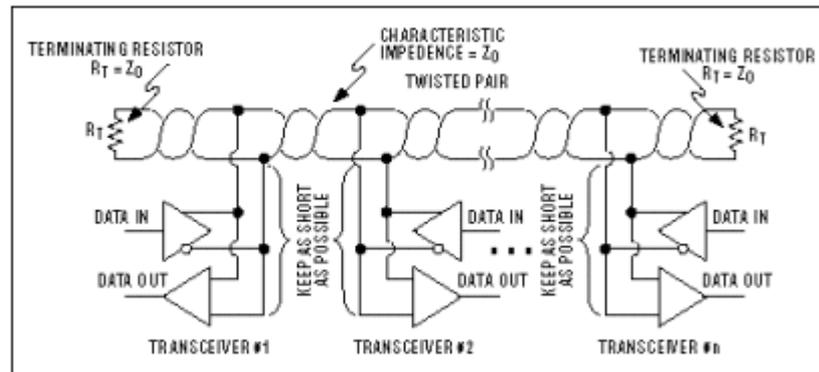
Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

4.5.3

INTERFACE RS232

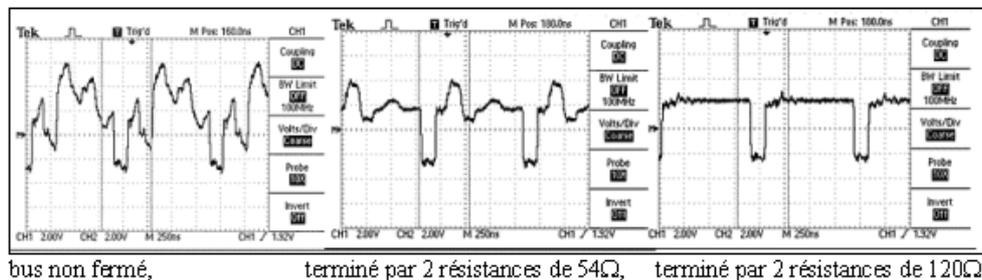
4.5.4 INTERFACE RS485

Câblage correct du bus :

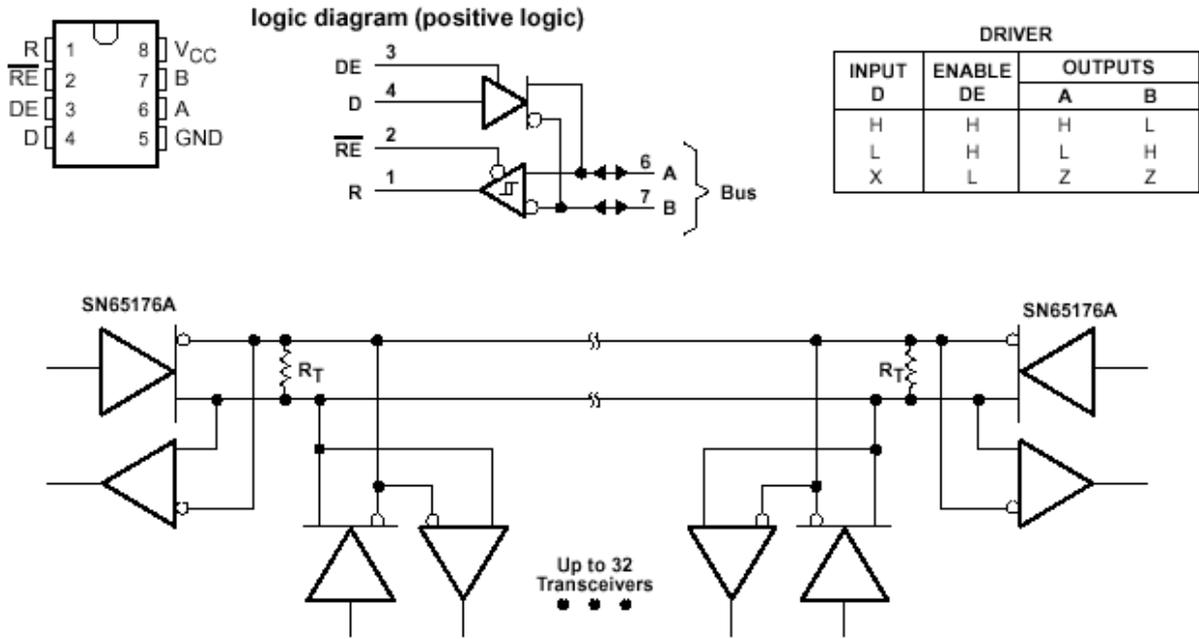


Le bus doit être fermé aux 2 extrémités par des résistances de 120 pour réduire les perturbations. Le bus doit être linéaire et non en étoile. Il est conseillé que la longueur des raccords en « T » soit réduite au maximum.

Influence des résistances de fin de bus :



Une interface RS485 peut aisément être conçue avec l'utilisation des composants MAX232 qui assure l'adaptation TTL - +/-12V et MAX487 (ou 65176A ou 75176A) qui réalise la conversion TTL - Différentiel.



NOTE A: The line should be terminated at both ends in its characteristic impedance ($R_T = Z_0$). Stub lengths off the main line should be kept as short as possible.

Prendre $R_T = 100 \text{ Ohms}$

4.5.5

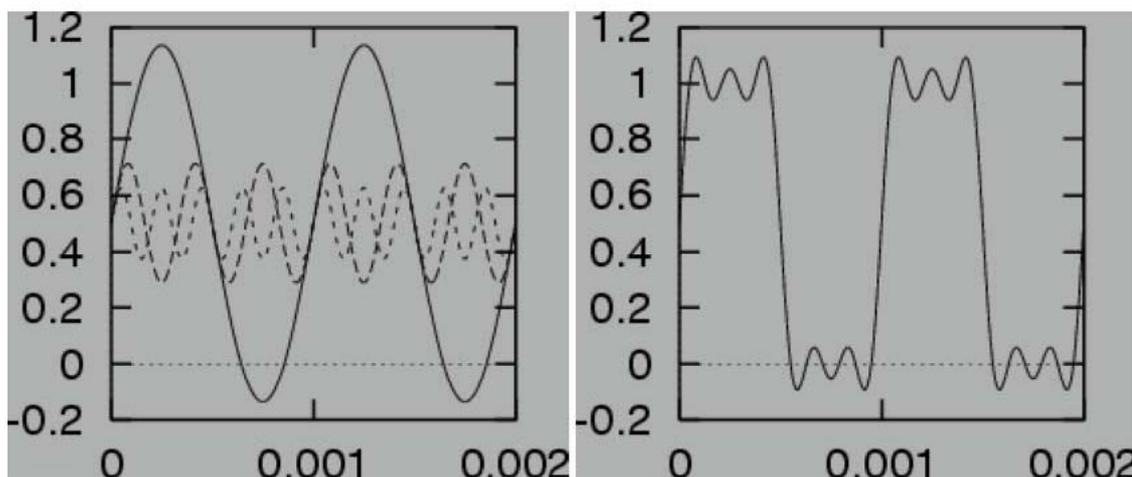
MODES DE TRANSMISSION

4.5.5.1 Transmission en bande de base.

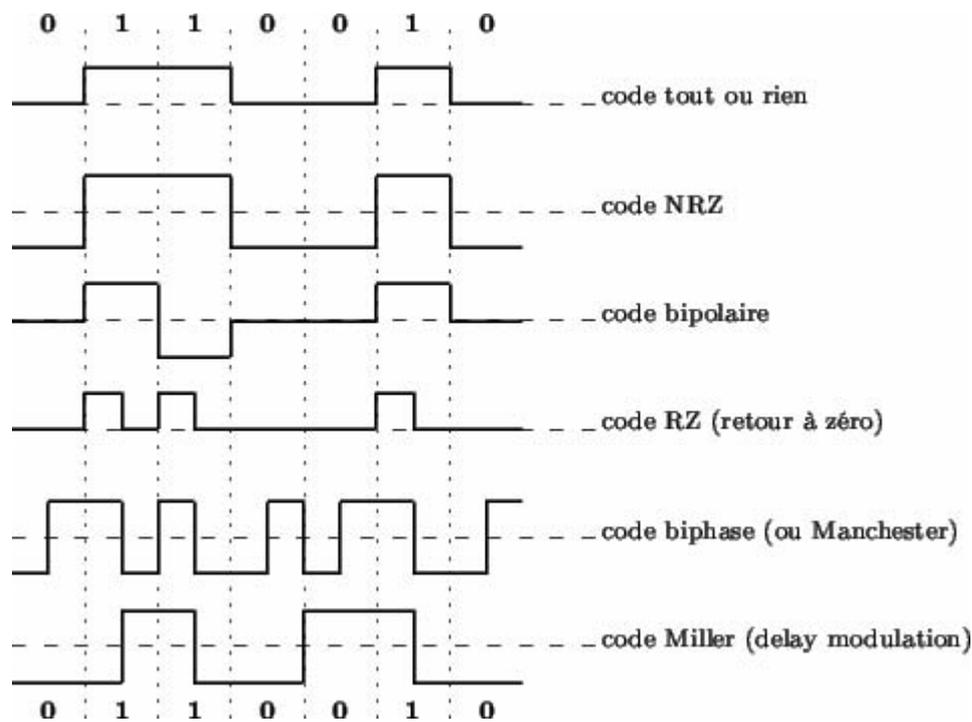
La transmission en bande de base consiste à envoyer directement les suite de bits sur le support à l'aide de *signaux carrés* constitués par un courant électrique pouvant prendre 2 valeurs (5 Volts ou 0 par exemple).

Le signal peut-être décomposé en une somme infinie d'*harmoniques*, la première étant dénommée *fondamentale* (Fourrier).

Cependant, le câble sur lequel est émis le signal possède une **bande passante** qui est l'intervalle des fréquences possibles sur ce support, donc à la réception, toute les harmoniques du signal d'origine ne sont pas forcément reçues, et dans la plupart des cas un signal carré décomposés sera très déformé. Par exemple, le câble téléphonique a une bande passante de 300 à 3400 Hz, donc tous les signaux de fréquence inférieure à 300 ou supérieure à 3400 seront éliminées.



La figure ci-avant présente les harmoniques et transformée de Fourier de la séquence de bits 1010. Dans la figure nous trouvons à gauche les 3 premières harmoniques et on remarque que plus la fréquence augmente plus l'amplitude diminue. À droite nous avons le signal réellement perçu par le récepteur si l'on considère que le câble ne laisse passer que ces 3 harmoniques-ci. Dans ce cas le signal reçu reste assez proche du carré émis et le récepteur n'aura pas trop de mal à le décoder.



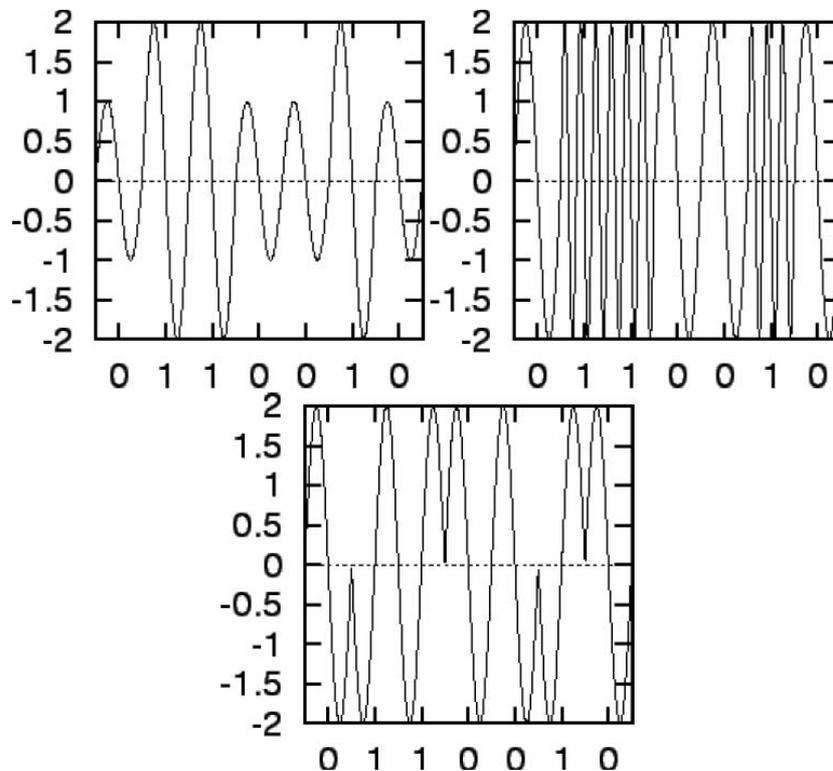
Dans la figure nous trouvons quelques exemples de codage de l'information pour une transmission en bande de base.

- le **code tout ou rien** : c'est le plus simple, un courant nul code le 0 et un courant positif indique le 1
- le **code NRZ** (non retour à zéro) : pour éviter la difficulté à obtenir un courant nul, on code le 1 par un courant positif et le 0 par un courant négatif.
- le **code bipolaire** : c'est aussi un code tout ou rien dans lequel le 0 est représenté par un courant nul, mais ici le 1 est représenté par un courant alternativement positif ou négatif pour éviter de maintenir des courants continus.
- le **code RZ** : le 0 est codé par un courant nul et le 1 par un courant positif qui est annulé au milieu de l'intervalle de temps prévu pour la transmission d'un bit.
- le **code Manchester** : ici aussi le signal change au milieu de l'intervalle de temps associé à chaque bit. Pour coder un 0 le courant sera négatif sur la première moitié de l'intervalle et positif sur la deuxième moitié, pour coder un 1, c'est l'inverse. Autrement dit, au milieu de l'intervalle il y a une transition de bas en haut pour un 0 et de haut en bas pour un 1.
- le **code Miller** : on diminue le nombre de transitions en effectuant une transition (de haut en bas ou l'inverse) au milieu de l'intervalle pour coder un 1 et en n'effectuant pas de transition pour un 0 suivi d'un 1. Une transition est effectuée en fin d'intervalle pour un 0 suivi d'un autre 0.

4.5.5.2 Transmission modulée.

Le principal problème de la transmission en bande de base est la dégradation du signal très rapide en fonction de la distance parcourue, c'est pourquoi elle n'est utilisée qu'en réseau local (<5km). Il serait en effet trop coûteux de prévoir des *répéteurs* pour régénérer régulièrement le signal. C'est pourquoi sur les longues distances on émet un

signal sinusoïdal qui, même s'il est affaibli, sera facilement décodable par le récepteur. Ce signal sinusoïdal est obtenu grâce à un **modem** (modulateur-démodulateur) qui est un équipement électronique capable de prendre en entrée un signal en bande de base pour en faire un signal sinusoïdal (modulation) et l'inverse à savoir restituer un signal carré à partir d'un signal sinusoïdal (démodulation). Autrement dit il permet de passer de signaux numériques discrets (0 ou 1) à des signaux analogiques continus.



Modulations d'amplitude, de fréquence et de phase de la séquence de bits 0110010 .

Il existe trois types de modulation décrits dans la figure

- la **modulation d'amplitude** envoie un signal d'amplitude différente suivant qu'il faut transmettre un 0 ou un 1. Cette technique est efficace si la bande passante et la fréquence sont bien ajustées. Par contre, il existe des possibilités de perturbation (orage, lignes électriques...), car si un signal de grande amplitude (représentant un 1) est momentanément affaibli le récepteur l'interprétera à tort en un 0.
- la **modulation de fréquence** envoie un signal de fréquence plus élevée pour transmettre un 1. Comme l'amplitude importe peu, c'est un signal très résistant aux perturbations (exemple la radio FM) et facile à détecter.
- La **modulation de phase** change la phase du signal suivant qu'il s'agisse d'un 0 (phase montante) ou d'un 1 (phase descendante). Dans notre exemple, 180° .

Dans les exemples donnés ci-dessus on a seulement 2 niveaux possibles à chaque fois, donc on a uniquement la possibilité de coder 2 valeurs différentes à chaque instant, dans ce cas 1 baud = 1bit/s. De manière plus sophistiquée il existe des modems capables de moduler un signal suivant plusieurs niveaux, par exemple 4 fréquences différentes que le modem récepteur saura lui aussi distinguer. Dans ce cas, chaque signal envoyé code 2 bits donc 1 baud = 2bit/s. Il est même possible de transmettre des signaux mêlant les différentes modulations présentées comme dans le cas de la norme

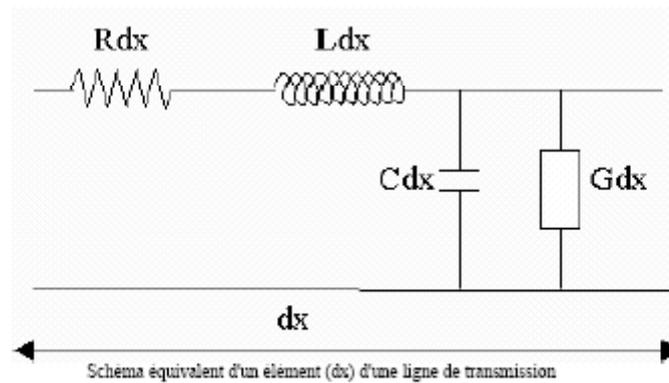
V29 qui module à la fois l'amplitude du signal sur deux niveaux et la phase sur huit niveaux. En combinant les 2 modulations, on obtient ainsi 16 signaux différents possibles à chaque instant, permettant de transmettre simultanément 4 bits à chaque top d'horloge (1 baud = 4 bit/s).

4.5.6 SUPPORTS DE TRANSMISSION

4.5.6.1 Représentation électrique d'un tronçon de ligne

Un tronçon de ligne de longueur dx peut être représenté par le circuit électrique ci-dessous. Les éléments R, C, L et G sont appelés paramètres primaires de la ligne. Ils s'expriment en unité de longueur (généralement en km). Ces éléments représentent :

- La résistance linéique des deux conducteurs de la ligne en série pour R [Ω/m]
- La capacité linéique formée par le diélectrique et les deux conducteurs pour C [F/m]
- L'inductance linéique due au champ magnétique inter et intra conducteurs pour L [H/m]
- La conductance transversale linéique due aux défauts d'isolation et aux pertes diélectriques pour G [S/m]



R: résistance linéique
L: inductance linéique
C: capacité linéique
G: conductance linéique

4.5.6.2 Classes et catégories

Classes et catégories

En dehors des limites de la bande passante, la qualité de la transmission n'est plus garantie.

C'est pourquoi, les propriétés de la ligne de transmission ont été classées

- en *classes* (selon l'ISO);

- en *catégories* (selon TIA).

La bande passante requise constitue la différence principale entre les classes ou catégories.

Classe ISO	Catégorie TIA	Applications
A	1	1 à 100 kHz
B	2	1 à 1 MHz
C	3	1 à 16 MHz
	4	1 à 20 MHz
D	5	1 à 100 MHz
	5E	1 à 100 MHz
E	6	1 à 200 MHz
F	7	1 à 600 MHz
?	?	?

4.6 Couche liaison

4.6.1 NOTION DE PROTOCOLE

Les protocoles informatiques définissent les règles de communication et d'échange d'informations entre deux machines communicantes sur liaison série, parallèle ou réseau.

Ces règles incluent :

- les conventions d'adressage
- les codes de services ou fonctions et données associées
- les formats des trames
- les contrôles de validité des trames
- les règles de dialogue (connexion, invitations, maître-maître, maître-esclave...)
- le séquençement des trames et les aspects temporels.
- les traitements des modes dégradés (reprises, temps d'attentes)

Un protocole peut être de type Full-duplex (communication simultanée dans les deux sens, chaque système ayant une responsabilité égale vis à vis de la liaison), ou half-duplex (simplex, discussion à l'alternat).

En automatisme, on parle également de protocole de type maître-maître ou maître-esclave : ces appellations différencient la mécanique d'invitation à communiquer (maître-maître : les systèmes communiquent lorsqu'ils le souhaitent, spontanément, un arbitrage du réseau étant effectué ; maître-esclaves : un système gère les invitations à communiquer pour les autres systèmes).

4.6.2 DEUX EXEMPLES DE PROTOCOLES, COUCHES OSI, ET AUTOMATES ASSOCIES

4.6.2.1 Le protocole MODBUS RTU

Etude des documents sur MDOBUS RTU et TCP/IP. Couches liaison et application.

Discussion autour de la problématique des « modes dégradés ».

4.6.2.2

Le protocole UNITEL-WAY

4.6.2.2.1 Niveau « liaison »

Au niveau de la liaison (couche basse de communication du protocole), le protocole est de type « maître – esclave » fonctionne en « half-duplex » : à un instant T donné, un message est émis, soit dans le sens « maître vers esclave(s) » ou soit dans le sens « esclave vers maître », en sachant que la transmission d'un esclave fait toujours suite à une autorisation de communication issue du maître (le maître arbitre les dialogues).

Sur un réseau donné, une station (un PC ou un équipement) joue le rôle du maître, et, à ce titre, a l'initiative des échanges sur la liaison. Les esclaves sont « adressables », en sachant que 253 esclaves sont au maximum supportés (adresse de 1 à 253). L'adresse 255 correspond à une diffusion.

Suivant les sollicitations des applications, le maître transmet des données vers les stations ou scrute tour à tour les stations esclave afin de les inviter à émettre leur données.

Le protocole repose sur des échanges de messages entre le maître et les esclaves. Les messages utilisent les caractères de contrôles suivants :

MNEMONIQUE	Code hexadécimal
DLE	10
STX	2
EOT	4
ACK	6
NACK	15
ENQ	5

Les messages, échangés à des fins d'échange de données, sont des suites de caractères (octets), calqués sur le format général de trames suivant (les caractères < et > sont introduits à des fins de lisibilité, mais ne font pas partis des messages) :

<DLE><STX><ADRESSE_liaison><Longueur><DONNEES_liaison><BCC>

La transparence, c'est à dire les mécanismes utiles à la possibilité d'utiliser des caractères de contrôle dans les champs de données, est assurée par l'insertion et le retrait de caractères <DLE>. Si un caractère DLE est présent dans les données, il est dupliqué dans le champs <DONNEES_LIAISON>. Si le champ <longueur> vaut <DLE>, il est doublé. Les champs <ADR> et <BCC> ne sont jamais doublés. Les caractères <ENQ>, <STX> sont toujours précédés d'un <DLE>.

L'intégrité au niveau des caractères (octets) est assuré par le bit de parité, par caractère et un octet de contrôle <BCC> appelé « checksum ». Le « checksum » se fait après bourrage des DLE dans la trame : les DLE entrent en compte dans le calcul de checksum.

Le calcul de la longueur se fait avant bourrage des DLE (donc or DLE introduits pour la transparence).

Le « checksum » BCC est la « somme modulo 256 des octets de la trame (PDU), du DLE d'entête au dernier octet de < DONNEES_liaison>.

Le dialogue propre à l'émission de données par le maître est le suivant:

Maître : <DLE><STX><ADR_liaison><longueur><DONNEES_liaison><BCC>

Esclave : <ACK> ou <NACK> ou <silence> ou <indéfini>

Le dialogue propre à la scrutation d'un esclave par le maître est le suivant :

Maître : <DLE><ENQ><ADR liaison>

Esclave : <EOT> ou <silence> ou <indéfini> ou
<DLE><STX><ADR liaison><long><DONNEES_liaison><BCC>

Maître (dans le cas d'une réception de données) :
<ACK> ou <NACK> ou <silence> ou <indéfini>

Le paramétrage préconisé le la transmission est :

- o 1 bit Start,
- o 8 bits de données, 1 bit parité IMPAIRE,
- o 1 Bit de stop,
- o vitesse de 9600 bauds.

Niveau « application »

Principaux services UNI-TE:

Services UNI-TE	Codes des requêtes (Hexadécimal)
LECTURE DE BITS INTERNES.	00
LECTURE D'UN MOT INTERNE	04
LECTURE D'UN MOT CONSTANT	05
LECTURE D'UN MOT SYSTEME	06
LECTURE D'UN MOT COMMUN	07
ECRITURE D'UN BIT INTERNE	10
ECRITURE D'UN MOT INTERNE	14
ECRITURE D'UN MOT SYSTEME	15
ECRITURE D'UN MOT COMMUN	16
RÉSERVATION	1D
LIBÉRATION	1E
MISE EN MARCHÉ (RUN)	24
MISE A L'ARRET (STOP)	25
ENTRETIEN	2D
VERSION PROTOCOLE	30
LECTURE DU STATUT	31
AUTO TEST	32
LECTURE D'OBJETS STRUCTURES	34
ECRITURE D'OBJETS STRUCTURES	35
LECTURE GÉNÉRALE D'OBJETS	36
ECRITURE GÉNÉRALE D'OBJETS	37
CHARGEMENT DE PROGRAMMES	3A, 3B, 3C
DECHARGEMENT DE PROGRAMMES	3D, 3E, 3F
LECTURE DES COMPTEURS D'ERREURS	A2
LECTURE DE L'ÉTAT STATION	A3
SUPPRESSION DE PROGRAMMES	F5
TEST DE LA COMMUNICATION	FA
DONNÉES NON SOLLICITÉES	FC

4.6.2.2.4 Automate liaison Maître.

//Etat de départ : ETAT_NEUTRE

{ETAT_WAIT_ACK_POLLING,	ActPolling},	Evénement:
{ETAT_WAIT_ACK_SELECTING,	ActSelecting},	// EVT_M_POLLING
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_SELECTING
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_ACK
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_NACK
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_EOT
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_LPDU_OK
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_LPDU_ERROR
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_SILENCE
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_INDEFINI
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_MAX_RETRY
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_WAIT
{ETAT_NEUTRE,	ActInattendu},	// EVT_M_DISCONNECT

//Etat de départ : ETAT_WAIT_ACK_POLLING

{ETAT_WAIT_ACK_POLLING,	ActInattendu},	Evénement - cascade
{ETAT_WAIT_ACK_POLLING,	ActInattendu},	// EVT_M_POLLING
{ETAT_WAIT_ACK_POLLING,	ActRetry},	// EVT_M_SELECTING
{ETAT_WAIT_ACK_POLLING,	ActRetry},	// EVT_M_ACK - EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_POLLING,	ActRienADire},	// EVT_M_NACK - EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_POLLING,	ActPDUAuPollingOk},	// EVT_M_EOT - EVT_WAIT
{ETAT_WAIT_ACK_POLLING,	ActPDUAuPollingNok},	// EVT_M_LPDU - EVT_WAIT
{ETAT_WAIT_ACK_POLLING,	ActRetry},	// EVT_M_LPDU_ERROR-EVT_M_INDEFINI
{ETAT_WAIT_ACK_POLLING,	ActRetry},	// EVT_M_SILENCE-EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_POLLING,	ActRetry},	// EVT_M_INDEFINI-EVT_M_MAX_RETRY
{ETAT_NEUTRE,	ActMaxRetryPolling},	// EVT_M_MAX_RETRY
{ETAT_NEUTRE,	ActWait},	// EVT_M_WAIT
{ETAT_NEUTRE,	ActDeconnecte},	// EVT_M_DISCONNECT

//Etat de départ : ETAT_WAIT_ACK_SELECTING

{ETAT_WAIT_ACK_SELECTING,	ActInattendu},	Evénement - cascade
{ETAT_WAIT_ACK_SELECTING,	ActInattendu},	// EVT_M_POLLING
{ETAT_WAIT_ACK_SELECTING,	ActAckSelecting},	// EVT_M_SELECTING
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_ACK-EVT_M_WAIT
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_NACK
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_EOT-EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_LPDU-EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_LPDU_ERROR
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_SILENCE-EVT_M_MAX_RETRY
{ETAT_WAIT_ACK_SELECTING,	ActRetry},	// EVT_M_INDEFINI-EVT_M_MAX_RETRY
{ETAT_NEUTRE,	ActMaxRetrySelecting},	// EVT_M_MAX_RETRY
{ETAT_NEUTRE,	ActWait},	// EVT_M_WAIT
{ETAT_NEUTRE,	ActDeconnecte},	// EVT_M_DISCONNECT

4.6.2.2.5

Automate liaison Esclave.

```

//Etat de départ : ETAT_NEUTRE

{ETAT_WAIT_ACK,      ActPolling},      Evénement - Cascade
{ETAT_NEUTRE,        ActInattendu},    // EVTS_POLLING - (INHIBIT)
{ETAT_NEUTRE,        ActInattendu},    // EVTS_ACK
{ETAT_NEUTRE,        ActInattendu},    // EVTS_NACK
{ETAT_NEUTRE,        ActAcqSelecting},  // EVTS_LPDU_OK
{ETAT_NEUTRE,        ActNackSelecting},  // EVTS_LPDU_ERROR
{ETAT_NEUTRE,        ActInattendu},    // EVTS_SILENCE
{ETAT_NEUTRE,        ActInattendu},    // EVTS_INDEFINI
{ETAT_NEUTRE,        ActInattendu},    // EVTS_MAX_RETRY
{ETAT_NEUTRE,        ActInattendu},    // EVTS_DISCONNECT
{ETAT_NEUTRE,        ActMasterFailure}, // EVTS_MASTER_FAILURE

//Etat de départ : ETAT_WAIT_ACK

{ETAT_WAIT_ACK,      ActInattendu},      Evénement - cascade
{ETAT_NEUTRE,        ActPDUPollingSend},  // EVTS_POLLING
{ETAT_NEUTRE,        ActRetry},           // EVTS_ACK
{ETAT_WAIT_ACK,      ActInattendu},      // EVTS_NACK-EVTS_MAX_RETRY
{ETAT_WAIT_ACK,      ActInattendu},      // EVTS_LPDU
{ETAT_WAIT_ACK,      ActInattendu},      // EVTS_LPDU_ERROR
{ETAT_NEUTRE,        ActRetry},           // EVTS_SILENCE-EVTS_MAX_RETRY
{ETAT_NEUTRE,        ActRetry},           // EVTS_INDEFINI-EVTS_MAX_RETRY
{ETAT_NEUTRE,        ActMaxRetryPolling}, // EVTS_MAX_RETRY
{ETAT_NEUTRE,        ActDeconnecte},     // EVTS_DISCONNECT
{ETAT_NEUTRE,        ActMasterFailure},  // EVTS_MASTER_FAILURE

```

4.7 Exemples de Couches « application »**4.7.1 OPC : « STANDARD DE FAIT » ACTUEL INCONTOURNABLE**

OPC (OLE for Process Control) offre une solution flexible d'utilisation des produits d'automatisation standard sous différents environnements Windows. Les interfaces OPC permettent d'échanger facilement et de normaliser des données entre des automates programmables et des applications PC telles que des systèmes de supervision ou des applications de bureautique. Sur le plan technique, OPC utilise une architecture client/serveur mettant l'accent sur l'objet, ce qui facilite l'échange de données entre les différents logiciels. Les objets du serveur sont décrits à l'aide d'interfaces d'objets dont les fonctions sont accessibles au client.

Les interfaces du serveur OPC ne varient pas et sont indépendants du système (bus de terrain, automates, cartes d'acquisition, système de commande...) utilisé par le serveur OPC. Conformément aux normes OPC, les applications clientes et serveurs peuvent être développées séparément.

La norme OPC ne permet pas d'accéder à des éléments uniques. Cet accès n'est possible que lorsqu'un élément appartient à un groupe. Ces groupes peuvent être constitués de plusieurs éléments, un élément pouvant faire partie de plusieurs groupes. Le client utilise des groupes définis pour communiquer avec le serveur. L'accès peut être synchrone ou asynchrone. Avec un accès synchrone, la fonction de lecture/ écriture ne s'achève que lorsque l'édition est terminée. Les appels asynchrones placent uniquement des tâches dans une liste d'attente de serveur et retournent immédiatement au client.

Les fabricants de matériel informatique définissent leur bus de terrain ou leur système de commandes par rapport à un serveur OPC. Les fabricants de logiciel utilisent l'interface OPC comme une application cliente. L'OPC client accède toujours à la même interface avec des

objets fixes. Les objets visibles du serveur OPC (items) ainsi que leurs attributs et méthodes correspondants sont définis suivant la norme OPC. Il peut s'agir de variables simples ou de structures de données.

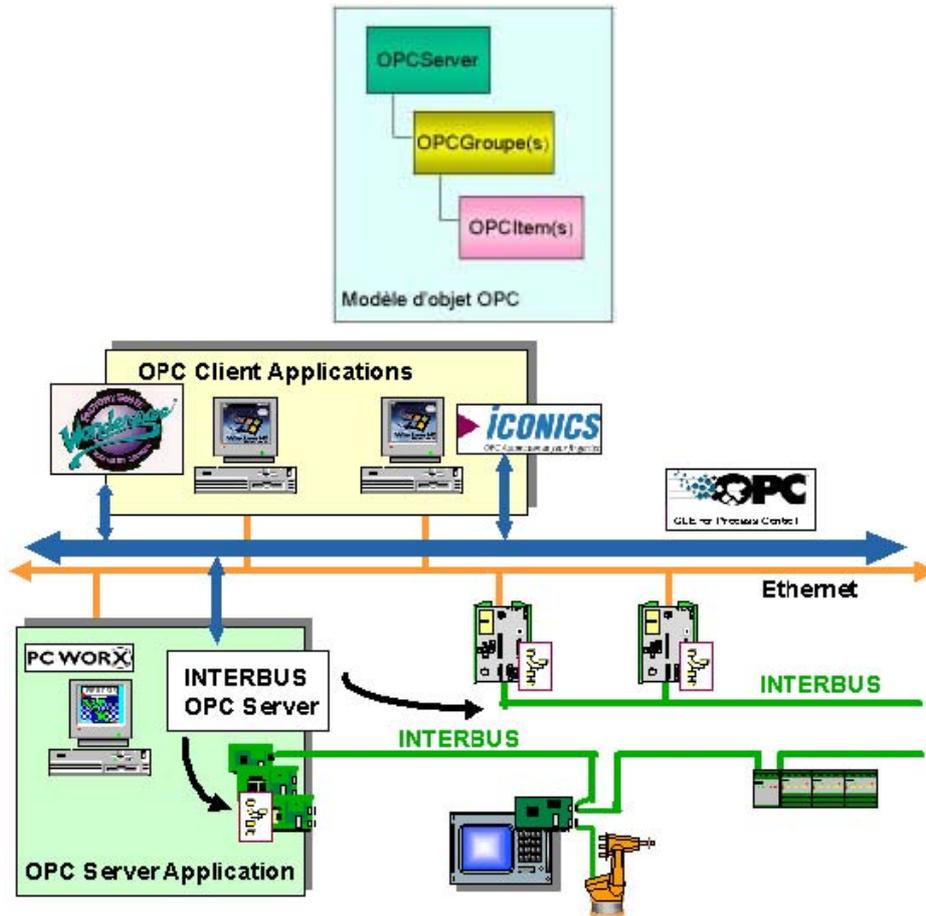


Figure 1 : Exemple : Utilisation du serveur OPC INTERBUS
(extrait document commercial PHOENIX Contact)

5 LES RLI ET BUS DANS LE MONDE INDUSTRIEL

5.1 Avantages

- Réduction des coûts :
 - de câblage (matériels et temps d'installation) : un seul câble en général (ou deux en redondance) pour tout les équipements.
 - de maintenance : complexité moindre, diagnostics en ligne, outils de test intégrés, flexibilité d'extension.
- Performances accrues :
 - facilité de conception des applications : les données sont disponibles en tout point du réseau.
 - possibilité de communication entre deux équipements, sans notion de « superviseur ».
 - Interopérabilité importante grâce au soucis de standardisation :
 - Possibilité de connexion d'équipements de différents fournisseurs : choix pour l'utilisateur final (système « ouvert », non privatif)
 - Existence de nombreuses « passerelles »
 - Existence de « Kits », et de « composants » hard/soft pour les constructeurs (exemple ANYBUS).
 - Accès au bus : conflit, arbitrage, temps de latence...
 - Sécurité des informations transportées : gestion des erreurs, redondance d'accès...
 - Topologie : longueur, débits

5.2 Historique de l'évolution des concepts – Standards

5.2.1 HISTORIQUE

- 40 : apparition du standard boucle analogique 4-20mA
- 70 : boom des processeurs – contrôle centralisé
- 80 : contrôle distribué – capteur intelligent – réseaux de terrain et informatique. RLI propriétaires.
- 92 Standardisation de la couche physique
- 94 : WorldFIP (World Factory Information Protocol, Europe) et ISP (Interoperable System Project, USA fusionnent pour donner la Fieldbus Foundation (FF).
 - couche liaison, application : planifiées fin 98 !! plus de dix ans de normalisation (un standard de communication UTI met 3 ans à sortir...)
 - l'idée était d'avoir un standard avant la sortie des produit commerciaux
 - lobbying actif des groupes d'intérêt => échec de la normalisation fin 1998.

Deux types de Bus :

- Standards de fait : INTERBUS-S, ASI, Lonworks
- Standards internationaux :
 - WorldFip NFC 46-600 (Cegelec, ALSTOM, Télémacanique, Honeywell, EDF...)
 - PROFIBUS (Process Field BUS, Allemagne) DIN 19245 intégré à ISP (Interoperable System Project USA) : Siemens, Fisher Controls...

Autres bus : AS-I : (Actuator Sensor Interface) 1993, CAN : Controller Area Network en 1993, Controlnet : 1995, DeviceNet : Milwaukee 1994, Interbus : 1984, LonWorks : Californie 1991, P-Net : Danemark, SDS : Smart Distributed System en 1989.

5.2.2 LA BOUCHE 4-20 MA

Se reporter à la note « B&B Electronics – Current Loop Application note ».

www.bb-elec.com/tech_articles/current_loop_app_note

5.3 Critères de comparaison

Le **CIAME** (Comité Interprofessionnel pour l'Automatisation et la Mesure) propose un ensemble de **critères** (46) pour la comparaison des RLI. Ils sont répartis en deux sous-groupes : **Techniques et Stratégiques**.

5.3.1 LES CRITERES TECHNIQUES

5.3.1.1 Topologiques

Longueur maximale : Longueur maximum du réseau en fonction du nombre de répéteurs et du type de médium utilisé.

Topologie : Architecture physique et implantation des noeuds connectés au réseau, structure de câblage de toutes les stations.

5.3.1.2 Temporels

Vitesse de transmission : Vitesse de transmission physique maximale possible pour le réseau. Différente du débit réel dépendent de l'efficacité du protocole.

Temps de réaction maximal : Délai maximal possible qui peut survenir lors de l'envoi d'informations. Ce temps dépend du temps de cycle, du nombre d'abonnés, de la longueur du réseau, du médium et de la vitesse physique de transmission.

5.3.1.3 Autres

Nombre maximum d'équipements : Nombre d'équipements qui peuvent être connectés au réseau. Il dépend en partie du nombre de répéteurs utilisés.

Efficacité du protocole : $LU / LT \times 100\%$, avec LU=Longueur des données Utiles et LT=Longueur Totale du message (données utiles + bits de trame).

Détection d'erreurs : Mécanisme de détection d'erreurs (parité, CRC...)

5.3.2 LES CRITERES STRATEGIQUES

5.3.2.1 Standards

Couches OSI : Définition des couches du modèle de référence OSI utilisé.

Certification : Entité responsable de la réalisation de tests et de la certification (tests des produits développés pour assurer la compatibilité).

Exemple : le standard **IECC 61158** qui regroupe 8 bus de terrain différents sous le même standard : Interbus, Profibus, WorldFip, Foundation Fieldbus - H1, Foundation Fieldbus – HSE, ControlNet, SwiftNet , P-Net.

Comme pour toute nouvelle technique, l'arrivée des bus de terrain a souffert d'un manque de normalisation. L'émergence de nouveaux produits et de nouvelles fonctionnalités ont creusé encore plus l'écart entre le début de la normalisation et la réalité technique. Ce manque qui rendait les clients frileux malgré tous les avantages que pouvait leur apporter les bus de terrain s'expliquait par la **Crainte de faire le choix d'une solution propriétaire qui lierait le client à son fournisseur voire même d'une solution qui ne serait pas pérenne ou instable**. Bien sur, le retour sur investissement existe bien, cependant mettre en place une solution de ce type nécessite un investissement de départ qu'il faut prendre en compte ainsi que le maintien en vie de la solution. Pour garantir le bon fonctionnement du bus de terrain, il faut sans cesse assurer la maintenance, mettre à jour le matériel existant et ajouter de nouveaux matériels. Choisir une solution normalisée garantie une solution mûrement réfléchie, stable et pérenne. Elle garantit aussi une meilleure interopérabilité entre les matériels de différents constructeurs (théorique en tout cas...).

5.3.2.2 Disponibilité de composants, de logiciels et de prestation de services

Disponibilité des composants existants, spécifications des fabricants.

5.3.2.3 Autres

Diffusion, nombre d'installations : Nombre d'installations en fonctionnement.

Perspectives pour l'avenir : Présomption empirique du rôle joué par le réseau dans l'avenir.

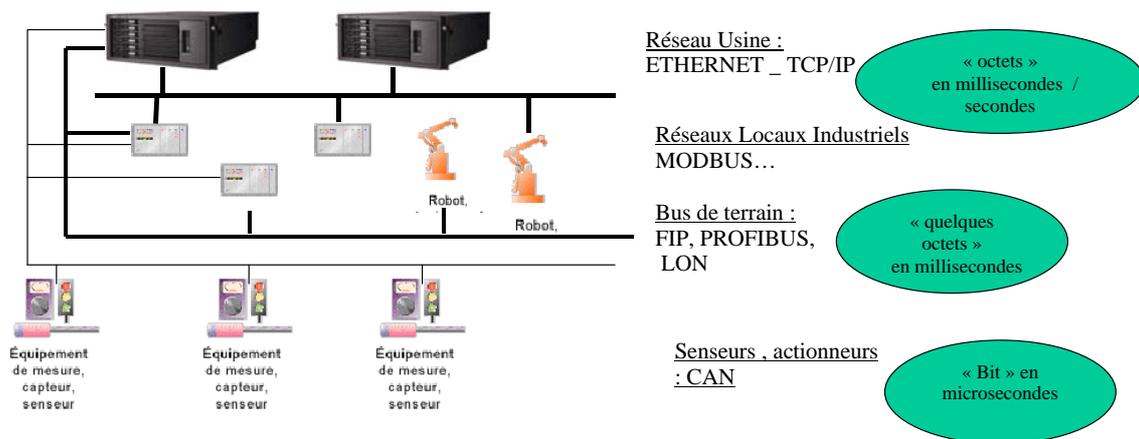
5.4 Comparaison entre les principaux bus

www.dachs.net/fb_comp.htm

5.5 Hiérarchie des RLI et BUS

Généralement on regroupe sous le terme BUS de TERRAIN tous les Bus de communication... :

- Les Bus « Usine » : réseau local sur ETHERNET
- Les Bus de Terrain
- Les Bus de niveau bas (Sensor Aktor Bus) : capteurs et actionneurs.



« **Bus Inter-usine** » : INTERNET – INTRANET, Réseaux de transport (Frame Relay, ATM, X25)

“ **Bus d ’usine** “ : Réseau Local Industriel basé sur ETHERNET. Un système sert de passerelle entre le réseau informatique et le bus de terrain

Bus de terrain : Relie des unités intelligentes qui coopèrent. Temps de réaction de quelques millisecondes. Communication de type Maître/Esclaves ou MultiMaîtres. Possibilités d ’accès au niveau inférieur.

Bus senseurs /acteurs (ou Capteurs /Actionneurs) : relie entre eux des nœuds à intelligence limitée ou nulle(E/S déportées). Temps de réaction très rapide. Communication Maître/Esclaves

5.6 Constat

- un bus de terrain est en général supporté par un ou des constructeurs... et adoptés par les autres sous la forme d'options.
- en ce qui concerne la continuité des solutions proposées depuis l'automate jusqu'aux capteurs – actionneurs, en passant par divers périphériques déportés, certains systèmes se voient, fautes de performances techniques suffisantes, recourir à un ensemble de solutions partielles distinctes, résumées sous un même nom (illusion d'une continuité) :exemple PROFIBUS FMS, DP, DP+, PA.

5.7 WorldFIP

Source WEB des paragraphes 5.7.1 à 5.7.3

Conservatoire national des arts et métiers
Informatique industrielle A 10
Bus WorldFip et Profibus
Année 2004-2005

5.7.1 ORIGINE DE WORLDFIP

Worldfip (World Factory Instrumentation Protocol) est une organisation française à but non lucratif fondée en 1988, elle fonctionne grâce au financement des membres de l'organisation, à la mise en place de formations, à l'assistance technique et au support. Un consensus entre les utilisateurs finaux, les instituts de recherche et les fabricants est mis en place avant de prendre toute décision visant à faire évoluer le bus. Worldfip est membre de « FieldBus Organisation » et est très actif dans le projet Européen NOAH (Network Oriented Applications Harmonisation). Les buts de Worldfip sont de réduire les coûts de connexion, d'obtenir un maximum de fonctionnalités, d'obtenir un bon débit et d'assurer une interopérabilité avec le plus grand nombre possible de matériel.

5.7.2 FONCTIONNEMENT DE WORLDFIP

Termes utilisés :

- **Variable** : la plus petite unité d'échange entre les stations.
- **Message** : unité d'échange plus grande et plus explicite que les variables.
- **Station** : élément du bus pouvant être, un ordinateur, un automate, un capteur ou un actionneur.
- **Unité de traitement** : automate programmable, ordinateur ou régulateur.
- **Consommateur** : station qui a besoin de lire une variable.
- **Producteur** : station qui fournit une variable aux autres stations.
- **Identifiant** : adresse globale d'une variable.
- **Arbitre de bus** : chef d'orchestre du bus, sans son accord les stations ne peuvent ni émettre des informations ni en recevoir.

5.7.2.1 Topologie

La topologie réseau est une topologie bus classique, on peut cependant par le biais de répéteur cascader d'autres bus on obtient ainsi plusieurs bus principaux. On peut aussi y ajouter des dérivateurs multiples appelés boîtes de jonctions pour connecter des éléments en un point unique sur le bus principal ou secondaire.

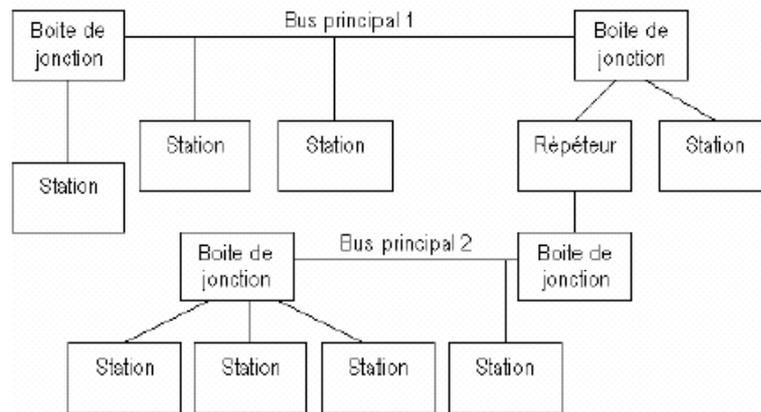
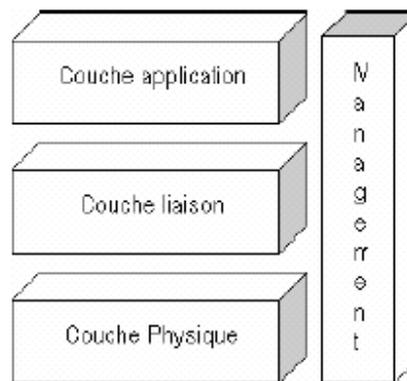


Figure 2 : Exemple de topologie

Les différentes couches du protocole Le protocole WorldFip est composé 3 couches de communications, une couche physique (physical layer), une couche liaison (data link layer) et une couche application (application layer). Il peut être complété d'une couche transverse assurant le management du réseau.



5.7.2.2 Médias utilisés

WorldFip utilise deux types de média comme support de transmission, la paire torsadé ou la fibre optique.

Les médias peuvent être redondés (DUAL FIP).

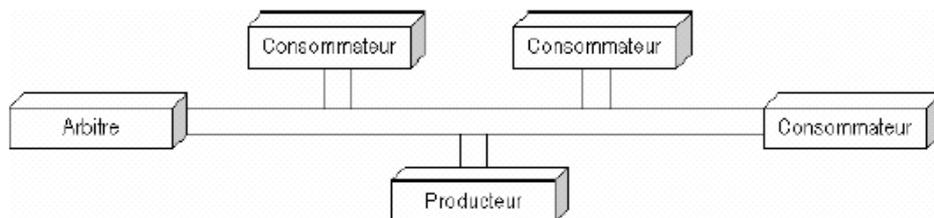
5.7.2.3 Principe de fonctionnement

WorldFip peut être comparé a une base de données distribuée, le procédé utilisé pour mettre à jour les variables garantit une cohérence des données. Une variable possède un identifiant unique sur tout le bus. L'identifiant des variables étant codé sur 16 bits on peut avoir jusqu'à 65536 identifiant sur le bus. Une variable ne peut être

produite que par une seule station et ce indéfiniment dans le temps, les autres stations ne pourront y accéder qu'en lecture seule. Il ne peut y avoir qu'un seul arbitre de bus sur tout le bus à un moment donné. Le rôle d'arbitre est assuré par une station. La fonction **arbitre de bus** est assurée par le service *ABAS* de la couche application. Il existe deux méthodes d'échange de données, une **cyclique**, est une autre **acyclique**. Il existe aussi deux types d'échange, un **échange de variable** et un **échange de message**. Les messages étant plus longs que les variables. L'échange de variable cyclique et acyclique est assuré par le service *MPS* de la couche application. L'échange de messages est assuré par le service *subMMS* de la couche application. Lors de la configuration du bus, une liste des variables est établie. Pour chaque variable on définit qui en est le producteur et à quelle périodicité la variable sera utilisée sur le bus. Cette liste est détenue par l'arbitre du bus.

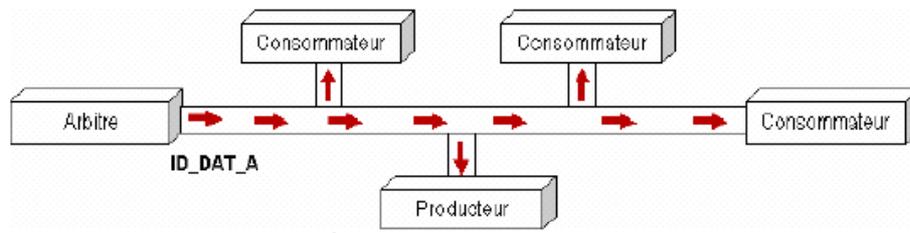
Variable	Périodicité en ms	Type	Temps en μ s
A	5	OSTR_32	190
B	10	INT_8	150
C	10	INT_16	200
D	15	INT_8	160
E	20	OSTR_32	320
F	25	OSTR_32	200
G	30	INT_8	140

Les schémas utilisés dans les exemples suivants comporteront un bus avec 5 stations, comme ci-dessous :

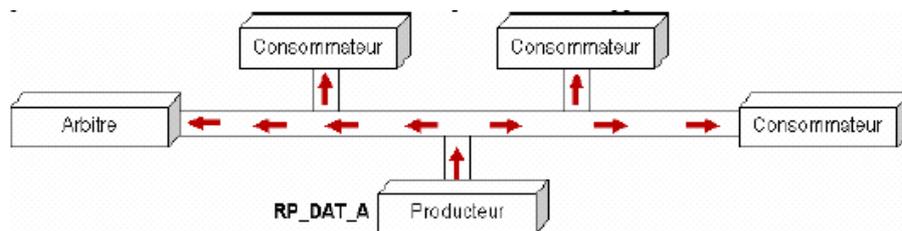


5.7.2.3.1 Cas d'un échange de variable cyclique

L'arbitre envoie « par diffusion » une trame « Question » sur le bus pour indiquer qu'il veut lire le contenu de l'identifiant « A », toutes les stations reçoivent le message mais comme il ne peut y avoir qu'un seul producteur pour chaque variable seule la station producteur peut répondre à cette question.

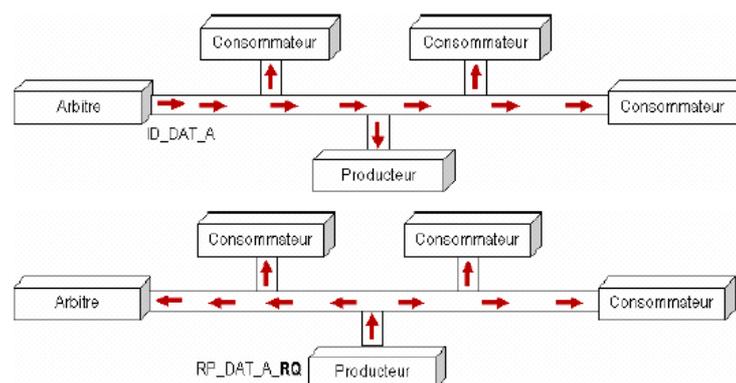


Le producteur donne sa « réponse » en envoyant la valeur de l'identifiant, toutes les stations reçoivent la valeur, les stations intéressées (consommateur) peuvent consommer cette variable ce procédé « s'appelle transfert de buffer ».



5.7.2.3.2 Cas d'un échange de variable acyclique

La demande de transfert de variable acyclique se fait en utilisant le procédé de transfert de variable cyclique. Comme précédemment l'arbitre envoie sa question, quant à lui, le producteur répond mais en positionnant un bit dans la trame indiquant qu'il a une demande de requête acyclique à formuler. On peut si nécessaire ajouter un drapeau de priorité urgent.



Maintenant, l'arbitre sait que la station qui vient de répondre a une ou plusieurs variables acyclique a transférer. Il retransmet la même question que précédemment en ajoutant un bit requête pour indiquer à la station de lui transmettre la liste des variables à demander. La liste peut contenir jusqu'à 64 identifiants.

Une fois que l'arbitre reçoit la réponse, il demande la lecture d'une des variables de la liste, le processus continue ainsi comme pour une variable cyclique. Seul un producteur peut demander le transfert de variable acyclique, par contre il peut demander la lecture de variables dont il n'est pas le propriétaire. Il est possible de faire une demande spécifiée. Le producteur choisi de déclencher sa demande de transfert acyclique sur production d'une variable spécifique.

5.7.2.3.3 Cas d'un échange de message sans acquittement

Les transferts de messages sans acquittement se font de façon acyclique, ils s'adressent à une station ou à plusieurs stations. Le début du processus de transfert de message se fait comme pour celui d'une variable acyclique, la différence commence quand le producteur répond, il positionne un bit indiquant qu'il a un message à transférer.

L'arbitre indique au producteur qu'il peut adresser son message et attend la trame de fin de transfert. Le producteur émet son message en précisant s'il attend ou pas un accusé de réception. Une fois celui-ci émit, il indique à l'arbitre que le transfert est terminé.

Pour éviter que le transfert s'éternise, l'arbitre arme un timer.

5.7.2.3.4 Cas d'un échange de message avec acquittement

Les transferts de messages avec acquittement se font de façon acyclique, ils s'adressent à une station. Le début du processus de transfert de message se fait comme précédemment, la différence commence quand le producteur envoie son message il positionne un bit indiquant que le message est avec accusé de réception.

Un message perdu ne peut être ré-émis que deux fois. L'arbitre dispose toujours d'un timer pour éviter d'être bloqué par un transfert trop long.

5.7.2.3.5 Classe de services

Tous les équipements ne peuvent pas assurer tous les services énumérés ci-dessus, pour avoir un aperçu rapide de leurs possibilités, les services ont été regroupés par classe :

Services / Classes	1	2	3	4	5	6	7	8	9
Transfert de buffer	X	X	X	X	X	X	X	X	X
Ecriture de buffer	X	X	X	X	X	X	X	X	X
Lecture de buffer		X	X	X	X	X	X	X	X
Demande explicite libre				X	X			X	X
Demande explicite spécifiée			X		X				X
Message sans acquittement						X	X	X	X
Message avec acquittement							X	X	X

5.7.3 CRITERES DE COMPARAISON

Les critères de comparaison

Les critères techniques

Topologiques

Longueur maximale : 4000m (avec 3 répéteurs = 1 par km à 1 Mbit/s)

Topologie : Bus ou étoile

Temporels

Vitesse de transmission : 1 Mbit/s typique

Temps de réaction maximal : Variable périodique : aucun car le temps de cycle est prédéfini.
Variable aperiodique : dépend du taux d'occupation du bus, de la priorité de la variable, du remplissage de la file d'attente.

Autres

Nombre maximum d'équipements : 256 (adresse sur 1 octet)

Efficacité du protocole : de 3% à 85% en fonction de nombre de données utiles.

Détection d'erreurs :
- Détection de la durée d'occupation
- Contrôle de trame (Check)
- Surveillance des identifiants (identifiant demandé = identifiant émis).

Les critères stratégiques

Standards

Couches OSI : 1, 2, 7

Certification : S'adresser au centre technique WorldFip.

Disponibilité de composants, de logiciels et de prestation de services

Composants : 4 fabricants, une dizaine de composants.

Autres

Diffusion, nombre d'installations : France, Italie, Angleterre, Amérique du nord.
350 000 noeuds installés.

Perspectives pour l'avenir : Utilisés par plusieurs sociétés mais le nombre de noeuds installés est significativement très inférieur à d'autres bus.

Source

F. Hoppenot (1999) Informatique industrielle
Introduction aux Réseaux Locaux Industriels

5.8 Profibus

Source WEB des paragraphes 5.8.1 à 5.8.3

Conservatoire national des arts et métiers
Informatique industrielle A 10
Bus WorldFip et Profibus
Année 2004-2005

5.8.1 ORIGINE DE PROFIBUS

Profibus (PROcess Field BUS) est le résultat d'un projet lancé en 1989 par le ministère fédéral allemand de la recherche et de la technologie, il a été développé et financé par des entreprises d'automatismes comme Siemens.

5.8.2 FONCTIONNEMENT DE PROFIBUS

PROFIBUS est composé de deux protocoles de transmission appelés profils de communication DP (Decentralized Periphery) et FMS (Fieldbus Message Specification) et de profils applicatif tels que PA (Process Automation), BA (Building Automation) ou Profisafe. Ces profils de communication définissent la façon dont les données sont transmises sur le support physique. Le profil de communication FMS est destiné à la communication entre équipements dits intelligents tels que des ordinateurs ou des automates, tandis que le profil DP est plutôt destiné à la communication entre équipements intelligents et équipements basiques tels que des capteurs et actionneurs.

5.8.2.1 Topologie

D'un point de vue physique, la topologie utilisée est celle d'un bus mais d'un point de vue logique on a en plus une topologie anneau.

5.8.2.2 Les couches du protocole

Le protocole PROFIBUS est composé de 3 couches de communications, une couche physique (physical layer), une couche liaison (data link layer) et une couche application (application layer).

5.8.2.3 Médias utilisés

PROFIBUS peut être utilisé avec trois types de média, la paire torsadée (RS485), le bifilaire (CEI 1158-2) et la fibre optique, cette dernière a pour avantage de fournir des débits élevés,

d'avoir une forte immunité aux parasites et peut être utilisée pour des grandes distances, par contre elle a pour inconvénient d'être plus fragile et plus coûteuse. A noter cependant que le futur de PROFIBUS prévoit son utilisation sur des infrastructures déjà existantes tel que le réseau Ethernet.

5.8.2.4 Principes de fonctionnement

Composition d'une trame

- Syn = Synchronisation
- SD2 = Délimiteur de début
- LE= Longueur
- DA= Adresse destination
- SA= Adresse source
- FC= Code fonction
- DU= Données
- FCS = Checksum
- ED= Délimiteur de fin

Il existe deux types de stations sur le bus, les stations maîtres appelées aussi stations actives et les stations esclaves appelées également stations passives. Toutes deux peuvent recevoir ou émettre des messages, cependant, seul le maître peut émettre librement tandis que l'esclave n'émet que sur demande du maître. Il existe aussi deux méthodes de communications différentes, une réservée pour la communication entre les maîtres appelée « méthode du jeton » et une autre réservée pour la communication entre les maîtres et leurs esclaves appelée « mode maître-esclave ». Pour communiquer entre elles les stations ont la possibilité de diffuser leurs messages en mode point à point, en mode broadcast (diffusion générale) ou en mode multicast (diffusion sélective)

5.8.2.4.1 Méthode du jeton

Il ne faut pas que certains maîtres aient le monopole de la parole, pour cela il faut instaurer un système garantissant dans un temps défini que chaque maître puisse avoir la possibilité de parler. Les maîtres connectés au bus forment un anneau à jeton virtuel et sont classés par rapport à leur adresse de la plus petite à la plus grande. C'est au démarrage du réseau que la liste des maîtres est élaborée, ensuite le maître possédant l'adresse la plus petite acquiert le jeton en premier, il peut ainsi communiquer avec les autres maîtres ou avec ses esclaves ou tout simplement passer son tour, il retransmet ensuite le jeton au maître ayant l'adresse supérieure. La liste est mise à jour automatiquement lors d'un retrait ou d'une insertion d'un maître. Une station maître possédant le jeton est une station active.

5.8.2.4.2 Mode maître-esclave

C'est sur ordre de leurs maîtres que les esclaves reçoivent ou transmettent des messages.

5.8.2.4.3 Profil de communication DP

Le profil de communication DP destiné à la communication entre les maîtres et les esclaves est utilisé pour des échanges de données courts (244 octets) et rapide. Il peut aussi bien se faire de façon cyclique que de façon acyclique.

La connexion et la déconnexion d'éléments DP s'effectuent de façon simple et rapide (Plug and Play). Il peut y avoir un ou plusieurs maîtres sur le bus. En configuration mono-maître, l'unique maître s'occupe de tous les esclaves présents sur le bus. En configuration multi-maîtres, les maîtres peuvent avoir des esclaves communs, cependant le droit en écriture sur ceux-ci n'est pas partagé, seul un d'entre eux possède un droit en écriture sur un esclave, cette station « DPM1 » est désignée lors de la configuration. De même, il existe un maître DPM2 qui est une station de configuration, de service et d'analyse pour les différents esclaves.

Transmission cyclique : Le DPM1 possède la liste de ses esclaves et une liste des variables à transférer avec leurs périodicités. DPM1 peut s'adresser directement à un esclave, à un ensemble d'esclaves ou voire tous les esclaves en même temps. Dans les deux derniers cas, il est obligé d'assurer la synchronisation des entrées et des sorties entre tous ses esclaves. Pour cela, il dispose de deux commandes « Synchro » et « Freeze »

Transmission acyclique : Pendant le temps libre entre deux transmissions cycliques, DPM1 peut envoyer des messages acycliques, il dispose pour cela cinq fonctions, MSAC1_Read pour lire les données de l'esclave, MSAC1_Write pour écrire des données vers l'esclave, MSAC1_Alarm pour transmission d'une alarme vers le maître avec attente d'accusé de réception, MSAC1_Alarm_Acknowledge pour envoi d'accusé de réception vers l'esclave et MSAC1_Status pour informer le maître de l'état de l'esclave.

5.8.2.4.4 Profil de communication FMS

La priorité pour ce profil destiné aux échanges entre maîtres est la quantité de données échangées au détriment de la rapidité. Le profil FMS possède un dictionnaire d'objets contenant la description, la structure, le type de données ainsi que la relation entre l'adressage interne et l'adressage externe complété par sa désignation connue par les autres stations du bus. Ces objets qui sont des objets de communication peuvent être des variables, des tableaux (suite de variables de même type), des structures (suite de variables de type différents), des domaines (zones mémoire) ou des événements (message d'alarmes).

Index	Type	Adresse interne	Désignation
10	Var	4100	Température
11	Var	3600	Pression
...			

5.8.2.4.5 Les profils applicatifs

Ils décrivent comment utiliser les profils de communications et les supports physiques par les applications ou par les équipements. PA : Basé sur le profil de communication DP, PA

utilise le support de transmission CEI 1158-2 avec dans la plupart des cas une télé-alimentation des équipements. Grâce à des interfaces décrivant le comportement de l'équipement sur le bus et une connaissance détaillée des différents équipements du marché, on peut interchanger des équipements de marques différentes sans intervention sur l'équipement. Profisafe : Basé sur le profile de communication DP, Profisafe assure la sécurité (intégrité) des transmissions (perte de trame, erreur de séquence, ...). Il définit aussi le raccordement d'équipement de sécurité tel qu'un bouton d'arrêt d'urgence, BA : Basé sur le profile de communication FMS, BA définit les règles sur tout ce qui a trait à l'exploitation d'un bâtiment, surveillance, exploitation, traitement d'alarme... . Ces profils applicatifs ont de par leurs règles déjà définies, pour rôle de faire baisser les coûts engendrer par une étude sur l'installation d'un bus de terrain en fonction de l'application métier.

5.9 CAN

5.9.1 ORIGINE DE CAN

A l'origine le CAN fut développé pour l'usage automobile par Bosch et aujourd'hui la plupart des constructeurs mettent au point des systèmes entièrement multiplexés utilisant la technologie CAN.

La technologie CAN trouve sa place dans de nombreuses industries notamment grâce à ses qualités de fiabilité et d'architecture temps réel. Toute transmission des données à travers le bus est bornée grâce à un temps maximal normalisé. De plus l'architecture CAN possède un système d'erreur simple et efficace qui permet de l'utiliser à des fins médicales ainsi que dans le cadre de toute application mettant en jeu la sécurité des personnes.

Les contrôleurs CAN sont physiquement petits, peu coûteux et entièrement intégrés. Ils sont utilisables à des débits importants, en temps réel et dans des environnements difficiles. Enfin, les transmissions ont un haut niveau de fiabilité. C'est pourquoi ils ont été utilisés dans d'autres industries que l'automobile et des applications utilisant le CAN sont aujourd'hui disponibles dans l'agriculture, la marine, le matériel médical, les machines textiles, etc...

5.9.2 FONCTIONNEMENT DE CAN

5.9.3 CRITERES DE COMPARAISON

Les critères de comparaison

Les critères techniques

Topologiques

Longueur maximale : 1000m à 50kbit/s, 40m à 1Mbit/s. Pas de répéteur.
Topologie : Bus avec résistance de terminaison de ligne.

Temporels

Vitesse de transmission : 1 Mbit/s au maximum, 20kbit/s au minimum.
Temps de réaction maximal : Indéfini (bus non déterministe).

Autres

Nombre maximum d'équipements : 32 stations avec une RS485 (couche physique). Pas de limitation par le mode d'adressage.
Efficacité du protocole : de 0% à 53% en fonction du nombre de données utiles.
Détection d'erreurs :
 - CRC
 - Format Check
 - Bit check : chaque station écoute ce qu'elle envoie.

Les critères stratégiques

Standards

Couches OSI : 1, 2, 7
Certification : Disponible ou en préparation pour CANopen, DeviceNet et SDS.

Disponibilité de composants, de logiciels et de prestation de services

Composants : 22 fabricants... Difficile de citer tout le monde.

Autres

Diffusion, nombre d'installations : Europe, USA, Japon.
 + de 9 000 000 de noeuds installés dans l'automobile.
 + de 6 000 000 de noeuds installés dans les automatisme industriels.

=> Fait partie des bus de terrain les plus répandus.

Perspectives pour l'avenir :
 - Composants bon marché
 - Beaucoup de fournisseurs
 - Beaucoup d'applications

Source du paragraphe :

P. Hoppenot (1999)

Informatique industrielle

Introduction aux Réseaux Locaux Industriels

5.10 InterBus

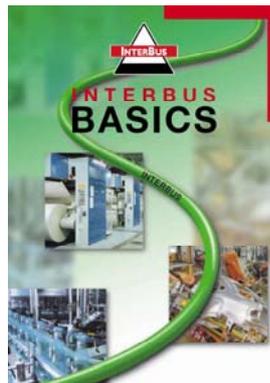
InterBus est un bus pour les capteurs et les actionneurs : il se veut assurer la continuité depuis le niveau automate, jusqu'au dernier « fin de course ». Le bus est spécifié par les normes DIN E 19 258 et EN 50 254.

Plus de 700 fabricants de matériels ont adoptés INTERBUS (variateurs, codeurs,...).

5.10.1 ORIGINE D'INTERBUS

5.10.2 FONCTIONNEMENT D'INTERBUS

Le document « Interbus BASIC » de PHOENIX Contact, fourni en annexe, décrit les fonctionnalités de ce BUS.



5.10.3 CRITERES DE COMPARAISON

Les critères de comparaison

Les critères techniques

Topologiques

<u>Longueur maximale :</u>	Bus inter-stations :	12.8 km (400m entre deux stations successives).
	Bus périphérique :	10 m.
	Inter-loop :	100 (capteurs et actionneurs).
<u>Topologie :</u>	Anneau.	

Temporels

<u>Vitesse de transmission :</u>	Bus inter-stations :	500 kbit/s.
	Bus périphérique :	300 kbit/s.
	Inter-loop :	500 kbit/s.

Temps de réaction maximal : Calculable. Dépend directement du nombre d'abonnés.

Autres

Nombre maximum d'équipements : Bus inter-stations : 1 carte maître et 256 points de connexion.
512 modules (4096 E/S) dont maximum 8 par bus périphérique.
Inter-loop : 64 équipements maximum.

Efficacité du protocole : de 10% (1 équipement, 8 bits de données) à 98% en fonction de nombre de données utiles.

Détection d'erreurs :
- CRC
- Vérification des longueurs

Les critères stratégiques

Standards

<u>Couches OSI :</u>	1, 2, 7
<u>Certification :</u>	Institut Fraunhofer Karlsruhe.

Disponibilité de composants, de logiciels et de prestation de services

Composants : Quelques fabricants, quelques composants.

Autres

Diffusion, nombre d'installations : + de 125 000 applications
+ de 1 700 000 de composants produits

Perspectives pour l'avenir : Grande acceptation du marché, en expansion.

Source du paragraphe :

P. Hoppenot (1999) Informatique industrielle
Introduction aux Réseaux Locaux Industriels

5.11 LON

5.11.1 ORIGINE DE LAN

5.11.2 FONCTIONNEMENT DE LON

5.11.3 CRITERE DE COMPARAISON

Les critères de comparaison

Les critères techniques

Topologiques

Longueur maximale : 6.1 km à 5 kbit/s.

Topologie : Dépend du médium : bus ou topologie libre.

Temporels

Vitesse de transmission : Maximum 1.25 Mbit/s, typiquement 78 kbit/s.

Temps de réaction maximal : Indéfini (bus non déterministe).

Autres

Nombre maximum d'équipements : 32385 par domaine (255 sous-réseau de 127 stations).

Efficacité du protocole : de 6% à 93% en fonction du nombre de données utiles.

Détection d'erreurs : - CRC

Les critères stratégiques

Standards

Couches OSI : 1, 7

Certification : Société Echelon.

Disponibilité de composants, de logiciels et de prestation de services

Composants : 2 fabricants, 4 composants.

Autres

Diffusion, nombre d'installations : Nombreuses applications bâtiment, énergétique.

Perspectives pour l'avenir : Décentralisation => Attente d'un marché en forte croissance.

Source du paragraphe :

P. Hoppenot (1999)

Informatique industrielle

Introduction aux Réseaux Locaux Industriels

5.12 Exemple de comparatif de bus : WorldFIP / PROFIBUS

Les bus de terrain FIP et PROFIBUS sont tout deux entièrement normalisés sous la norme EN 50170.

Profibus ressemble à un système de communication client-serveur entre le maître et ses esclaves où les esclaves seraient les serveurs et le maître le client. Le nombre de maîtres constituant l'anneau logique ne doit pas être trop important car il diminuerait le temps de communication disponible entre les maîtres et leurs esclaves (Cette contrainte n'existe pas dans Worldfip).

Le recouvrement de la perte d'un maître dans Profibus n'est pas soulevé dans les documentations, on peut se poser la question sur le devenir des esclaves appartenant au maître défaillant.

Worldfip ne fonctionnant pas en maître/esclave le point de défaillance pourrait être l'arbitre de bus mais la perte de celui-ci est prévue, en effet il existe sur le bus plusieurs arbitres capable de prendre le relais mais un seul peut être actif à un moment donné, le basculement se fait sur défaillance de l'actif par un algorithme d'élection embarqué qui se fait en fonction du poids de l'adresse.

6 COMMUNICATION INTER-USINE : LES RESEAUX DE TRANSPORT

6.1 Notions de base

L'interconnexion de sites distants passe par la mise en œuvre :

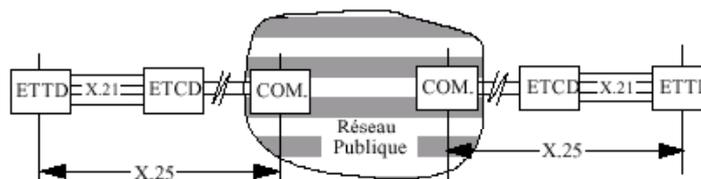
- de liaisons louées
- d'un réseau de transport (ou sous réseau réel au sens de l'OSI)

Les réseaux de transports regroupent les liens et les ressources partagées entre plusieurs utilisateurs à des fins de communication. Ils peuvent être de type privé (géré par une organisation de droit privé à ses fins propres) ou public (géré par un opérateur et soumis à des redevances d'utilisation).

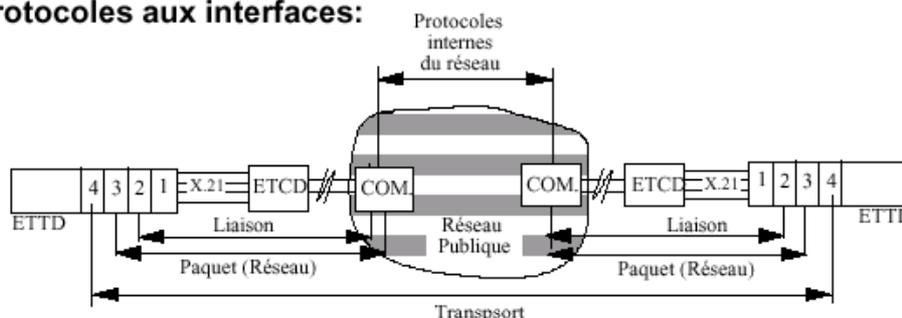
Un **réseau de transport** est caractérisé par :

- des **protocoles d'accès au réseau de transport** : ces derniers peuvent être différents des protocoles internes utilisés par le réseau.
- Les **modes d'accès** (liens permanents, liaisons spécialisées, liens temporaires comme le réseau téléphonique commuté)
- La **qualité de service** (débit, taux d'erreur, temps de traversée...)

Interfaces avec le réseau:



Protocoles aux interfaces:

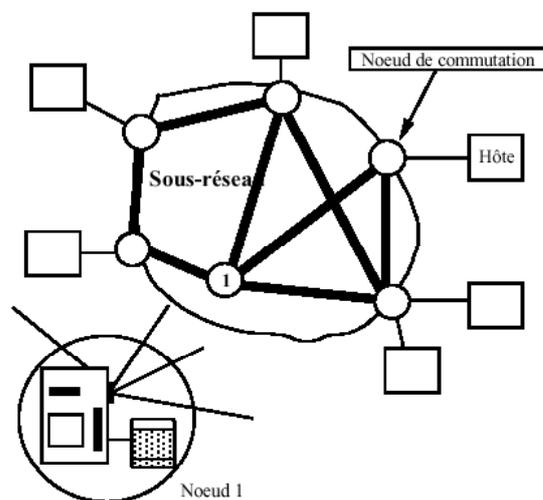


6.2 Les principes de commutation

Le réseau de transport est constitué de nœuds. La mise en relation de deux systèmes connectés au travers du réseau passe par le routage de l'informations entre ces nœuds. les nœuds peuvent être constitués par des équipements « intelligents » dotés de programmes et de tables de routage ou constitué par de « simples relais » (cas du RTC).

Parfois, pour des raisons d'efficacité, il est nécessaire de découper chaque message.

Ces techniques sont reliées aux problèmes des commutateurs.



6.2.1 COMMUTATION DE CIRCUITS - EXEMPLE : RTC

Un lien physique est établi à la connexion entre les nœuds (continuité métallique). Il est réservé durant tout l'échange. Les étapes de dialogue sont en effet les suivantes :

- 1) connexion (réservation du chemin)
- 2) échange de l'information
- 3) déconnexion (relâche du chemin)

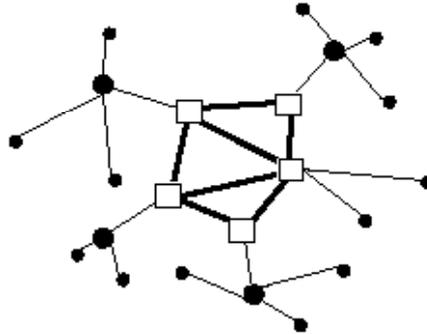
Durant ces étapes :

- les ressources nœuds sont monopolisées
- il y a présence physique permanente des deux abonnés
- il n'y a pas de stockage intermédiaire.
- La régulation de trafic est réalisée à la connexion

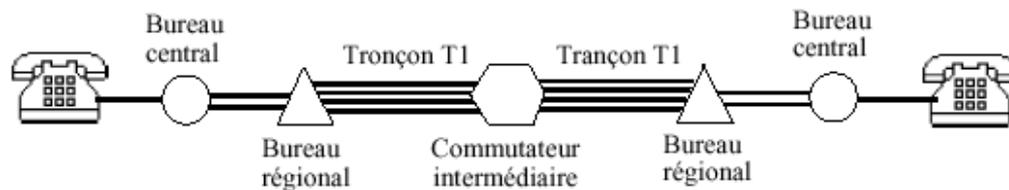
- Les nœuds de commutation sont de « simples relais »
- La facturation est « au temps de connexion et à la distance » (ce qui permet à l'opérateur d'amortir assez vite l'infrastructure)
- La résistance aux erreurs est variable

Ce type de commutation est très utilisé en téléphonie.

Le réseau téléphonique est organisé sous forme hiérarchique comme suit:



Un circuit est établi selon le routage hiérarchique:

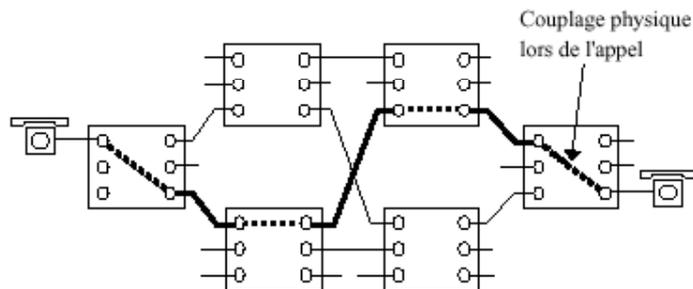


Dans la commutation de circuits, un **circuit** (fréquentiel ou temporel) est **établi** entre l'émetteur et le récepteur. Ce mode se caractérise essentiellement par la **réserve** des **ressources** de communication: on parle de réservation de **Bande Passante**.

Les **applications** classiques de ce type de réseau sont celles à **contrainte temporelle** (délai de **traversée constant**) telles que le service **téléphonique** et toutes les applications "**streaming**".

L'inconvénient majeur est le **gaspillage** possible de la Bande Passante. En effet, **Réserver** n'est pas **Utiliser**.

Un chemin fixe établi au moment de l'appel, permet de sélectionner un circuit. On établit donc une connexion physique. Ça prend du temps (environ 10 sec):



Il n'y a pas de congestion car sur chaque circuit il y a un utilisateur à la fois.

Le délai de propagation est de l'ordre de 6 msec/1000 Km

6.2.2 LA COMMUTATION DE MESSAGES - EXEMPLE : TELEX, EMAIL

Dans la Commutation de Messages, il n'y a **pas de réservation de ressources**. Ainsi, les messages qui arrivent dans le noeud de commutation sont traités selon l'ordre d'arrivée: **file FIFO** (First In First Out). S'il y a trop de **trafic**, il y a attente dans la file. Donc le temps de **traversée du réseau n'est pas constant** et dépend des **temps d'attente** qui est fonction du trafic.

La technique utilisée est le **Store & Forward**. Si on rajoute au traitement de routage, le traitement d'erreurs et d'autres traitements pour assurer un service fiable de transmission, le temps d'attente augmente.

L'avantage de cette technique est une meilleure utilisation des ressources puisqu'il n'y a pas de réservation. Ce mode de commutation est adaptée à un trafic sporadique et non continu n'ayant pas de contrainte de temps telles que les **applications informatiques classiques** (ex. transfert de fichiers).

L'inconvénient est le temps d'attente.

Un message est un bloc d'information (ou une unité de transfert).

Dans ce mode de commutation, le chemin entre les nœuds est élaboré à chaque transfert de message.

- acheminement individuel des messages, choix du circuit à chaque transmission
- message mémorisé intégralement par chaque nœud avant retransmission vers le nœud suivant
- chaque nœud a une mémoire de masse importante
- le débit est donc limité
 - il y a une meilleure utilisation des lignes
- le transfert peut avoir lieu même si le correspondant est non connecté ou occupé
- facturation au temps d'utilisation effectif
- résistance aux erreurs mauvaise (retransmission message)

$$\text{temps transit} = N * \text{longueur message} / \text{débit}$$

6.2.3 COMMUTATION DE PAQUETS - EXEMPLE : TRANSPAC (X25), INTERNET (TCP/IP)

La commutation de messages peut être **amélioré** en **découpant** le **message** en unités de données en **paquets** (taille variable mais ayant un maximum). En effet, la même technique (Store & Forward) est utilisée avec deux avantages:

- Effet "**pipe line**": on peut commencer à transmettre un paquet pendant qu'on reçoit un autre paquet du même message;
- **Temps d'émission plus réduit**: la taille du paquet étant limitée, une meilleure gestion de la file d'attente et un meilleur multiplexage des données est effectué.

Le **problème** à résoudre est le **réassemblage** du message avant de le donner à la couche supérieure.

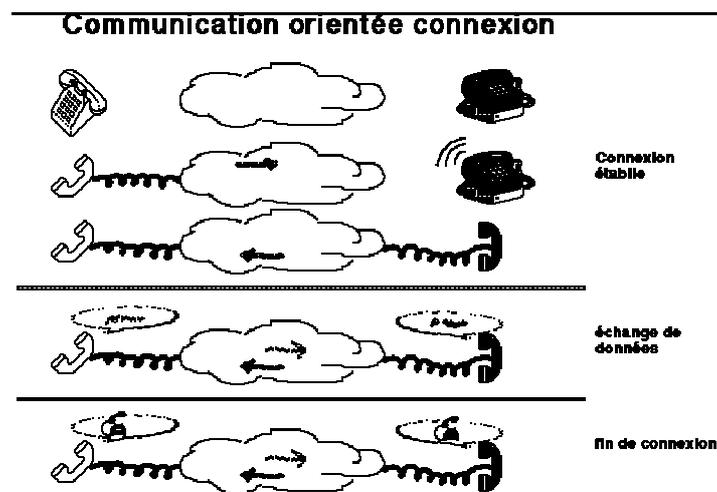
Un paquet est un fragments de messages.

- Les paquets sont envoyés indépendamment les uns des autres sur les nœuds
- Des paquets de différentes sources sont multiplexés sur un même lien
- Le séquençement est non garanti à l'arrivée :cela nécessite l'ordonnancement, et la mémoire de masse à l'arrivée doit être importante
- Ce mode nécessite la mémorisation de chaque paquet en attente d'acquittement
- les débits sont limités
- l'utilisation des liens est optimisée
- la conversion de débit et de taille des paquets est effectuée par le réseau
- le réseau paquet supporte la commutation de messages

Transferts successifs indépendants les uns des autres

- si destinataire absent, paquet perdu
- pas d'accusé de réception
- routes variables,
- adressage complet de chaque paquet
- séquençement des informations non garanti,
- répartition de charge optimisée sur le réseau
- mécanisme réseau simplifiés, réception complexe
- mode remise pour le mieux
- mode datagramme

6.2.6 MODE CONNECTÉ (CONS)



Transferts successifs sur le même chemin physique

6.2.7 MODE ORIENTÉ CONNEXION

Exemple : X25, ATM

- Transferts successifs sur le même chemin virtuel
- Etablissement préalable d'une liaison physique (commutation de circuits) ou virtuelle (commutation de paquets) et réservation des ressources (tampons, voies) - délai (Transpac 1,5s)
- Séquencement assuré
- Délivrance garantie, accusé de réception, reprise si incident
- Adressage simplifié (adresse de voie) durant la communication, performances améliorées, protocole simplifié

6.2.8 COMMUTATION POUR LE RESEAU TRANSPAC

A des fins de performance, les réseaux se sont orientés sur des techniques dites « de commutation de paquets », cas de TRANSPAC.

Le constat est en effet le suivant : pour relier totalement N abonnés, il faudrait $N(N-1)/2$ liens !. Seuls quelques liens existent, chaque abonné a une ligne de raccordement au réseau de commutation. La ressource « liens / noeuds » est donc rare, il faut optimiser le partage de cette ressource par les techniques de commutation.

Le temps de transit dans un réseau de transmission de paquets est notoirement plus faible que dans un réseau de transmission de messages. C'est très intéressant avec un grand nombre de liens et une taille de paquet faible. De plus, s'il y a des erreurs de transmission l'avantage donné à la transmission de paquets augmente.

6.2.9 CIRCUIT VIRTUEL

Le service de circuit virtuel permet l'établissement d'une « liaison logique » entre les systèmes d'extrémité. Il intervient en phase préalable à la communication. Un circuit virtuel (CV) va reposer sur l'utilisation de voies logiques, établies entre les nœuds du réseau de transport. Une même liaison physique (entre deux nœuds) peut alors être utilisée pour deux liaisons logiques, utiles à deux CV différents (**notion de multivoies**).

Durant la phase d'établissement du circuit logique (CV), les ressources utiles au transfert de données sont réservées (buffers, voies...) Les messages échangés durant cette connexion seront tous orientés sur la voie logique (donc physique) établie. La voie physique est cependant partagée par plusieurs circuits virtuels. Le **circuit virtuel correspond à l'association des voies logiques effectuée.**

Les circuits virtuels peuvent être de type Permanents (CVP) ou Commuté par appel (CVC). Un circuit Permanent peut être considéré comme un circuit Commuté implicitement lors du raccordement initial de l'abonné au réseau de transport. La déconnexion ayant lieu lors de la résiliation du contrat d'attachement au réseau.

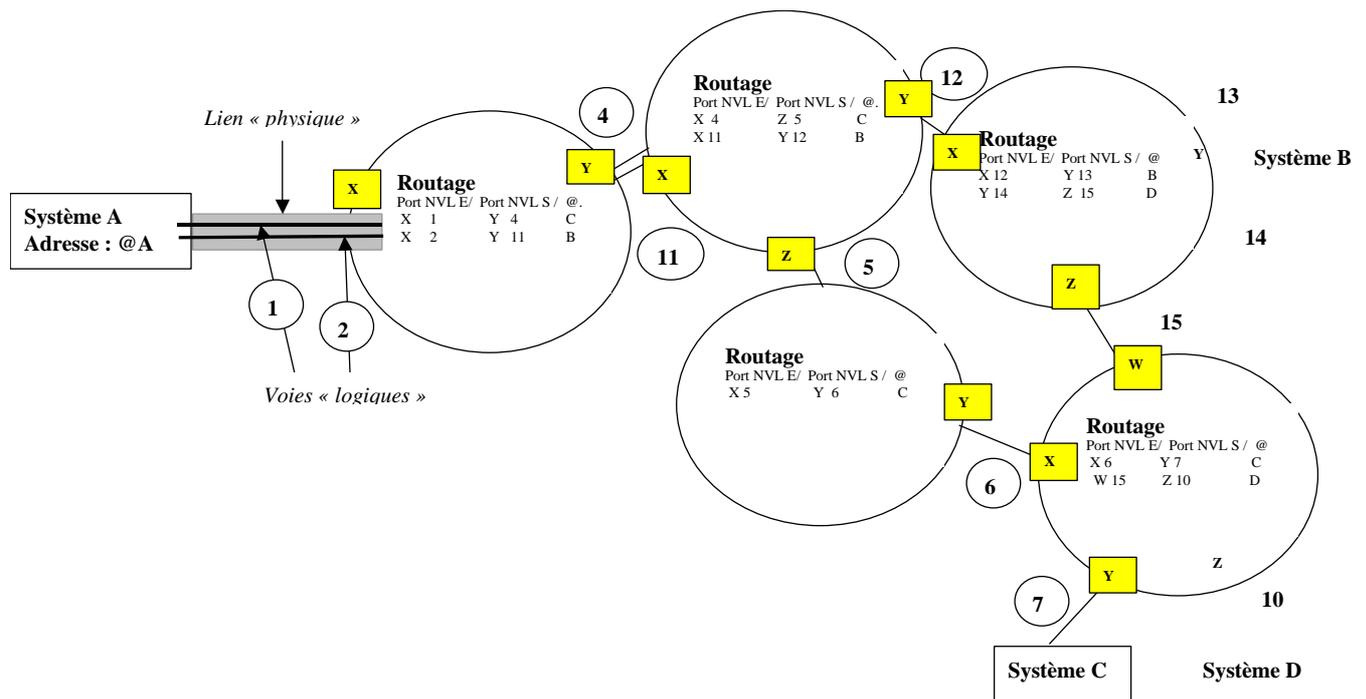


Figure 3 : Circuits Virtuels et liens physiques

6.2.10 EXEMPLE D'ETABLISSEMENT D'UN CIRCUIT VIRTUEL

En tenant compte de la figure ci-dessus, on considère l'établissement d'un circuit virtuel entre les systèmes A et B, à l'initiative de A :

- « A » envoie au commutateur de rattachement un paquet d'établissement de connexion contenant l'adresse du demandé @B et le numéro de voie logique NVL qu'il lui attribue. Dans le cas présent, par exemple NVL=2.
- Le commutateur mémorise l'adresse @B et le NVL=2 vis à vis du port physique de raccord de l'appelé.
- Le commutateur en fonction de sa table de routage et de l'état des liens (connaissance des liens physiques de sorties permettant d'atteindre B, et routage adaptatif), détermine un port de sortie et attribue un NVL de sortie pour ce port. Exemple NVL=11.
- Les informations (adresse du destinataire / port d'entrée / NVL d'entrée / port de sortie / NVL de sortie) constitue une nouvelle entrée pour la table de routage, qu déterminera l'itinéraire des paquets, le routage étant fixe
- Le commutateur d'entrée envoie l'adresse de destination, l'adresse source et le NVL=11 au nœud suivant.
- L'opération se poursuit de nœud en nœuds jusqu'au commutateur de raccord de B. ce dernier fourni à B l'adresse de l'appelant @A et le NVL qu'il lui attribue localement NVL=13.
- La connexion étant dans notre exemple supposée être acceptée, par suite, les systèmes peuvent s'échanger leur donnée en ne spécifiant que les NVL dans leur trames d'informations : chaque nœud répercute l'adresse correspondante dans la table de routage avant de transiter le paquet sur le port physique correspondant.

6.2.11 LE ROUTAGE DANS LES NŒUDS

Le *routing* des paquets dans un réseau maillé consiste à fixer par quelle ligne de sortie chaque commutateur réexpédie les paquets qu'il reçoit. Ceci se fait en fonction de la destination finale du paquet et selon une table de routage qui indique pour chaque destination finale quelles sont les voies de sortie possible.

Pour l'exemple de la figure on pourrait avoir la table de routage suivante :

destination finale	voie de sortie
D1	A1, A2
D2	A2
D3	A2, A3
D4	A3

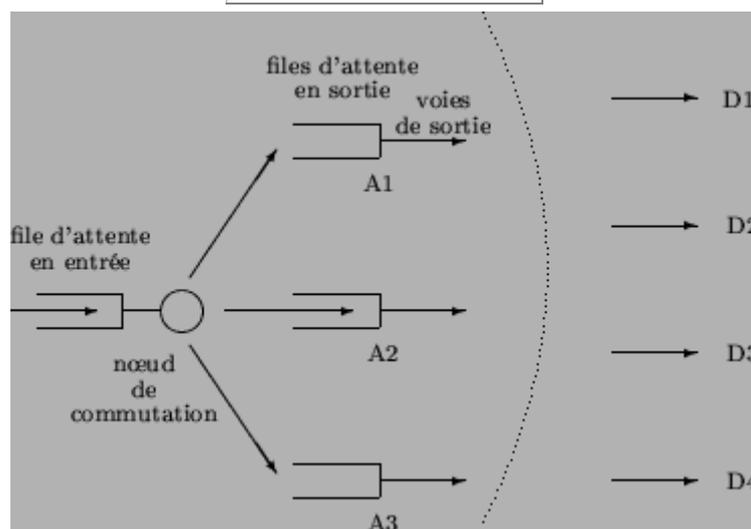


Figure 1.17: Routage de paquets.

D'une manière générale le routage est un ensemble de processus algorithmiques devant prendre des décisions dispersées dans le temps et dans l'espace. Les différents algorithmes sont répartis sur chaque nœud du réseau et l'ensemble peut fonctionner de manière centralisée ou répartie.

- Le *routage centralisé* est géré par un nœud particulier du réseau qui reçoit des informations de chacun des nœuds du réseau et leur envoie leur table de routage. Pour fixer ces tables on prend en compte notamment le coût des liaisons, le coût de passage dans un nœud, le débit demandé, le nombre de nœuds à traverser, la sécurité de transport de certains paquets, l'occupation des mémoires des nœuds de commutation,... Le plus souvent un algorithme de plus court chemin donne de bons résultats en fixant à 1 le coût de franchissement d'un nœud (on peut également pondérer plus fortement les nœuds qui sont les plus occupés). La mise à jour des tables de routage peut se faire de manière

- *fixe* : en fait il n'y a pas de mise à jour, la table de routage est fixée une fois pour toute en fonction de la topologie du réseau.
- *synchrone* : toutes les tables sont mises à jour au même moment par le centre de contrôle qui reçoit des informations de la part de tous les noeuds à intervalles réguliers (toutes les 10 sec par exemple).
- *asynchrone* : les tables sont mises à jour indépendamment les unes des autres dans certaines parties du réseau, chaque noeud envoyant un compte-rendu de son état au centre de contrôle lorsqu'il observe des changements significatifs.

Mais le routage centralisé dans un réseau à grande échelle est peu performant, car un routage est d'autant meilleur qu'il réagit rapidement aux informations qui lui parviennent. De plus, si une panne survient dans l'ordinateur qui assure ce contrôle, c'est tout le réseau qui tombe en panne.

- Le *routage décentralisé* ne possède pas de centre de contrôle et les règles de passage d'un paquet (paquet d'appel pour établissement d'un circuit virtuel) sont :
 - *l'inondation* : à la réception d'un paquet celui-ci est renvoyé sur toutes les lignes de sortie. Cette technique simpliste et rapide est efficace dans les réseaux à trafic faible et où le temps réel est nécessaire mais elle est pénalisante en flux de données, inadaptée aux réseaux complexe et au circuit virtuel.
 - la technique «*hot potatoes*» : un paquet reçu est renvoyé le plus tôt possible par la première ligne de sortie vide. On améliore ce principe en affectant des coefficients à chaque ligne de sortie en fonction de la destination voulue. On tient compte de l'état des noeuds voisins, sans utiliser de paquet de contrôle, mais simplement en comptabilisant le nombre de paquets reçus de chacun d'eux. Ils peuvent également envoyer de manière synchrone ou asynchrone un compte-rendu de leur état, permettant ainsi de choisir la «meilleure» ligne à un instant donné. Mais ceci reste local et une panne du réseau localisée au-delà du premier noeud voisin ne pourra pas être prise en compte.
 - le *routage adaptatif à la fois dans l'espace et dans le temps* demande, de la part de chaque noeud, une connaissance complète du réseau. Les différents noeuds s'échangent donc des messages, mais si chacun envoie des messages à tous les autres le trafic va augmenter de manière beaucoup trop grande. C'est pourquoi un noeud ne transmet un compte-rendu qu'à ses voisins immédiats qui doivent en tenir compte dans leur propre compte-rendu.

D'une manière générale, le routage doit éviter l'usage d'algorithmes adaptatifs trop complexes et limiter les dialogues de services entre les noeuds sinon l'effet obtenu sera à l'opposé de celui recherché.

7 RESUME DES RECOMMANDATIONS DU CCITT

Les recommandations V traitent des transmissions sur les circuits telephoniques

- V1 - Equivalence entre la notation binaire et sa transposition en un code a deux etats caracteristiques.
- V2 - Niveaux de puissance pour la transmission de donnees sur lignes telephoniques .
- V3 - Alphabet International No5.
- V4 - Structure generale des codes de l'Alphabet International No5 en vue des transmissions de donnees sur le reseau commute public.
- V5 - Standardisations des vitesses de modulations et des debits binaires employes dans les transmissions de donnees sur le reseau commute public.
- V6 - Standardisations des vitesses de modulations et des debits binaires employes dans les transmissions de donnees sur les lignes specialisees.
- V10 - Caracteristiques electriques des circuits d'echange en double-courant asymetrique, pour une generalisation d'equipements de transmission de donnees a circuits integres (EIA RS-423).
- V11 - Caracteristiques electriques des circuits d'echange en double-courant symetrique, pour une generalisation d'equipements de transmission de donnees a circuits integres (EIA RS-422).
- V13 - Simulateur de reponse automatique.
- V15 - Utilisation de coupleurs acoustiques pour la transmission de donnees.
- V16 - Recommandations sur les modems destines a la transmission de donnees medicales analogiques.
- V19 - Modems pour la transmission de donnees parallele, utilisant des frequences de signalisation telephonique.
- V20 - Modems pour la transmission de donnees parallele sur reseau commute public.
- V21 - Modem full-duplex a 300 bauds utilises sur reseau commute public.
- V22 - Modem full-duplex a 1200 bits/s sur ligne 2 fils, standardise pour un usage general sur le reseau telephonique commute public et sur les lignes specialisees 2 fils.
- V22 bis - Modem full-duplex a 2400 bits/s full-duplex, utilisant le procede de partage de bande de frequences, pour un usage general sur le reseau telephonique commute public et sur les lignes specialisees 2 fils.
- V23 - Modems 600-1200 bits/s avec retour a 75 bits/s, utilise sur le reseau telephonique commute.
- V24 - Definitions des circuits d'echanges entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) EIA-RS-232C.
- V25 - Appareils avec appel et/ou reponse automatique, relies au reseau telephonique commute.
- V26 - Modems 2400 bits/s utilises sur lignes specialisees 4 fils point-a-point.
- V26 bis - Modems 1200/2400 bits/s standardise pour l'utilisation sur le reseau telephonique commute.
- V27 - Modems 4800 bits/s utilises pour lignes specialisees.
- V27 bis - Modems 2400/4800 bits/s avec egaliseur automatique, standardise pour une utilisation sur lignes louees.
- V27 ter - Modems 2400/4800 bits/s avec egaliseur automatique, standardise pour une utilisation sur le reseau telephonique commute.
- V28 - Caracteristiques electriques des circuits de donnees en differentiel de tension asymetriques (EIA RS-232C).
- V29 - Modems 9600 bits/s pour lignes specialisees.

- V31 - Caracteristiques electriques des circuits simple courant controles par fermeture de contacts.
- V32 - Modem 9600 bits/s standardise pour une utilisation sur le reseau telephonique commute.
- V32 bis - Modem 14400 bits/s standardise pour une utilisation sur le reseau telephonique commute.
- V35 - Transmission de donnees a 48 kbits/s utilisant des circuits primaires dans la bande 60-108 kHz.
- V36 - Modems synchrones pour la transmission de donnees sur circuits primaires dans la bande 60-108 kHz.
- V40 - Indication d'erreurs avec des equipements electro-mecaniques.
- V41 - Systeme de controle d'erreur independant du code.
- V42 - Protocole de correction d'erreur de donnees gere par les modems. Version standardise du protocole MNP4.
- V42 bis - Protocole de compression des donnees gere par les modems. Version standardise du protocole MNP5.
- V50 - Limites standards de la qualite des transmissions de donnees.
- V51 - Organisation de la maintenance des circuits telephoniques internationaux destines a la transmission de donnees.
- V52 - Appareillage pour la mesure du taux d'erreur et de la distorsion en transmission de donnees.
- V53 - Limites de maintenance des circuits telephoniques en transmission de donnees.
- V54 - Equipements de bouclage pour modems.
- V55 - Specifications d'un appareil de mesure du bruit impulsionnel sur les circuits telephoniques.
- V56 - Tests comparatifs de modems utilisant des circuits telephoniques.
- V57 - Equipements de tests complets pour hauts debits numeriques.

Les recommandations X traitent des reseaux de donnees.

X1 - Classes de service internationales dans les reseaux publics de transmission de donnees.

X2 - Installations internationales dans les reseaux publics de transmission de donnees.

X3 - Equipements d'Assemblage/Desassemblage de Paquets (PAD) utilises dans les reseaux publics de transmission de donnees.

X4 - Structure generale des signaux de l'Alphabet International No5 pour la transmission de donnees dans les reseaux publics de transmission de donnees.

X20 - Interface entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) dans les reseaux publics de transmission de donnees en mode Start-Stop.

X20 bis - Interface compatible V21 entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) dans les reseaux publics de transmission de donnees en mode Start-Stop.

X21 - Interface d'usage general entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) dans les reseaux publics de transmission de donnees en mode synchrone.

X21 bis - Utilisation d'equipements terminaux prevus pour employer des modems normalises dans la serie V, des reseaux publics de transmission de donnees.

X24 - Definitions des circuits d'echange entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) dans les reseaux publics de transmission de donnees en mode paquets.

X25 - Interface entre equipement terminal de donnees (DTE) et equipement terminal de communication (DCE) dans les reseaux publics de transmission de donnees en mode paquets.

X26 - Caracteristiques electriques des circuits d'echange en double-courant asymetrique, pour une generalisation d'equipements de transmission de donnees a circuits integres (EIA RS-423).

X27 - Caracteristiques electriques des circuits d'echange en double-courant symetrique, pour une generalisation d'equipements de transmission de donnees a circuits integres (EIA RS-422).

X28 - Interface DTE/DCE pour un equipement terminal en mode start-stop, raccorde a un PAD d'un reseau public de transmission de donnees situe dans le meme pays.

X29 - Procedures pour l'echange d'informations de controle et de donnees utilisateur entre un DTE en mode paquets et un PAD

X92 - Hypothese de connexions de reference pour les reseaux publics de transmission de donnees synchrones.

X95 - Parametres de reseau dans les reseaux publics de transmission de donnees.

X96 - Cheminement d'appel dans les reseaux publics de transmission de donnees.

X200 - Modele de reference d'OSI (Open Systems Interconnection) pour les applications CCITT.

X210 - Conventions de definition des couches OSI.

8 BIBLIOGRAPHIE

- Interbus Basic brochure de PHOENIX CONTACT
- Réseaux locaux, normes et protocoles 3^{ième} édition revue et augmentée - Pierre Rolin HERMES
- Manuel utilisateur de l'interface UNITELWAY Dev/O TECHNOLOG

Et sur le WEB ... :

1. Claude Servin, Réseaux et Sécurité Dunod
2. François Flückiger Cours de Logiciels et Réseaux Informatique 1^{er} année
3. http://www.cyber.uhp-nancy.fr/demos/GRT-002/cha_3/cha_3_2.html
4. <http://www.info.univ-angers.fr/pub/pn/poly/node7.html>
5. <http://www.univ-paris12.fr/lac/tan/Reseau/newintro/sld051.htm>
6. http://www.cyber.uhp-nancy.fr/demos/GRT-002/cha_2/cha_2_3_1.html
7. <http://www.univ-irem.fr/commissions/ci3m/CDci3m/marseille/fichiers/fourcar.doc>
8. <http://isc.cemif.univ-evry.fr:8080/~hoppenot/enseignement/cours/rli/ls.pdf>
9. <http://www.epsic.ch/cours/BT.php>
10. <http://www.arcelect.com/rs232.htm>
11. <http://www.urec.curs.fr/cours/Physique/xmitdata/sld022.htm>
12. <http://www.commentcamarche.net/transmission/transanalog.php3>

-