COMMUNICATIONS

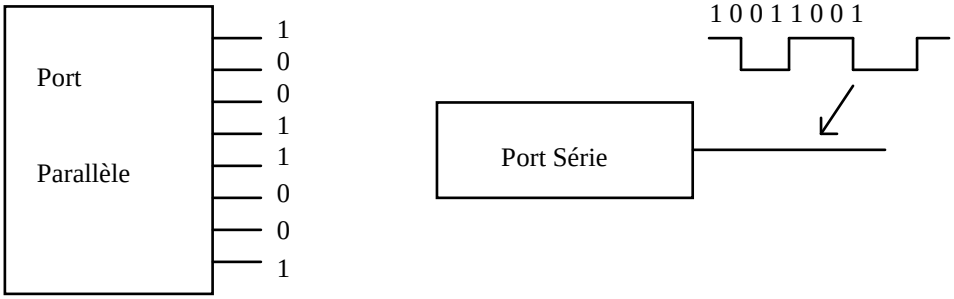
SERIES

1) PRESENTATION GENERALE

a) Principe de base

Les communications séries servent à transmettre des informations d'un équipement à un autre.

Les bits sont transmis en série les uns à la suite des autres :



Les communications sont définies par des normes les avis : V du CCITT ou RS de la norme américaine.

Les communications séries sont définies par 2 critères :
- le format de transfert des données de communication.
- l'interface physique de communication.

Le format de transfert des données de communication peut être divisé en 2 types :
- le format synchrone.
- le format asynchrone.

Ce document n'aborde que les transmissions asynchrones.

Les communications asynchrones sont définies par plusieurs paramètres :
- les signaux de communications.
- la vitesse de transmission ou baud rate.
- le format des données.
- le mode Full ou Half Duplex.

b) Les interfaces

L'interface physique de communication peut être très divers mais nous n'aborderons que 3 types d'interface :
- l'interface RS232C.
- l'interface RS422.
- l'interface RS485.

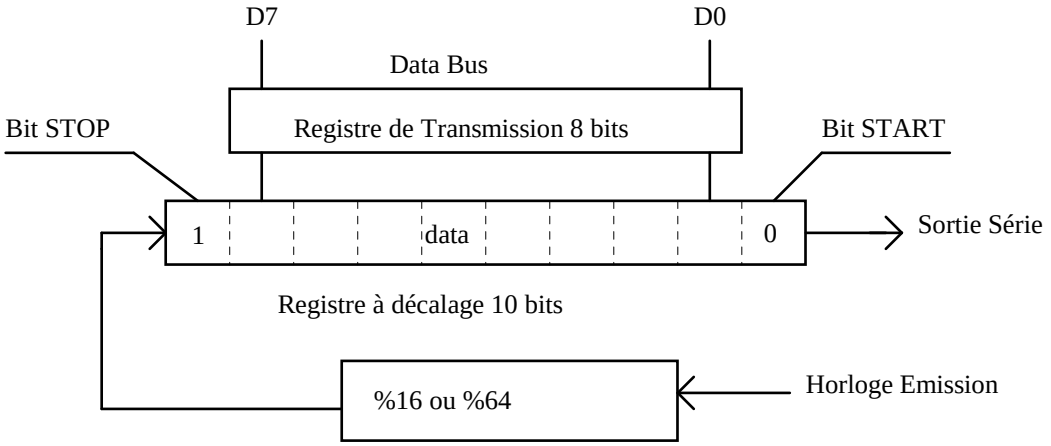
Le brochage et le type des connecteurs font partie de la norme.
Pour la norme RS232C il existe deux types de connexions :
- DTE : Data Terminal Equipement (Mode Terminal).
- DCE : Data Communication Equipement (Mode Modem).

Ces deux modes diffèrent par le sens de transfert des données sur les lignes de réception et de transmission.

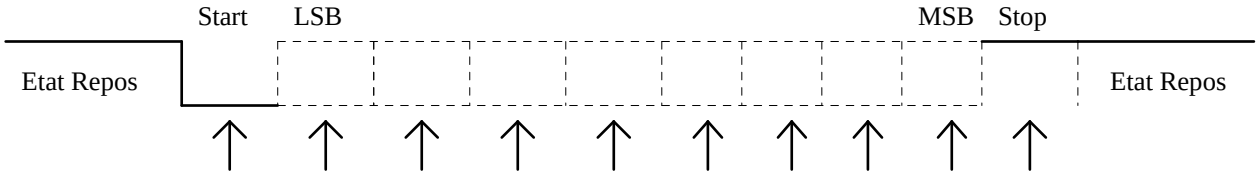
2) PRINCIPE DE TRANSMISSION

a) Principe de l'émission asynchrone.

La transmission de données s'appuie sur le principe des registres à décalage.



Tout composant périphérique d'entrée/sortie série comprend un registre 8 bits qui reçoit les codes ASCII à transmettre.
Pour que le récepteur puisse détecter le début et la fin de transmission, le composant périphérique ajoute, au mot de 8 bits, un bit de START (0) et un bit de STOP (1).
Ce mot de 10 bits est transmis dans un registre à décalage qui assure la transmission.
Le premier bit transmis est le bit de poids faible.



Echantillonage Par le Récepteur

b) Registre de contrôle et registre d'état.

Dans chaque composant périphérique il existe un registre de contrôle qui permet de:

- fixer le format des mots transmis (7 ou 8 bits)
- fixer le facteur de division de l'horloge (%16, %64 ou autre).
- le nombre de bits de start et de stop.
- les bits de test de parité.
- préciser le fonctionnement des interruptions.
- etc...

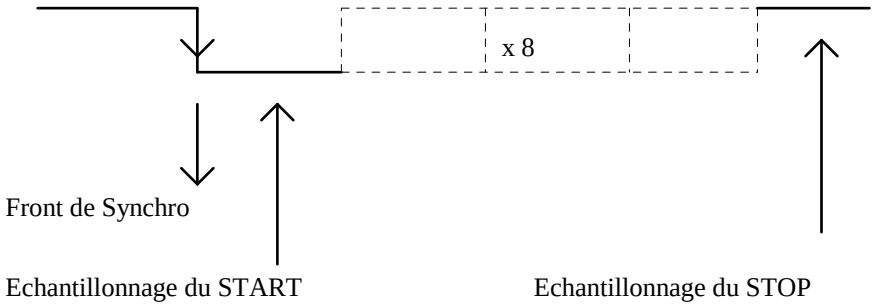
Dans chaque composant périphérique il existe un registre d'état qui permet de:

- savoir si une transmission est en cours.
- savoir si une réception est en cours.
- connaître l'état de lignes de contrôle.
- connaître l'état des interruptions
- etc...

c) Principe de la réception asynchrone.

Le récepteur possède une horloge qui doit être à la même fréquence que l'horloge d'émission.
Pour pouvoir lire un bit il faut synchroniser les deux horloges : c'est le rôle du bit de start.

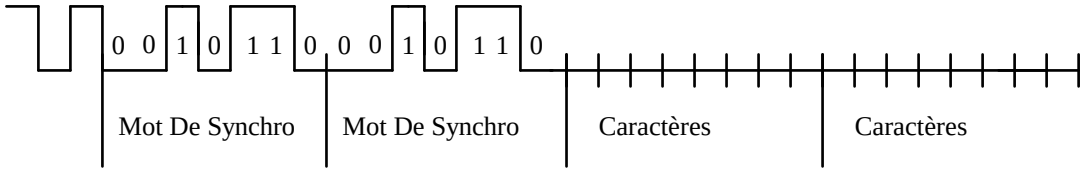
- Le récepteur attend le premier front descendant de la ligne de donnée pour synchroniser l'horloge de réception.
- Il vient tester le premier bit après 1/2 période d'horloge.
  si ce bit vaut 1 on a un mauvais bit de start
  si ce bit vaut 0 on a un vrai bit de start.
- Le récepteur échantillonne les bits de données à chaque période.
- Le récepteur test le bit de stop
  si ce bit vaut 0 on a une erreur d'encadrement (framing error).
  si ce bit vaut 1 on valide le bit de réception plein.



d) principe de la transmission synchrone.

Dans une liaison synchrone la transmission ne se fait plus individuellement caractère par caractère, mais par blocs de caractères (128, 256 ou 512).
Le début et la fin des blocs est signalée par des caractères de synchronisation.
Il faut que l'horloge de réception et l'horloge d'émission soit exactement les mêmes et synchrones.
Pour cela on peut:

- transmettre l'horloge.
- synchroniser l'horloge de réception sur le signal par une PLL.
- utiliser des caractères de synchronisation.



Le format des blocs est plus complexe qu'en transmission asynchrone.
Les vitesses atteintes peuvent aller jusqu'à 10 Méga Bits/s.

3) FORMAT DES COMMUNICATIONS ASYNCHRONES.

a) Signaux de Communications entre 2 équipements.

Description des Signaux.

Les signaux de communications comportent :

- des signaux de données et horloges
- des signaux de contrôle de flux de transmission
- des signaux de signalement de présence
- des références de potentiel

SIGNAUX DE DONNEES ET HORLOGES :

- TXD transmit data — Données dans un sens.
- RXD receive data — Données dans l'autre sens.

- TXC transmit clock — Horloge de transmission.
- RXC receive clock — Horloge de réception.

SIGNAUX DE CONTROLE DE FLUX DE TRANSMISSION:

- RTS request to send — Demande à émettre
- CTS clear to send — Prêt à recevoir

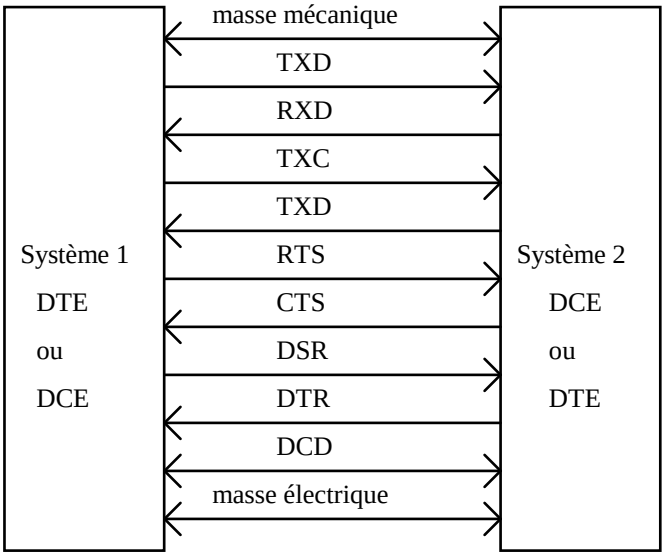
Ces signaux permettent d'indiquer d'un équipement à l'autre l'état des buffers de réception: si l'équipement est prêt à recevoir les données.

SIGNAUX DE SIGNALEMENT DE PRESENCE :

- DTR data terminal ready — Equipement 1 prêt
- DSR data set ready — Equipement 2 prêt
- DCD data carrier detect — Présence de la porteuse de transmission

Ces signaux sont rarement tous utilisés et rarement dans leur fonction prévue par la norme.

Il existe d'autre signaux qui ne sont utilisés que pour les transmissions par modem. Ils ne sont pas détaillés dans ce document.



b) La vitesse de transmission des données.

La vitesse de transmission des données ou **baud rate** définit le nombre de bits transmis par seconde si la ligne est utilisée à son maximum de vitesse. Ceci n'est vrai que si les émetteurs et récepteurs sont toujours prêts à émettre ou à recevoir.
Les vitesses les plus utilisées :

2400 bauds
4800 bauds
9600 bauds
19200 bauds
38400 bauds

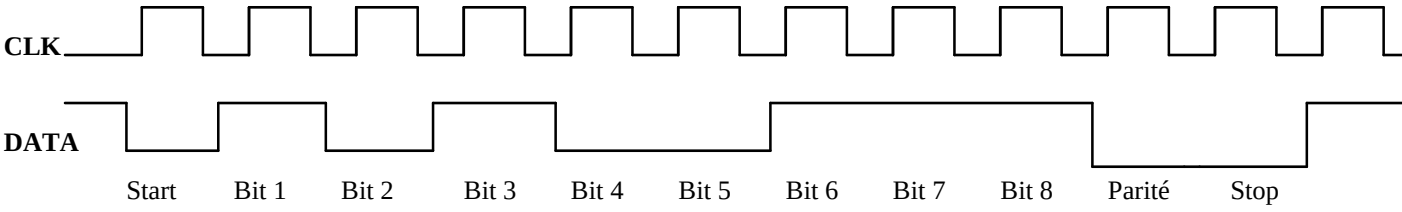
Ce baud rate peut être différent en émission et réception. Il est généralement programmable dans les contrôleurs de communications série type 6850, 68681 ou 2691 ou dépendant des horloges externes RXC et TXC.

c) Le format des données

Le format des données est défini par :

1 bit de start
7 ou 8 bits de données
1 bit de parité
1ou 2 bits de stop

Les signaux sont actifs à zéro.



Chronogramme transmission asynchrone

Il est évident que lorsque 2 équipements veulent communiquer il faut que leurs configurations soient identiques.

d) Les modes HALF et FULL DUPLEX

Ces modes sont utilisés par exemple en connexion d'un terminal sur une carte unité centrale.

- **mode Half Duplex.**

La transmission est bidirectionnelle entre deux équipements, mais à un instant donné la transmission ne se fait que dans un sens.

Exemple : un terminal en Half Duplex

Lorsque l'utilisateur frappe une touche sur le clavier:
- le terminal envoie le code ASCII de la touche vers la liaison série.
- le terminal affiche immédiatement le caractère sur l'écran:
  il n'attend pas de réponse en écho pour procéder à l'affichage.
- la transmission ne se fait que du terminal vers la carte CPU.

- **mode Full Duplex.**

La transmission est bidirectionnelle entre deux équipements, mais à un instant donné la transmission peut se faire dans les deux sens.

Exemple : un terminal en Full Duplex

Lorsque l'utilisateur frappe une touche sur le clavier:
- le terminal envoie le code ASCII de la touche vers la liaison série.
- le terminal n'affiche le caractère sur l'écran que lorsqu'il en reçoit un identique sur sa broche de réception.
  il attend une réponse en écho pour procéder à l'affichage.
- pendant qu'il attend une réponse il peut envoyer un second caractère.

4) CONTRÔLE DE FLUX DE DONNEES

Dans toute transmission il faut pouvoir contrôler les flux entre les équipements pour éviter de perdre des informations.

a) Contrôles avant la transmission.

Avant d'envoyer des données le transmetteur doit vérifier que le récepteur est prêt à recevoir les données. Même chose pour le récepteur.

C'est le rôle des lignes DTR, DSR DCD.

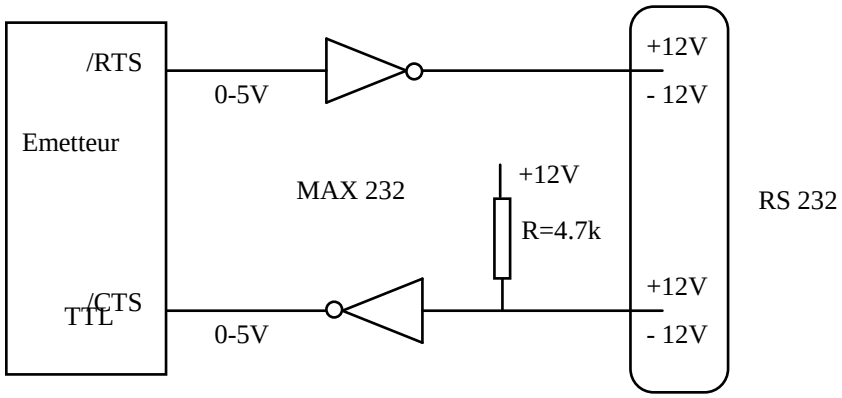
Si l'une de ces lignes devient inactive pendant la transmission il faut être capable de le détecter (soit par scrutation soit par interruption).

b) Contrôles matériels pendant la transmission.

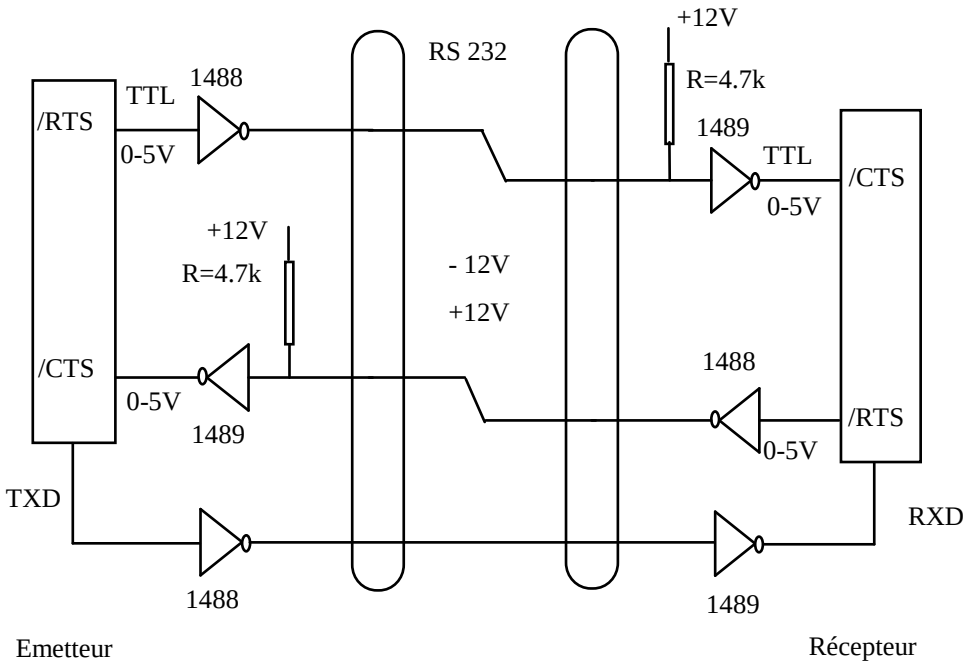
Il est possible contrôler le flux de données entre deux caractères par deux lignes de "handshake" RTS et CTS.

L'émetteur informe le récepteur qu'il est prêt à émettre par RTS.
La Broche Request To Send (demande d'émission) est une sortie de l'émetteur pour les interfaces séries. Elle est active au niveau bas.

Le récepteur informe l'émetteur qu'il est prêt à recevoir par CTS.
La broche Clear to Send (Inhibition de l'émetteur est une entrée de l'émetteur). Elle est active au niveau bas. Un niveau haut bloque l'émetteur.



Sur toute les cartes on dispose de résistances de Pull Up pour imposer un niveau Actif Bas sur les entrées. Entre un émetteur et un récepteur on a le câblage suivant:



REMARQUE :

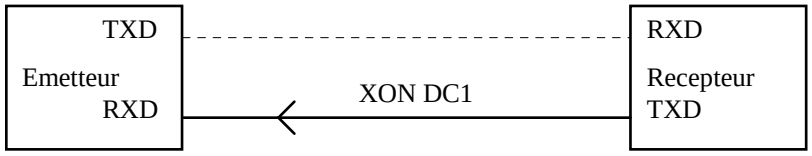
Certain logiciels nécessitent simplement le bouclage de RTS sur CTS.

c) Contrôles logiciels pendant la transmission.

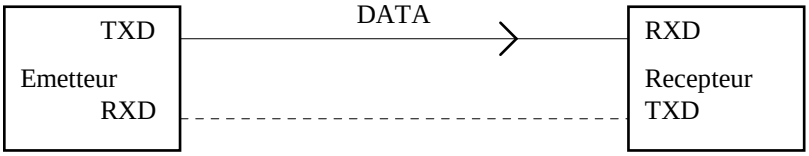
Il est possible contrôler le flux de données par deux codes ASCII "XON" et "XOFF" (code DC1 DC3). Ces deux codes sont transmis comme des caractères standards

- XOFF (CTRL S) demande l'arrêt d'une transmission.
- XON (CTRL Q) demande le départ d'une transmission.

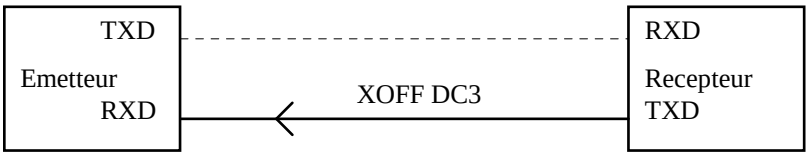
Autorisation de Transmission



Transmission des Données



Demande d'Arrêt de Transmission



2) CONTRÔLE DE PARITE

Il peut y avoir des erreurs pendant la transmission de données. Pour contrôler les bits transmis on peut ajouter un bit de parité par octet transmis.

La parité d'un nombre binaire est obtenue en calculant le nombre de bits à 1 du mot.

Si ce nombre est pair la parité est paire.
Si ce nombre est impair la parité est impaire.

Dans une transmission asynchrone il est possible de choisir une parité paire ou impaire.
Le bit de parité vaut 1 si la parité est respectée.

Les deux équipements calculent chacun de leur coté la parité. Le récepteur vérifie le bit de parité reçu à celui qui est calculé et indique une erreur éventuelle.

3) INTERFACES PHYSIQUES

Trois types d'interfaces sont abordés dans ce document :

- les interfaces RS232C.
- les interfaces RS422.
- les interfaces RS485.

Le choix d'un interface est fonction :

- de la vitesse de transmission.
- de la longueur du câble.
- de l'immunité aux parasites.

a) Interfaces RS232C.

- **Niveau des tensions.**

  Les interfaces RS232C utilise des composants du types 1488 ,1489, LT 1180 ou MAX 232.

  Ils permettent de détecter des transitions de +3V ou -3V.
  Ils sont alimentés +12V et - 12V mais peuvent aussi être alimenté en +5V et - 5V, 8V.....(la tension maximale en réception est de 25 V).

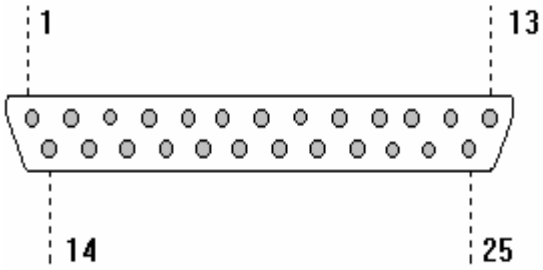
  La tension d'alimentation permet d'avoir plus de longueur de câble et des tensions différentielles de masse entre équipement plus importantes.

  Les interfaces RS232C sont bien adaptés pour communiquer jusqu'à 9600 bauds à une distance maximum de 12 mètres.

- **Les connecteurs.**

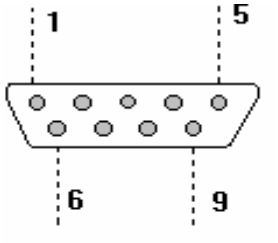
Connecteurs RS232 SUBD 25 points :

Broche	Signal	Broche	Signal
1	masse mécanique	14	
2	TXD	15	TXC
3	RXD	16	
4	RTS	17	RXC
5	CTS	18	
6	DSR	19	
7	masse électrique	20	DTR
8	DCD	21	
9		22	
10		23	
11		24	
12		25	
13			

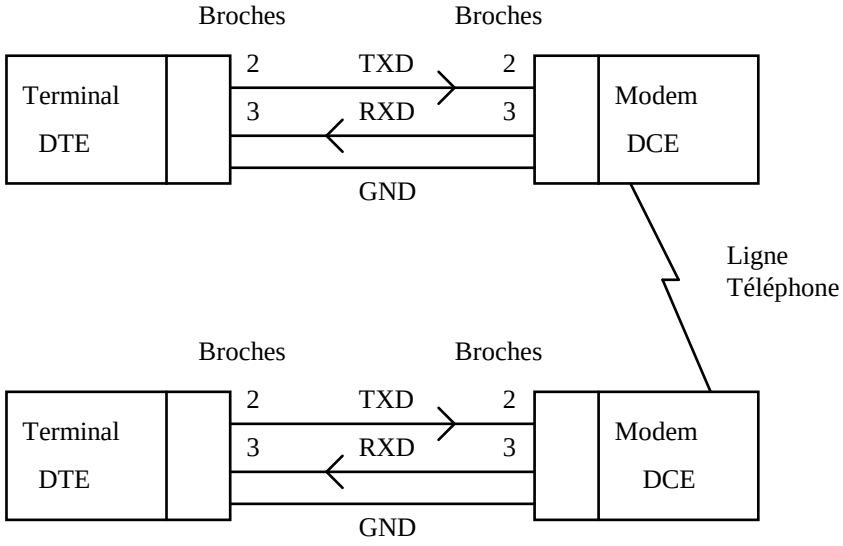


Connecteurs RS232 SUBD 9 :

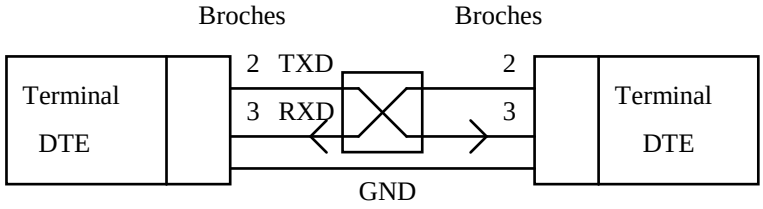
Broche	Signal	Broche	Signal
1	DCD	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	
5	masse électrique		



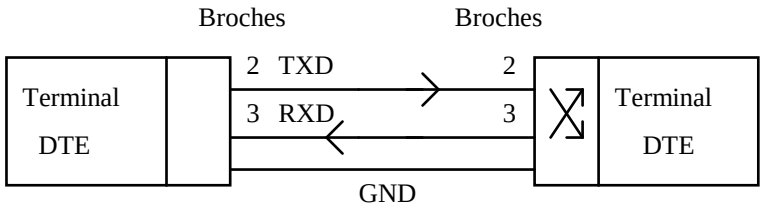
- **Connexion entre un Terminal et un Modem**



- **Connexion entre deux terminaux**



Connexion DTE-DTE Null Modem entre deux "Terminaux"



Connexion DTE-DTE sans Null Modem entre deux "Terminaux"

b) Interface RS422.

Les interfaces RS422 utilisent des buffers de types 26LS31 et 26LS32

Les transmissions RS422 sont des transmissions différentielles de bout en bout sur câble chargé par des faibles impédances donc des forts courants :

charges types 100 ohms à l'extrémité donc sous 5V 50 mA.

Les buffers d'émission présentent en sortie en même temps le signal direct et le signal inversé.

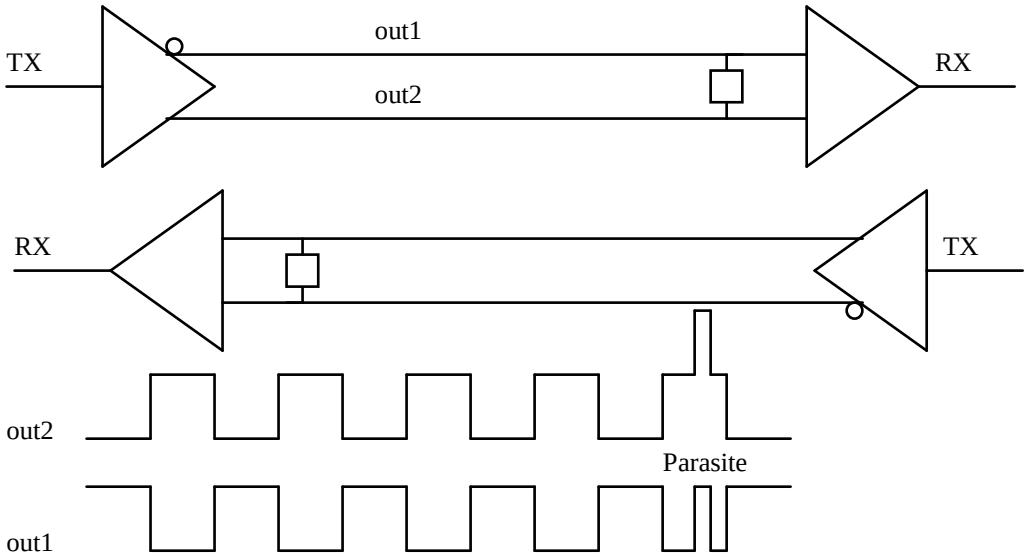
Les buffers de réception mesurent la différence de potentiel entre les deux entrées.

Ces buffers permettent des tensions de mode commun entre équipement jusqu'à 7V. (Elimination des parasites par soustraction de signaux)

Ils peuvent travailler à :
- 10 Méga bits/secondes à des distances de 12m.
- 100 K bits/secondes à des distances de 1200m.

Ils sont protégés contre les courts-circuits en sortie.

Entre les émetteurs et la ligne il peut y avoir un isolement galvanique par opto-coupleurs.



Synoptique connexion RS422

Les connecteurs RS422 ne sont pas définis clairement par la norme

c) Interface RS485.

Les interfaces RS485 sont prévus pour être utilisés dans des applications réseau où plusieurs équipements sont interconnectés par un seul câble type paire blindée. Les caractéristiques de transmission sont semblables à celles des buffers RS422.

Les interfaces RS485 utilisent des buffers types DS 75176.

Les transmissions RS485 sont des transmission différentielles multi-points bidirectionnelles sur câble chargé par des faibles impédances donc des forts courants.
charges types 100 ohms à chaque extrémité donc sous 5V 100 mA.

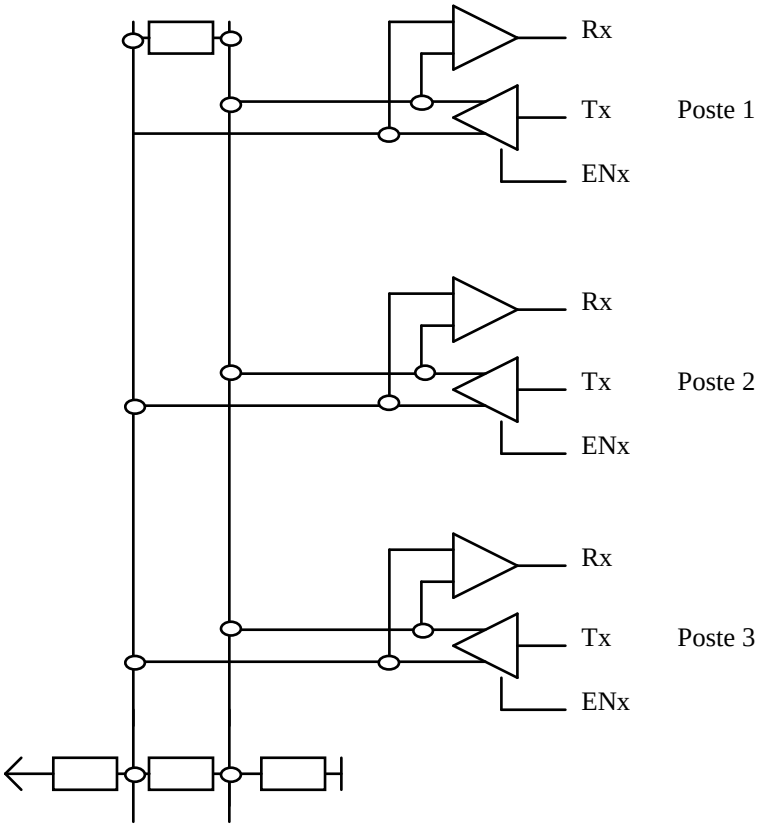
Ces buffers permettent des tensions de mode commun entre équipement jusqu'à 7V.

Les Buffers d'émission peuvent se mettre en état de haute impédance.

Ils peuvent travailler :
10 Méga bits/secondes à des distances de 12m
100 K bits/secondes à des distances de 1200m

Ils sont protégés contre les courts-circuits en sortie. Entre les émetteurs et la ligne il peut y avoir un isolement galvanique par opto-coupleurs.

BUS SERIE



Synoptique interconnexion RS485

L'émission et la réception ne peuvent se faire en même temps ce qui implique que les communications se font en

HALF DUPLEX Maitre Esclave

TABLE DE CODES ASCII

CARAC	HEXA	CARAC	HEXA	CARAC	HEXA	CARAC	HEXA
NULL	$00	SP	$20	@	$40	`	$60
SOH	$01	!	$21	A	$41	a	$61
STX	$02	"	$22	B	$42	b	$62
ETX	$03	#	$23	C	$43	c	$63
EOT	$04	$	$24	D	$44	d	$64
ENQ	$05	%	$25	E	$45	e	$65
ACK	$06	&	$26	F	$46	f	$66
BEL	$07	'	$27	G	$47	g	$67
BS	$08	(	$28	H	$48	h	$68
HT	$09	)	$29	I	$49	i	$69
LF	$0A	*	$2A	J	$4A	j	$6A
VT	$0B	+	$2B	K	$4B	k	$6B
FF	$0C	,	$2C	L	$4C	l	$6C
CR	$0D	-	$2D	M	$4D	m	$6D
SO	$0E	.	$2E	N	$4E	n	$6E
SI	$0F	/	$2F	O	$4F	o	$6F
DLE	$10	0	$30	P	$50	p	$70
DC1	$11	1	$31	Q	$51	q	$71
DC2	$12	2	$32	R	$52	r	$72
DC3	$13	3	$33	S	$53	s	$73
DC4	$14	4	$34	T	$54	t	$74
NAK	$15	5	$35	U	$55	u	$75
SYN	$16	6	$36	V	$56	v	$76
ETB	$17	7	$37	W	$57	w	$77
CAN	$18	8	$38	X	$58	x	$78
EM	$19	9	$39	Y	$59	y	$79
SUB	$1A	:	$3A	Z	$5A	z	$7A
ESC	$1B	;	$3B	[	$5B	{	$7B
FS	$1C	<	$3C	\	$5C	|	$7C
GS	$1D	=	$3D	]	$5D	}	$7D
RS	$1E	>	$3E	^	$5E	~	$7E
US	$1F	?	$3F	_	$5F	DEL	$7F

Les réseaux TCP/IP

1. Le réseau Ethernet TCP/IP

a. Terminologie réseaux de base

Un réseau local d'ordinateurs, **« intranet »,** comporte à la fois le matériel et le logiciel pour permettre aux ordinateurs de communiquer entre eux. Chaque ordinateur connecté à un réseau est un **« host ».**

Les « hosts » peuvent être et sont souvent situés dans des sites différents. Les « hosts» peuvent être de type différents.

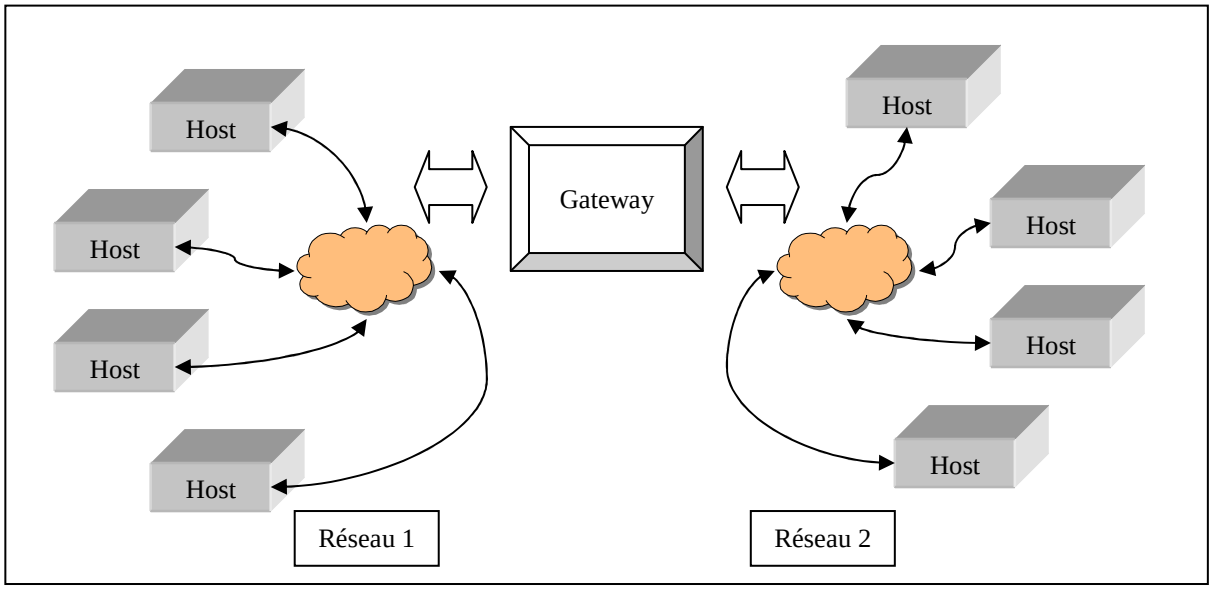
Par exemple il est possible d'avoir des ordinateurs personnels connectés à des stations de travail ou des serveurs. Les systèmes d'exploitation peuvent être variés : Windows, Unix, VAX/VMS ou autres.

Un **internet** consiste en la connexion de deux ou plusieurs réseaux locaux qui permet aux ordinateurs des différents réseaux de de communiquer entre eux. On parle parfois de **internetwork.**

Les réseaux sont connectés entre eux pas des passerelles **« gateways »**.

Une passerelle **« gateways »** achemine les messages d'un réseau à un autre réseau

Le shéma ci-desous présente une passerelle entre deux réseaux :

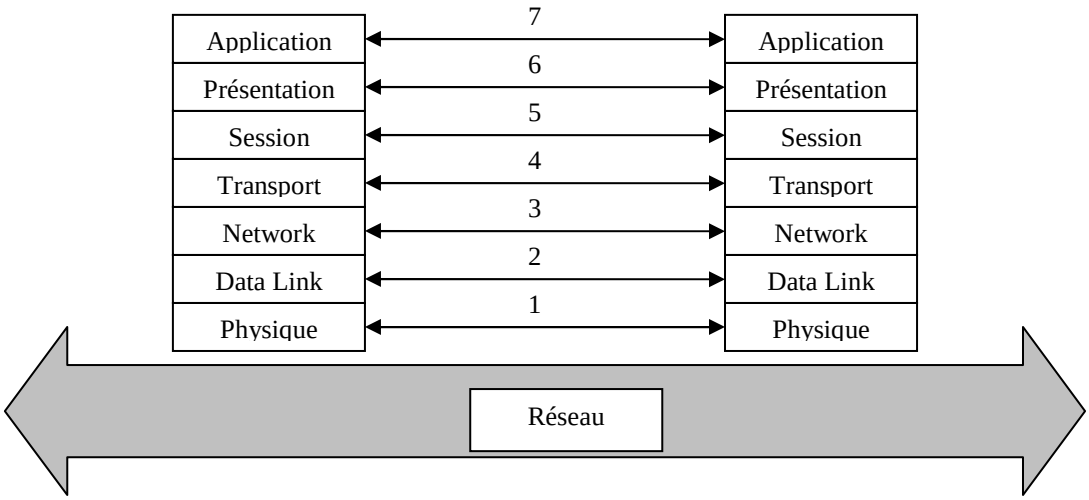


b. Modèle OSI pour les réseaux

L'Organisation Internationale pour la Standardisation « **OSI »** a crée un modèle, sous le nom **« Open System Interconnection »**, ou encore OSI model, comme un cadre conceptuel pour developper les protocoles standards.

Un protocole est un ensemble de règles et de conventions qui permet la modularité et la portabilité des systèmes. **Ethernet TCP/IP** est un protocole de communication.

Le modèle **« OSI »** comprend sept couches conceptuelles organisées comme ci-dessous :



c. La couche Physique

La couche physique spécifie le modèle d'interconnexion, y compris les caractéristiques de courant et de tension. C'est la couche la plus basse de l'architecture. La norme **«Ethernet »** précise ces caractéritiques.

d. La couche Data Link

La couche Data Link est la couche d'interface matérielle. Le matériel informatique produit un flux de bits qui est mis en forme par cette couche sous la forme de trames. Cette couche définit le format des trames et précise comment deux machines détectent le début et la fin de trames. Comme il peut y avoir des erreurs dans la transmission cette couche définit un principe de détection d'erreur sous forme d'un **« checksum ».** On peut imaginer cette couche comme une fenêtre vers les composants matériels tels qu'une carte réseau Ethernet.

e. La couche Network

La couche réseau définit le format des trames d'un réseau particulier. Elle comporte les fonctions qui complètent l'interaction entre le « host » et le réseau. Elle définit la forme de la trame de base et du principe de transfert dans le réseau. Elle comprend les concepts d'adresse **« source »** de **« destination »** et de **« routage »**. Le protocle **« IP », « Internet Protocol »**, en est un exemple.

f. La couche Transport

La couche transport permet la liaison point à point entre deux machines. Elle permet à l'ordinateur **« source »** de se connecter à l'ordinateur **« destination »**. Le protocole **« TCP », « Transmission Internet Protocol »,** en est un exemple.

g. La couche Session

La couche session maintient la connexion ouverte pendant qu'une application est active.

h. La couche Présentation

La couche présentation réalise l'interface entre le réseau et l'application. Elle met en forme les données pour que l'application puisse les analyser. Elle convertit les données sous forme de standards en respectant des règles de codage.

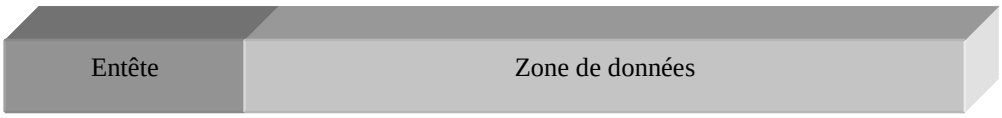
i. La couche Application

La couche application communique directement avec le programme qui utilise les informations issues du réseau.

2. Les datagrams

Quand une information transite sur un réseau entre deux ordinateurs, situés soit sur le même réseau, soit au travers d'un « gateway », les données sont transmises sous forme de paquets **« packets ».** Un paquet est un ensemble de données physiques qui traverse les couches réseau. Un **« datagram »** caractérise le type de paquets et la forme de l'unité de base d'information qui transite à travers le réseau. L'Internet appelle cette unité **« Internet Datagram »** ou **« IP Datagram »**

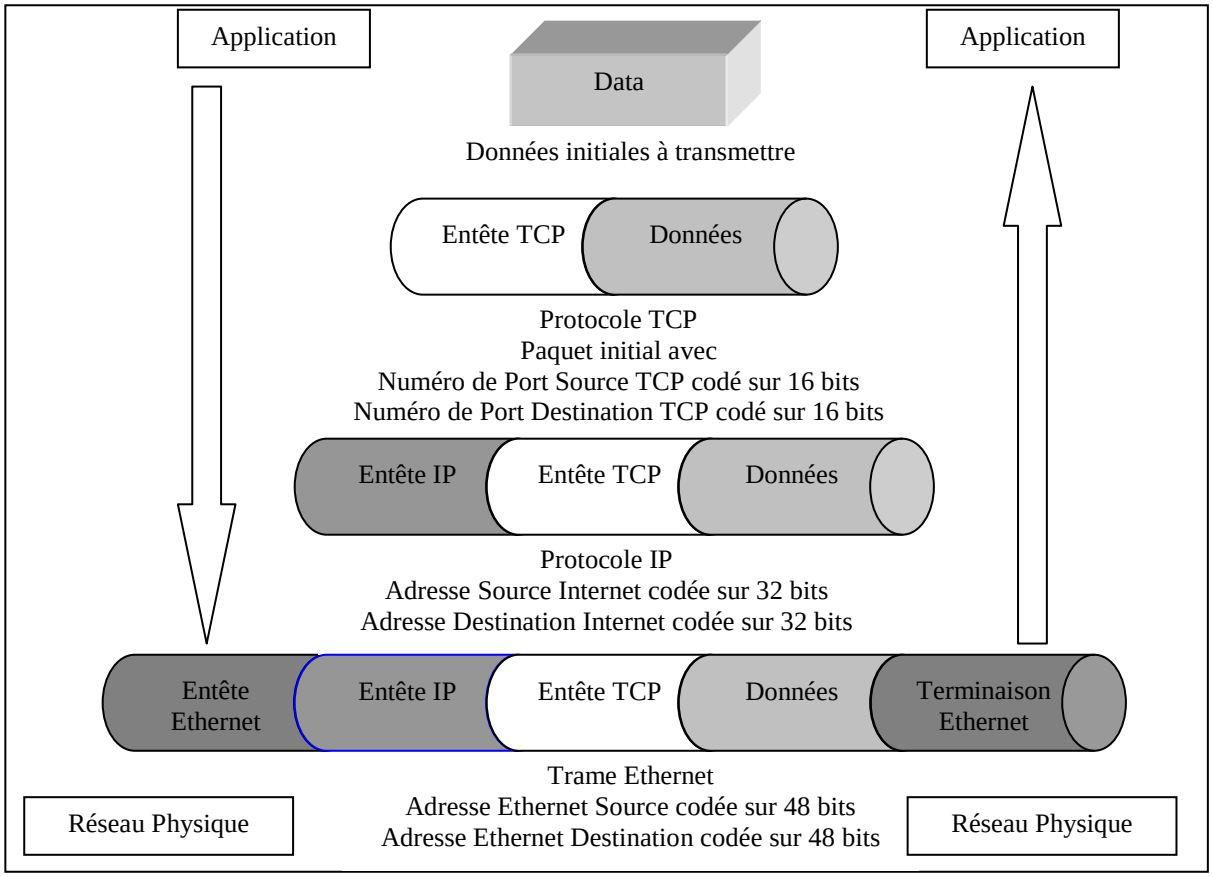
Un Datagram est composé d'une entête et d'une zone de données. L'entête contient l'adresse de source et de destination des adresses de l'Internet Protocol **« IP »** et un champ qui identifie le contenu du « datagram »



Le datagram dépend du **« MTU », Maximum Transfert Unit,** du réseau. Chaque réseau possède différents MTU. Quand un datagram doit passer sur un réseau de faible MTU, Internet décompose les grands datagram en plus petits fragments. Ce principe s'appelle la fragmentation.

La fragmentation apparaît souvent sur une « gateway » quelque part entre l'adresse de source et l'adresse de destination. Le « gateway » reçoit un datagram d'un réseau à large MTU et doit le transmettre à un autre de plus faible MTU. La taille du fragment est calculée sur la base de la plus petite trame accessible sur le réseau de plus faible MTU.

Une trame traverse la couche « data link » et contient un datagram encapsulé. Quand on ajoute une information à un datagram, cette opération s'apelle « l'encapsulation ». Les datagram sont encapsulés à chaque passage dans une couche réseau. Le schéma ci-dessous explique ce principe.



Les processus clients serveurs

Les termes **« client »** et **«serveur** » apparaissent souvent dans une documentation réseau.

- Un **processus** est une tâche qui tourne sur une machine. Elle est construite autour d'un programme d'application et de tout son environnement de travail, code machine, mémoires, variables, entrées sorties…
- Un **serveur** est un processus installé sur un ordinateur qui fournit un service spécifique sur un réseau.
- Un **client** est toute tâche implantée sur un ordinateur qui utilise un service founit par un serveur.

3. Les protocoles

Quand les processus clients et serveur communiquent sur un réseau, les deux processus doivent suivre un ensemble de règles et de conventions communes. Ces règles sont connues sous le terme de **« protocole »**. Sans protocole les ordinateurs ne peuvent pas communiquer, il doivent parler le même language.

Les protocoles utilisés sur l'Internet sont :

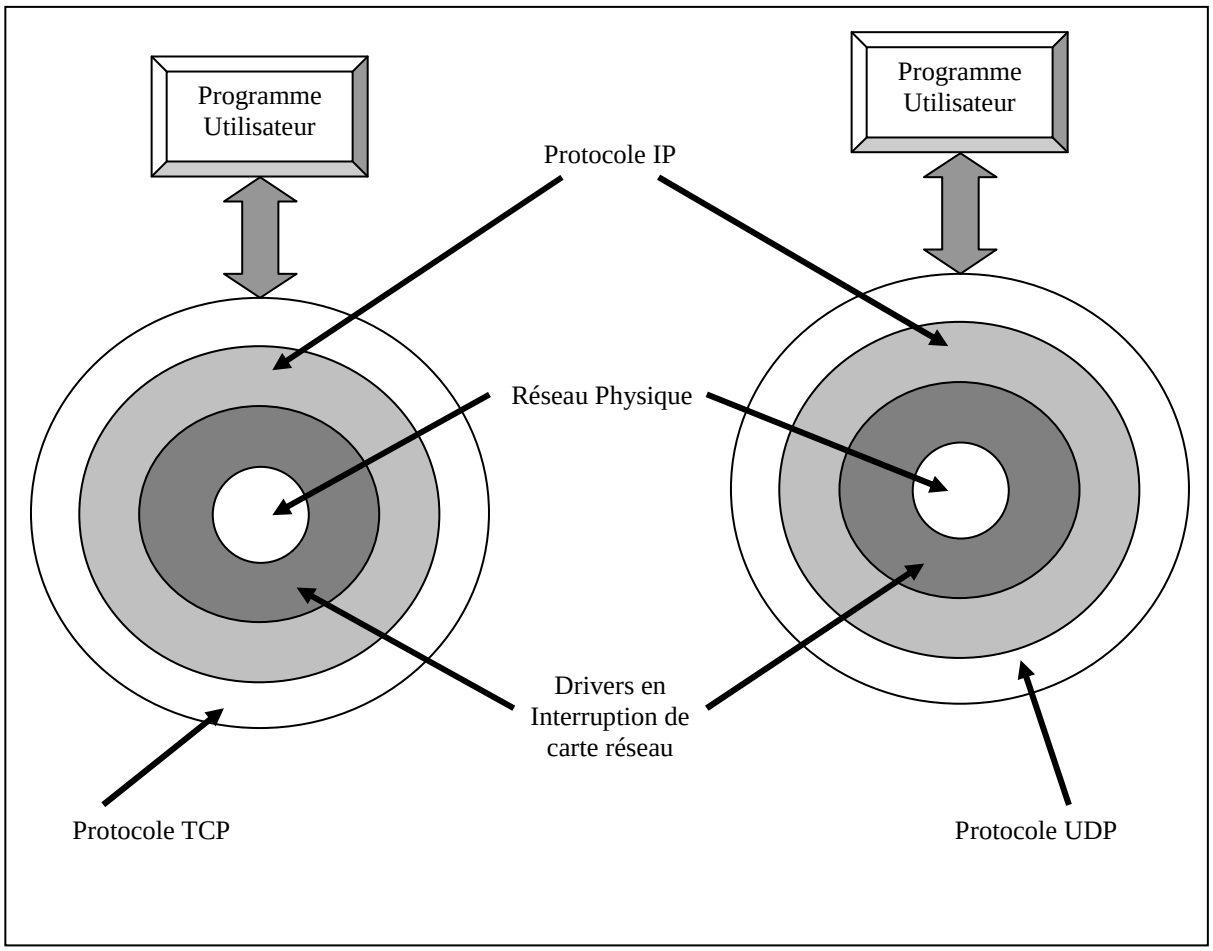
- Le **P**rotocole **I**nternet : **IP.**
- Le **P**rotocole de **C**ontrôle de **T**ransmission : **TCP**.
- Le **P**rotocole de **D**atagram **U**tilisateur : **UDP**.

a. Internet Protocole : IP

Le protocole Internet (IP) est le protocole qui transmet les datagram sur internet. IP est protocole de bas niveau situé au dessus des « drivers » de l'interface physique réseau est au dessous des protocoles de niveau supérieurs tels que « User Datagram Protocole » UDP ou « Transmission Control Protocol » TCP.

IP est donc un protocole qui fournit les paquets de données aux protocoles de niveaux supérieurs tels que TCP et UDP.

Le programmes utilisateurs utilisent donc IP au travers des protocoles tels que UDP et TCP.



Compte tenu de sa situation la couche **IP,** les datagram traverssent cette couche dans les deux sens.

- Depuis le réseau IP vers l'application utilisateur
- Depuis l'application utilisateur vers l'intérieur du réseau.

La couche IP construit un « checksum » dans une partie de l'entête, mais ne fournit pas les données à transmettre. Le « checksum » est ajoutée au datagram lors de la transmission et est aussi testé lors de la réception.

Un cheksum est un entier qui est le résultat d'un calcul sur les données transmises entre deux machines. Il permet de grantir l'intégrité des données.

La couche IP supporte la fragmentation et l'assmblage. Si le datagram est plus grand que ce que peut supporter le MTU du réseau, les datagram sont fragmentés en sortie. En réception les datagram sont éliminés de la file d'attente si le datagram complet n'est pas reconstruit en un minimum de temps.

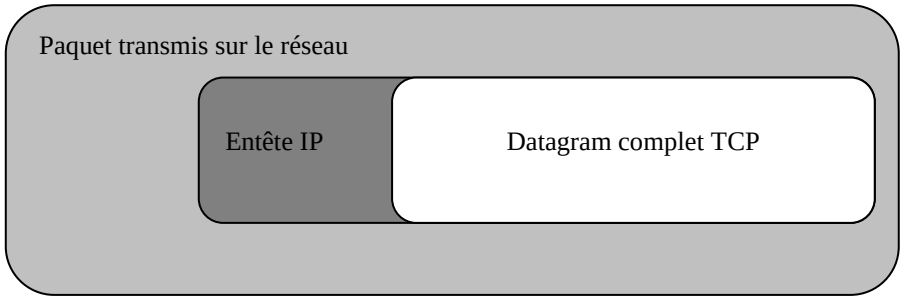
Si une erreur apparait pendant quelle est la couche driver de l'interface réseau, elle est transmise au programme utilisateur.

b. Transmission Control Protocol : TCP

Le Protocole de Contrôle de Transmission est situé au dessus de la couche IP. C'est un standard de protocole de niveau communication qui permet à un programme utilisateur situé sur une machine de transmettre un « flux de données » vers une autre machine. TCP fiabilise, contrôle le flux, ordonne, la transmission de données dans les deux sens entre deux processus connectés. Il est possible de fermer un sens de transfert de flux de données dans une connexion TCP tout en laissant l'autre sens actif (mode simplex).

Les logiciels qui exécutent TCP résident au cœur de l' « opérating system » (OS) et utilisent IP pour transmettre les informations au travers de l'Internet sous-jacent. TCP suppose que le service de datagram sous-jacent n'est pas forcément fiable. C'est pourquoi il ajoute un checksum sut toutes les données pour assurer la fiabilité de la transmission. TCP utilise l'adressage de niveau IP des machines et ajoute sa propre adresse de port de communication. A ce point une connexion TCP est identifiée par la combinaison d'une adresse IP et d'un numéro de port TCP.

Les paquets TCP sont encapsulés dans les datagram IP



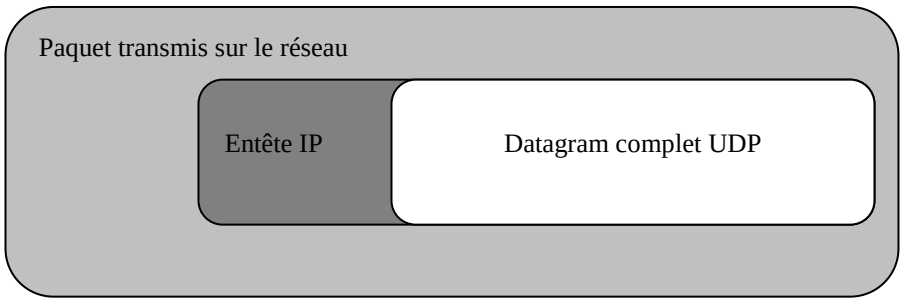
c. User Datagram Protocol : UDP

Le Protocole de Contrôle de Transmission est situé aussi au dessus de la couche IP. UDP est simple protocole non fiabilisé. C'est un standard de protocole de niveau communication qui permet à un programme utilisateur situé sur une machine de transmettre un « datagram » vers une autre machine en utilisant IP. La différence avec TCP est que UDP inclue un « protocol port number », qui permet à l'émetteur de distinguer les multiples applications actives sur la machine distante (remote host).

Comme TCP, UDP utilise un numéro de port en complément de l'adresse IP pour identifier le point final de communication.

Les datagram UDP ne sont pas fiabilisés. Ils peuvent être perdus ou abandonnés de plusieurs façons, incluant un échec du mécanisme de communication sous-jacent. UDP construit aussi un checksum sur la portion de données du paquet transmis. Si le checksum du paquet reçu est incorrect, il est éliminé et aucune information n'est envoyée à l'application. La file d'attente a une capacité limitée, si un datagram arrive une fois que la capacité de la file d'attente soit atteinte, il est détruit.

Les paquets UDP sont encapsulés dans les datagram IP



4. Adresse Internet

Indépendemment du protocole utilisé pour transmettre les données sur un réseau, on affecte à chaque machine une adresse Internet, **« IP adress »,** codée sur 32 bits.

Une adresse IP contient :

- Un champ réseau (netid).
- Un champ machine (hostid).

Les adresses IP sont représentées par des nombre décimaux, où chaque nombre décimal est codé sur un octet (8 bits) et correspond à une portion de l'adresse totale. On appel cela une notation à point (dot notation). Une adresse IP peut avoir l'une des formes suivantes :

- a.b.c.d
- a.b.c
- a.b
- c

Quand les quatres parties sont spécifiées, chaque élément représente un octet de données. De gauche à droite les quatres octets forment l'adresse IP totale.

Quand une adresse est constituée de trois parties, le dernier nombre est un mot de 16 bits et les deux octets de gauche représentent l'adresse du réseau.

Quand une adresse est constituée de deux parties, le dernier nombre est un mot de 24 bits et l'octet de gauche représentent l'adresse du réseau.

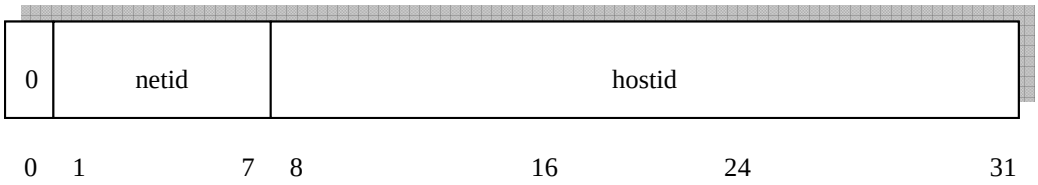
Quand une adresse est constituée d'une seule partie, la vlaeur correspondante est une adresse réseau de 32 bits sans décomposition en octet.

Chaque entier doit être choisi de manière précise pour permettre au routage d'être efficace. Classiquement une adresse IP indentifie un réseau sur lequel une machine est connectée

Les réseaux sont séparés en plusieurs classes. Il existe trois classes primaires :

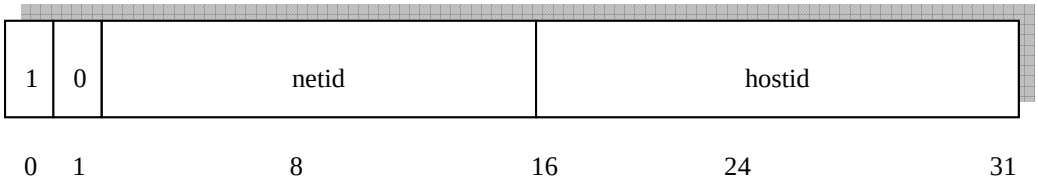
a. Classe A

La classe A est utilisée pour les réseaux qui ont plus de 65536 machines. Sept bits sont utilisés pour identifier le réseau et 24 bits pour chaque machine. Par exemple, la valeur hexadécimal 32 bits 0x0102ff04 est égale à 1.2.225.4. Cela correspond à un numéro de réseau « netid » de 1 et un numéro de machine de 0x02ff04. En classe A l'adresse se présente sous la forme **« net.host »**



b. Classe B

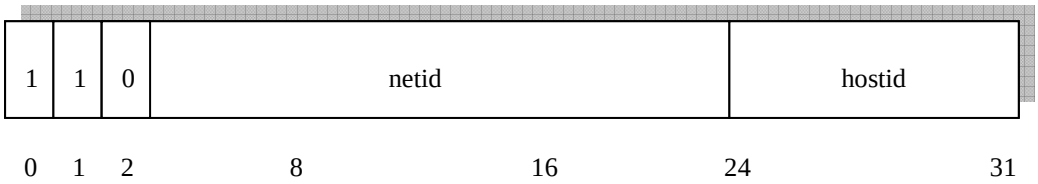
La classe B est utilisée pour les réseaux qui ont entre 256 et 65536 machines. Quatorze bits sont utilisés pour identifier le réseau et 16 bits pour chaque machine. Par exemple, en classe B une adresse est 128.100.0.5.



En classe A l'adresse se présente sous la forme **«128. net.host »**

c. Classe C

La classe B est utilisée pour les réseaux qui moins de 256 machines. Vingt et un bits sont utilisés pour identifier le réseau et 8 bits pour chaque machine. Par exemple, en classe C une adresse est 192.100.2.10.



Conclusion : les 3 bits de poids fort indiquent le type de classe utilisée. L'adresse IP a été construite pour extraire le numéro du réseau « netid » et le numéro de la machine « hostid ».

Cette adresse est unique pour toute machine connectée au réseau mondial. Le nombre d'adresses étant limité, les serveurs de réseaux locaux gèrent un ensemble d'adresse qui ne sont pas visible de l'extérieur. Lors d'une connexion une adresse temporaire est alors associée à la machine.
Les passerelles gateway gèrent le chemin sur la base du « netid » et dépendent de l'éfficacité de l'analyse des adresses.

5. Port de communication

En complément avec les adresses IP on utilise de numéros de port. Un numéro de port permet faire la distinction entre toutes les applications qui utilisent le réseau et tournent en même temps sur la même machine. Les numéros de port sont gérés par l'utilisateur quand les « socket », canaux de communication internet, sont ouverts et fermés.
Les numéros de port sont des valeurs codées sur 16 bits, la valeur maximale est donc de 65535.

Les numéros de port inférieurs à 1024 sont réservés. L'utilisateur doit choisir un numéro supérieur pour chaque application active.