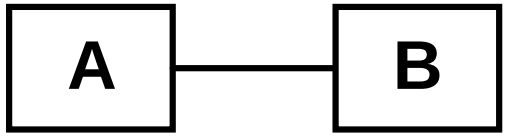
Liaisons asynchrones

- **Schéma fonctionnel**
- **Codage de l'information, code ASCII**
- **Transmission parallèle et série, UART**
- **Modes de transmission *full duplex* et *half duplex***
- **Modems normalisés UIT**
- **Terminaux, logiciels d'émulation**
- **Echo, contrôle d'erreur, contrôle de flux**
- **Labo 1 : liaison PC - PC**
- **Analyseur de trafic**
- **Labo 2 : PC – serveur de terminaux – serveur Alpha**
- **Système embarqué**

Schéma fonctionnel (1)

Une liaison de données relie 2 équipements terminaux désirant échanger des informations sous la forme de messages binaires; distinguons 2 cas :

Distance “faible”



Null modem

Distance “élevée”

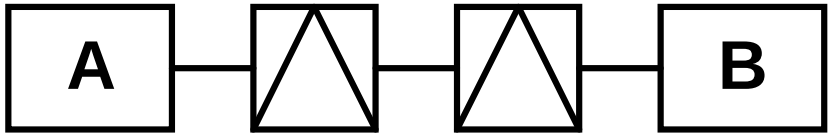
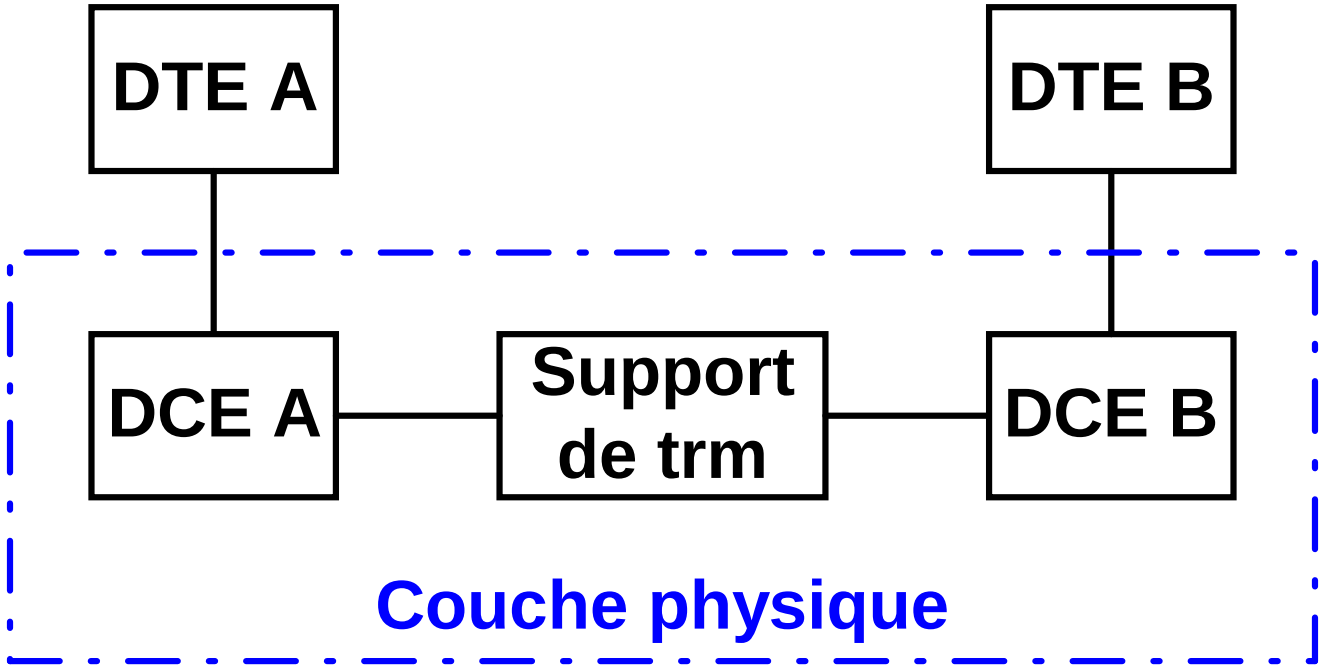


Schéma fonctionnel (2)

- **L'industrie des télécommunications a évidemment normalisé le second cas :**

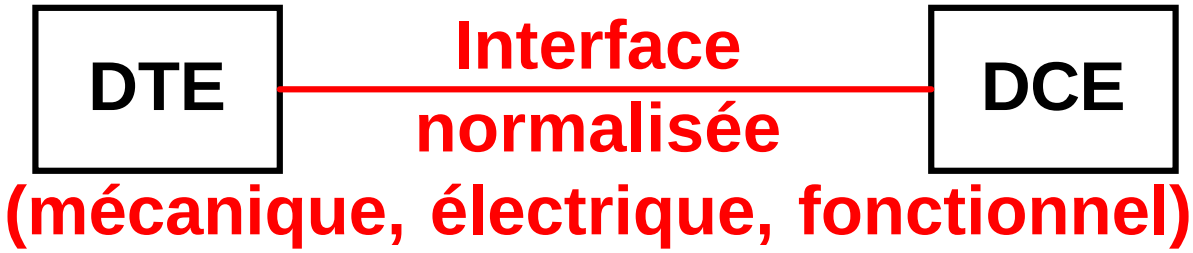


- **DTE** *Data Terminal Equipment*
- **DCE** *Data Circuit terminating Equipment*

Schéma fonctionnel (3)

- **La couche physique comprend, outre le support de transmission, des équipements assurant la conversion et l'adaptation des signaux dans chaque sens de transmission**

- **Ces DCEs, couramment appelés modems (contraction de modulateur-démodulateur), sont reliés aux DTEs par l'intermédiaire d'une interface normalisée**



Codage de l'information (1)

- **Les données numériques sont produites par une source discrète (par exemple un clavier), qui produit de l'information à partir d'un nombre fini N de caractères (lettres, chiffres, signes,...)**

- **Chacun de ces caractères est désigné par une expression logique codée, correspondant généralement à une seule combinaison binaire**

- **Le tableau de correspondance bilatérale entre les N caractères et les combinaisons binaires constitue le code**

- **Il permet les opérations de codage et décodage**

Codage de l'information (2)

Les différents codes se distinguent par :

- **leur richesse : majuscules, minuscules, signes spéciaux**
- **leur efficacité au sens de l'économie de configurations binaires qu'ils requièrent pour représenter un caractère**

Les critères de décision permettant le choix d'un code dépendent de l'application :

- **compatibilité avec d'autres équipements (norme)**
- **nature des informations à échanger**

Code ASCII (1) *à distribuer*

- ***American Standard Code for Information Interchange***

- **Code 7 bits → nombre de combinaisons = ?**

- **Tableau ASCII**

Dec	Hex	Code	CTRL
49	31	1	
50	32	2	
13	0D	CR	<CTRL><M>
17	11	DC1	<CTRL><Q> = Xon
19	13	DC3	<CTRL><S> = Xoff

Code ASCII (2)

- **Sur un clavier (environ 60 touches), les 128 combinaisons sont générées de la façon suivante :**

Colonne	00-1F	20-3F	40-5F	60-7F
	caractères de commande	chiffres signes	lettres maj.	lettres min.
			← SHIFT ←	
	←	CONTROL ←		←

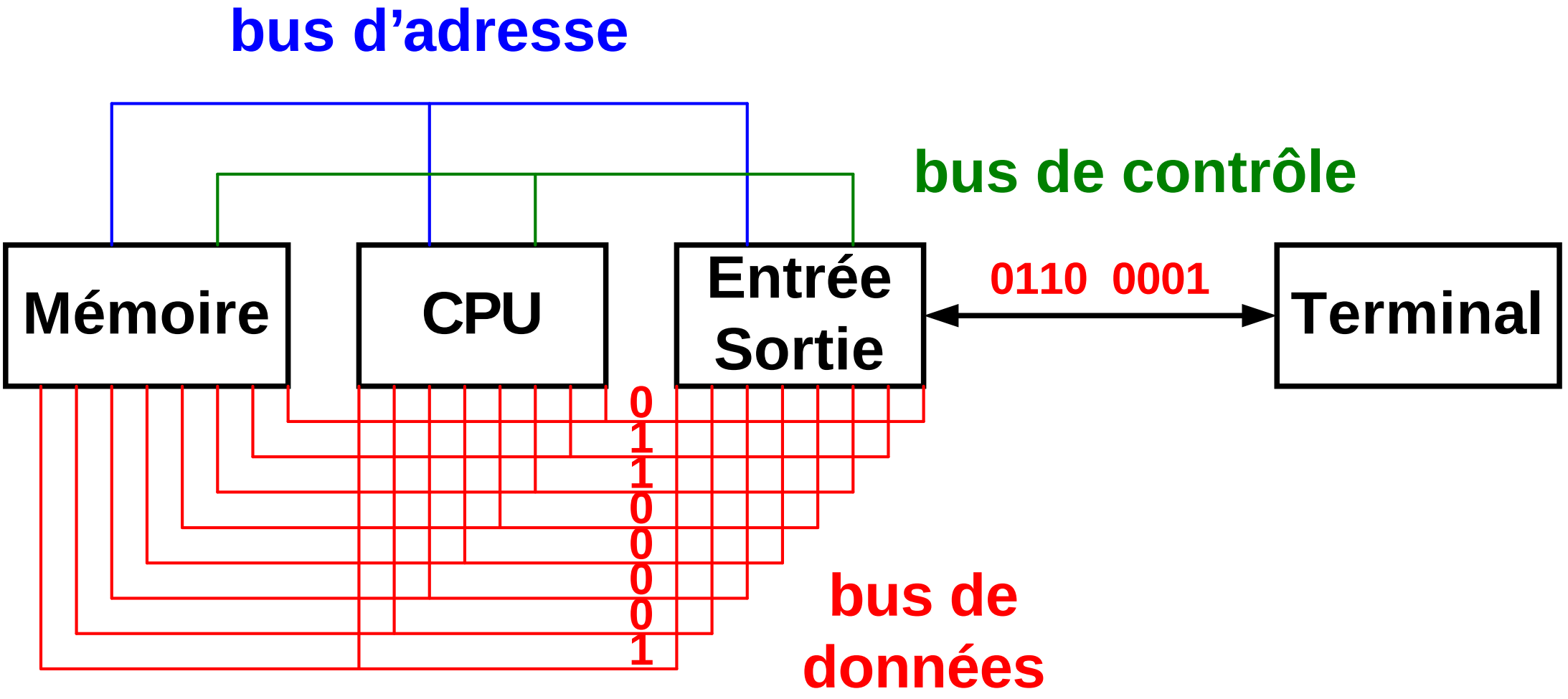
- **Les 2 premières colonnes (0-1) sont affectées à des caractères de commande *(control character)* pas imprimables**

Code ASCII (3)

NUL	*Null character*	DLE	*Data Link Escape*
SOH	*Start of Header*	DC1	*Dev. Cont. 1* (Xon)
STX	*Start of Text*	DC2	*Device Control 2*
ETX	*End of Text*	DC3	*Dev. Cont. 3* (Xoff)
EOT	*End of Transmission*	DC4	*Device Control 4*
ENQ	*Enquire*	NAK	*Neg. Acknowledgem.*
ACK	*Acknowledge*	SYN	*Synchronize*
BEL	*Bell*	ETB	*End of Text Block*
BS	*Bachspace*	CAN	*Cancel*
TAB	*Tab*	EM	*End of Media*
LF	*Line Feed*	SUB	*Substitute*
VT	*Vertical Tab*	ESC	*Escape*
FF	*Form Feed*	FS	*Form Separator*
CR	*Carriage Return*	GS	*Group Separator*
SO	*Shift Out*	RS	*Record Separator*
SI	*Shift In*	US	*Unit Separator*

Transmission parallèle et série (1)

- **Les ordinateurs traitent l'information sous forme de mot**
- **Les échanges à l'intérieur de la machine sont généralement de nature parallèle**



Transmission parallèle et série (2)

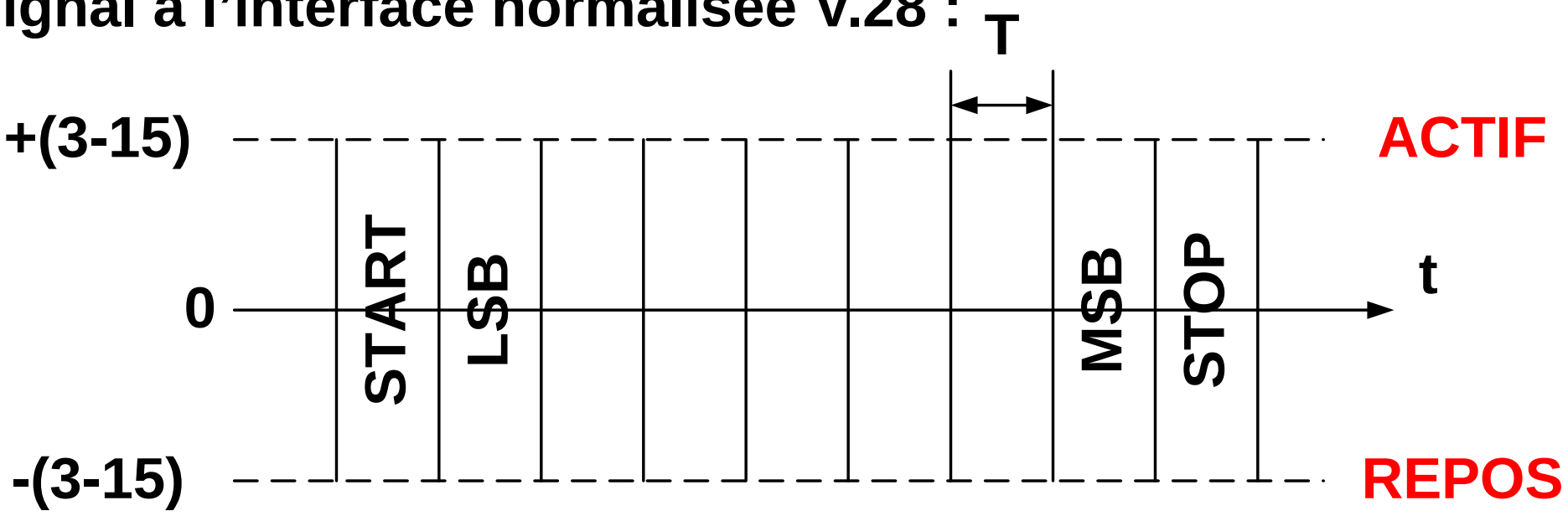
- **En effet, le bus de données comporte autant de fils qu'il y a de bits par mot**
- **Les bits sont transmis en parallèle, mais les mots le sont en série; l'un après l'autre**
- **Ce mode de liaison convient parfaitement lorsque les distances sont courtes (bus)**
- **Pour des distances plus grandes, on préfère passer par une liaison série; où le mot parallèle est converti en mot série pour être transmis**
- **Les bits sont transmis en série selon un format approprié (généralement le LSB en premier)**
- **Le débit binaire D est le nombre de chiffres binaires émis par seconde**
- **C'est une caractéristique des équipements DTE (terminal, ordinateur, imprimante,...) qui s'exprime en bit/s**

Transmission parallèle et série (3)

- **Dans une transmission asynchrone, émetteur et récepteur choisissent en commun un débit binaire D et un nombre n de bits par caractère**

- **Choix : D = 9600 bit/s et n = 7 → Durée du bit T = 104 «s**

- **Signal à l'interface normalisée V.28 :**

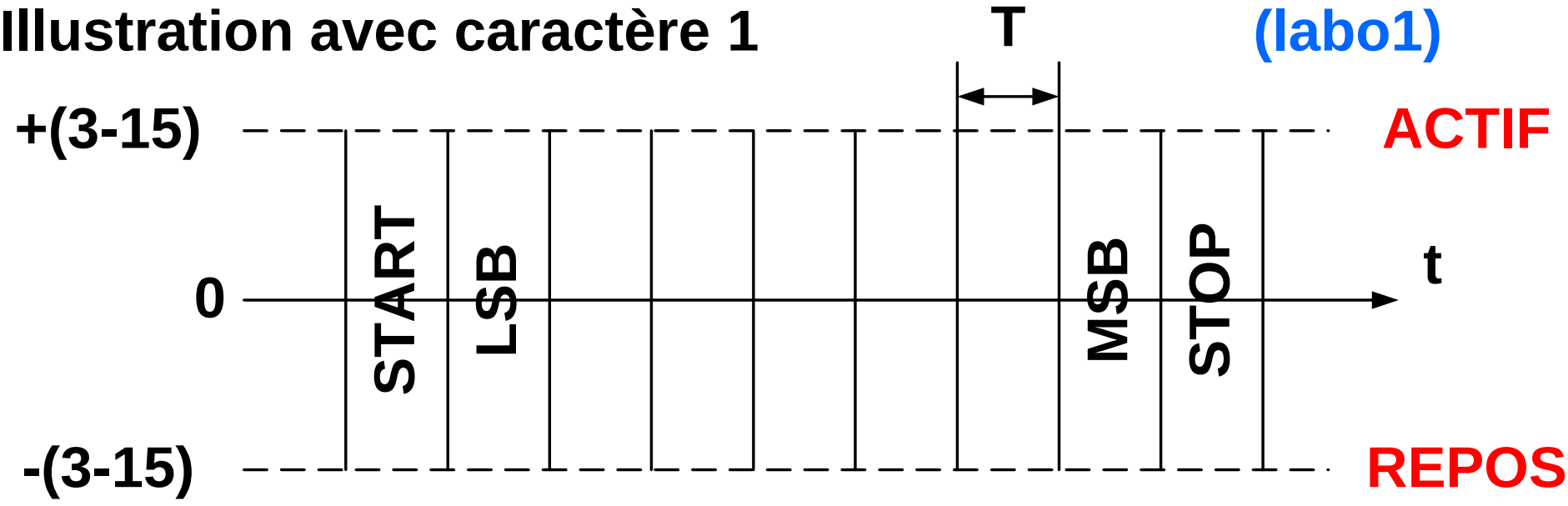


Transmission parallèle et série (4)

- **En l'absence de donnée à émettre, le signal est à l'état de repos correspondant à état logique 1**

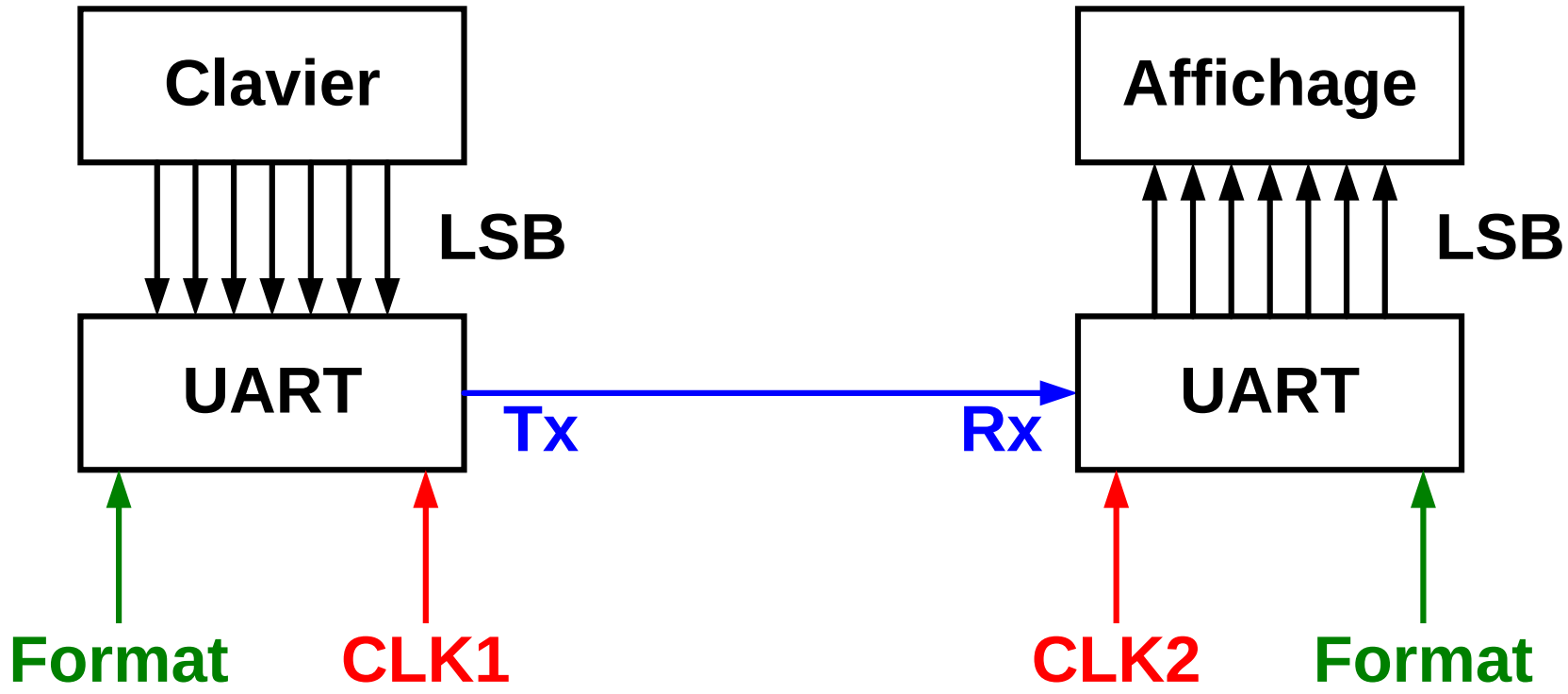
- **Format du caractère :** **1 bit de START état actif "0"**
  7 bits de DONNEE ASCII
  1 bit de STOP état repos "1"

- **Illustration avec caractère 1** **(labo1)**



UART

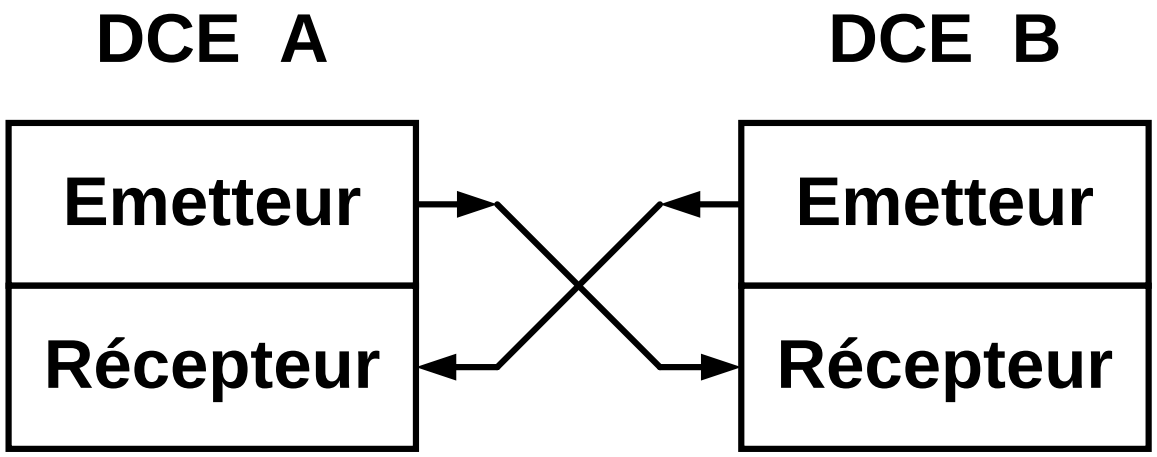
- **Conversion parallèle - série**



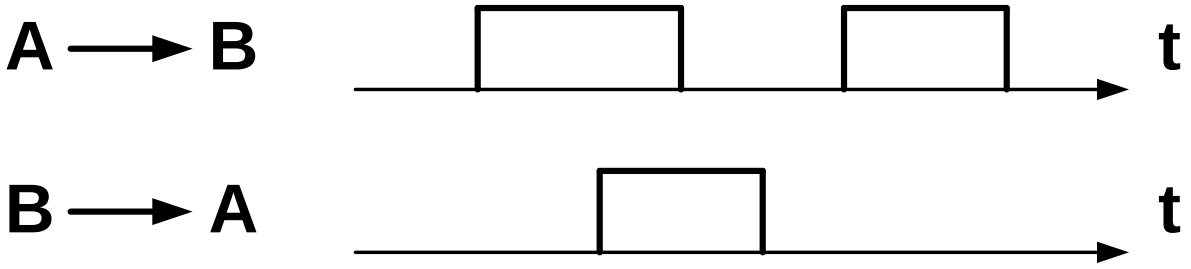
- ***Universal Asynchronous Receiver Transmitter (UART)***

Duplex intégral (full duplex)

- **Dans le mode duplex intégral, chaque sens dispose d'un canal de transmission :**

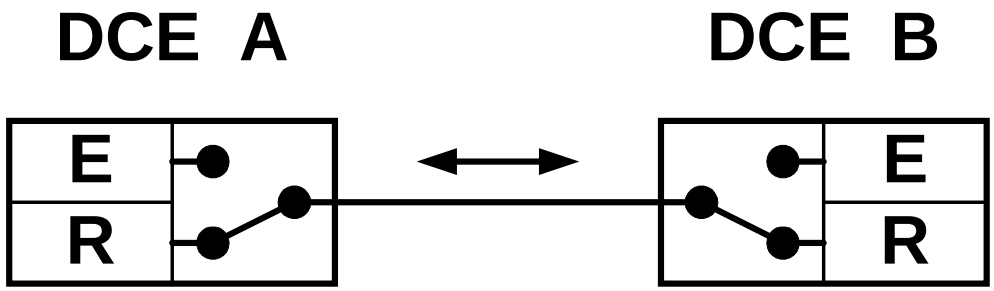


- **Les 2 extrémités A et B peuvent donc émettre et recevoir simultanément**

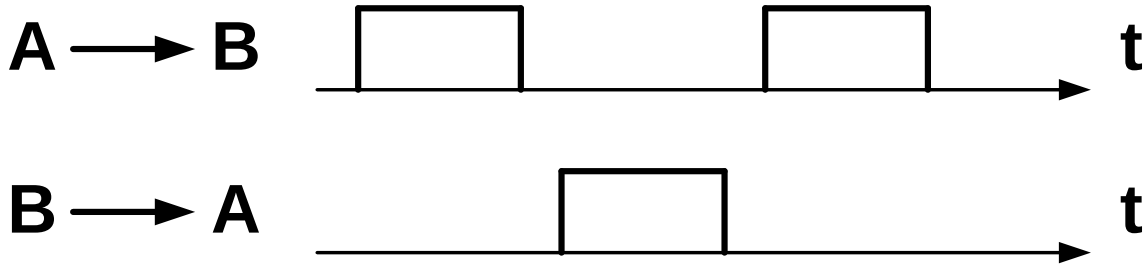


Semi-duplex (half duplex)

- **En semi-duplex , un seul canal est partagé entre les 2 sens de transmission :**



- **Ce mode requiert une discipline de dialogue (protocole)**
- **Chaque poste est normalement en état de réception; état qu'il quitte pour émettre (principe du *talky-walky*)**



Modems normalisés UIT

- **Ce tableau précise les caractéristiques essentielles de la couche physique utilisant les modems normalisés par l'UIT (Union Internationale des Télécommunications)**

Année	Norme	D [bit/s]	R[baud]	Mode
1964	V.21	300	300	Full
1964	V.23	1200	1200	Half
1980	V.22	1200	600	Full
1984	V.32	9600	2400	Full
1991	V.32bis	14400	2400	Full
1994	V.34	28800	2400 - 3200	Full
1998	V.90	57600		Full

Historique des terminaux http://vt100.net/vt_history

- **Terminaux non-intelligents (sans système de fichiers)**

  Teletype (TTY) — **Equipement électromécanique style machine à écrire**

  VT 52, VT100, VT220, … — **Equipements électroniques clavier – écran (25x80 caract.) caract. semi-graphiques, …**

- **Terminaux intelligents (avec système de fichiers)**

  Logiciels d'émulation — **Hyperterminal (PC)**

  L'émulation du terminal VT100 consiste à présenter un PC comme un terminal reconnu du serveur

  ➔ **le PC émule le terminal VT100**

VT 100 escape sequences

- **Le serveur a besoin d'identifier le type de terminal (TTY, VT52, VT100, …)**

C ← S	ESC [c	What Are You
C → S	ESC [?1 ;0c	VT100

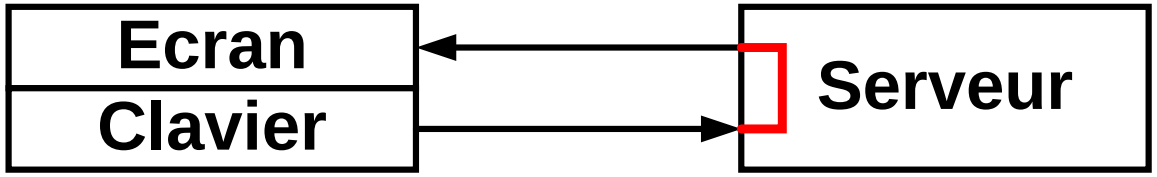
- **Quelques commandes *escape***

C ← S	ESC [Pn A	Cursor Up
C ← S	ESC [Pn B	Cursor Down
C ← S	ESC [Pn C	Cursor Right
C ← S	ESC [Pn D	Cursor Left
C ← S	ESC [Pl ;Pc H	Direct Cursor Addressing
C ← S	ESC c	Reset

Echo

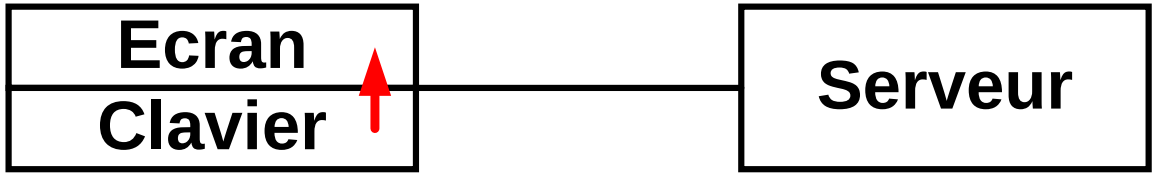
- **L'utilisateur du terminal doit voir les caractères envoyés (afin de ne pas travailler à l'aveugle)**

- **Echo distant *(remote echo)***



→ **Le support de transmission doit être duplex intégral**

- **Echo local *(local echo)***



→ **Le support de transmission peut être semi-duplex**

- **L'écho n'est pas toujours nécessaire !**

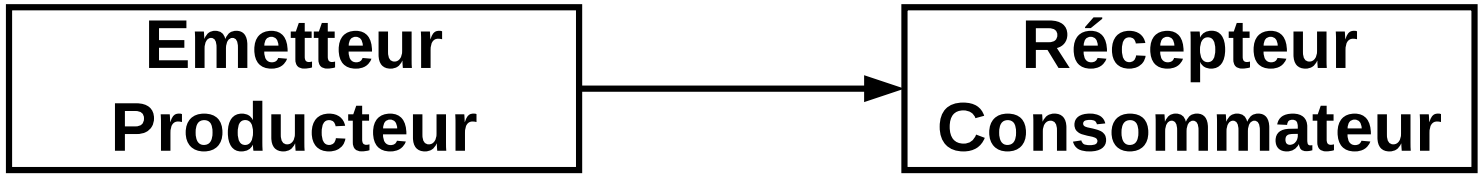
Contrôle d'erreur (error control)

- **Illustration avec bit de parité paire *(even parity)***

  Un mot est pair s'il contient un nombre pair de "1"

  Pour le caractère 1 : 1011 0001

- **L'émetteur génère ce bit de parité**



- **Le récepteur contrôle ce bit de parité afin de détecter d'éventuelle(s) erreur(s) de transmission**

Contrôle de flux (flow control) (1)

- **Problème à résoudre :**

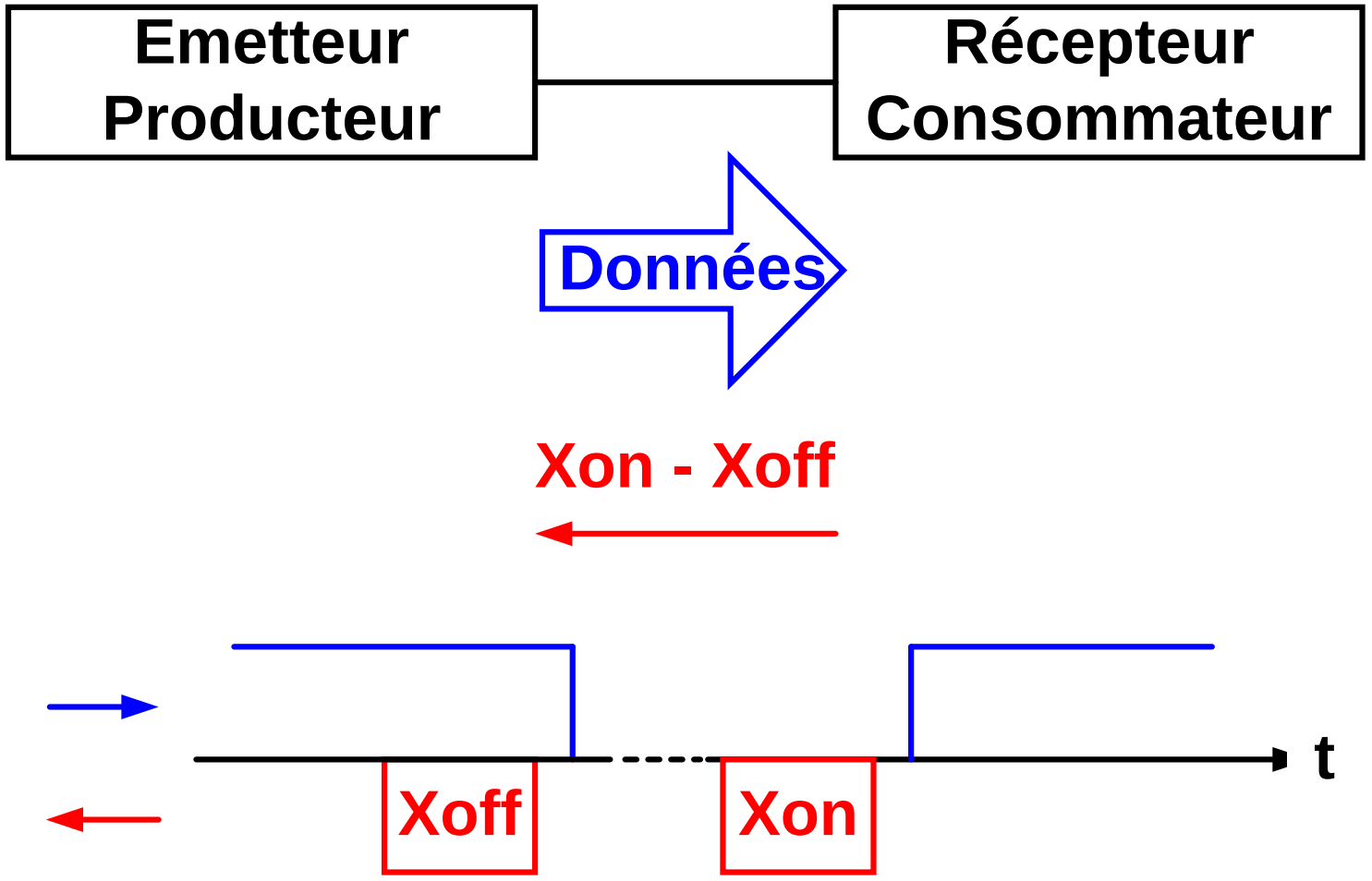
  L'émetteur produit 960 caractères/s (D=9600 bit/s, 1 bit start, 8 bit data, 1 bit stop); alors que le récepteur ne peut consommer que 150 caractères/s au maximum (cas d'une imprimante par exemple)

- **Ne pas mélanger débit utile *(throughput)* exprimé en caractère/s et débit binaire (bit/s) !!!**

Contrôle de flux (flow control) (2)

- **Solution :**

  Le récepteur, disposant d'une mémoire FIFO de 500 octets par exemple, va commander l'envoi des données :

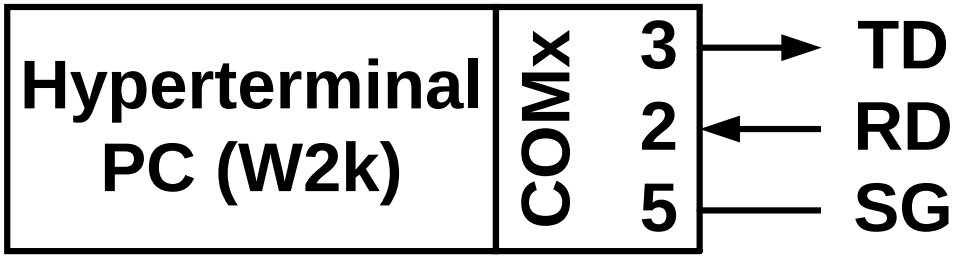


Contrôle de flux (flow control) (3)

- **La transmission doit être bidirectionnelle**
- **Elle doit même être …**
- **Le récepteur envoie Xoff (<CTRL><S>) si FIFO plein à 90 %**
- **Le récepteur envoie Xon (<CTRL><Q>) si FIFO plein à 10 %**

Labo 1 : liaison PC – PC (90 min)

- §1 **Câblage minimum 3 fils : émission, réception, potentiel de référence**

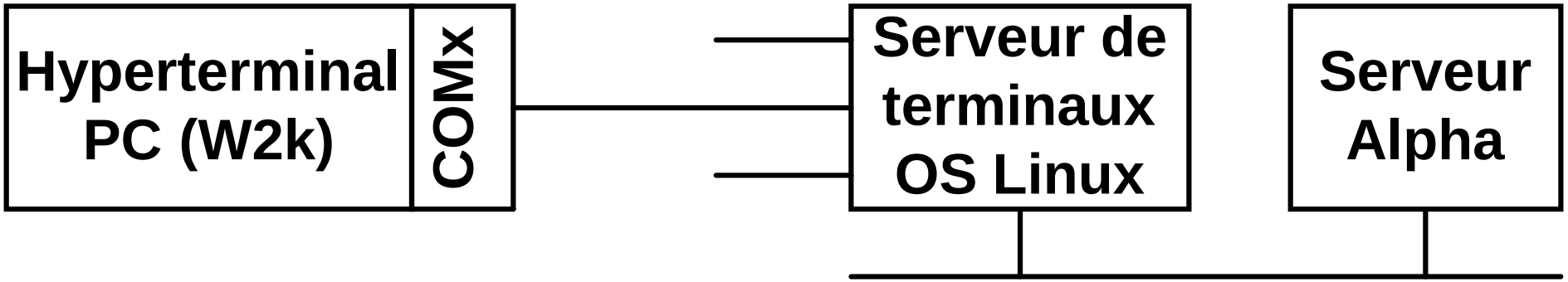


Quelle est la solution ?

- §2 **Configuration de HyperTerminal (émulation de terminal)**
- §3 **Mesure avec oscilloscope (slide 13)**
- §4 ***Connect directly to another computer → Host - Guest***
- §5 **Transfert de fichier → *Sharing***

Labo 2 : PC – serveur de terminaux – serveur Alpha

- **Le serveur de fichiers Alpha (OpenVMS) permet à l'utilisateur d'accéder à l'information (créer, modifier, afficher un fichier)**



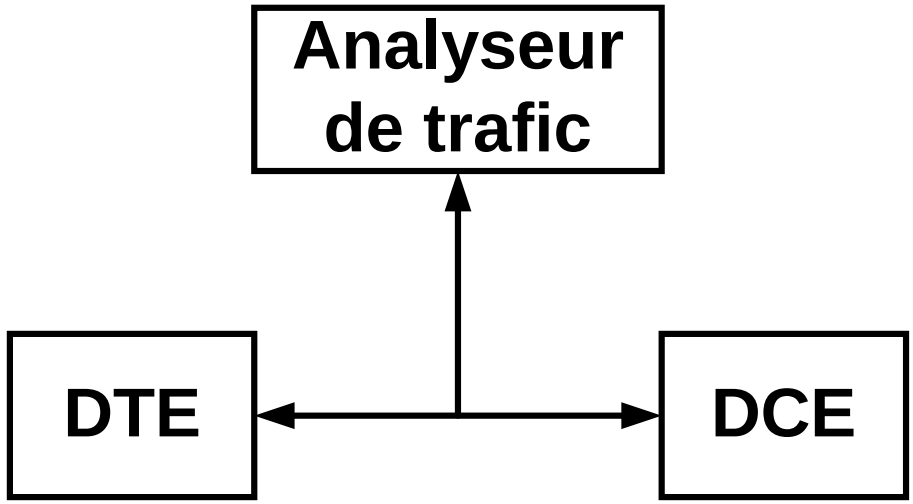
- **Modifier la configuration asynchrone : débit binaire, …**
- **Connexion au serveur telnet eig.unige.ch**
- **3 interlocuteurs !**

- **Config. étoile (liaisons point à point) et multipoint (LAN …)**

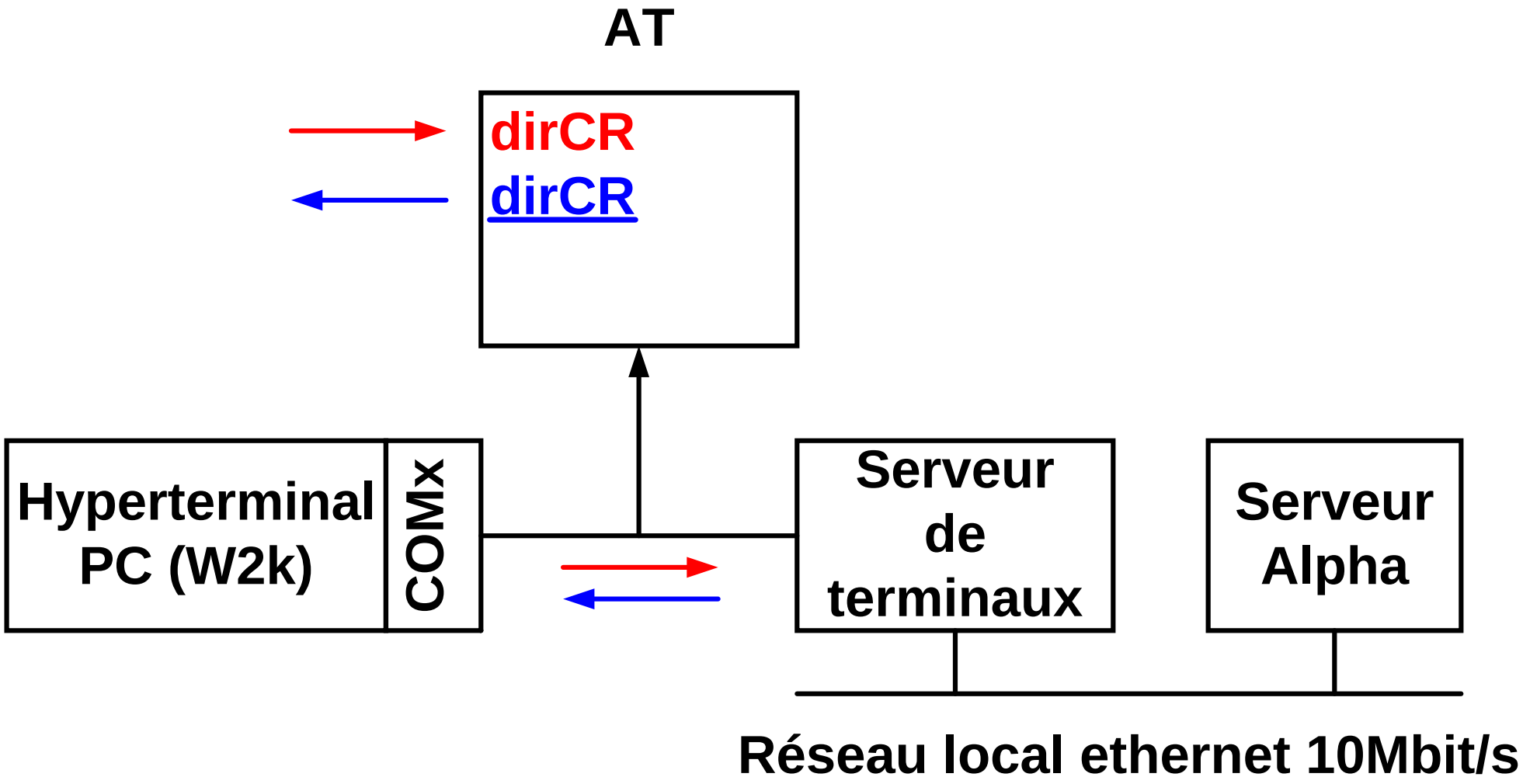
Analyseur de trafic (1)

- **Les caractères transmis sous forme sérielle nécessitent un appareil, appelé analyseur de trafic, capable d'afficher les caractères après avoir effectué l'échantillonnage et la conversion série-parallèle**

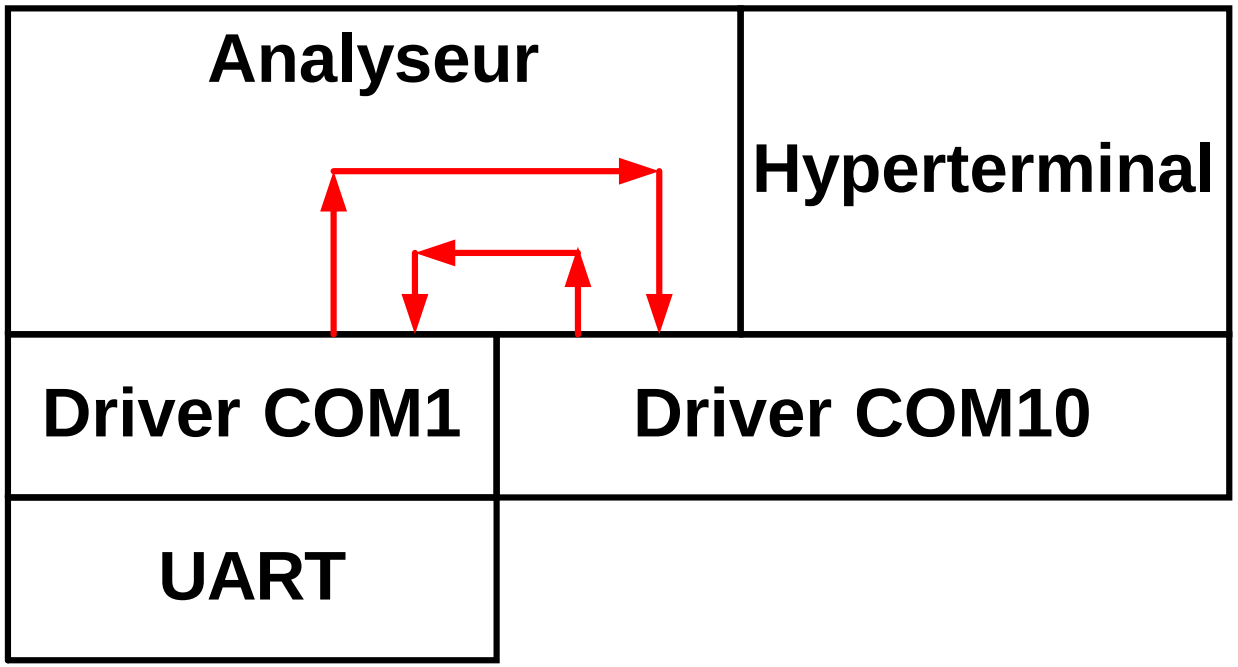
- **L'analyseur de trafic est un équipement passif (il ne génère aucun caractère) et se connecte entre DTE et DCE :**



Analyseur de trafic (2)



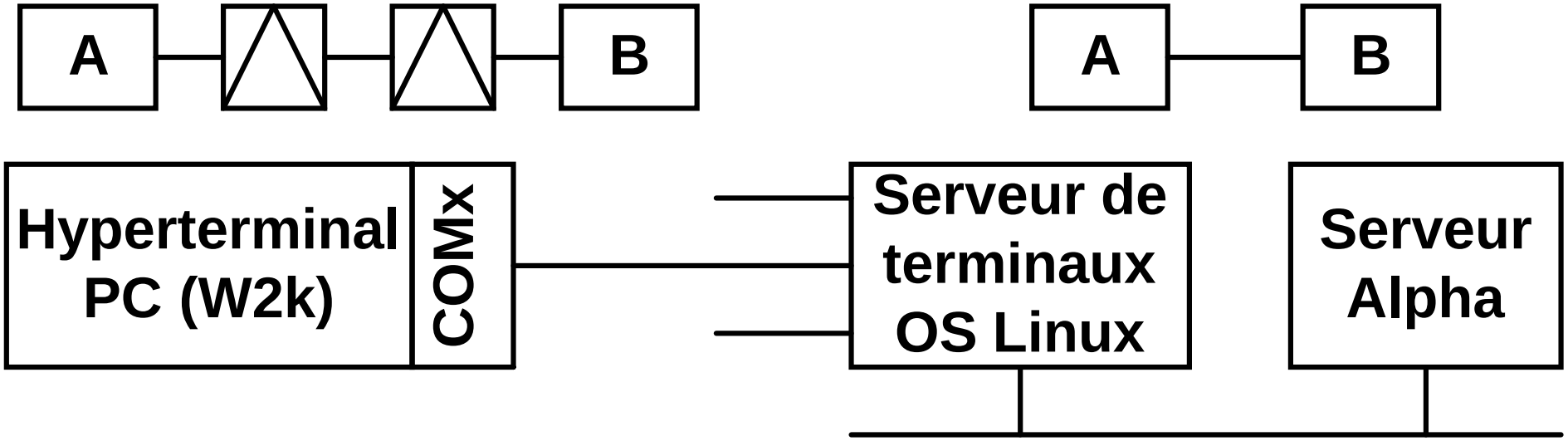
Redirection sur interface virtuelle



- 1 seul composant matériel UART
- Logiciel analyseur redirige le flux sur interface virtuelle
- Logiciel applicatif Hyperterminal

Transparent - virtual

- ***Something is transparent when it is physically here but seems not to be***
- **Exemple ?**



- ***Something is virtual when it is not physically but seems to be***
- **Exemple ?**
- **Interface COM 10 (slide précédent)**

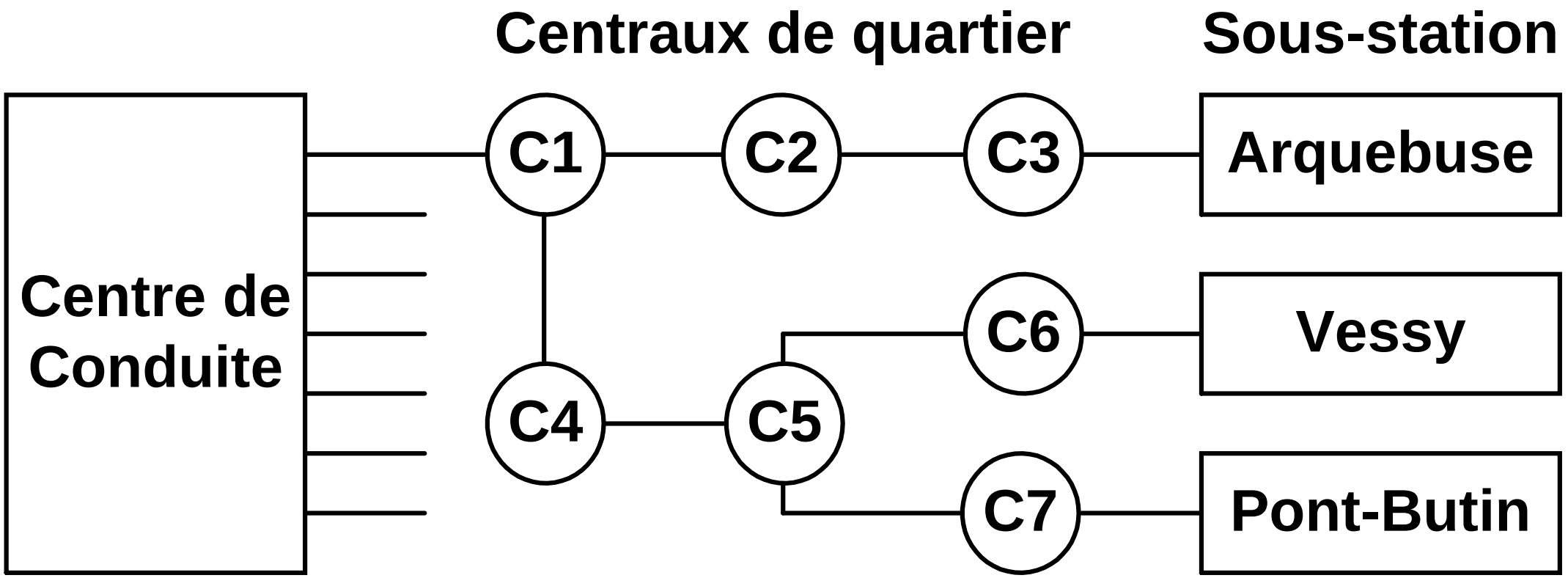
Système embarqué

- CPU 22,1 MHz
- SRAM 128 – 512 k
- Flash 256 – 512 k
- 4 x UART
- 1 x 10BaseT (ethernet)
- 26 I/O
- 5V – 135 mA

- Librairies UART, TCP/IP, …

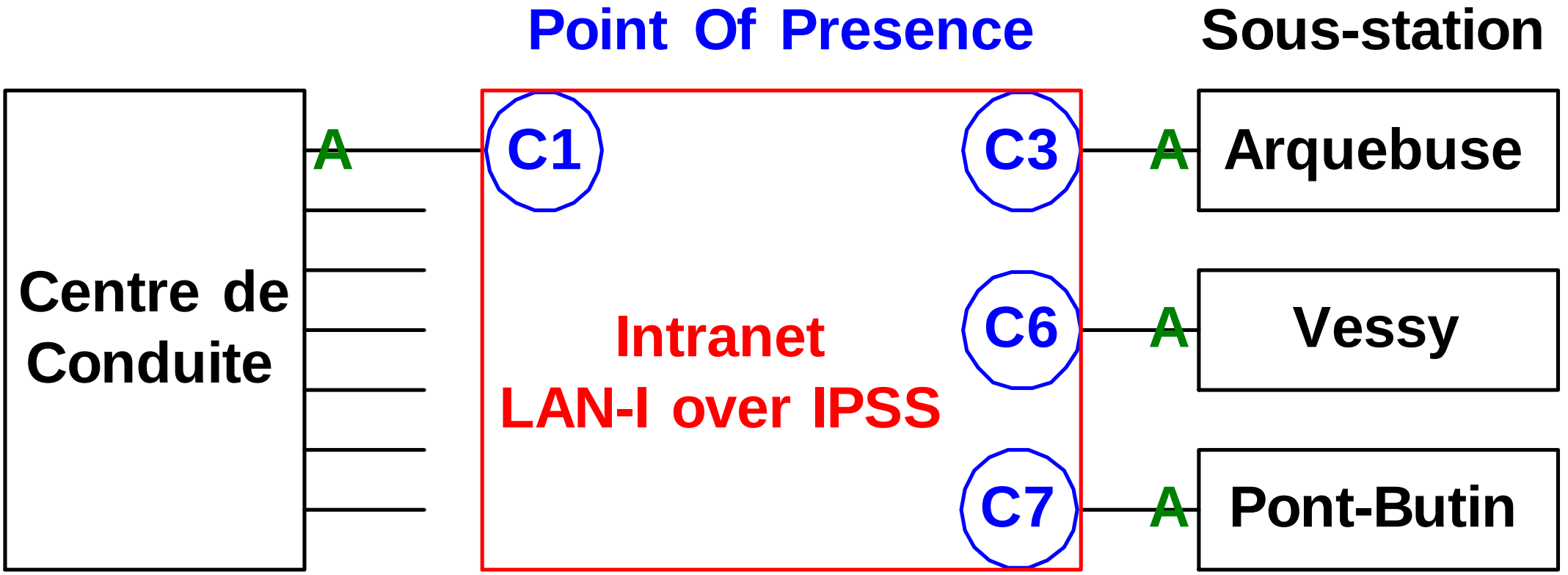
- Autres composants Motorola (68HC11, …), Mitshubishi, …

Migration du réseau SIG-Eau vers …



- **Multipoint cuivre + modems**
- **Transmission asynchrone à 600 bit/s**
- **Protocole de type *polling* (commande - réponse)**

Variante proposée



- Accès via liaison cuivre ADSL (64 – 2048 kbit/s)
- Accès fibre optique
- Aucun accès depuis internet
- Coût mensuel + frais installation

Adaptation © GL

Transmission parallèle et série (4)

- **En l'absence de donnée à émettre, le signal est à l'état de repos correspondant à état logique 1**

- **Format du caractère :** **1 bit de START état actif "0"**
  7 bits de DONNEE ASCII
  1 bit de STOP état repos "1"

- **Illustration avec caractère 1 : 31_H** **(labo1)**