

Ethernet – Applications Industrielles

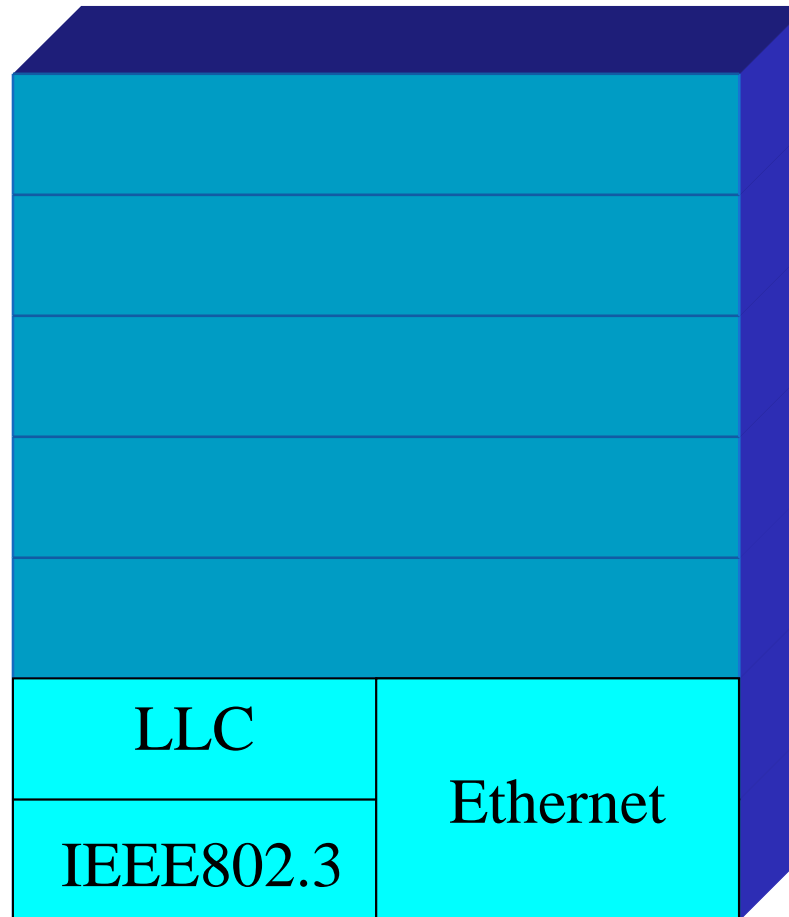


Ph. Meyne



Généralités

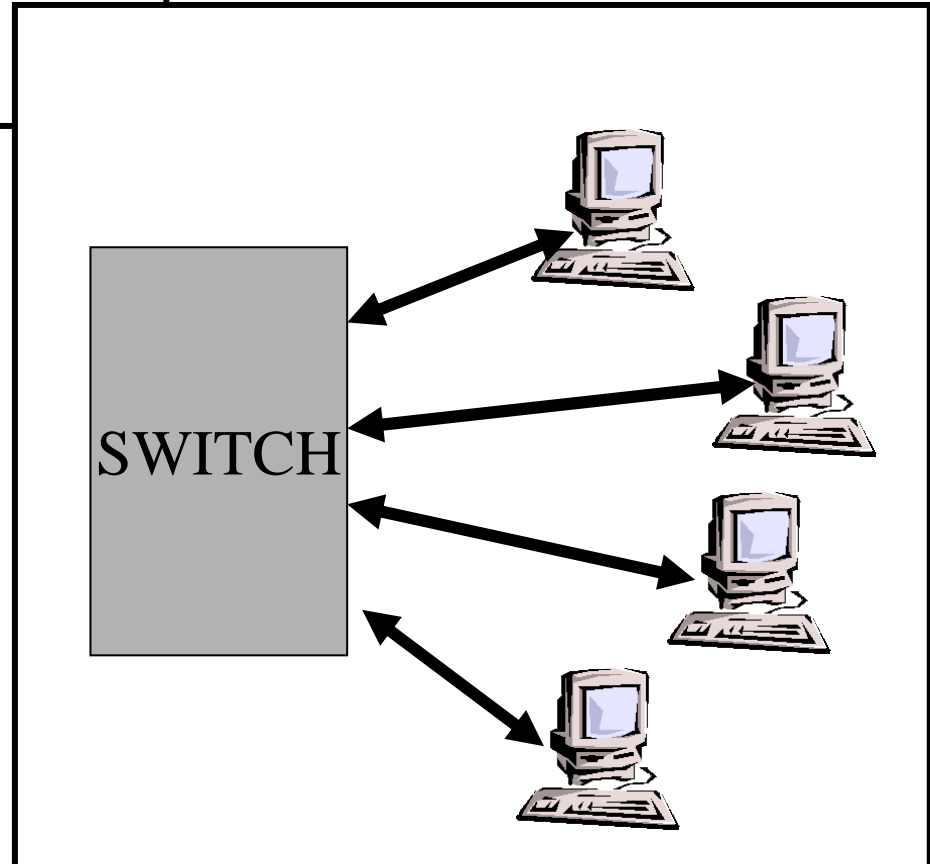
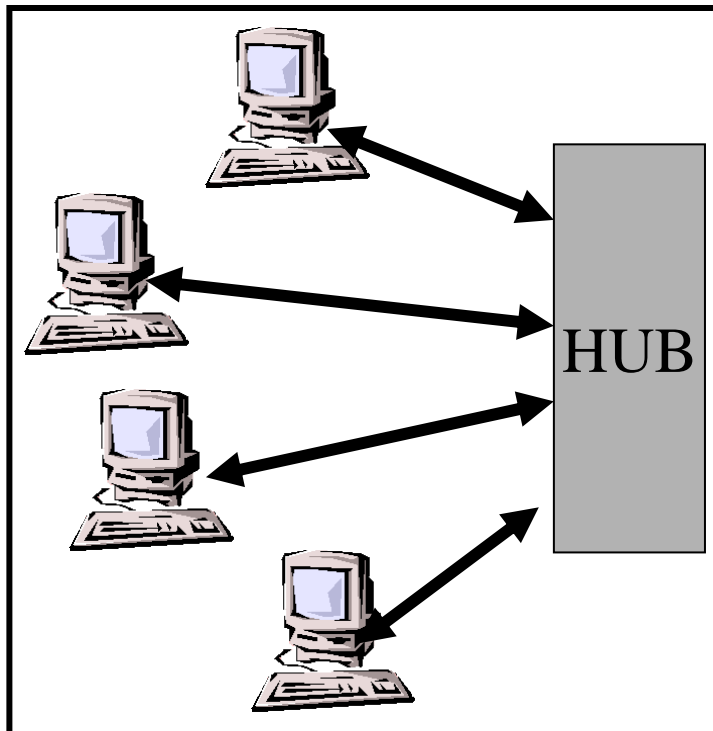
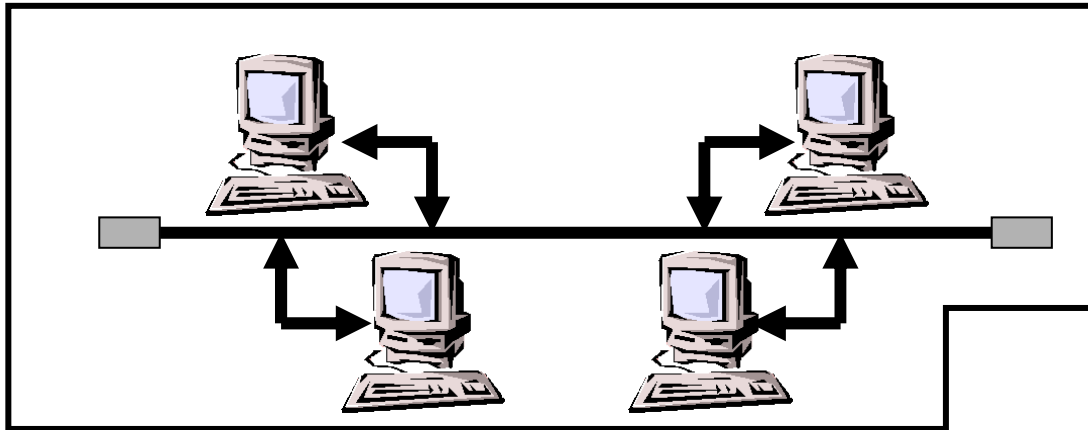
Modèle en couche





Généralités

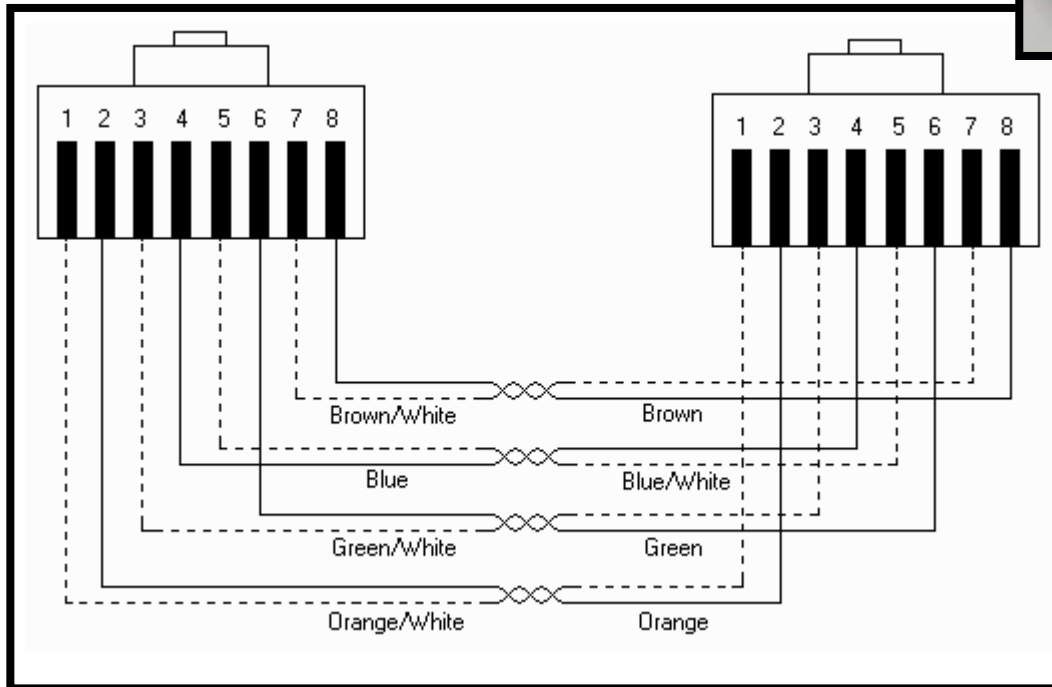
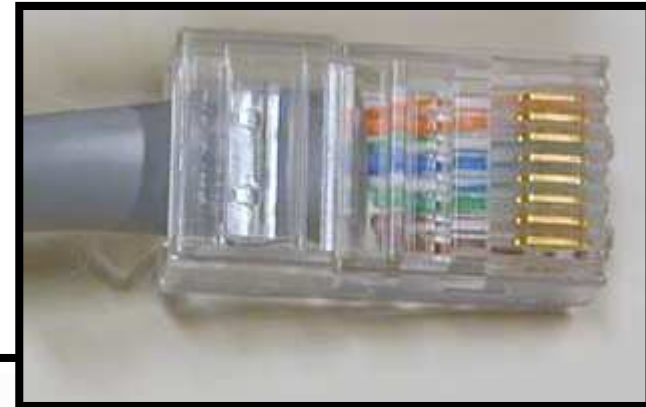
Topologie du réseau





Généralités

Connecteurs RJ 45





Généralités

Format de la trame : physique





Généralités

Format de la trame : physique

Symbole FDDI 5 bits	Nom	Signification
00100	H	Halt (arrête l'activité de l'anneau)
11111	I	Idle (bourrage et synchronisation)
11000	J	Délimiteur de trames
10001	K	Délimiteur de trames
00101	L	Délimiteur de trames (FDDI version 2)
00000	Q	Quiet (absence de signal)
00111	R	Zéro logique (Reset)
11001	S	Un logique (Set)
01101	T	Délimitation de trames

Symbole FDDI 5 bits	Binaire 4bits	Nom
11110	0000	0
01001	0001	1
10100	0010	2
10101	0011	3
01010	0100	4
01011	0101	5
01110	0110	6
01111	0111	7
10010	1000	8
10011	1001	9
10110	1010	A
10111	1011	B
11010	1100	C
11011	1101	D
11100	1110	E
11101	1111	F



Généralités

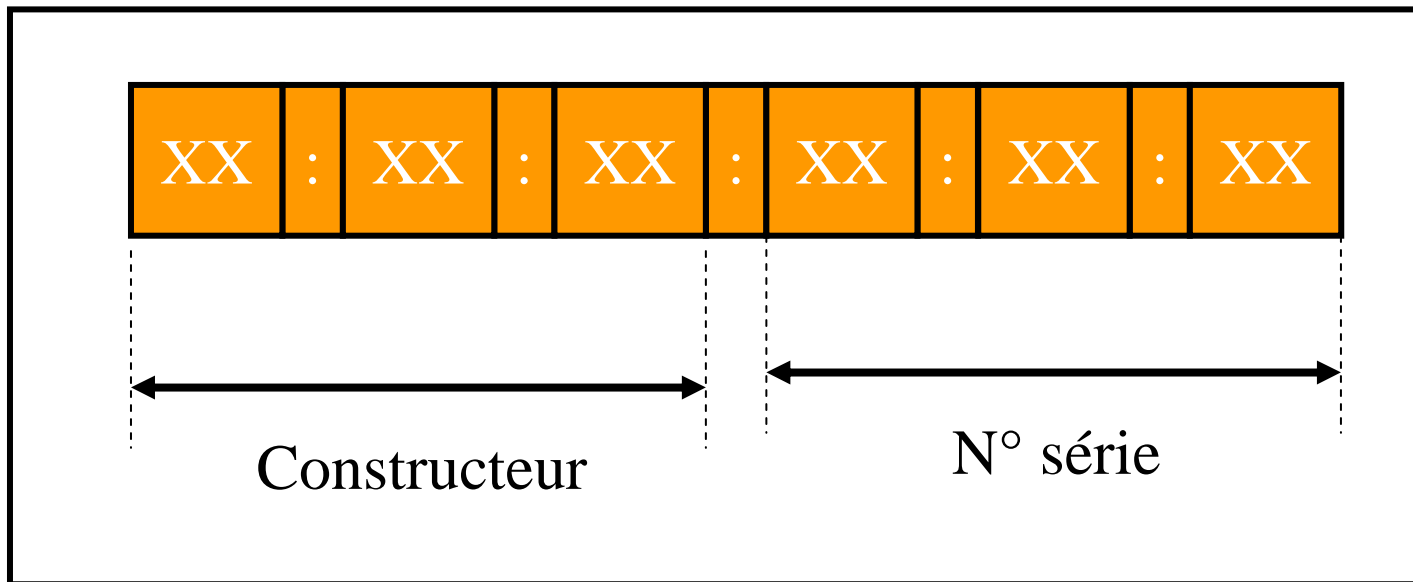
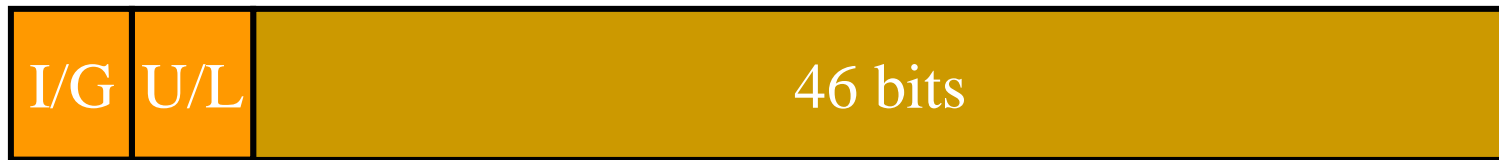
Format de la trame : MAC





Généralités

Format des adresses MAC





Généralités

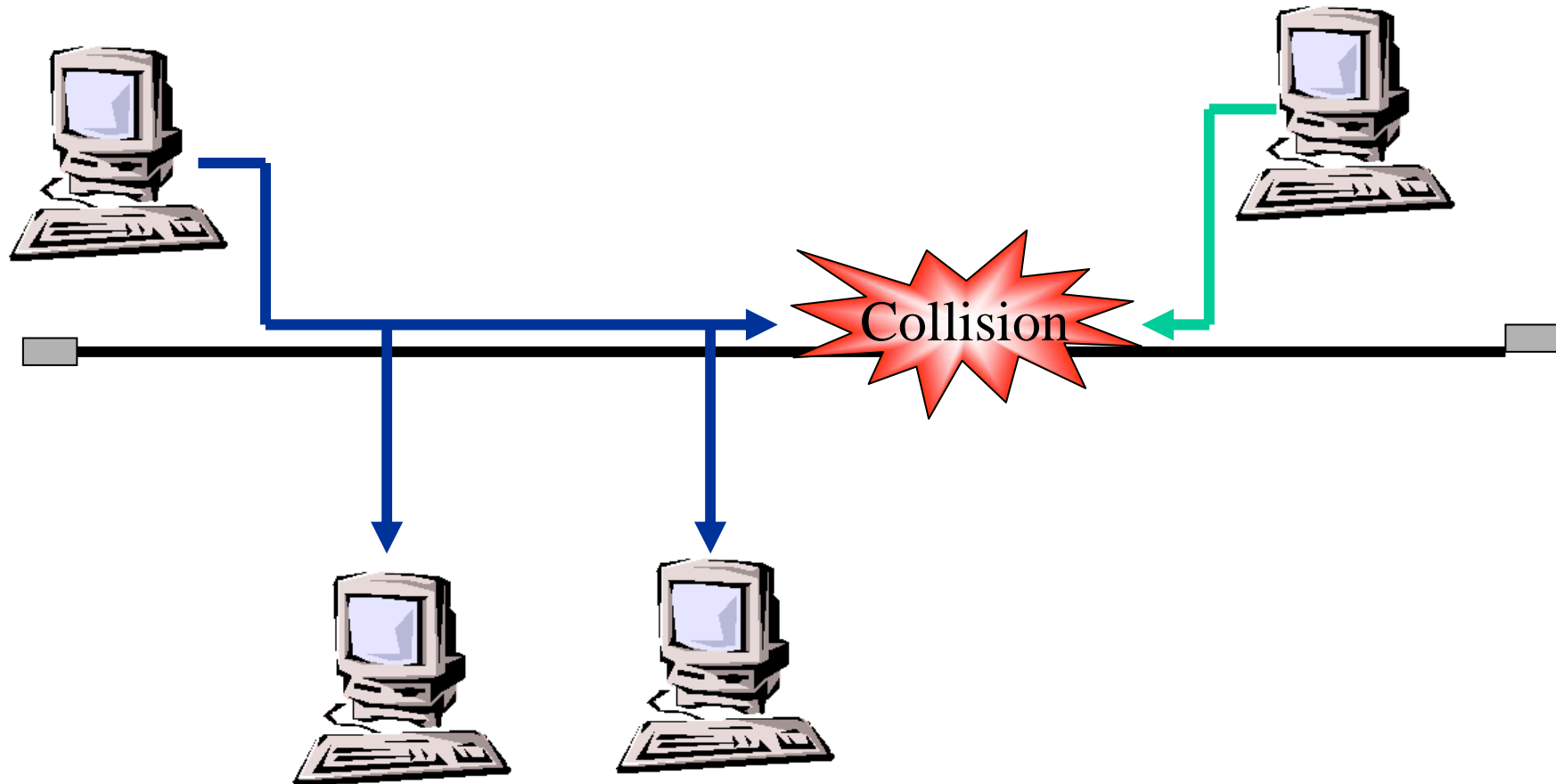
Exemple d'une trame

00 50 fc 9f 0d 20 00 20 ed b9 4e a0 08 00 45 00
00 3c 00 b0 00 00 80 01 b5 e6 c0 8e 01 10 c0 8e
01 fe 08 00 42 5c 02 00 09 00 61 62 63 64 65 66
67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
77 61 62 63 64 65 66 67 68 69



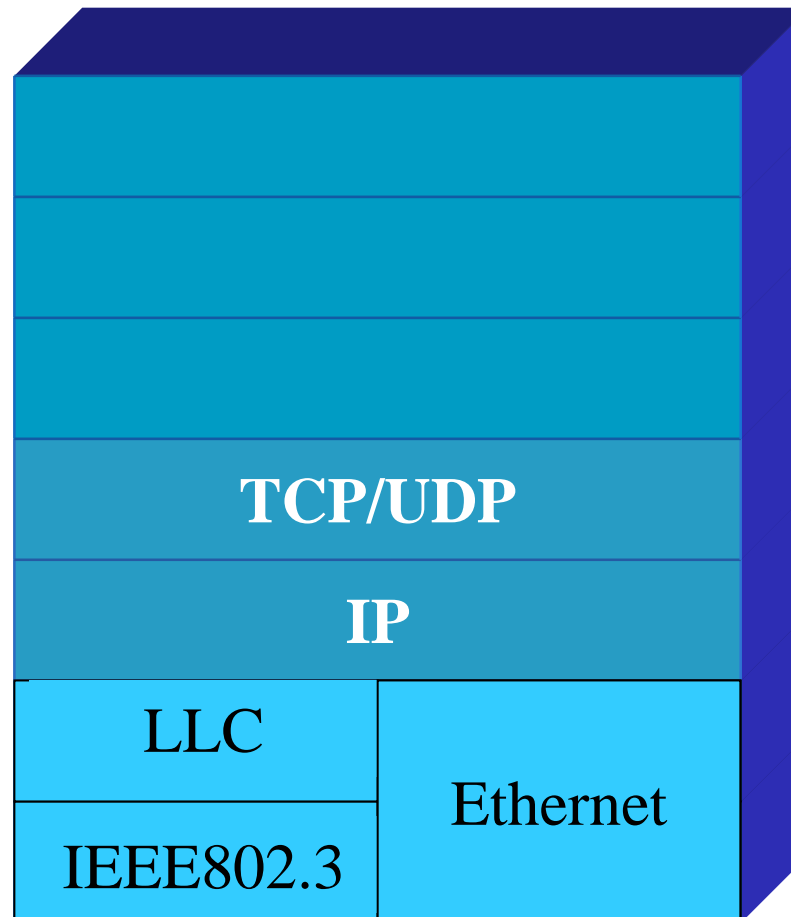
Généralités

Accès au réseau : CSMA/CD - Collision





Protocole de communication associé Présentation





Protocole de communication associé Datagramme IP

0	7	7	15	31
Version	Longueur	Type de services	Longueur totale du datagramme	
Identification			Drapeau	Place du fragment
Durée de vie	Protocole		Checksum	
Adresse de la source				
Adresse de la destination				
Option + bourrage				
Données				



Protocole de communication associé

Exemple de datagramme IP

00 50 fc 9f 0d 20 00 20 ed b9 4e a0 08 00 45 00
00 3c 00 b0 00 00 80 01 b5 e6 c0 8e 01 10 c0 8e
01 fe 08 00 42 5c 02 00 09 00 61 62 63 64 65 66
67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
77 61 62 63 64 65 66 67 68 69

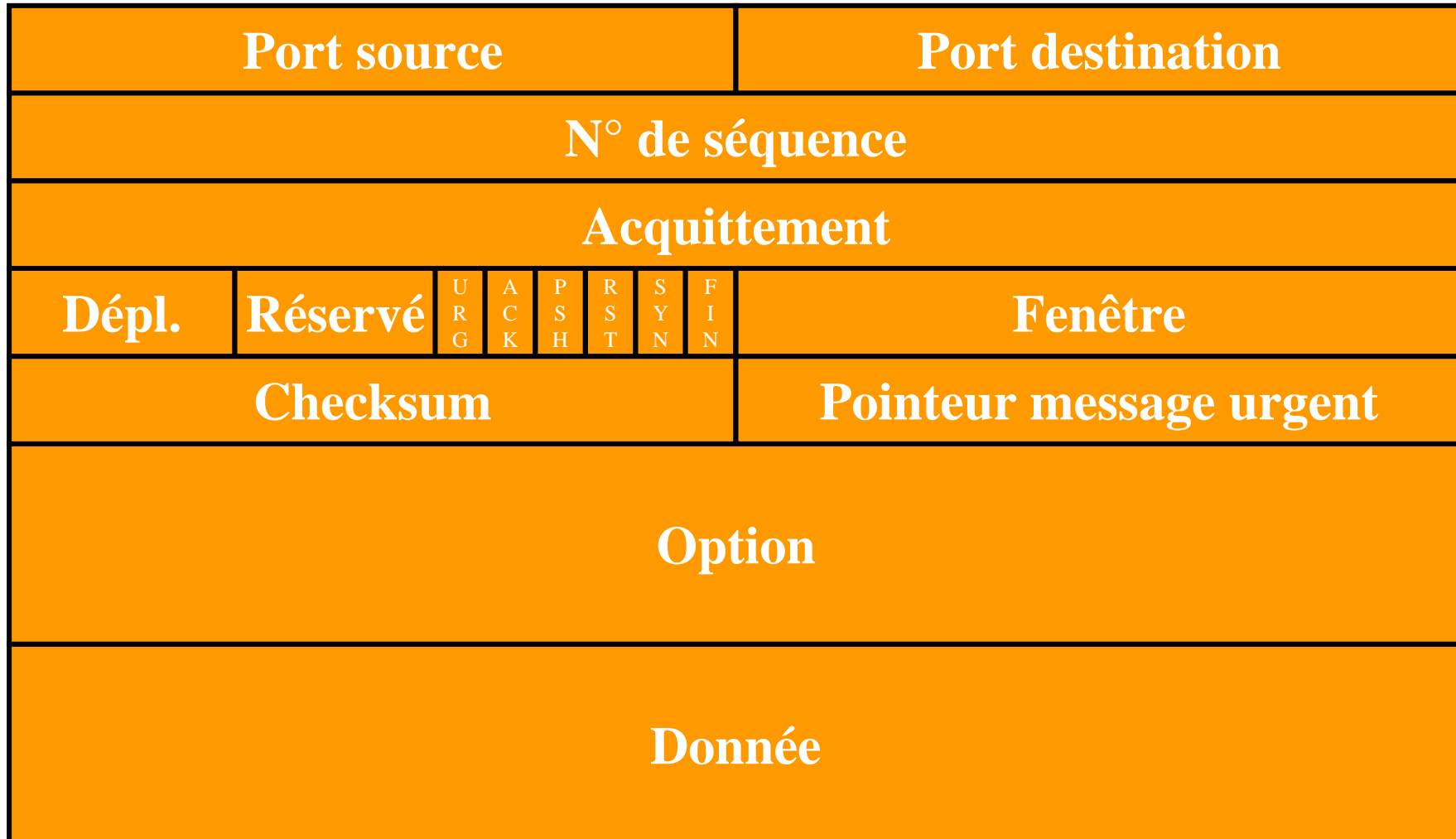


Protocole de communication associé Protocole TCP

0

15

31





Protocole de communication associé

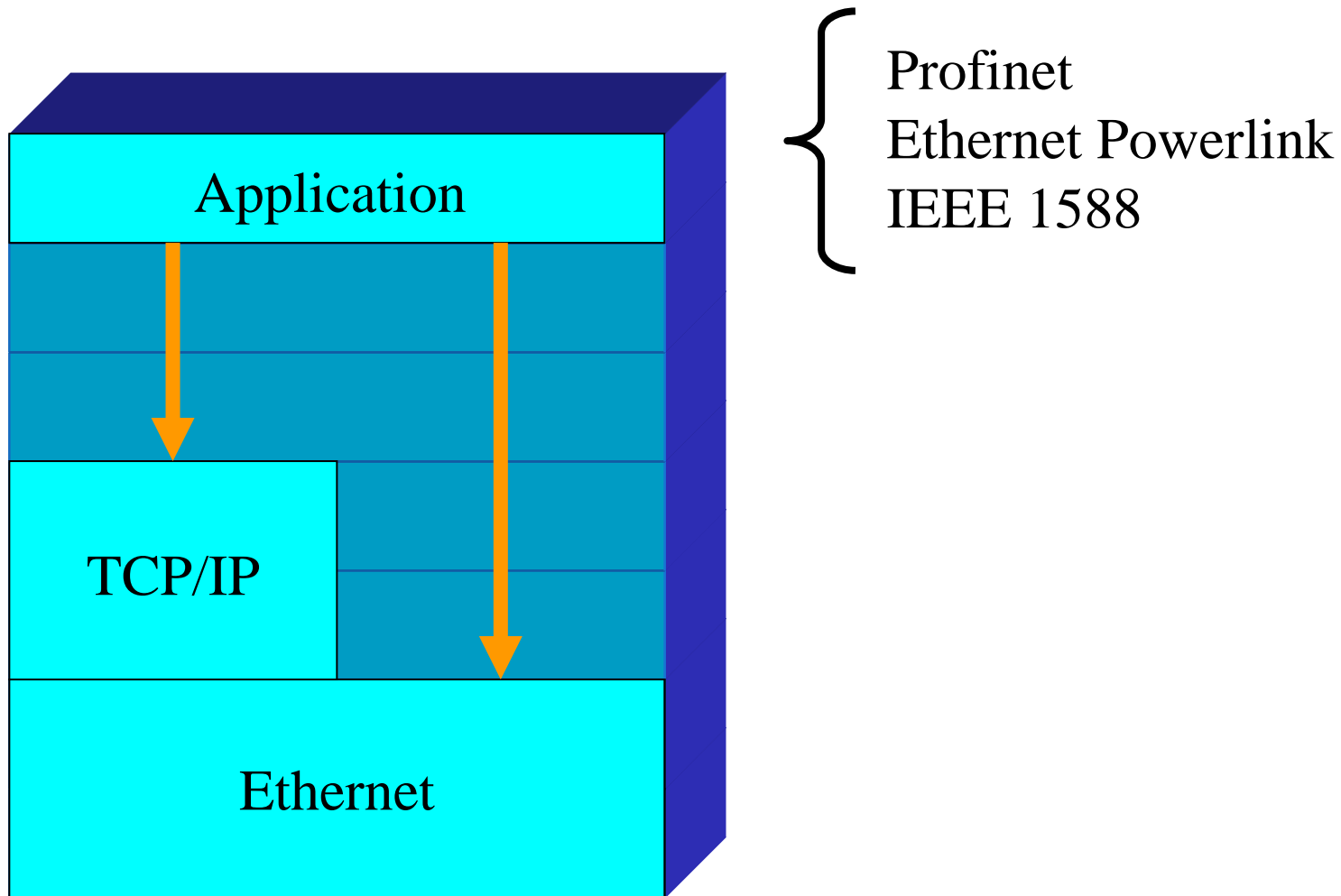
Exemple de trame TCP

00 50 fc 9f 0d 20 00 20 ed b9 4e a0 08 00 45 00 00 28 08
c3 40 00 80 06 6d e2 c0 8e 01 10 c0 8e 01 fe 04 1c 00 8b
a1 27 77 67 dc 9c e3 bb 50 10 43 33 0a e8 00 00



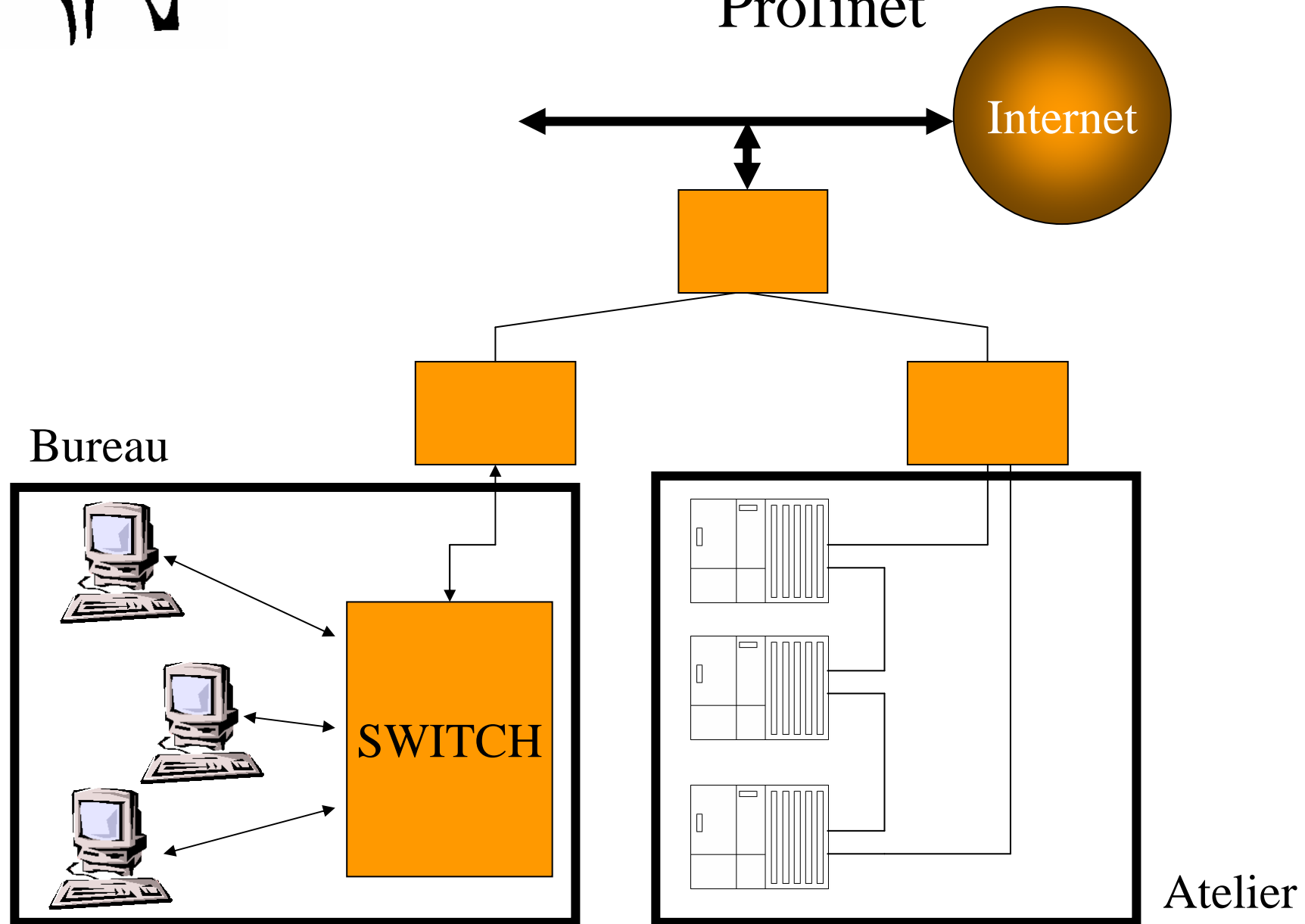
Mise en œuvre dans un cadre industriel

Principes



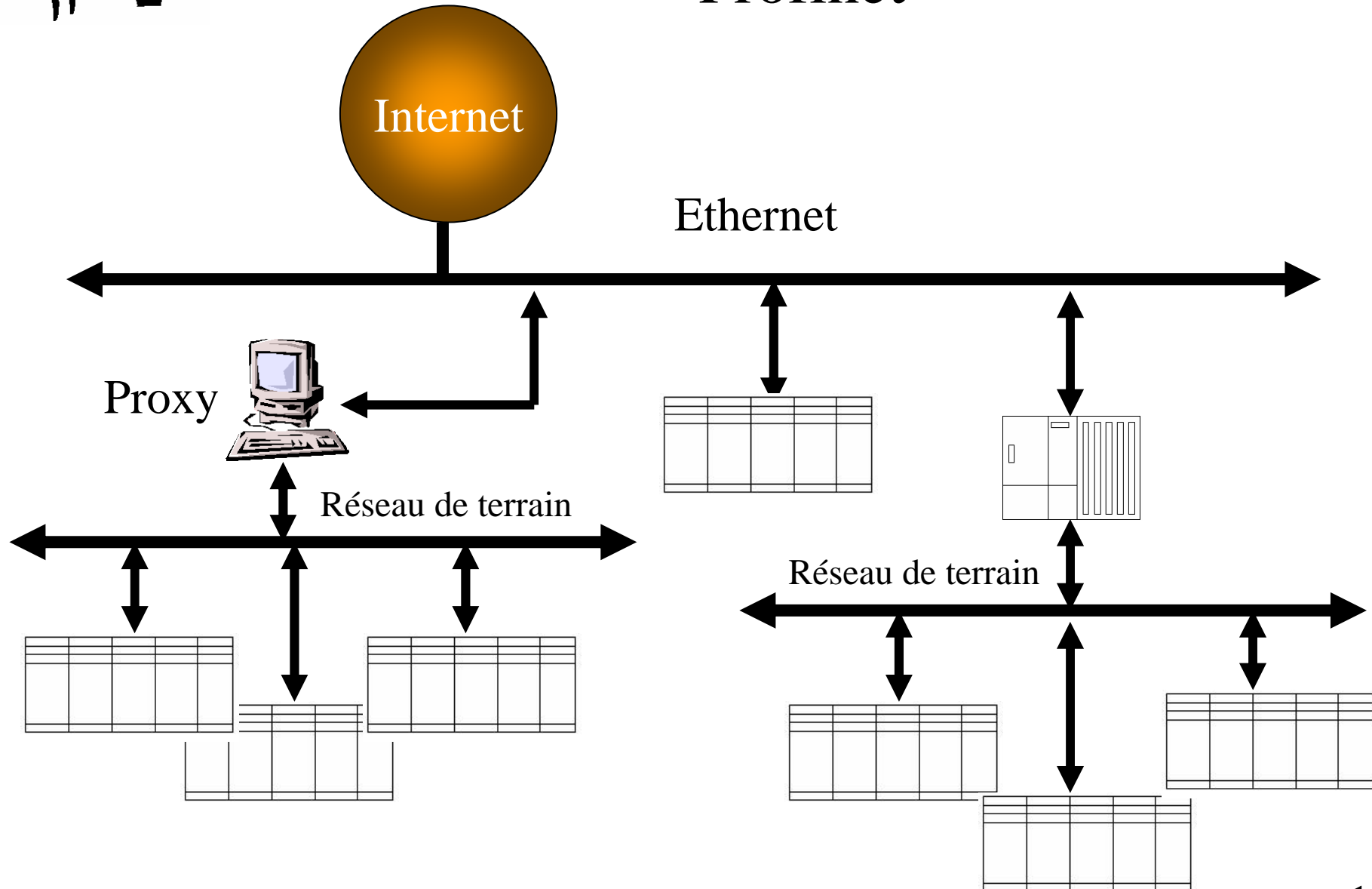


Mise en œuvre dans un cadre industriel Profinet



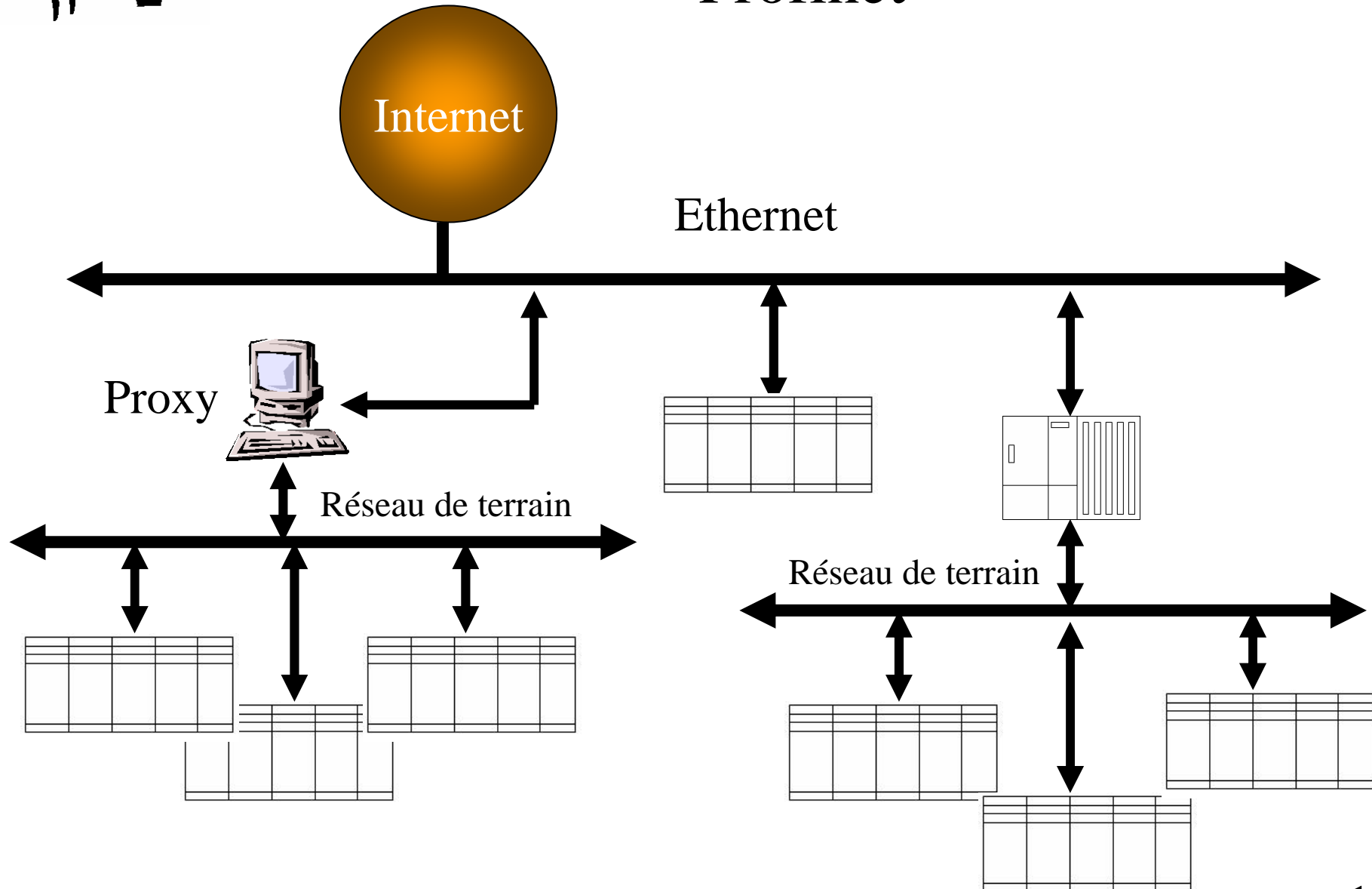


Mise en œuvre dans un cadre industriel Profinet





Mise en œuvre dans un cadre industriel Profinet



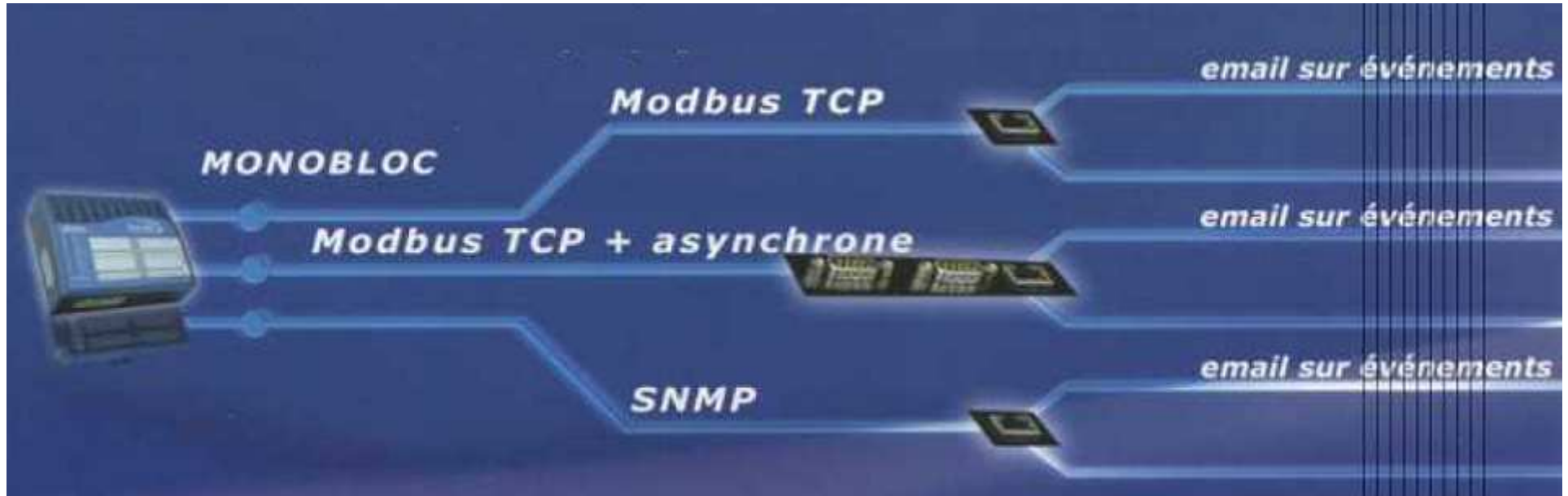


Mise en œuvre API et E/S déportée sous Ethernet





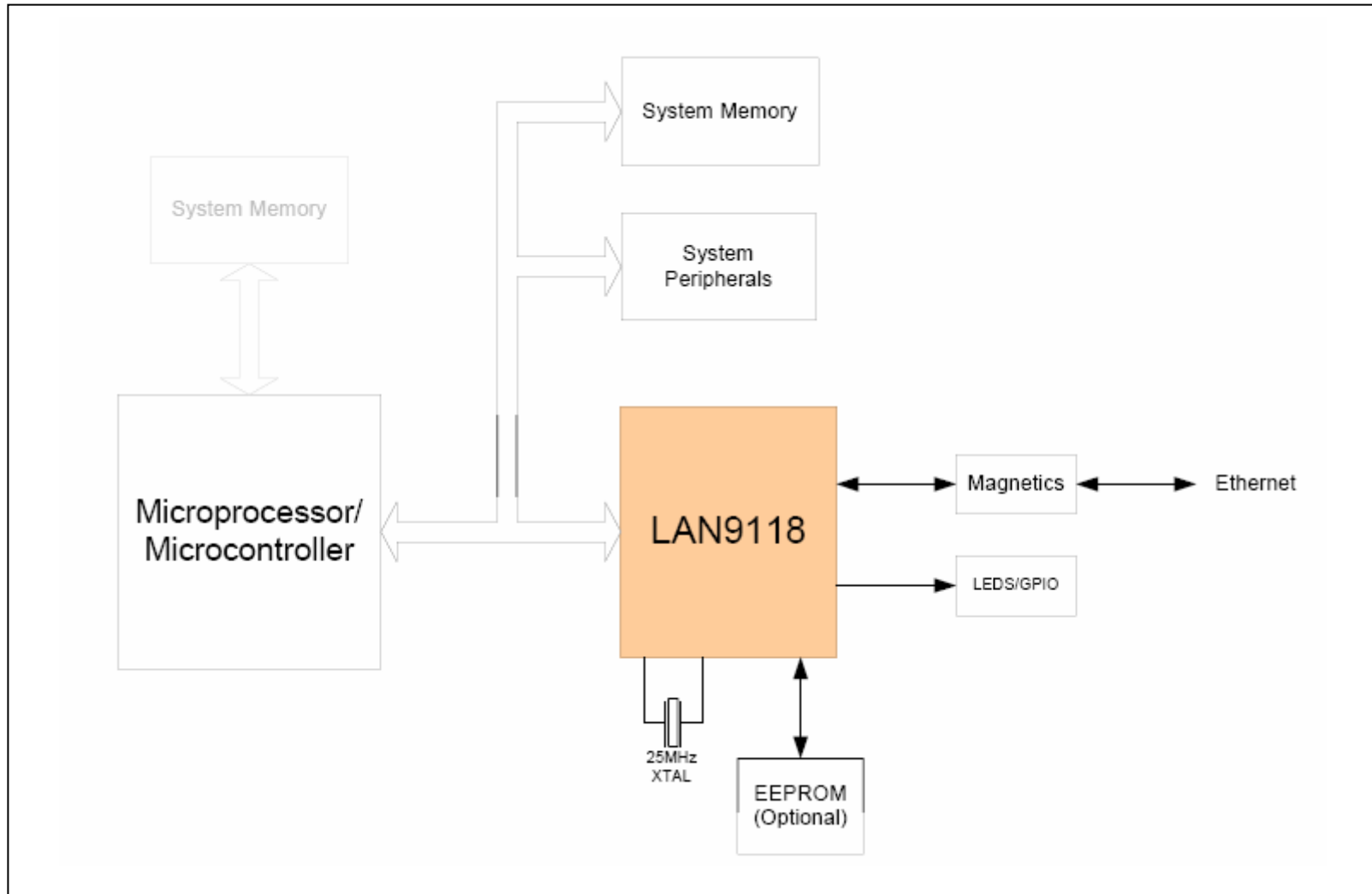
Mise en œuvre API et E/S déportée sous Ethernet





Mise en œuvre

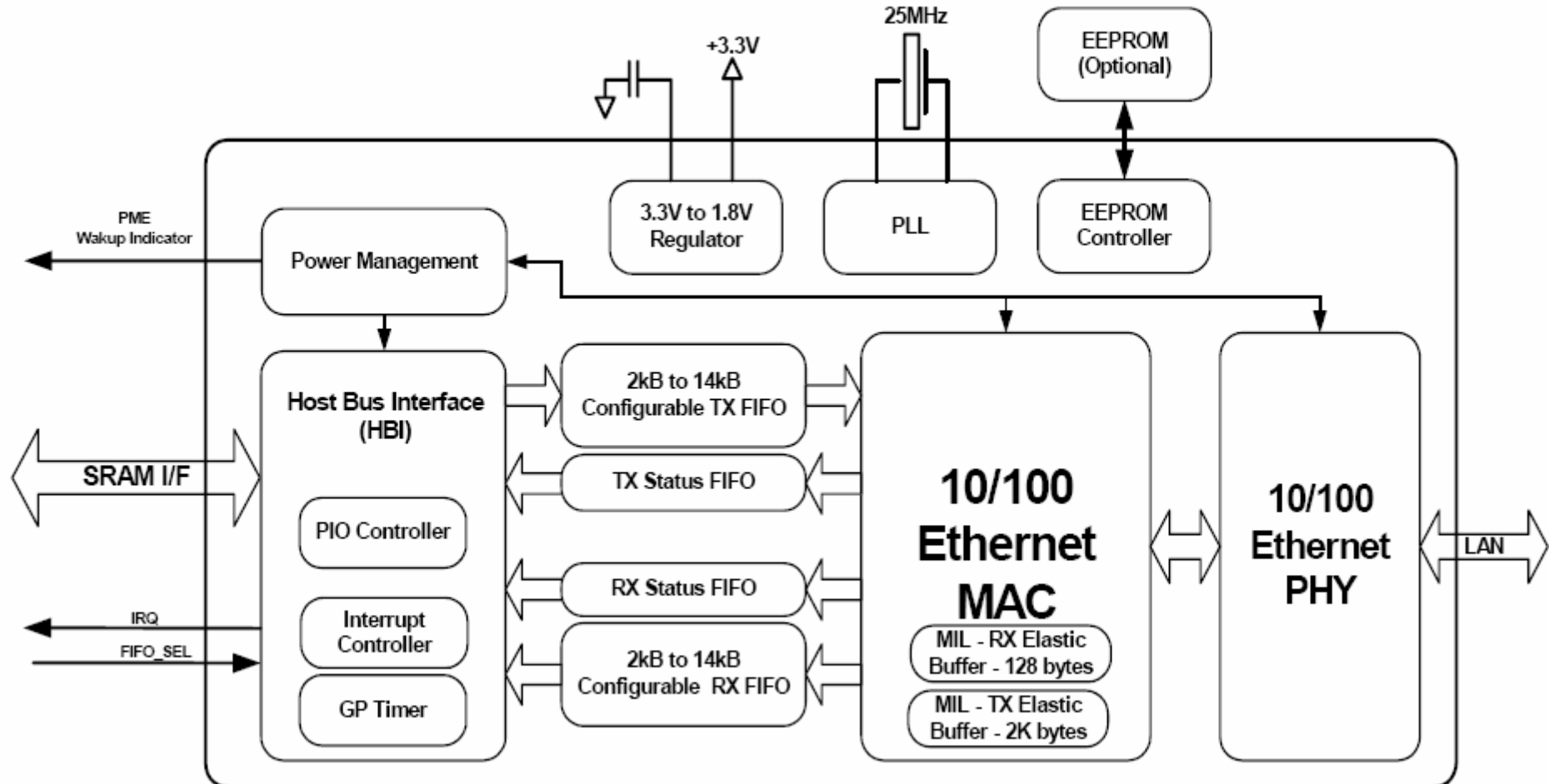
Contrôleur Ethernet – LAN9118





Mise en œuvre

Contrôleur Ethernet – LAN9118

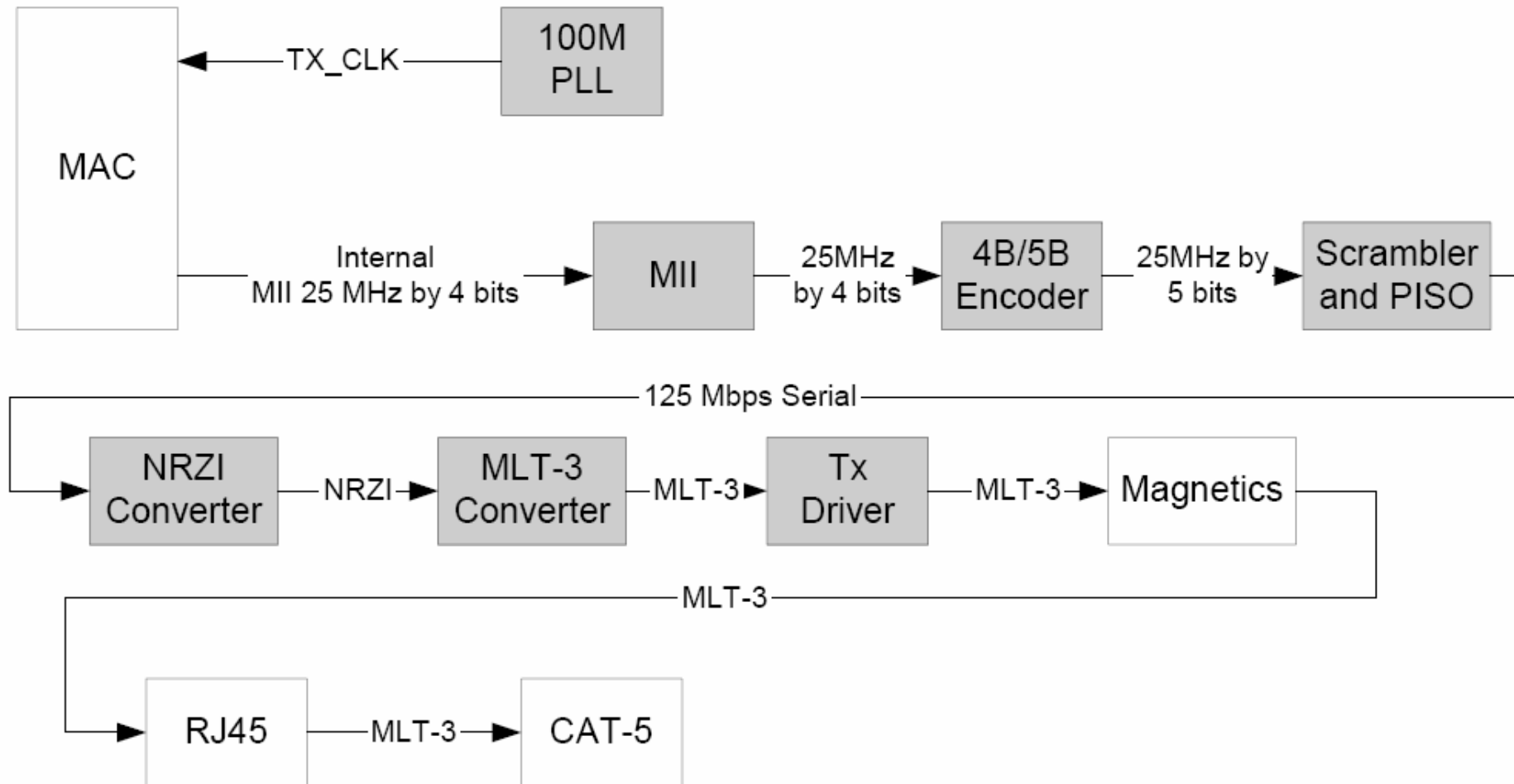




Mise en œuvre

Contrôleur Ethernet – LAN9118

Emission d'une trame

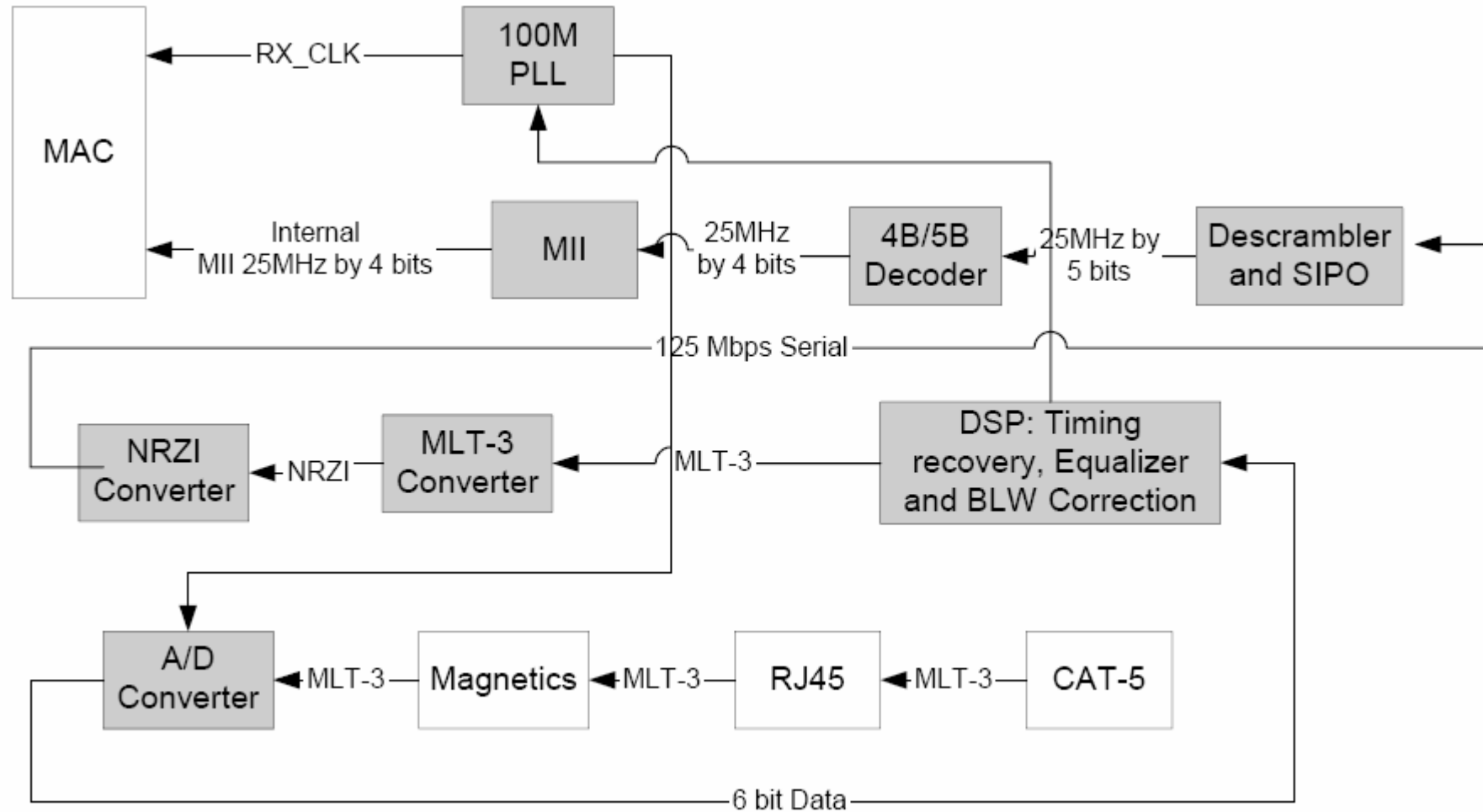




Mise en œuvre

Contrôleur Ethernet – LAN9118

Réception d'une trame

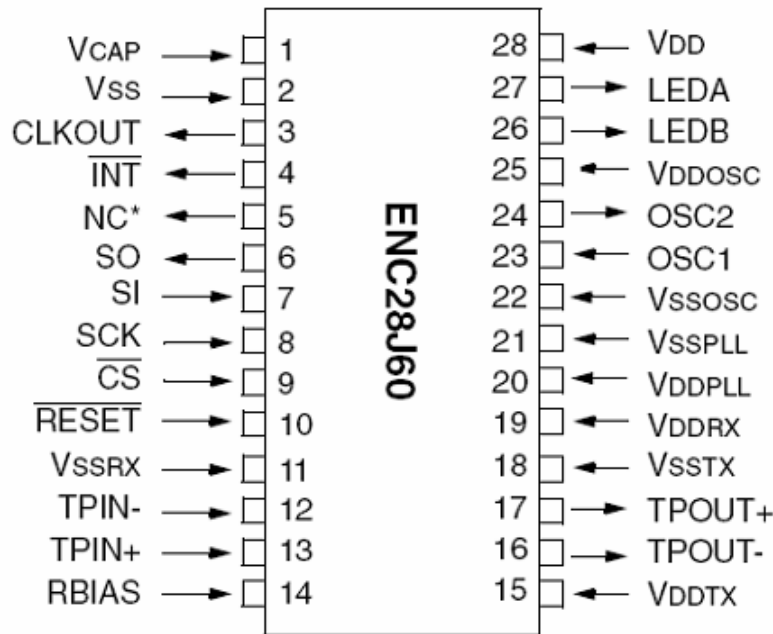




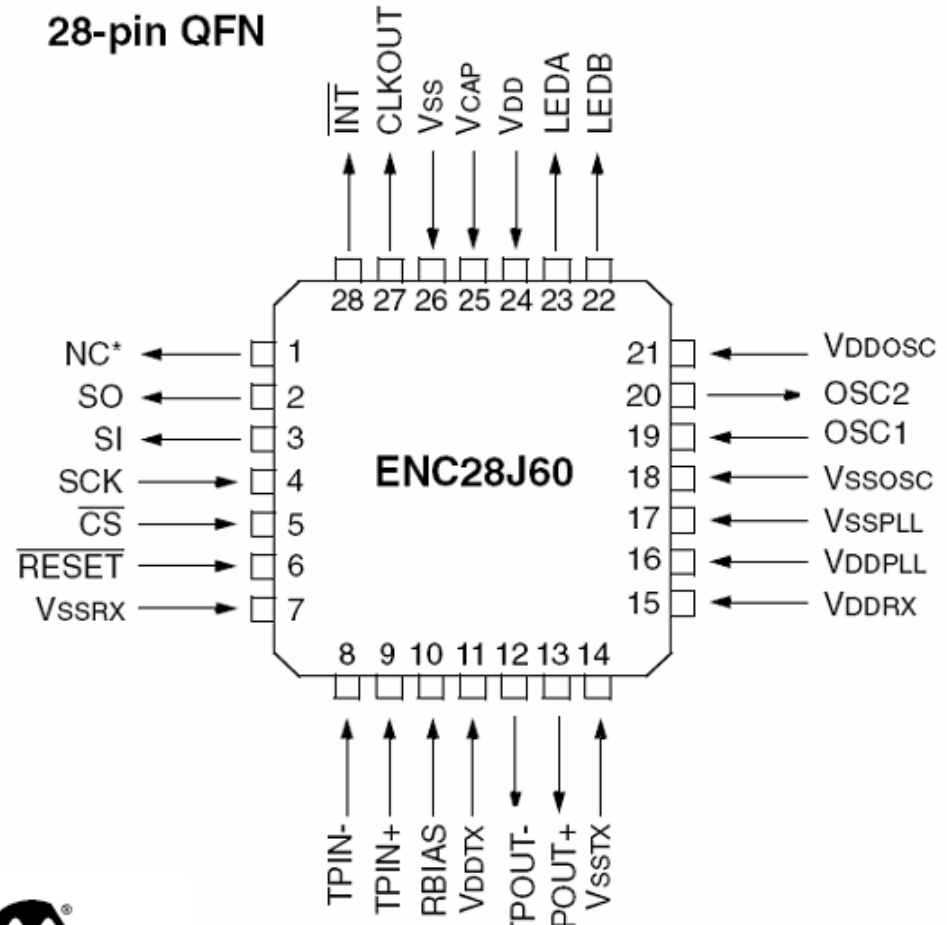
Mise en œuvre

Contrôleur Ethernet – ENC28J60

28-Pin SPDIP, SSOP, SOIC



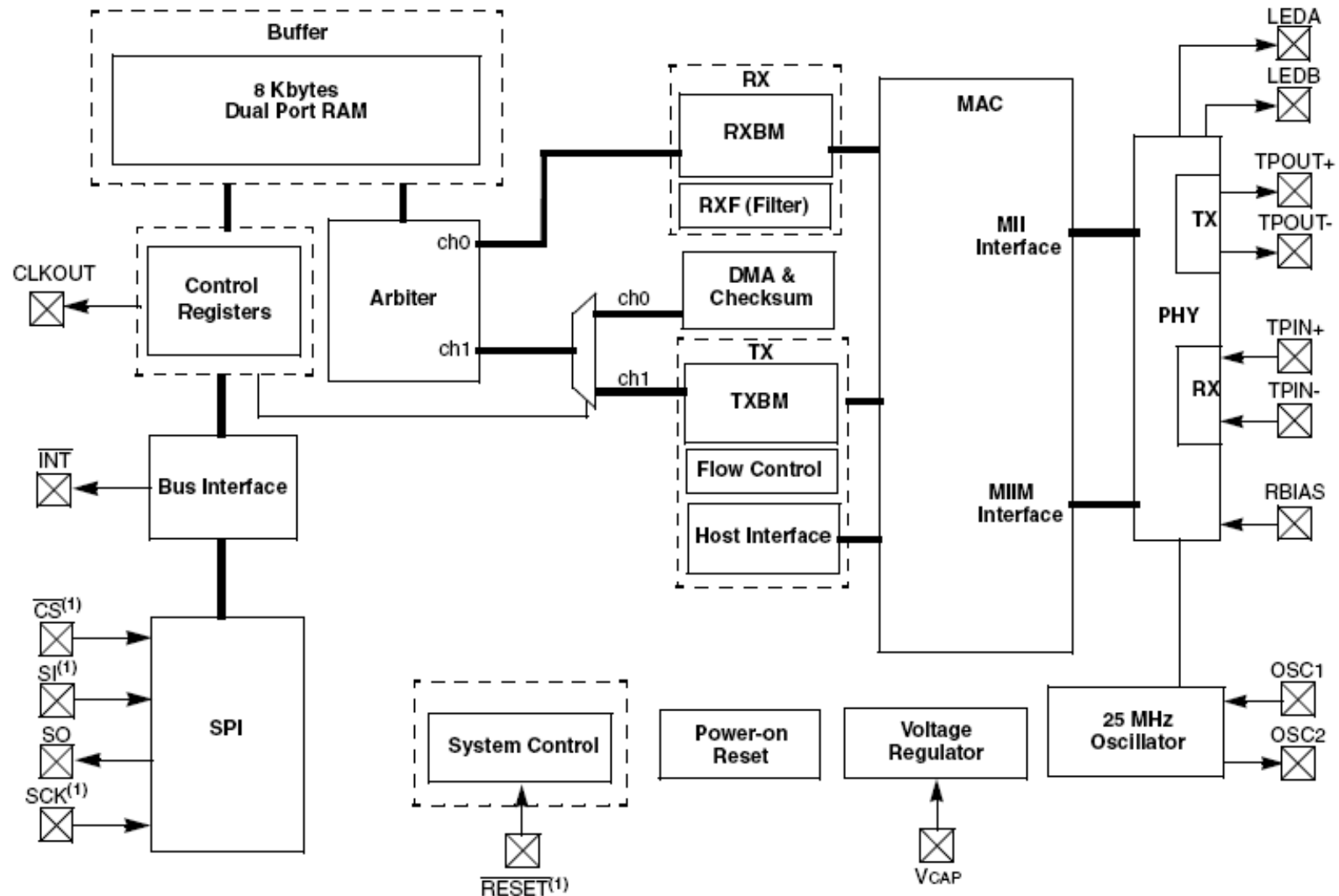
28-pin QFN





Mise en œuvre

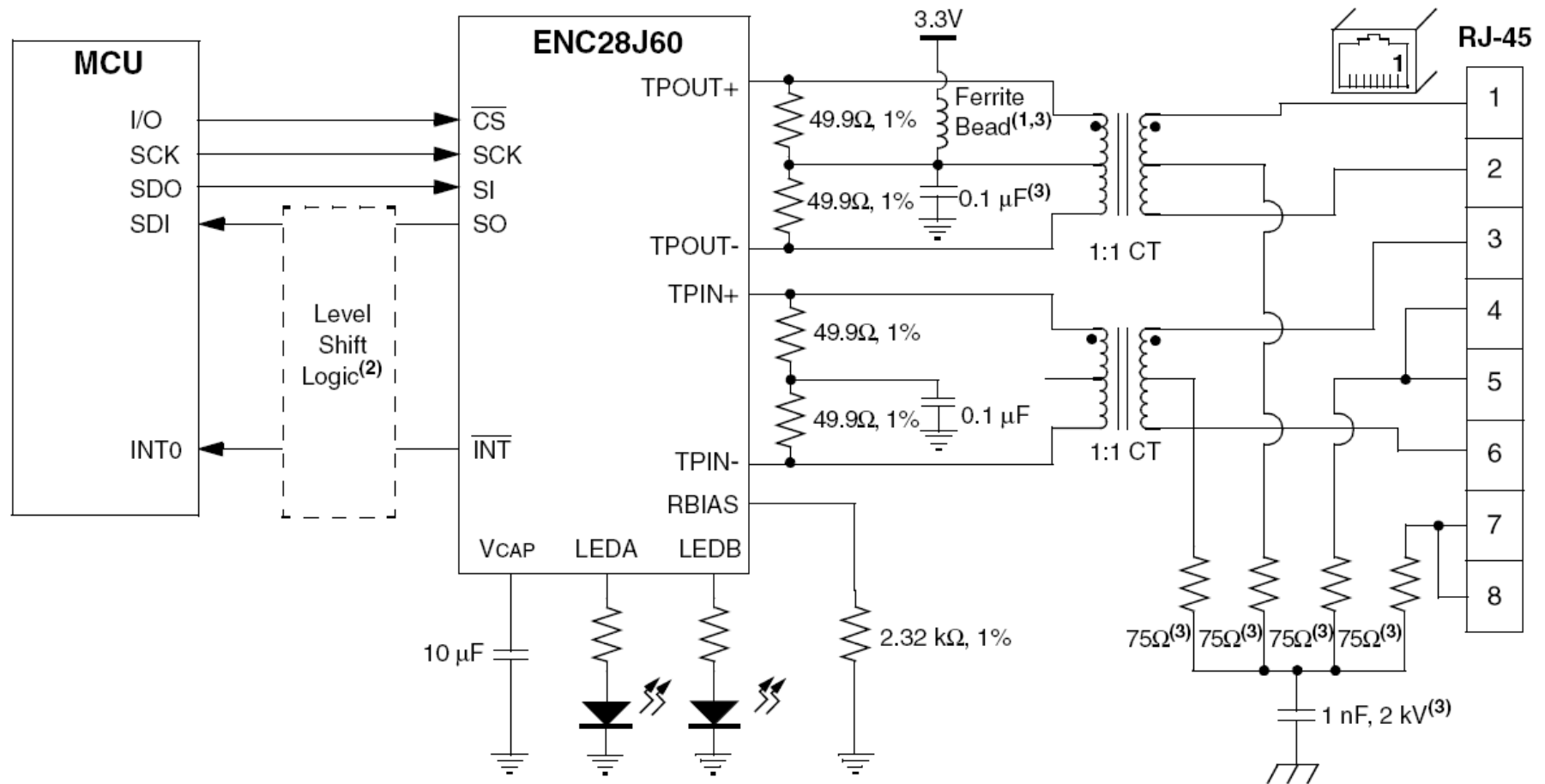
Contrôleur Ethernet – ENC28J60





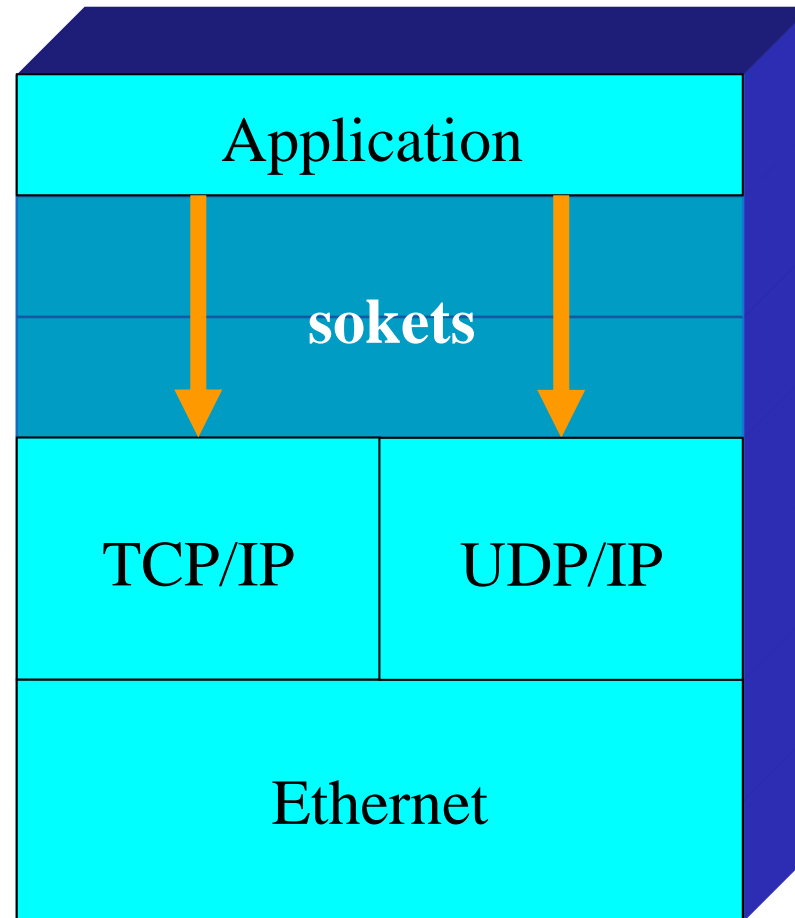
Mise en œuvre

Contrôleur Ethernet – ENC28J60





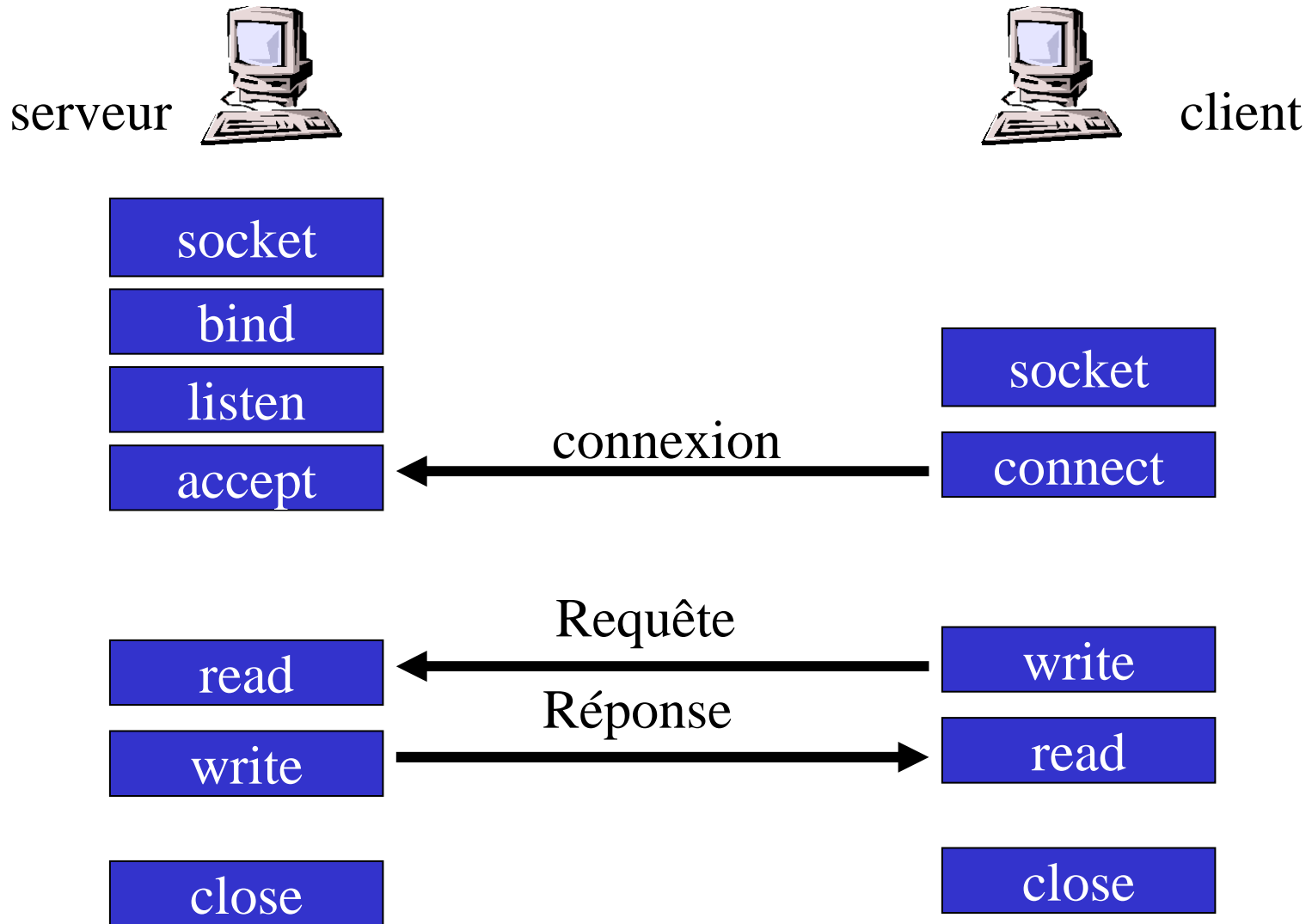
Mise en œuvre Linux et les sockets





Mise en œuvre

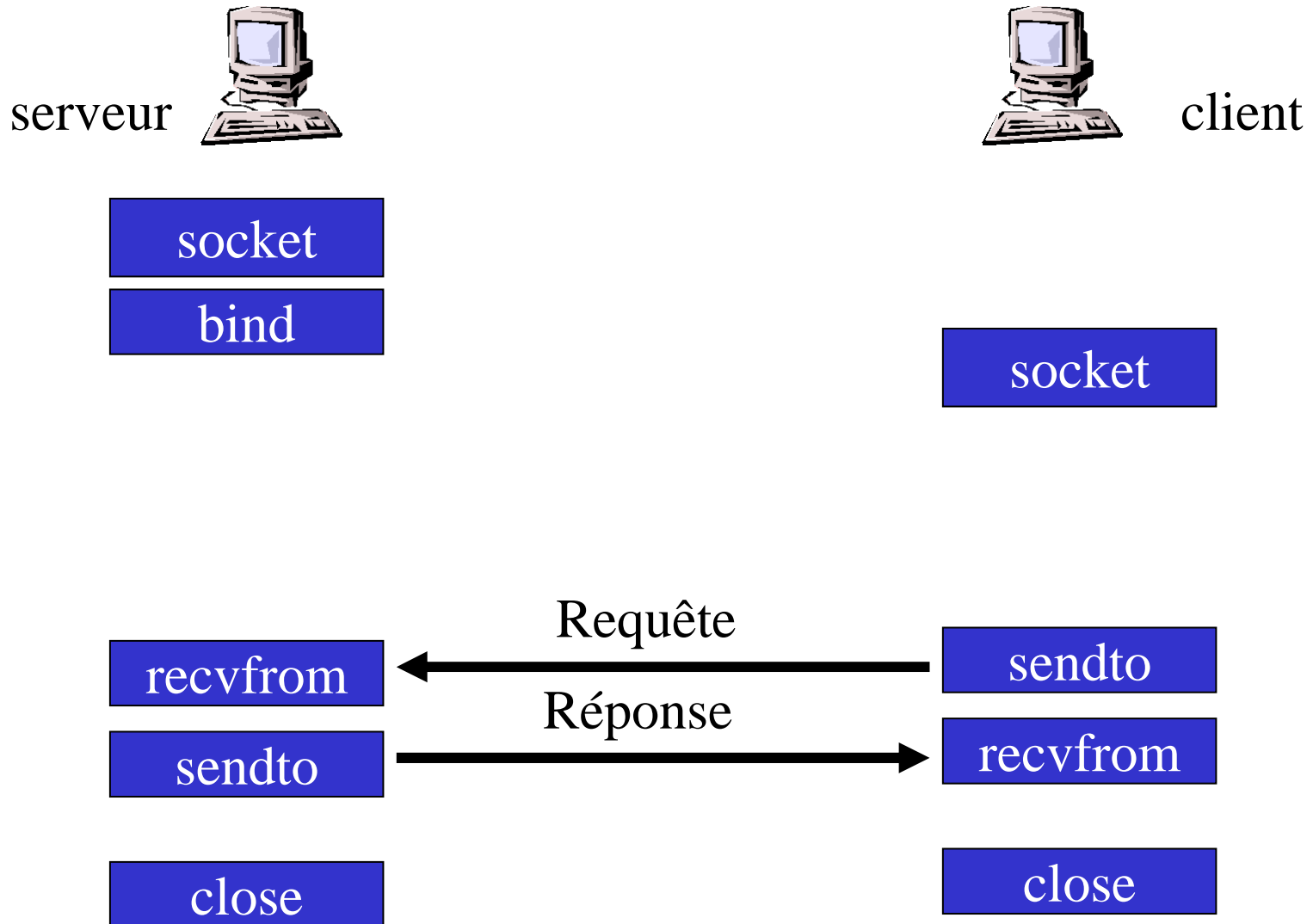
Linux et les sockets – Mode connecté





Mise en œuvre

Linux et les sockets – Mode connecté





Mise en œuvre Linux et les sockets

```
mysocket = socket(int socket_family, int socket_type, int protocol);
```

```
int bind(int sockfd, struct sockaddr *my_addr, socklen_t addrlen);
```

```
int connect(int sockfd, struct sockaddr *serv_addr, socklen_t addrlen);
```

```
int accept(int sock, struct sockaddr *adresse, socklen_t *longueur);
```



Mise en œuvre Linux et les sockets

```
ssize_t write(int fd, const void *buf, size_t count);
```

```
ssize_t read(int fd, void *buf, size_t count);
```

```
ssize_t sendto(int socket, const void *message, size_t length, int flags,  
const struct sockaddr *dest_addr, socklen_t dest_len);
```

```
ssize_t recvfrom(int socket, void *restrict buffer, size_t length, int  
flags, struct sockaddr *restrict address, socklen_t *restrict  
address_len);
```