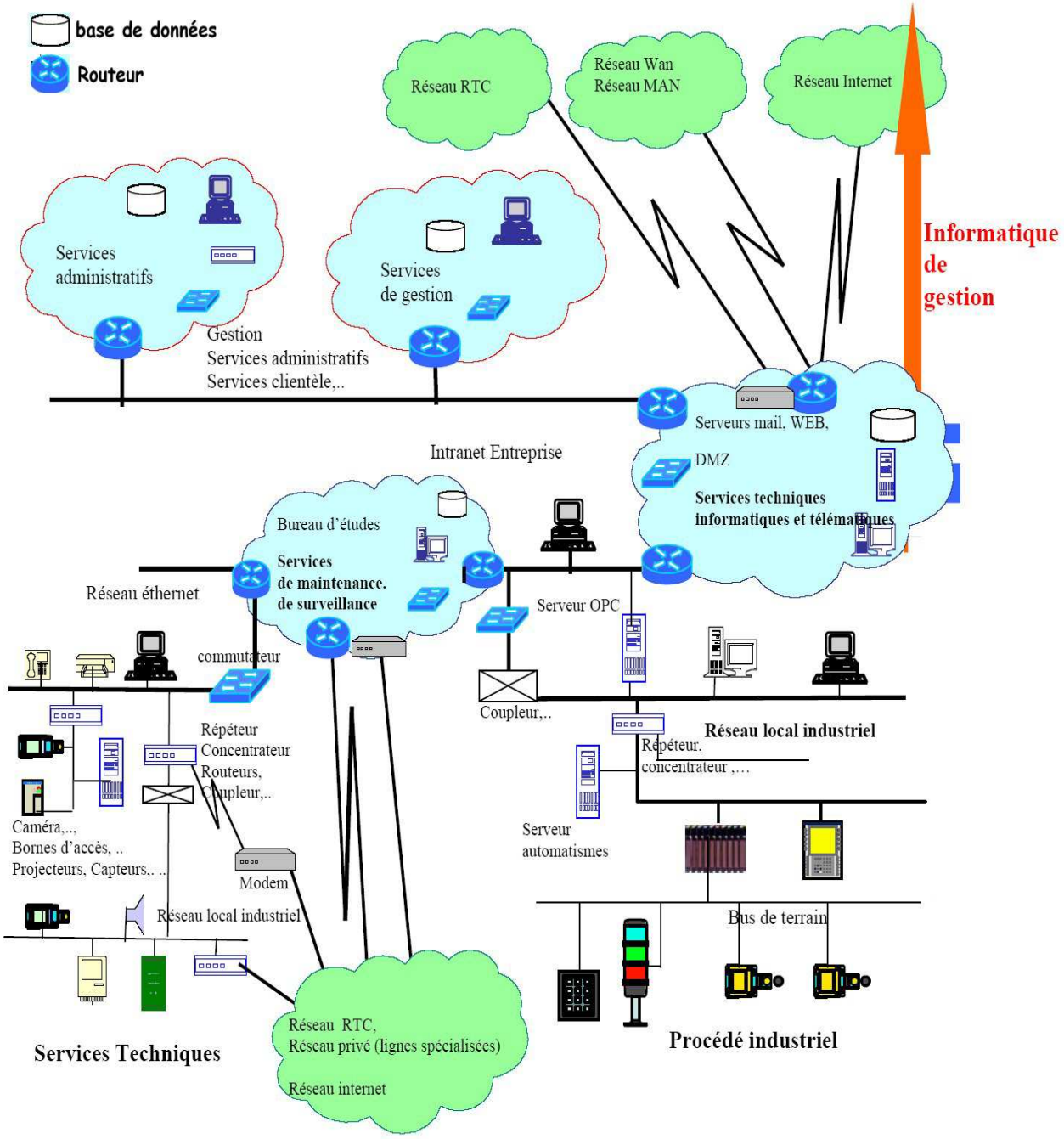
LE PROTOCOLE MODBUS

LE PROTOCOLE MODBUS

1) Introduction

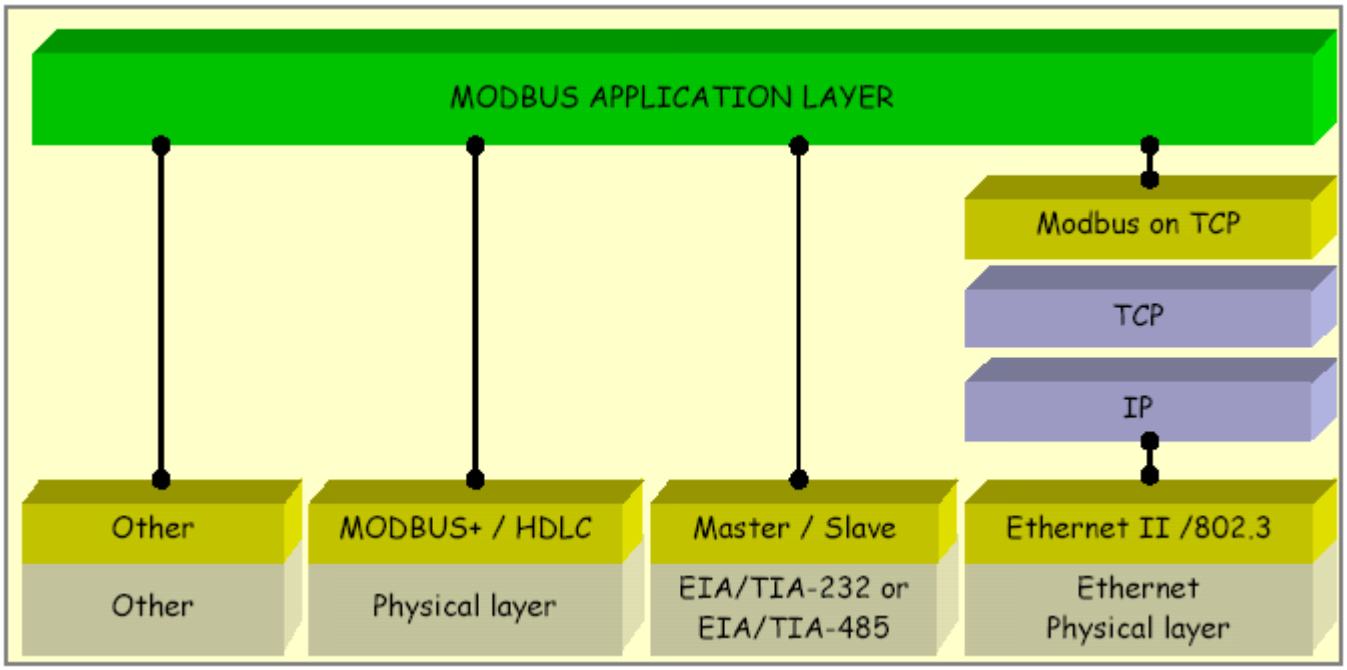
1.1 Position du réseau

Le protocole MODBUS est un moyen de communication dans les réseaux locaux industriels



Ces applications sont la supervision industrielle, gestion de production, commande de machines distantes.

1.2) couches du modèle OSI utilisées



En MODBUS + l'adresse MODBUS est convertie par l'émetteur en une adresse de noeud et de chemin d'accès sur le réseau. L'adresse destinataire est intégrée au protocole.

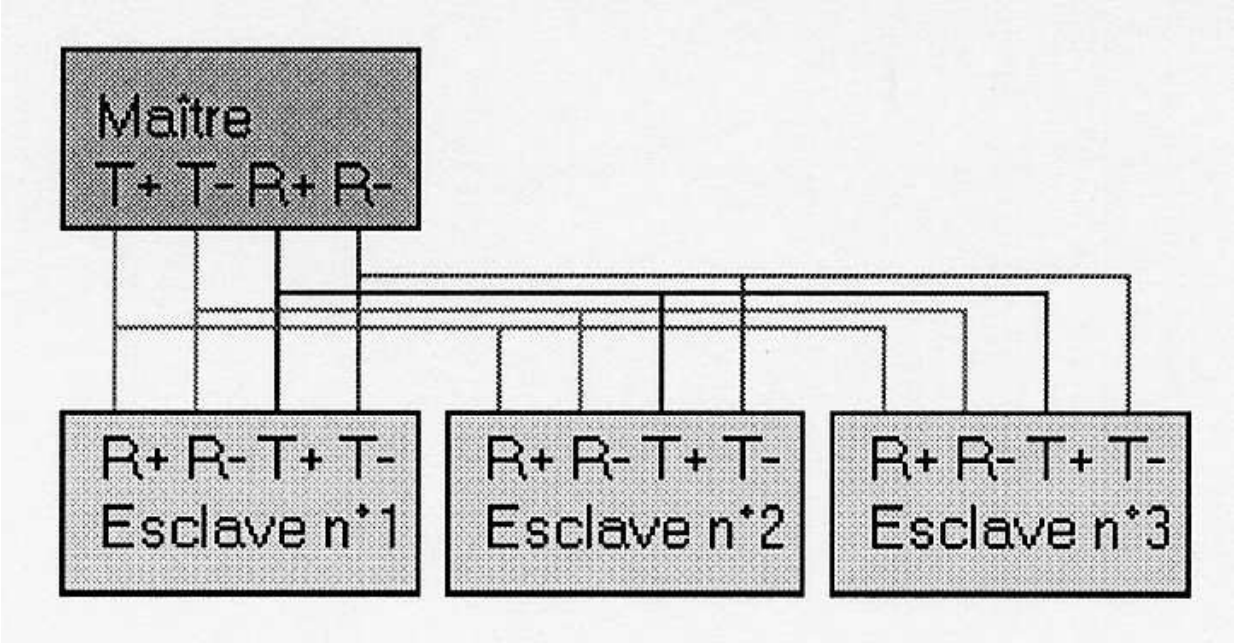
1.3) Le protocole MODBUS du type binaire ou RTU (remote terminal unit)

C'est la société Modicon (d'où le nom MODBUS) qui a développé ce protocole, c'est un protocole facile à implanter et très utilisé par un grand nombre de fabricant. Environ 95% des modules communicants sur MODBUS utilisent des trames "RTU".

2) Support de transmission

Il s'appuie sur la norme RS485 à 4 fils ou 2 fils (voir cours sur les transmissions séries) ou la norme RS232 si il y a qu'un seul esclave.

Exemple de câblage courant pour une liaison RS485 à 4 fils :



C'est une transmission half duplex, avec un débit maximum de 38400bits/s

3) Principes

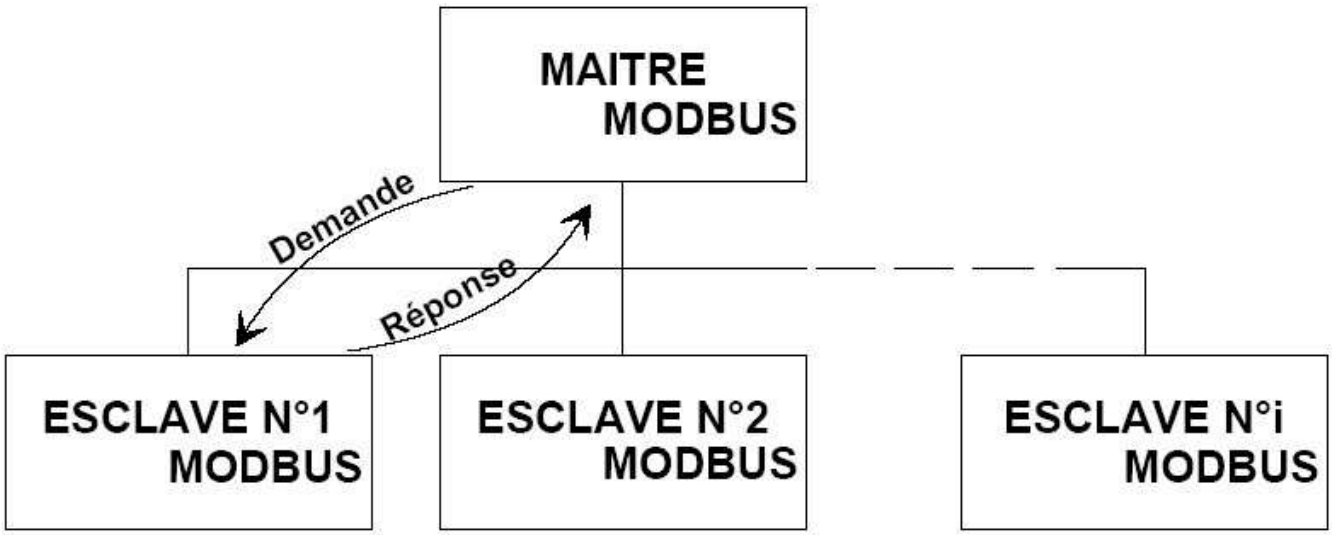
3.1) principe de diffusion

Ce type de protocole permet d'interroger depuis le maitre, un ou plusieurs esclaves intelligents. Une liaison multipoint relie entre eux, maitres et esclaves.

2 types de dialogues sont possibles entre maitre et esclaves.

- Le maitre parle à un esclave et attend sa réponse .
- Le maitre parle à l'ensemble des esclaves sans attente de réponse (principe de la diffusion générale).

Le maître gère les échanges et lui seul en a l'initiative. Ce maître réitère la question lors d'un échange erroné et décrète l'esclave absent après une non-réponse dans un temps enveloppe donné (time-out). Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre. Aucun esclave ne peut de lui-même envoyer un message sans y avoir été invité. Toutes communications latérales (esclave à esclave) ne peuvent exister que si le logiciel du maître a été conçu pour recevoir des données et les renvoyer d'un esclave à l'autre.



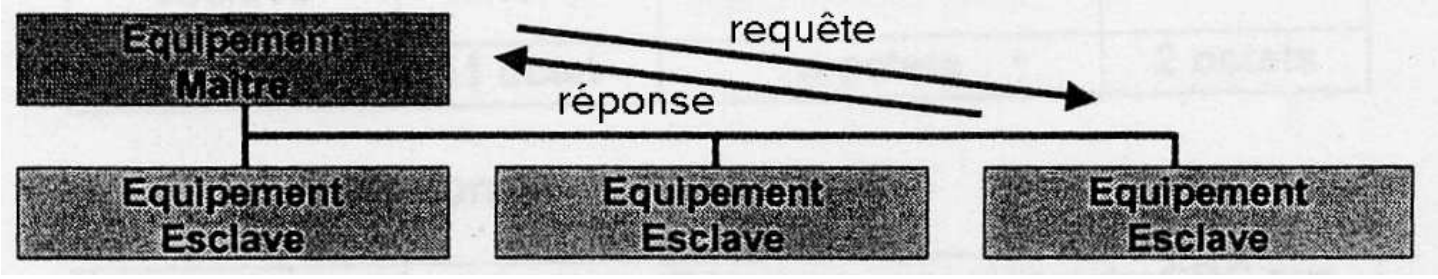
Le maître peut adresser **255 esclaves** repérés de l'esclave n°1 à l'esclave n°255.
Si le maître utilise le numéro d'esclave 0, cela correspond à une diffusion vers tous les esclaves (écriture seulement).
Le protocole MODBUS permet d'accéder aux équipements raccordés sur le même câble.

3.2) Méthode d'accès au support

La gestion des transactions incombent au maitre, 2 types de transations :

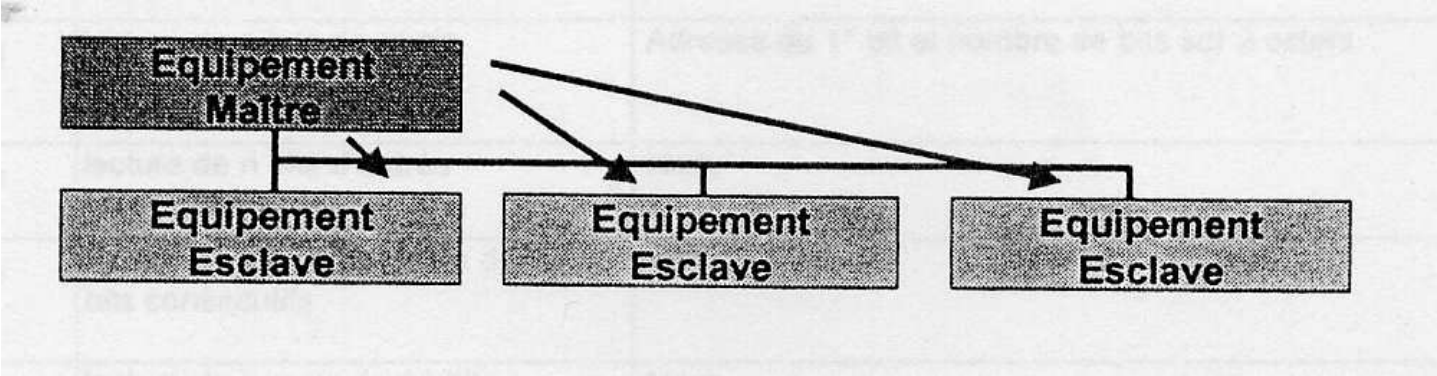
a) La transaction question-réponse

Le maitre émet une trame contenant une requête vers un équipement esclave particulier identifié par une adresse. Cette trame fait l'objet d'une trame réponse de la part de ce dernier. Une seule transaction question réponse peut avoir lieu à un instant donné.

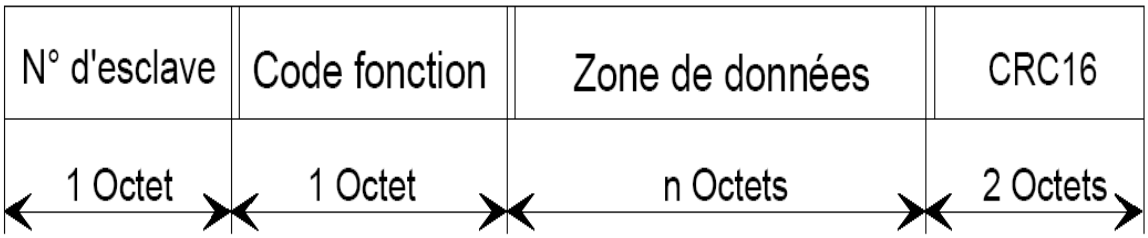


b) La transaction diffusion

Le maître émet une trame contenant une requête vers tous les équipements esclaves . Cette trame ne fait pas l'objet d'une trame réponse de la part de ces derniers.



3.3) Format d'une trame



N° d'esclave : 0 à 255 (0 = diffusion générale pour les fonctions d'écriture seulement).
Code fonction : Indique le type d'échange : 3 pour la lecture de n mots, 16 pour l'écriture de n mots par exemple.
Zone de données : Champ d'informations contenant les paramètres liés à la fonction: adresse mot, valeur de mot, nombre de mots...etc.
CRC16 : Mot utilisé pour détecter les erreurs de transmission (Cyclic Redundary Check).

La transaction Question/Réponse

La trame Question

N° équipement esclave	Code fonction	Paramètres associés au code fonction	CRC
1 octet	1 octet	n octets	2 octets

La trame Réponse

N° équipement esclave	Code fonction	Réponse associée au code fonction	CRC sur 2 octets
1 octet	1 octet	n octets	2 octets

La transmission s'effectue de manière asynchrone (bit de start---bit de stop), la longueur maximum d'une trame est de 261 octets.

3.4) Les principaux codes fonctions

Les principales fonctions (1)

Code	Signification des fonctions	Paramètres nécessaires à la fonction
$(1)_{16}$	lecture de n bits de sortie consecutifs	Adresse du 1° bit et nombre de bits sur 2 octets
$(2)_{16}$	lecture de n bits d'entrée consécutifs	Idem
$(3)_{16}$	lecture de n mots internes de 16 bits consecutifs	Idem
$(4)_{16}$	lecture de n mots de 16 bits consecutifs associés à des cartes d 'entrées	Idem

Les principales fonctions (2)

Code	Signification des fonctions	Paramètres nécessaires à la fonction
$(5)_{16}$	écriture d'un bit	Adresse du bit et valeur du bit sur 2 octets $(FF00)_{16}$ pour un et $(0000)_{16}$ pour 0
$(6)_{16}$	écriture d'un mot de 16 bits	Adresse du mot et valeur du mot sur 2 octets
$(7)_{16}$	lecture rapide d'un octet	Lecture d'un registre spécial
$(F)_{16}$	écriture de n bits consécutifs	Adresse du 1° bit, nombre de bits sur 2 octets, nombre d'octets (1 octet) du champ suivant, valeur des bits (jusqu'à 256 bits=32 octets)
$(10)_{16}$	Ecriture de n mots consécutifs	Adresse du 1° mot, nombre de mots sur 2 octets, valeur des mots, nombre d'octets du champ suivant, valeur des mots.

Le protocole MODBUS RTU du type MODICON

Structure de la trame	*Jbus*	*Modbus*
N° d'esclave	*1 à 255*	*1 à 247*
Code fonction	*1 à 16 (sauf 9 & 10)*	*1 à 24*
Longueur trame	*255 Octets max.*	*261 Octets max.*
CRC	*CRC16*	*CRC16*
*Détection des trames **	*silence > 3 caractères*	*silence > 1,5 ou > 3,5 caractères*
** Cette différence ne pose pas de problème à des débits supérieurs à 1200 bits/s car ces temps sont inférieurs au temps de traitement de l'équipement (temps de retournement).*		
Fonctions standard		
F1	*Lecture de n bits à @ 0 to FFFF (1 < n < 2000)*	*Lecture de n bits à @ 1 to 9999*
F2	*Lecture de n bits à @ 0 to FFFF (1 < n < 2000)*	*Lecture de n bits à @ 10001 to 19999*
F3	*Lecture de n mots à @ 0 to FFFF (1 < n < 125)*	*Lecture de n mots à @ 40001 to 49999*
F4	*Lecture de n mots à @ 0 to FFFF (1 < n < 125)*	*Lecture de n mots à @ 30001 to 39999*
F5	*Ecriture de 1 bit à @ 0 to FFFF*	*Ecriture de 1 bit à @ 1 to 9999*
F6	*Ecriture de 1 mot à @ 0 to FFFF*	*Ecriture de 1 mot à @ 40001 to 49999*
F7	*Lecture rapide de 8 bits utilisateur*	*Lecture du status d'exception (8 bits) Signale des défauts sur l'équipement*
F15	*F15 Ecriture de n bits à @ 0 to FFFF*	*Ecriture de n bits à @ 1 to 9999*
F16	*Ecriture de n mots à @ 0 to FFFF*	*Ecriture de n mots à @ 40001 to 49999*

3.5) Différents exemples de code fonction

a) Les codes fonctions 01 ou 02

■ Demande.

N° esclave	1 ou 2	Adresse du 1er bit à lire : PF Poids fort | Pf Poids faible	Nombre de bits à lire : 1 ≤ n ≤ 2000	CRC 16 pf* | PF*
1 octet	1 octet	2 octets	2 octets	2 octets

■ Réponse.

N° esclave	1 ou 2	Nombre d'octets lus	1er octet lu	Dernier octet lu	CRC 16 pf* | PF*
1 octet	1 octet	1 octet	n octets		2 octets

Détail d'un octet :

Dernier bit transmis							1er bit transmis

Nota : Les bits non utilisés dans l'octet sont mis à zéro.

Exemple :
Lecture des bits 204 à 211 de l'esclave n° 1.

■ *Demande.*

01	01	02 04	000E	

CRC 16

■ *En hexadécimal, le nombre de bits à lire de 204 à 211 se détaille comme suit :*
□ *0204, 0205, 0206, 0207, 0208, 0209, 020A, 020B, 020C, 020D, 020E, 020F, 0210, 0211,*
□ *soit 14 valeurs ou 0E en hexadécimal.*

■ *Réponse.*

01	01	02	10101001	00101110	

20B ← 204 211 ← 20C CRC 16

b) Les codes fonctions 03 ou 04

Lecture de n mots : fonction 3 ou 4

■ Demande.

N° esclave	3 ou 4	Adresse du 1er mot à lire : PF * | pf *	Nombre de mots à lire n ⩽ 125 : PF * | pf *	CRC 16 pf * | PF *
1 octet	1 octet	2 octets	2 octets	2 octets

■ Réponse.

N° esclave	3 ou 4	Nombre d'octets lus	Valeur 1er mot PF * | pf *	Valeur du dernier mot PF * | pf *	CRC 16 pf * | PF *
1 octet	1 octet	1 octet	2 octets	2 octets	2 octets

Exemple :
Lecture des mots 805 à 80A de l'esclave n° 2.

■ *Demande.*

02	03	0805	0006	

CRC 16

■ *Réponse.*

02	03	OC	XXXX	YYYY	

CRC 16
Valeur du mot 80A
Valeur du mot 805

c) **Ecriture d'un bit : fonction 5**

■ Demande.

N° esclave	5	Adresse du bit PF * | pf *	Valeur du bit	0	CRC 16 pf * | PF *
1 octet	1 octet	2 octets	1 octet	1 octet	2 octets
			bit forcé à 0_ bit forcé à 1_	écrire 0 écrire FF	

■ Réponse.

N° esclave	5	Adresse du bit PF * | pf *	Valeur du bit	0	CRC 16 pf * | PF *

Nota : Si le numéro d'esclave est 00, tous les esclaves exécutent le forçage sans émettre de réponse.

Exemple :
Forçage à 1 du bit 210 de l'esclave n° 2.

02	05	02 10	FF	00	

└ CRC 16

d) **Ecriture d'un mot: fonction 6**

■ Demande.

N° esclave	6	Adresse du mot		Valeur du mot		CRC 16	
		PF *	pf *	PF *	pf *	pf *	PF*
1 octet	1 octet	2 octets		2 octets		2 octets	

■ Réponse.

N° esclave	6	Adresse du mot		Valeur du mot		CRC 16	
		PF *	pf *	PF *	pf *	pf *	PF*

Nota : Si le numéro d'esclave est 00, tous les esclaves exécutent le forçage sans émettre de réponse.

Exemple :

Ecriture de la valeur 1000 dans le mot d'adresse 810 de l'esclave n°1.

01	06	810	1000	
				CRC 16

3.6) La Trame d'exception

En cas d'impossibilité d'executer la requête, l'équipement esclave renvoie une trame d'exception. Une trame d'exception est repérée par le code fonction. Le bit MSB de ce champ est à 1
Ex code lecture de n bits =$(1)_{16}$→ code exception $(81)_{16}$

N° équipement esclave	Code fonction	Code erreur	CRC sur 2 octets
1 octet	1 octet	1 octet	2 octets

Les différents code erreurs :

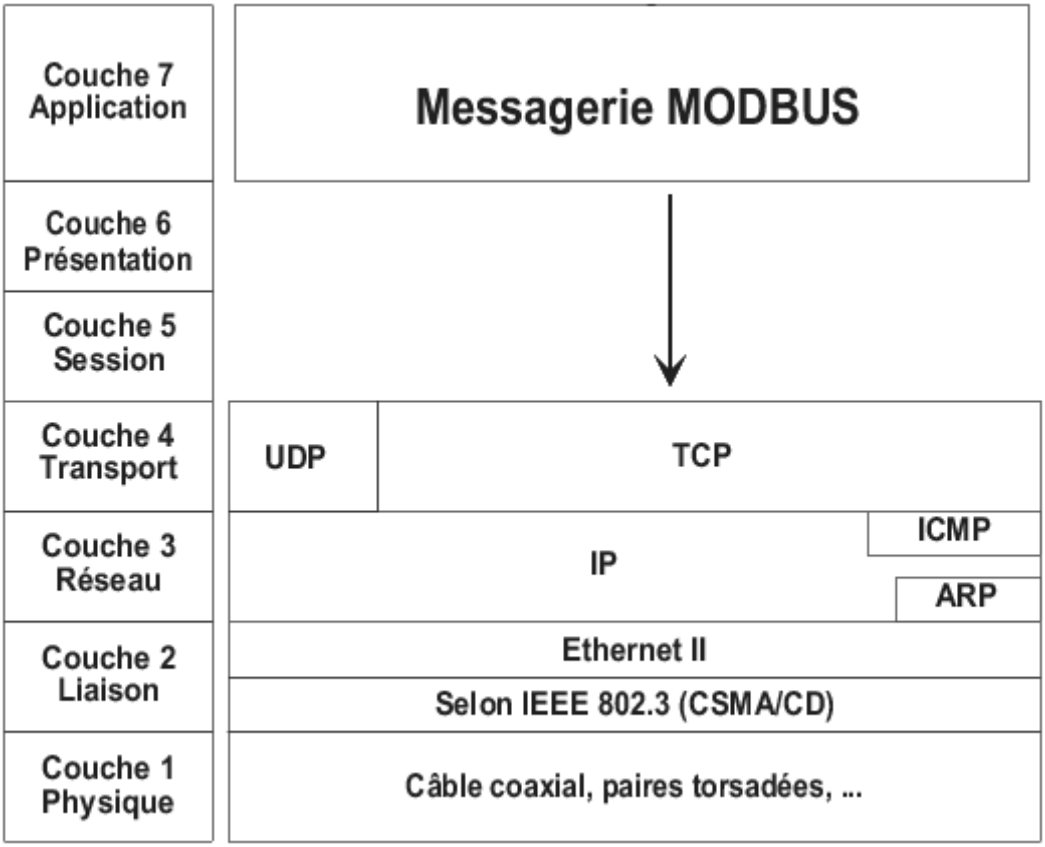
Code	***Nom***
01	Illegal function
02	Illegal data address
03	Illegal data value
04	Slave device failure
05	Acknowledge
06	Slave device busy
08	Memory parity error
0a	Gateway path unavailable
0b	Gateway target device failed to respond

4 - Le protocole MODBUS et TCP/IP

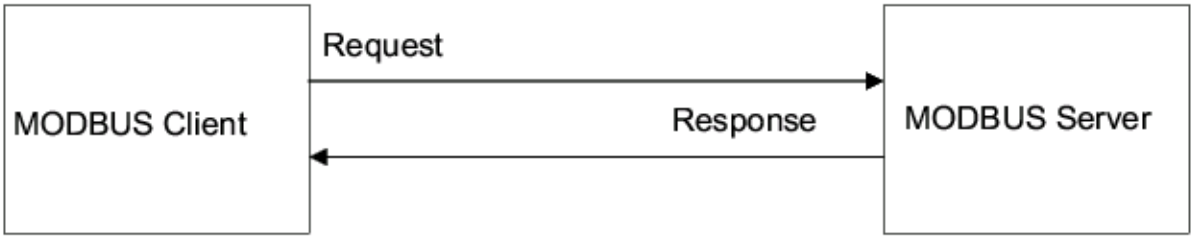
4.1) L'encapsulation

Le principe est simple, il s'agit d'encapsuler un message MODBUS dans une trame TCP/IP classique.

4.2) Modèle client/serveur



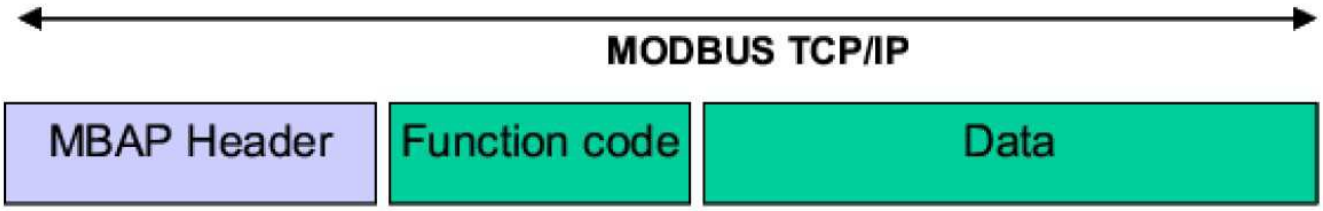
Le protocole MODBUS/TCP fonctionne en mode client/serveur comme ci-dessous :



Le client MODBUS effectue une requête et attend en retour une réponse du serveur MODBUS. Chaque équipement MODBUS/TCP qu'il soit client ou serveur sera identifié par 2 informations :

- adresse IP (donc adresse Ethernet via le protocole ARP) = à définir par l'utilisateur
- **port TCP = 502** (port réservé pour les applications MODBUS)

4.3) Format d'une trame



le premier champ "numéro de l'esclace" en modbus RTU est remplacé par un champ "MBAP Header" et le champ "Error check (CRC)" disparait car les couches TCP/IP effectuent déjà un contrôle d'erreur (CRC).

En-tête MBAP (ModBus Application Protocol header)

L'en-tête MBAP d'une longueur de 7 octets se décompose comme suit :

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

Voici la description des différents champs :

- **Transaction Identifier** : il s'agit en fait du numéro d'un message MODBUS. Chaque message (requête+réponse) est identifié par Un numéro différent
- **Protocol Identifier** : il s'agit du numéro du protocole pour les systèmes multiplexés qui en utilisent plusieurs (MODBUS = 0)
- **Length** : il s'agit de la longueur du message à suivre (donc Unit Identifier + Data).
- **Unit identifier** (numéro de l'esclave)