Protocoles TCP – IP
et
Modbus-TCP

Analyse de Trafic Ethernet - IP

Eddy BAJIC
IUT Nancy Brabois
eddy.bajic@iutnb.uhp-nancy.fr

Sommaire

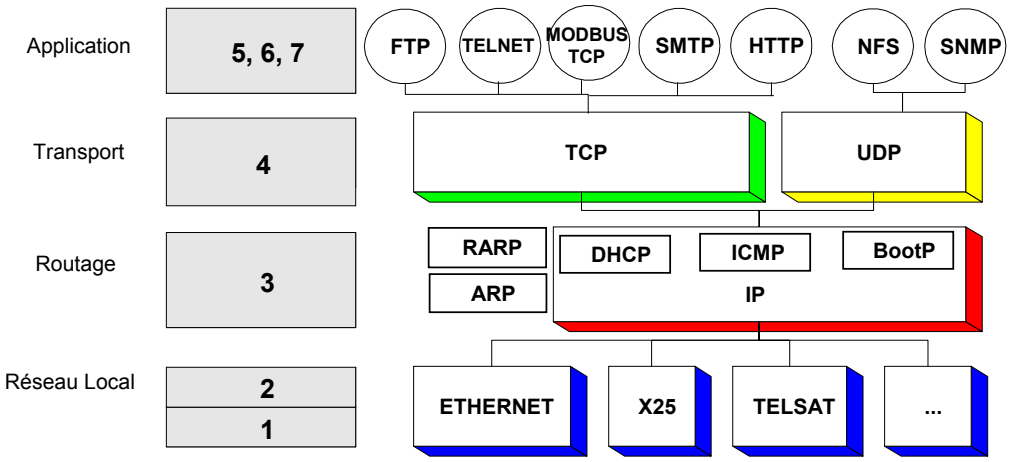
- *La pile de protocoles ou Stack TCP –IP*
- *Mécanisme d'Encapsulation et Trame Ethernet – TCP - IP*
- *ARP – DNS - DHCP*
- *Protocole de routage IP*
- *Protocoles de transport TCP - UDP*
- *Modbus – TCP*
- *Application sur Module E/S Wago*
- *Analyse de Trafic avec EtherReal*

Famille de Protocoles Internet

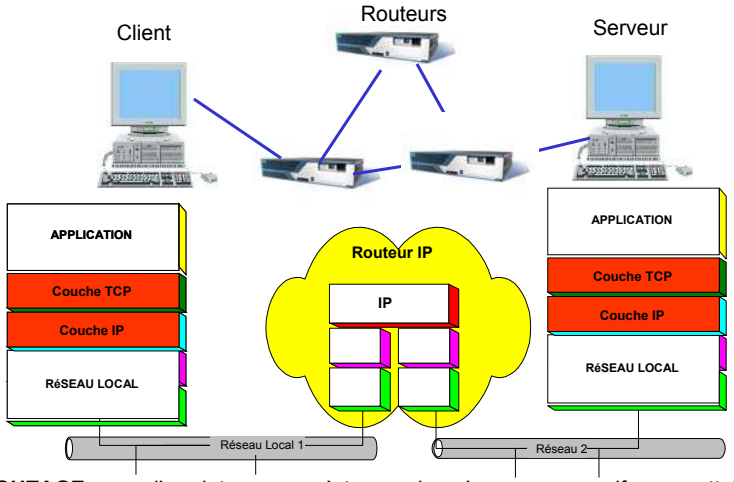
Le terme TCP-IP n'est pas réservé aux seuls protocoles TCP et IP,

il englobe un ensemble de protocoles assurant les mécanismes d'échanges d'information sur Internet

⇨ **Modèle réseau TCP-IP / Pile de protocoles TCP-IP (Stack TCP-IP)**



Interconnexion par protocoles TCP-IP



⇨ Le **ROUTAGE** d'un datagramme à travers les réseaux successifs pour atteindre le destinataire du message. (Repose sur un adressage logique mondial)

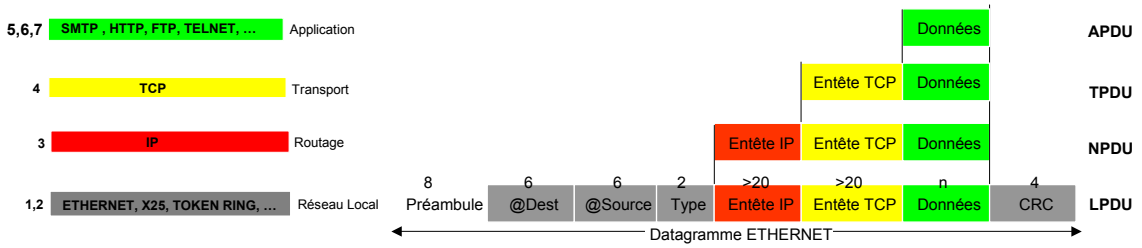
⇨ Le **CONTRôLE DE FLUX** des paquets pour garantir leur acheminement à travers les réseaux traversés (Acquittement / Réémission)

⇨ La **FRAGMENTATION / ASSEMBLAGE** des paquets pour correspondre aux tailles maximales des trames dans les réseaux traversés.

Mécanisme d ’encapsulation du Modèle TCP-IP

L'empilement des protocoles selon les couches du modèle TCP-IP permet d'installer l'ensemble des services nécessaires à l'utilisateur d'un réseau.

⌦ **pile de protocoles TCP IP** ou **Stack TCP IP**

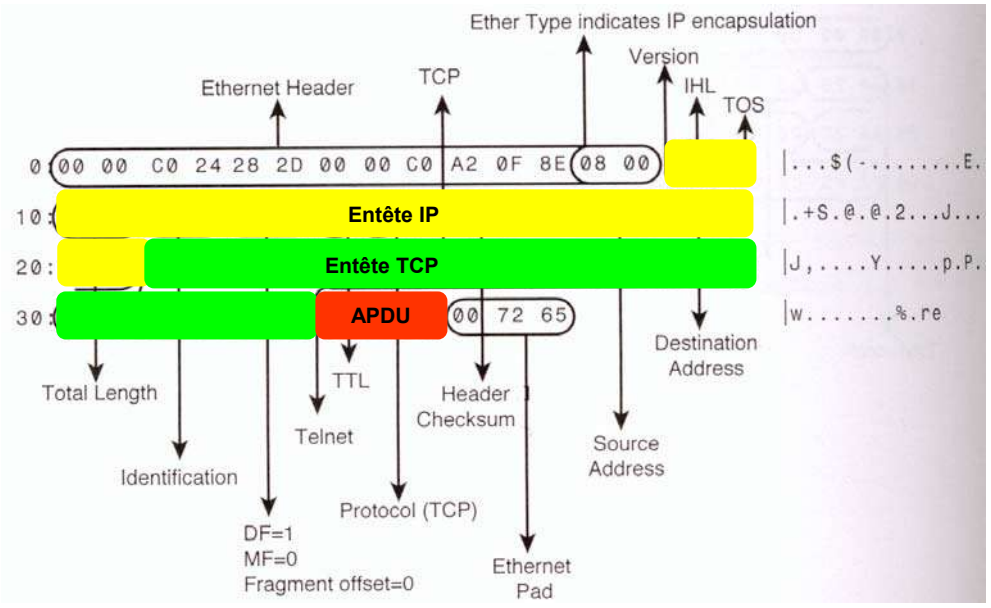


Chaque couche fournit des données encapsulées par la couche inférieure pour former finalement une trame réseau

⌦ **Encapsulation**.

Le paquet mise en forme par une couche est appelé : **PDU Protocol Data Unit**

Trame Ethernet - TCP -IP

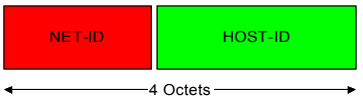


Adressage IP et Classes de Réseau Internet

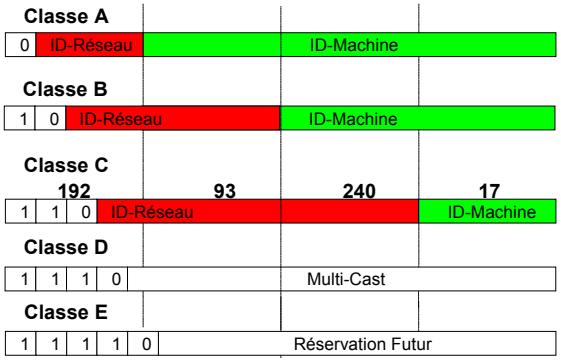
Une @ IP tient sur **4 octets** soit **32 bits**.

192	93	240	17

Une @ IP permet d'**identifier le réseau IP** et la **machine hôte**



3 classes de réseaux Internet basés sur la taille maximale du réseau (nb maxi de stations adressables)



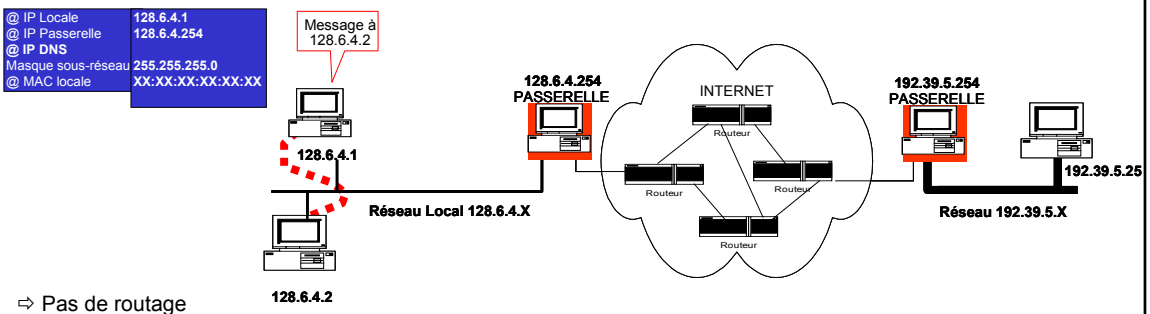
Classe	Plage d'Adressage	Nbre de Réseaux	Nbre de Stations
A	0.xxx.xxx.xxx à 127.xxx.xxx.xxx	127	16 777 214
B	128.0.xxx.xxx à 191. 255.xxx.xxx	16 383	65 534
C	192.0.0.xxx à 223.255.255.xxx	2 097 151	254

Adressage IPv4 (32 bits)
permet de définir **2,1 Millions de réseaux pour un total de 3,72 Milliards d'hôtes**.

Adressage IPv6 (128 bits)
permet d'adresser au minimum **1 Milliard de réseaux**

Réseau Local IP – Résolution d'Adresse

Les machines source et destination sont sur le même réseau local
⇨ **128.6.4.1** veut envoyer un message à **128.6.4.2**



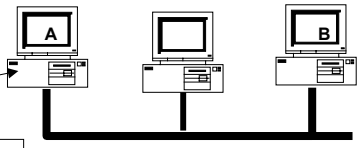
⇨ Pas de routage

⇨ La machine source doit connaître l'adresse physique (@MAC) de la machine dest. pour lui envoyer le message

⇨**ARP : Adress Resolution Protocol** (RFC 826) :
Protocole de résolution d'adresse @MAC @IP sur un réseau local

(RARP : Reverse Adress Resolution Protocol)

Protocole ARP



@ IP A : 192.39.5.27

@ MAC A : 00:80:20:FE:10:15

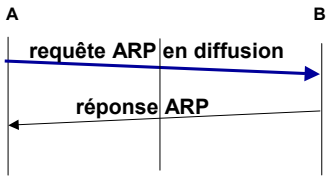
@ IP B : 192.39.5.25

@ MAC B : 00:80:20:FE:50:64

Cache ARP

Validité	@IP	@MAC
13:36	**192.39.5.25**	**00:80:20:FE:50:64:FE:64**

La machine A cherche l 'adresse MAC de la machine 192.39.5.25 :
➔ **Protocole ARP**



Mécanisme de résolution d'adresse ARP :

1. A émet sur le réseau local **une requête ARP** en diffusion contenant l'adresse IP cible recherchée

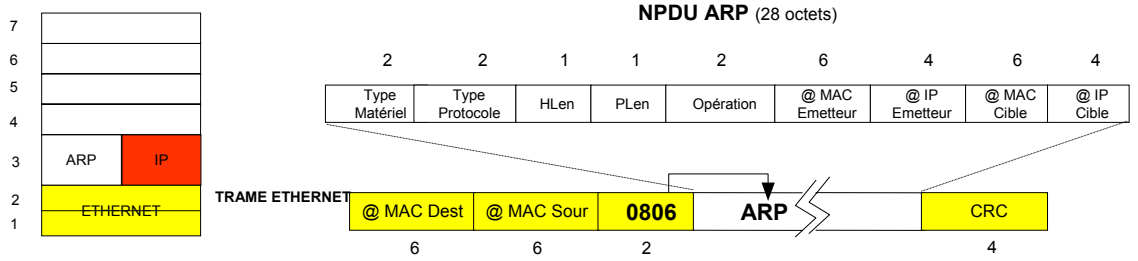
2. La machine B se reconnaît et envoie **une réponse ARP** en point à point à la machine demandeuse.

3. A mémorise **@MAC=@IP** dans une **mémoire cache interne :** Table ARP. Elle consultera cette table par la suite pour ne pas effectuer de requête ARP inutile.

Format de Trame ARP

Le protocole ARP est encapsulé par le protocole de couche 2 et pas par le protocole IP

⇨ **le protocole ARP ne peut pas être Routé**



Type Matériel : Identifie le réseau local sous jacent. 1: Ethernet, 7: Arcnet, 11: Localtalk, 16: ATM, …

Type Protocole : Identifie le protocole de couche supérieure pour lequel l'adresse MAC est recherchée (IP : 0800)

Hlen : Longueur octets de l'@ matérielle (6 pour Ethernet)

Plen : Longueur octets de l'@ de protocole de couche supérieure (4 pour IPV 4)

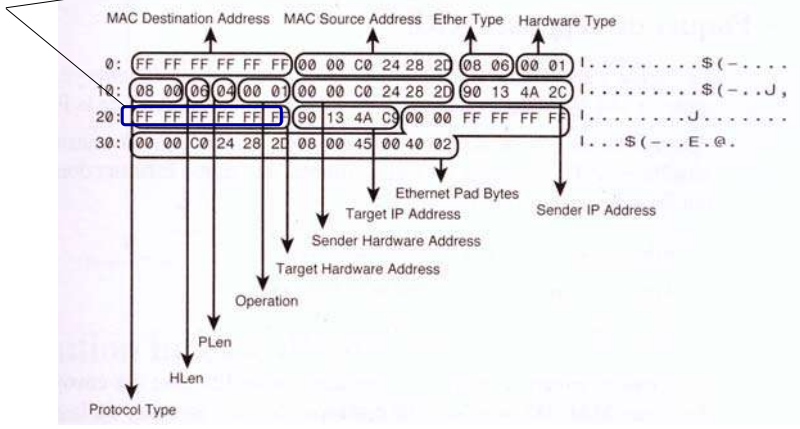
Opération : Type d'opération ARP demandée : 1: Requête ARP, 2: Réponse ARP, 10 : NAK ARP, 3 : Requête RARP, 4 : Réponse RARP, …

N.B : @MAC Cible initialisé à **FF.FF.FF.FF.FF.FF** dans une requête ARP

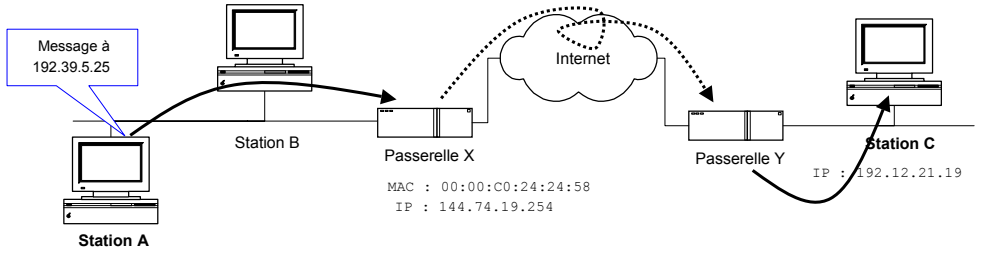
Analyse de trame ARP

```
Packet Number : 2                          6:38:38 PM
Length : 64 bytes
ether: ========== Ethernet Datalink Layer ==========
       Station:  00-00-C0-24-28-2D  ----> FF-FF-FF-FF-FF-FF
       TYPE:  0x0806 (ARP)
 arp: ========== Address Resolution Protocol ==========
       Hardware:  Ethernet
       Protocol:  08x0800  (IP)
       Operation:  ARP Request
       Hardware address length:  6
       Protocol address length:  4
       Sender Hardware Address:  00-00-C0-24-28-2D
       Sender Protocol Address:  144.19.74.44
       Target Hardware Address:  FF-FF-FF-FF-FF-FF
       Target Protocol Address:  144.19.74.201
```

Adresse MAC recherchée

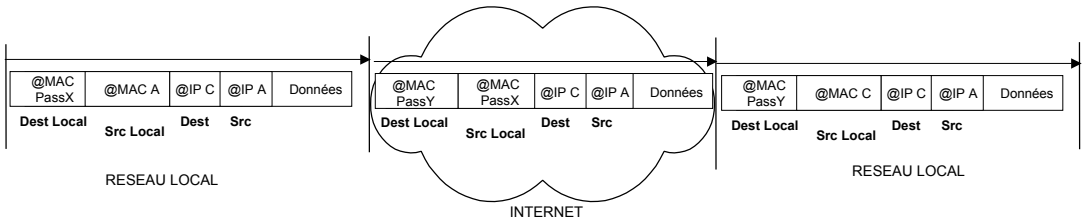


Scénario d 'échange Internet - Routage



⇨ La station source doit utiliser **une passerelle pour sortir** du réseau local.

⇨ La passerelle se charge de **router le message** vers le destinataire (Le datagramme est routé sur Internet)

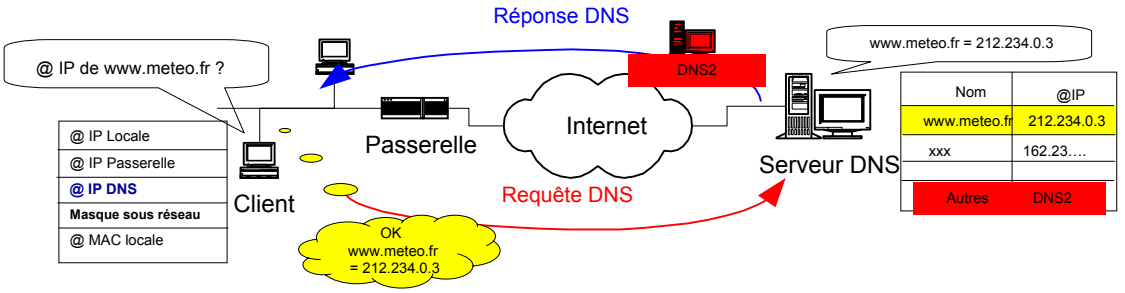


Service DNS : Domain Name Server

Les machines ou applications sur Internet sont dénommées soit par

- une adresse IP *195.220.155.17*
- un nom symbolique ou URL (Uniform Resource Locator) *www.cran.uhp-nancy.fr*

☞ Le service d'annuaire automatique est réalisé par des applications DNS (Domain Name Server)



☞ Protocole Résolution d 'adresse IP d 'un Nom Symbolique

❶ la **résolution de nom** : Le client interroge son serveur DNS pour avoir l'@IP du nom symbolique : *www.meteo.fr*

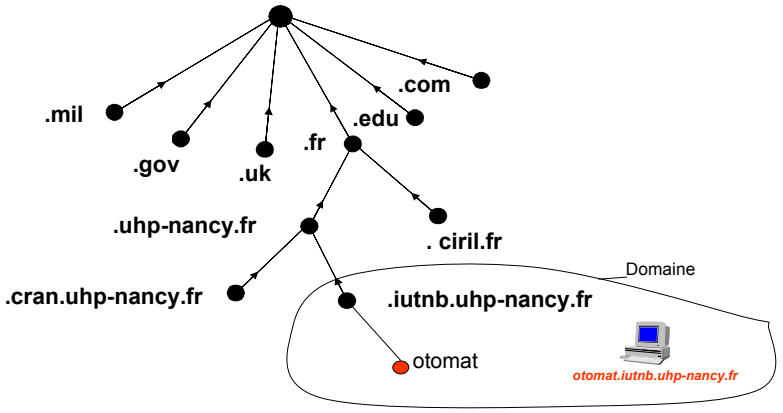
❷ le **serveur DNS interroge un autre DNS** jusqu'à ce que l'association nom de domaine ⇔ adresse IP, soit trouvée

❸ un **serveur DNS retourne l'adresse IP** du nom demandé : 212.234.0.3

❹ le **client peut maintenant contacter le destinataire** par son adresse IP : *212.234.0.3*

Domaines et Nommage des machines

Le nom d'une machine est : ***Machine . Sous-Domaine . Domaine***



Lorsqu'un client a besoin d'un nom ou d'une adresse (Nom ⇔ Adresse IP), il s'adresse au serveur DNS qui gère son sous-domaine.

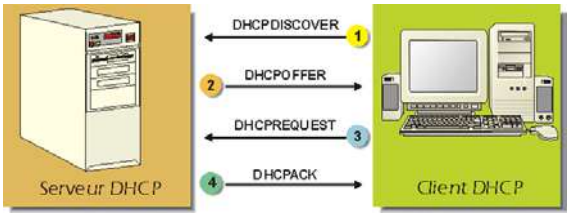
Le serveur DNS relaie la demande au niveau supérieur s'il ne connaît pas le nom recherché (arborescence)

DHCP : Dynamic Host Control Protocol (RFC 2131)

Attribution dynamique d'une adresse IP à un poste qui se connecte au réseau.

Le serveur DHCP fournit également d'autres informations, comme la passerelle par défaut et le DNS.

Ce système, évite à l'utilisateur d'avoir à configurer manuellement sa pile IP; Il permet également à l'administrateur du réseau de modifier son architecture sans avoir à prévenir tous ses clients.



1. Lorsque le client DHCP démarre, Il envoie une trame broadcast, "DHCPDISCOVER", destinée à trouver un serveur DHCP. N'ayant pas encore d'adresse IP, il adopte provisoirement l'adresse 0.0.0.0. et fournit sa "MAC Address".
2. Le, ou les serveurs DHCP du réseau qui vont recevoir cette trame vont répondre par un "DHCPOFFER". Cette trame contient une proposition de bail et la "MAC Address" du client, avec également l'adresse IP du serveur. Tous les DHCP répondent et le client normalement accepte la première réponse venue.
   Le "DHCPOFFER" sera un broadcast (Ethernet) ou non, suivant le serveur DHCP utilisé.
3. Le client répond alors par un DHCPREQUEST à tous les serveurs (donc toujours en "Broadcast") pour indiquer quelle offre il accepte.
4. Le serveur DHCP concerné répond définitivement par un DHCPACK qui constitue une confirmation du bail. L'adresse du client est alors marquée comme utilisée et ne sera plus proposée à un autre client pour toute la durée du bail

Protocole DHCP : Dynamic Host Control Protocol (2)

En-têtes de trames.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x6719436e
2	0.001182	192.168.0.253	192.168.0.9	ICMP	Echo (ping) request
3	0.342454	192.168.0.253	192.168.0.9	DHCP	DHCP Offer - Transaction ID 0x6719436e
4	0.344405	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x6719436e
5	0.348264	192.168.0.253	192.168.0.9	DHCP	DHCP ACK - Transaction ID 0x6719436e
6	0.353014	CIS_b9:49:37	Broadcast	ARP	Who has 192.168.0.9? Tell 192.168.0.9
7	0.571241	CIS_b9:49:37	Broadcast	ARP	Who has 192.168.0.9? Tell 192.168.0.9
8	1.571441	CIS_b9:49:37	Broadcast	ARP	Who has 192.168.0.9? Tell 192.168.0.9

1. Le client DHCP démarre, il n'a pas d'IP et utilise 0.0.0.0 pour faire un "broadcast général (255.255.255.255)". C'est le DHCP Discover.
2. Notre serveur DHCP, qui a l'intention d'offrir à ce client l'IP 192.168.0.9, fait un ping sur cette adresse, histoire de voir si elle est réellement disponible sur le réseau.
3. Comme il ne reçoit pas de réponse à son ping, il offre cette adresse au client.
4. Le client fait alors un DHCP Request
5. Le serveur accepte
6. Le client fait un broadcast ARP pour vérifier de son côté que l'IP 192.168.0.9 n'est pas dupliquée sur le réseau.
7. idem
8. idem

Protocole IP

Internetwork Protocol RFC 791

Protocole IP : Internetwork Protocol

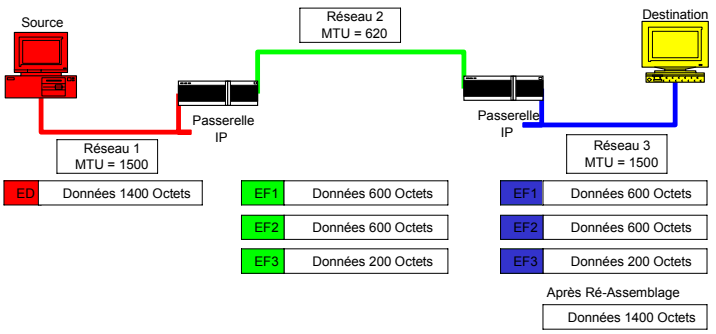
Le protocole **IP** (RFC 792) a été conçu pour s'adapter à des types de réseaux différents.
La taille maxi. des données admissibles par la couche 2 d'un réseau varie selon la nature du réseau

⌦ IP s'adapte par un mécanisme de **fragmentation** de paquets.

Maximum Transfert Unit (**MTU**) : Taille maximale d'un datagramme sur une machine IP, passerelle ou routeur.

Ethernet : MTU = 1500 octets **X25** : MTU = 1007 octets **Token Ring** : MTU = 4440 à 17940 octets f(temps conservation jeton) **ATM** : MTU = 32

IP est chargé d'adapter la taille des paquets de données pour s'adapter aux MTU des différents réseaux traversés, en découpant le paquet en paquets plus petits : c'est un rôle majeur des routeurs sur Internet appelé **fragmentation**



Le **ré-assemblage** est effectué par la couche IP du destinataire **jamais par les routeurs**.

Routage IP

Le routage est le processus permettant à un **datagramme** d'être acheminé vers son destinataire quand celui-ci n'est pas sur le même réseau physique que l'émetteur.

Le chemin parcouru est le résultat du **processus de routage** qui effectue les choix nécessaires afin d'acheminer le datagramme.

Les routeurs forment une structure coopérative de telle manière qu'un datagramme transite de routeur en routeur, jusqu'à ce que l'un d'entre eux le délivre à son destinataire.

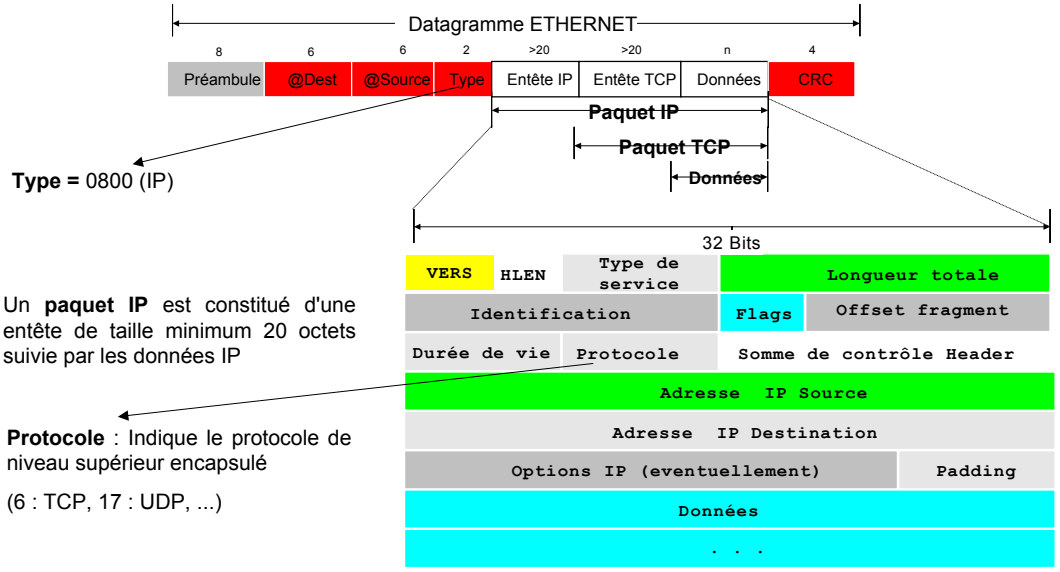
Un **routeur ne connaît jamais le chemin complet** pour atteindre la destination.

Le routage repose sur une **table de routage IP**, présente sur le routeur, indiquant la manière d'atteindre les destinations

Destination	Prochain Routeur	Interface
183.27.0.0	Connexion directe	P1
172.9.0.0	Connexion directe	P2
137.5.0.0	183.27.2.5	P1
140.4.0.0	172.9.1.2	P2
0.0.0.0	183.27.2.5	P1 (par défaut)

Format de l'Entête IP

Le protocole IP encapsule les données générées par les couches 7 à 4 pour former **un paquet IP**.



Un **paquet IP** est constitué d'une entête de taille minimum 20 octets suivie par les données IP

Protocole : Indique le protocole de niveau supérieur encapsulé

(6 : TCP, 17 : UDP, ...)

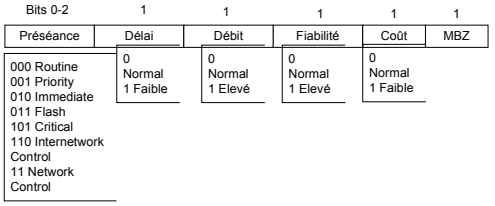
Champs de l ’Entête IP

VERS : Version IP (4 : IPV4, 6 : IPV6 IPng)

HLEN : Longueur de l'entête IP en mots de 32 bits

Type de Service : TOS : Qualité de service demandée
Longueur Totale : Long. de l'entête et des données IP en octets.

Identification : Numéro du datagramme IP.



Flags : DF =1 : Le Datagramme ne doit pas être fragmenté
MF = 1 le destinataire est informé que d'autres fragments vont arriver

Offset Fragment : Code sur 13 bits l'offset des données contenues dans le datagramme IP depuis le fragment originel.

Durée de Vie : Compteur de sauts (**hops**) décrémenté à la traversée de chaque routeur. Arrivé à 0 le routeur stoppe le paquet et renvoi un message ICMP. Limite la durée de vie des datagrammes sur Internet.

Protocole : Indique le protocole de couche supérieure (6 : TCP, 17 : UDP, 1 : ICMP,….)

Somme de contrôle header : Checksum complémenté à deux de tous les mots de 16 bits de l'entête sans les données IP et sauf le checksum lui-même. (Recalculé à chaque routeur car TTL change)

Adresses IP Source et Destination

Options : Indique le niveau de sécurité, information de routage, d'horodatage : rarement utilisées.

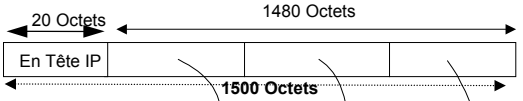
Padding : Sert à compléter l'entête IP en nombre de mots de 32 pleins.

Données : Données encapsulées à destination du protocole de niveau supérieur.

Fragmentation d'un datagramme IP

Dans un routeur, la couche IP peut se charger de fragmenter un datagramme afin de s'adapter à la longueur maximale des trames (MTU) du réseau qu'elle dessert.

Processus de fragmentation

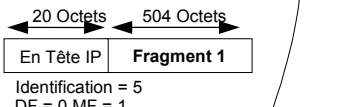


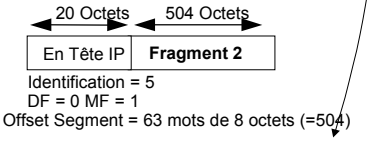
⇨ Les fragments appartiennent au même N° de paquet original : **Identification**.

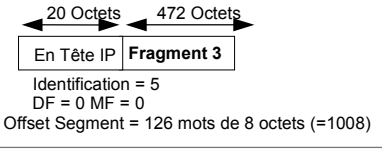
Si MTU = 524

⇨ Premier Fragment : **DF** = 0 et **MF** = 1

⇨ Dernier Fragment : **MF** = 0 (No More Fragment).







⇨Le réassemblage dans l’ordre sera effectué sur la base des **offset de fragment**.

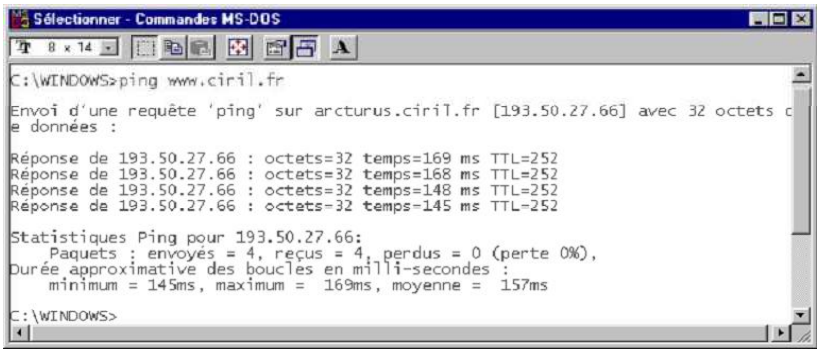
La commande PING

La commande **ping** (Packet Internet Groper) permet de vérifier la connexion d'un ou plusieurs ordinateurs distants.

• Envoi des paquets d'écho selon le protocole **ICMP** (Internet Control Message Protocol) vers l '@ cible.

• Attente des réponses, jusqu'à 1 seconde pour chaque paquet.

• Vérification des paquets reçus (32 octets de données ASCII 'a' à 'z' …) , et mesure du temps d'attente.

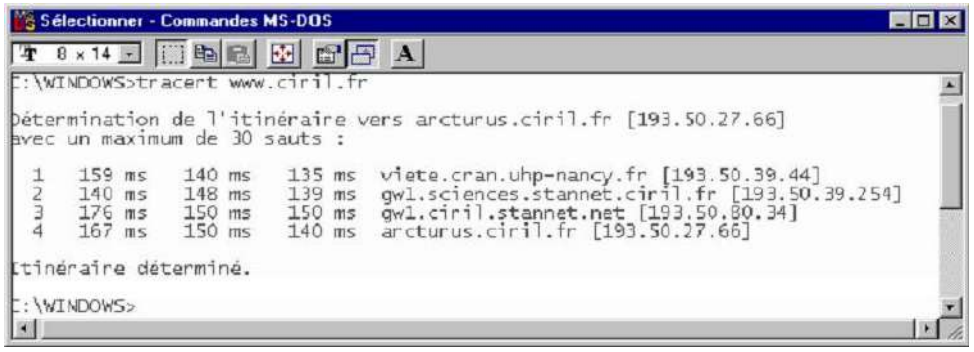


La commande Tracert

La commande **tracert** (TRACE RouTer) permet de connaître tous les routeurs traversés pour accéder à une station d '@IP donnée.

• Exécute un ping vers une station distante avec un paramètre interprété par chaque routeur traversé lui signifiant de répondre par son adresse.

• Attente des réponses et mesure du temps d'attente.



Protocole TCP

Transmission Control Protocol RFC 792

Protocole TCP (RFC 792)

☞ **Transmission Control Protocol** est un protocole de transport fiable:

- ➔ fiabilité de transmission de paquet de bout en bout : chaque octet transféré est acquitté
- ➔ transferts tamponnés : découpage en segments
- ➔ connexions bidirectionnelles et simultanées
- ➔ garantie de non perte de messages ainsi que de l'ordonnancement des paquets
- ➔ service en mode connecté

☞ TCP est un protocole de transport **orienté connexion**.

Une connexion de type circuit virtuel est établie avant que les données puissent être échangées :

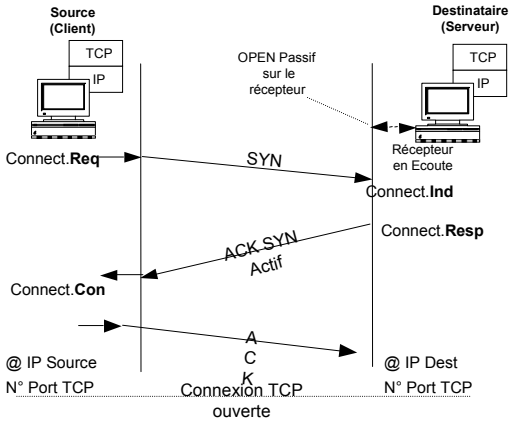
Ouverture connexion avec Négociation + Transferts + **Fermeture Connexion**

☞ Les services fournis par la couche TCP sont :

- **CONNECT** : ouverture d'une connexion
- **CLOSE** : fermeture d'une connexion
- **SEND** : envoyer les données sur une connexion ouverte
- **RECV** : recevoir les données sur une connexion ouverte
- **STATUS** : fournir des renseignements sur une connexion ouverte

Ouverture d'une connexion TCP

Une connexion TCP est établie en 3 temps de manière à assurer la synchronisation entre Client et Serveur



Le serveur se met en écoute : OPEN passif.
La couche TCP serveur se met en attente de connexions.

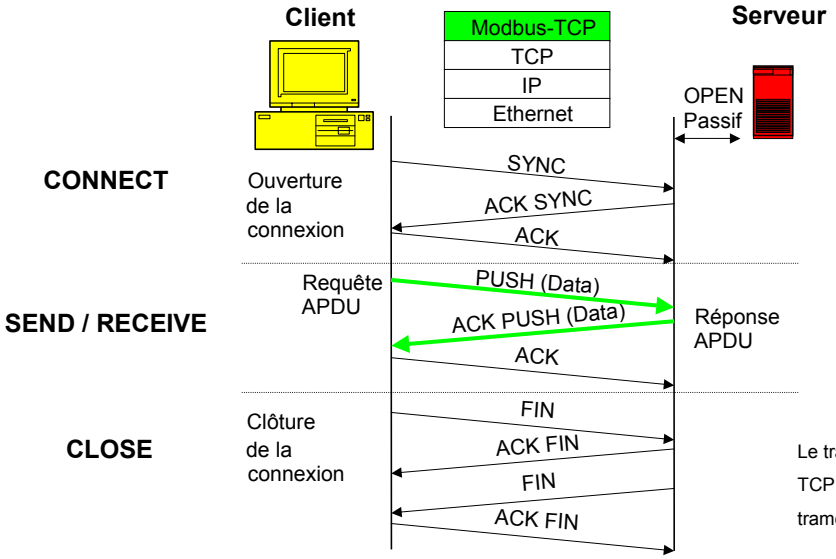
Un client fait une demande de connexion : OPEN actif.
La couche TCP cliente spécifie les ***sockets*** source et destination liées à cette connexion.
Le Serveur Accepte ou Refuse la connexion (Nb limité)

Une connexion est établie entre l'émetteur et le destinataire caractérisés chacun par un couple **(@ IP, N° port TCP**), appelé une **SOCKET**

Emetteur	**Destinataire**
(@ IP, port TCP) **(124.32.12.1 , 1512)**	(@ IP, port TCP) **(19.24.67.2 , 1538)**

Gestion d'une connexion TCP

L 'envoi de requêtes APDU est réalisé après **ouverture préalable d 'une connexion**.



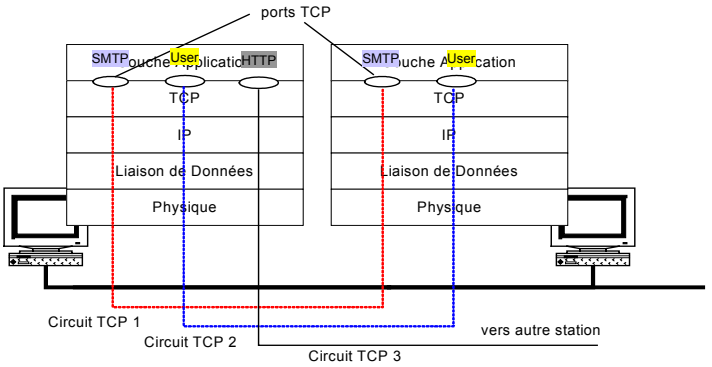
Le transport de données avec TCP est très fiable car chaque trame est acquittée.

Multiplexage TCP

☞ TCP peut servir plusieurs applications en même temps sur la même machine source :

⇨ Principe de multiplexage.

☞ TCP associe un numéro de port à chaque application qui utilise ses services.



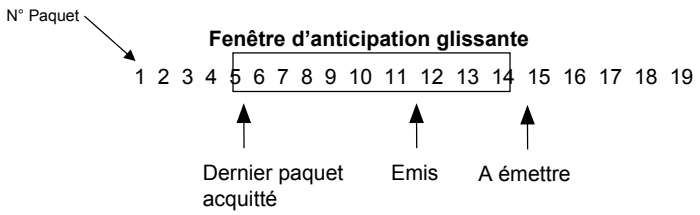
N° Port TCP réservés : 20, 21 : FTP, 23 : TELNET, 25 : SMTP, 80 : HTTP, 110 : POP3, 502 : MODBUS-TCP…

N° Port libres utilisateur > 1023

Contrôle de flux sous TCP

TCP contrôle le flux des paquets émis par l'intermédiaire d'acquittement de paquets.

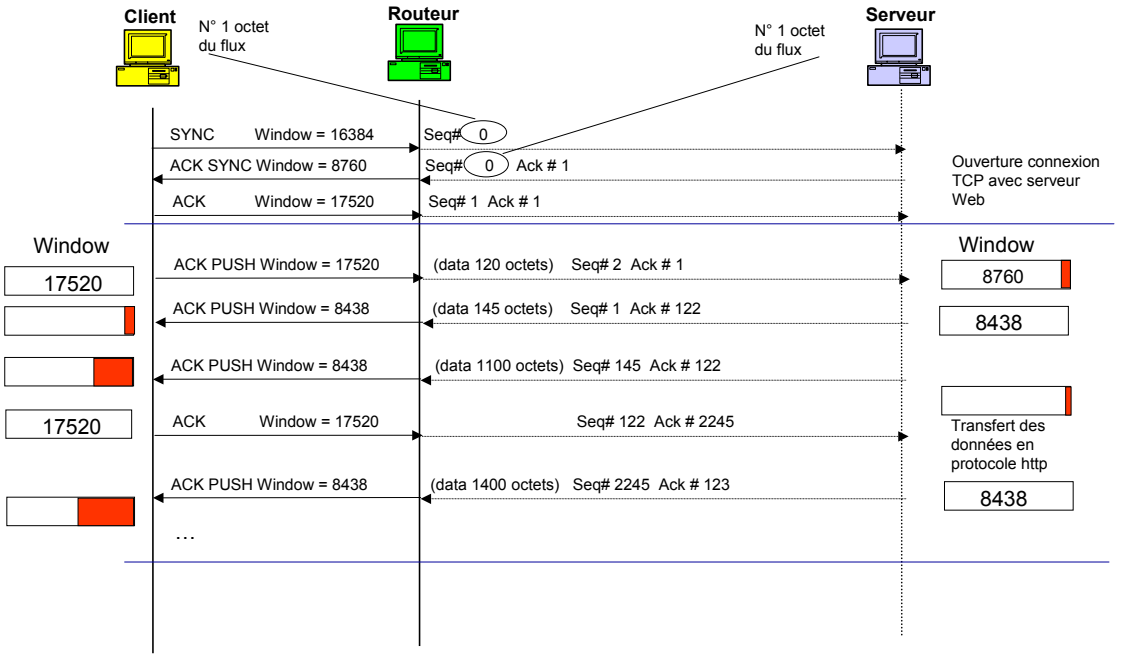
La procédure choisie est l'anticipation des acquittements sur une fenêtre glissante d'anticipation



☞ La **taille de la fenêtre est variable** afin d'ajuster le débit de l'émetteur à celui du récepteur.

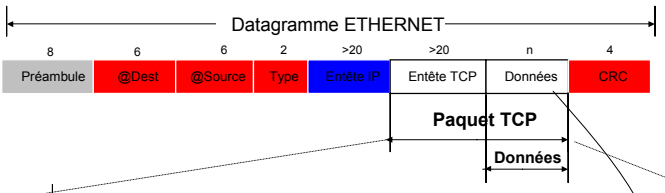
☞ Elle est **négociée lors de l'ouverture de la connexion** et **ré-ajustable** par l'émetteur à tout moment de la communication :
Pour cela, les accusés des paquets contiennent un champ (window) spécifiant la taille de la fenêtre de réception en nombre d'octets libres (et non pas en paquets).

Acquittement TCP par Numérotation des Octets



Format de l ’entête TCP

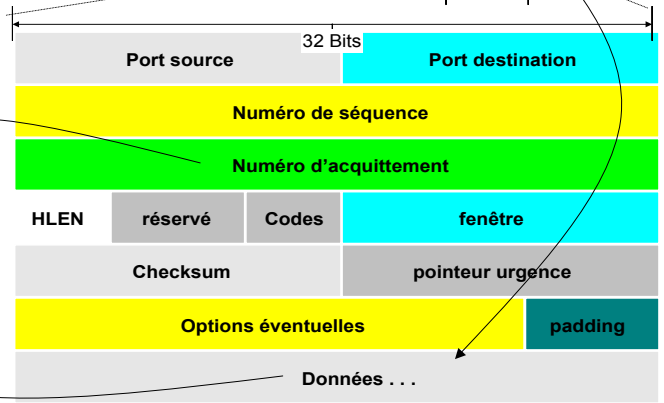
Le protocole TCP encapsule les données générées par les couches 7 à 5 pour former **un paquet TCP**



Un paquet TCP est constituée d'une entête de 20 octets minimum suivie par des données TCP

Chaque paquet TCP sera acquitté assurant un transport fiable de bout en bout

Les Données TCP sont constituées de l’APDU de la couche supérieure



Champs d ’entête TCP

L'entête TCP à une longueur minimale de 20 Octets, qui peut augmenter avec les options.

Port Source, Port Destination : ports en connexion des applications origine et destination.

Numéro de séquence : Le numéro du premier octet est le numéro de séquence.

Numéro d'acquittement : prochain numéro de séquence attendu par l’émetteur de cet acquittement.
Acquitte implicitement les datagrammes reçus avec les octets précédents.

HLEN : Code sur 4 bits la longueur de l'entête TCP en mots de 32 bits

Codes : 6 bits de drapeau

URG	: Indique l'envoi des données en urgence sans attente d'acquittement
ACK	: Indique que le champ "Numéro d'acquittement est valide"
PSH	: Indique la remise immédiate des données à la couche supérieure
RST	: Demande de ré-initialisation de la connexion
SYN	: Indique l'ouverture d'une connexion de circuit virtuel
FIN	: Indique la fermeture de la connexion

Fenêtre : Taille de la fenêtre d'anticipation.

Checksum : Somme complémentée à 1 des compléments à 1 de tous les mots de 16 bits du paquet TCP

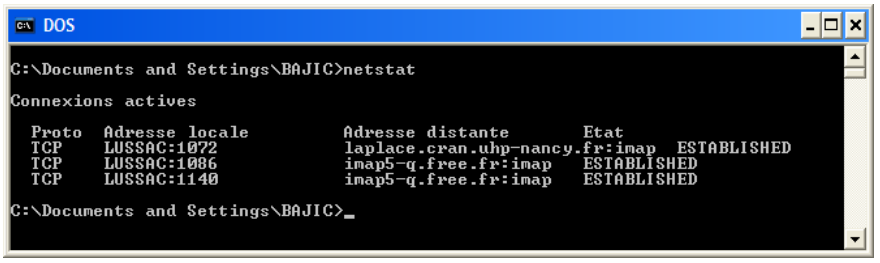
Pointeur d'urgence
Options : Permet de négocier la taille maximale des segments échangés. TCP calcule une taille maximale de segment de manière à ce que le datagramme IP résultant corresponde au MTU du réseau. La recommandation est de 536 octets.

Padding : Sert à complété en nombre de mots de 32 pleins l'entête TCP.

Données : Données encapsulées à destination du protocole de niveau supérieur.

Ports TCP ouverts

La commande NETSTAT.EXE permet de connaître les ports utilisés par la couche TCP



Protocole UDP

User Datagram Protocol RFC 768

Protocole UDP *(User Datagram Protocol RFC 768)*

- protocole de transport **non orienté connexion** donc moins robuste et moins fiable que TCP
- UDP effectue le multiplexage des processus d'application source et destination par des N° **Port UDP**
- UDP **ne séquence pas les données** : ⇨ Pas de fragmentation
- Dans la plupart des implémentations TCP-IP, la partie UDP est incorporée au module TCP

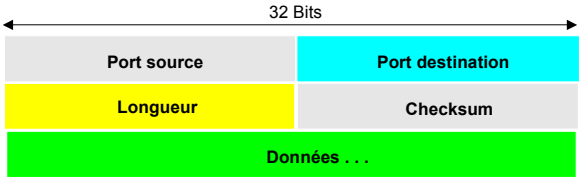
Caractéristiques d'utilisation de UDP

Malgré ces inconvénients face à TCP, UDP est utilisé par les applications DNS (Domain Name Server), SNMP (Simple Network Management Protocol), RIP (Routing Information Protocol), NFS (Network File System V2) :

- UDP convient bien à un environnement de réseau local.
- UDP est utilisé pour des applications de commande/réponse tenant dans un seul datagramme.
- UDP est simple de mise en œuvre et plus rapide que TCP car il n'y a pas de connexion à gérer.

Entête UDP

L'entête UDP, très simple, est de taille fixe de 8 octets



Port Source, Port Destination : Indiquent les N° port des applications émetteur et destinataire sur 16 bits .

Longueur : Longueur totale du paquet UDP, comprenant l'entête de huit octets plus les données UDP.

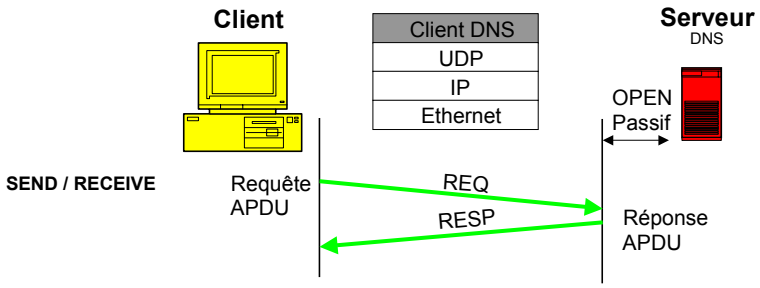
Checksum : Somme complémentée à 1 des compléments à 1 de tous les mots de 16 bits du paquet UDP, avec un remplissage de zéros pour obtenir une longueur multiple de deux octets, auquel est ajoutée une pseudo-entête telle que ci-dessous

Requête APDU en Mode non connecté UDP

L 'échange est de type Question/Réponse.

⇨ Le transport de bout en bout n'est pas fiable car la réception d'une trame n 'est jamais acquittée.

⇨ Seule la réponse est transmise (Time Out)



Les échanges UDP sont très rapides comparés à TCP :

⇨ **Pas de connexion**

⇨ **Pas de fragmentation de paquets**

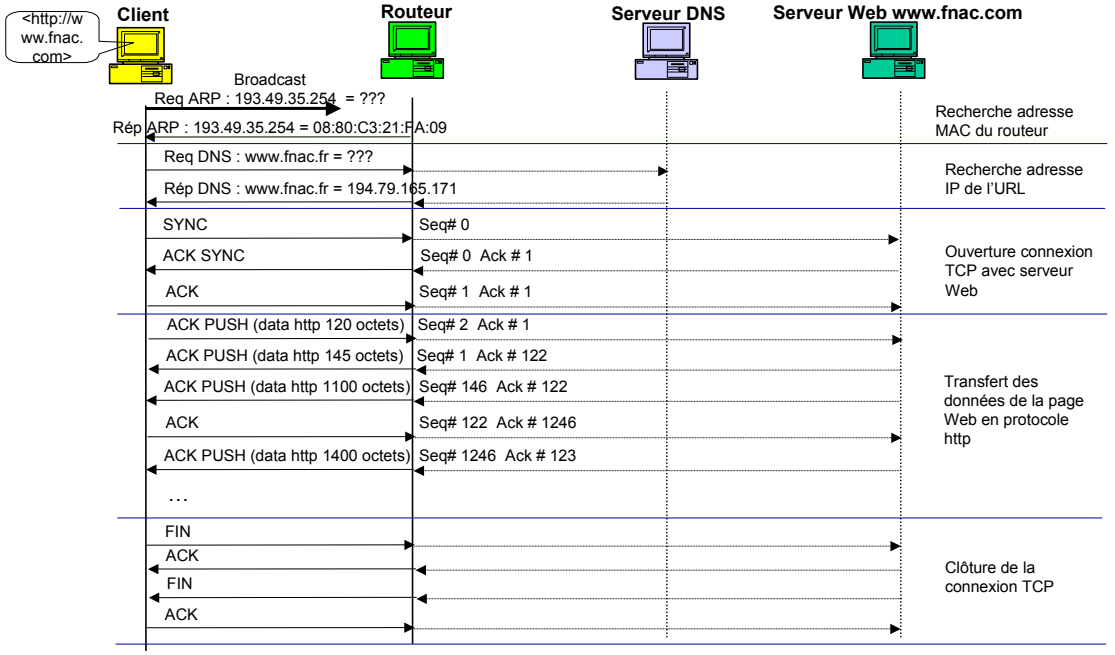
Analyse de trame : Requête DNS

Accès à www.radio-france.com depuis Netscape

```
From SRLI6 To Rtr:00d0d3:28d820   0800 == IP
- - - - - - - - - - - - - - - - - - - -
Internet Protocol
Datagram Length = 66 octets.
Version = 04
IP Precedence-Routine:Normal Delay,Normal Throughput,
Low Reliability
Identifier = 348f (hex)
Last fragment. - Fragment offset = 0 (Octets)
Time to live = 128 hops
Protocol = 17 == UDP
Header Checksum = 13691 : Checksum is Correct
Source Address - 193. 49. 35. 56
Destination Address - 193. 54. 43.  1
- - - - - - - - - - - - - - - - - - -
UDP
Source Port = 3361  Unknown
Destination Port = 53 Domain name server
Data Length = 46 - Checksum = 35523
- - - - - - - - - - - - - - - - - - -
DNS
QUERY MESSAGE
Identifier  2 : Code  0 == Standard Query
Authoratitive Answer Bit is NOT Set : Message was NOT
Truncated
Recursion Desired Bit is Set : Recursion Available Bit
is NOT Set
Error Status  0 == No Error Condition
Number of Questions  1
Number of Resource Records in Answer Section  0
Number of Name Server Resource Records in Authority
Records section  0
Number of Resource Records in Additional Records
section  0
Domain Name is  www.radio-france.com.
Type  1 == A Host Address
Class  1 == the Internet
- - - - - - - - - - - - - - - - - - -
```

```
From Rtr:00d0d3:28d820   To SRLI6 0800 == IP
- - - - - - - - - - - - - - - - - - - -
Internet Protocol
Datagram Length = 223 octets.
Version = 04
IP Precedence - Routine : Normal Delay, Normal Throughput
Low Reliability
Identifier = b23b (hex)
Do not fragment...
Last fragment. - Fragment offset = 0 (Octets)
Time to live = 254 hops
Protocol = 17 == UDP
Header Checksum = 63792 : Checksum is Correct
Source Address - 193. 54. 43.  1
Destination Address - 193. 49. 35. 56
- - - - - - - - - - - - - - - - - - -
UDP
Source Port = 53  Domain name server
Destination Port = 3361 Unknown
Data Length = 203 - Checksum = 39378
- - - - - - - - - - - - - - - - - - -
DNS
RESPONSE MESSAGE
Identifier  2 : Code  0 == Standard Query
Authoratitive Answer Bit is NOT Set : Message was
NOT Truncated
Recursion Desired Bit is Set : Recursion Available Bit
is Set
Error Status  0 == No Error Condition
Number of Questions  1
Number of Resource Records in Answer Section  3
Number of Name Server Resource Records in Authority Reco
section  3
Number of Resource Records in Additional Records
section  0
Domain Name is  www.radio-france.com.
Type  1 == A Host Address
Class  1 == the Internet
- - - - - - - - - - - - - - - - - - -
Offset from Start of Packet
0050    03 77 77 77 0C 72 61 64 69 6F 2D 66 72 61 6E 63
.www.radio-france.com
0060    65 03 63 6F
- - - - - - - - - - - - - - - - - - -
```


Datagrammes TCP-IP d'une navigation Internet

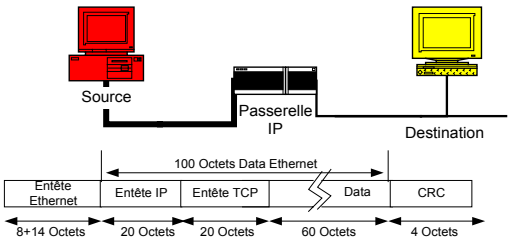


Rendement de Transmission

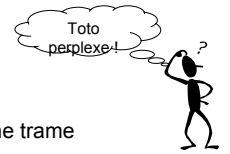
Une application informatique doit transmettre 60 octets de données APDU (Nom, prénom, adresse par exemple) vers une application destinataire sur Internet. En disposant de 2 réseaux Ethernet à 10 MBps, interconnectés par un routeur IP, quel va être le temps de transmission ?

« Je sais Maîtresse !
$T \approx 60 * 8 * 10^{-7}$ s !?
C 'est tout bête ! " dit Toto sans lever son doigt !

"Non Toto! " répond la maîtresse, "C'est bien raisonné !, mais tu supposes disposer d'un rendement de transmission de 100 %, ce qui n'est pas le cas mon p 'tit Toto et I 'on va voir pourquoi …."



1) Rendement Ethernet



à 10 MBps , 1Tbit = 10^{-7} s

Ninfo = 100 * 8 bits

➔ Tps Transmission d'une trame

T = ((8+14)*8+100*8+4*8) Tbits = 100,8 µs

➔ Temps Inter-trame Ethernet

Tig= 96 Tbits = 9,6 µs

➔ Taux de Transfert des Infos

TTI = Ninfo / (T + Tig) = 7,25 MBps ⇨ **72 %**

2) Rendement IP

Ninfo = 640 bits (100-20 octets entête IP)

➔ TTI = 640 / ((T + Tig) = 6 Mbps = 5.8 MBps

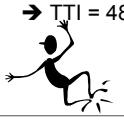
⇨ **60 %**

3) Rendement TCP

Ninfo = 480 bits (80-20 octets entête IP)

➔ TTI = 480 / ((T + Tig) = 4,35 Mbps

⇨ **43 %**



⇨ **RENDEMENT APDU 43%**

Modbus-TCP

Ethernet Industriel

Modbus-TCP

Selon Spécification Schneider Mars 1999

(www.modbus.org)

Pourquoi un Ethernet Industriel ?

Pourquoi Ethernet ?

Avantages

Standard Mondial, Rapide, Peu coûteux, Banalisé Multi-usage

Protocole TCP-IP et au dessus : HTTP (Web), SMTP (Mail), FTP (Fichiers) …

Inconvénients

Non déterministe, Non temps réel

Pour un Ethernet Industriel

*Fin des Domaines de Collision : **Commutation, Full Duplex** (Switch, ASIC switch)*

*Des Protocoles d'application Industriel : **Bataille pour un standard***

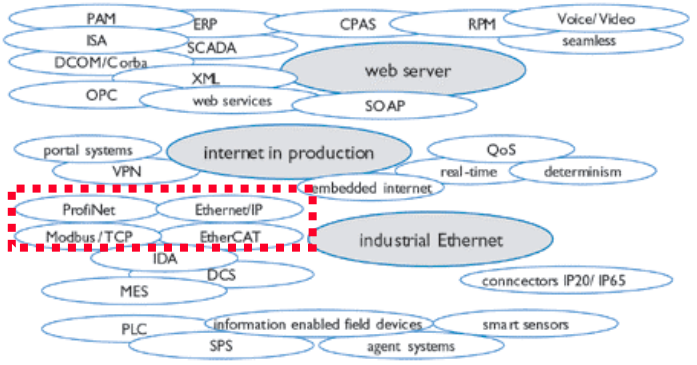
Protocoles Ethernet Industriels

SIEMENS : PROFINET

ROCKWELL : ETHERNET / IP

SCHNEIDER : MODBUS-TCP

BECKHOFF : ETHERCAT



Assoc promotion Ethernet dans l'automatisation www.iona.org

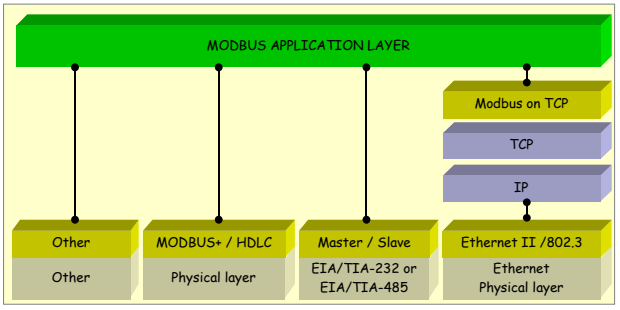
http://ethernet.industrial-networking.com

Stacks d'Implémentation de Modbus

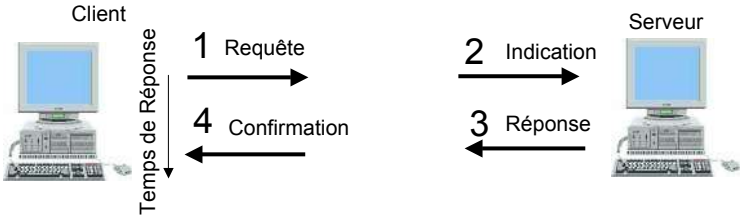
MODBUS est avant tout un Protocole d'Application, donc un ensemble de règles d'échange et de dialogue entre des équipements, et non pas réellement un réseau. Mais par abus de langage, un système de communication mettant en œuvre le protocole Modbus est couramment appelé "réseau Modbus".

Plusieurs implémentations de modbus existent :

- Modbus liaison série dit Modbus RTU (ASCII)
- Modbus Plus
- Modbus TCP



Modèle Client - Serveur



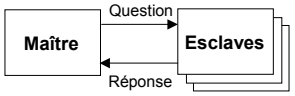
Serveur : Est à l'écoute des clients qui le sollicitent
Ne répond qu'à leurs requêtes

Client : Sollicite un service du Serveur

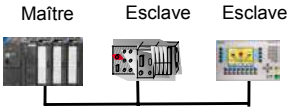
Temps : Le temps de réponse est indéterminé, il dépend de l'architecture réseau traversée

Modèle Maître / Esclave Vs Client / Serveur

Modèle Maître / Esclave (MODBUS-RTU)



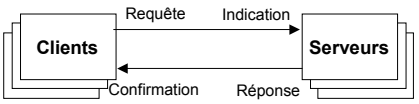
- Maître unique
- Plusieurs Esclaves (@1 à @63)
- Transactions séquentielles



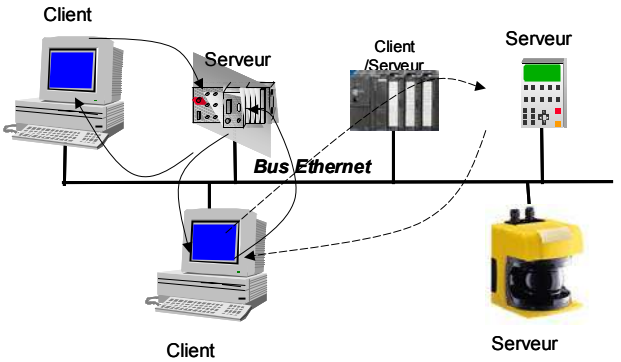
Bus RS 485

à 19 200 Bps ⇨ 1 transaction ≈ 100 ms

Modèle Client / Serveur (MODBUS-TCP)



- Clients Multiples (@IP)
- Serveurs Multiples (@IP)
- Transactions Simultanées



à 10 MBps ⇨ 1 transaction ≈ 1 ms

Performances

Les performances dépendent du réseau et du matériel :

MODBUS RTU liaison série :
0,6 ms / caractère (11 Bits à 19200 Bits/s) , Lecture 10 Mots # 48 ms (transaction réseau)
⇨ Soit 208 Mots par seconde (Supervision, commande processus lent)

MODBUS sous TCP/IP sur Intranet :
Ethernet 10 Base T offre un débit # 1.25 Mbytes/sec
Lecture 10 Mots , rendement encapsulation de 12/66 (Req) et 24/78 (Conf) # 25 % (transaction réseau)
⇨ Soit 1.25M / 2 * 25% = 156 250 Mots par seconde (Supervision, commande processus)

MODBUS sous TCP/IP Via Internet :
Temps de réponse lié à la charge du réseau Internet
Dizaine de millisecondes à plusieurs centaines de millisecondes (supervision, consultation, … à distance)

Ce sont des performances idéales au maximum des bandes passantes :

⇨ Il fait tenir compte des temps de cylce des équipements API, …

Test effectué par Schneider Automation, MOMENTUM Ethernet PLC, 4000 nœuds interrogés par secondes avec chacun 16 voies analogiques 12-bit ou 32 entrées TOR : 4 nœuds scannés / milliseconde

Le Stack MODBUS-TCP sur Ethernet

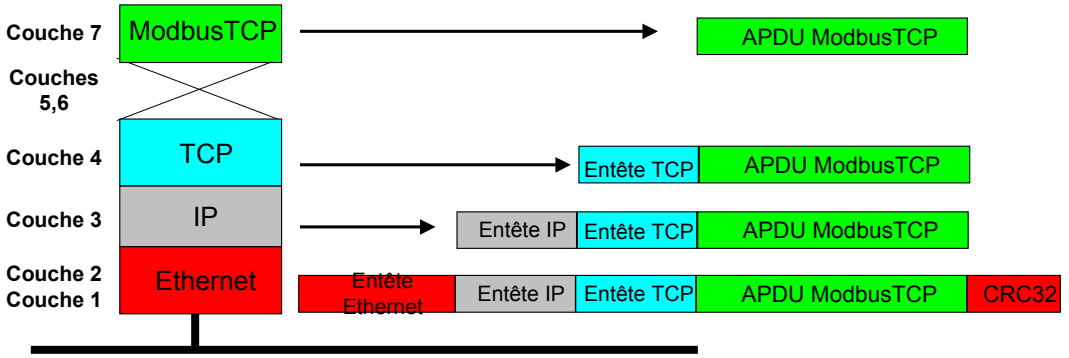
C'est une **couche application Modbus** sur un **réseau Ethernet/TCP-IP** pour communiquer dans une architecture d'automatisme industriel, proposée par **Schneider Electric** (Mars 1999)

.

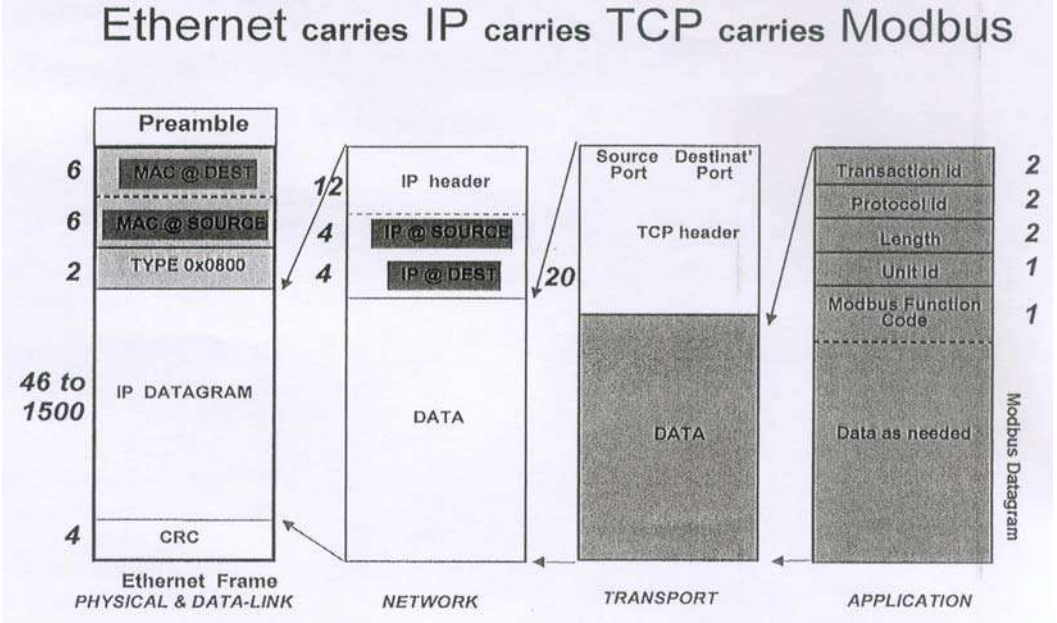
⇨ Bloc d 'Entrées/Sorties déportées, API, ….. sur Ethernet

La trame bien connue du protocole Modbus est partiellement réutilisée et complétée d'une entête spécifique.

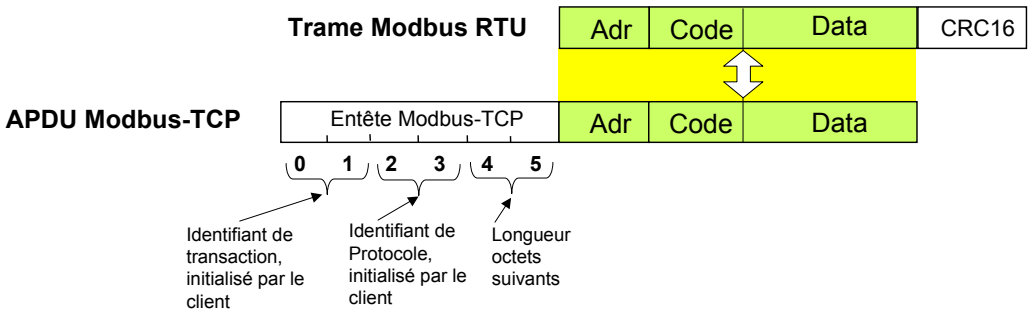
Le **CRC16 Modbus est supprimé** car le protocole TCP intègre un CRC32, IP aussi et Ethernet aussi !!!



Encapsulation Protocole Modbus-TCP



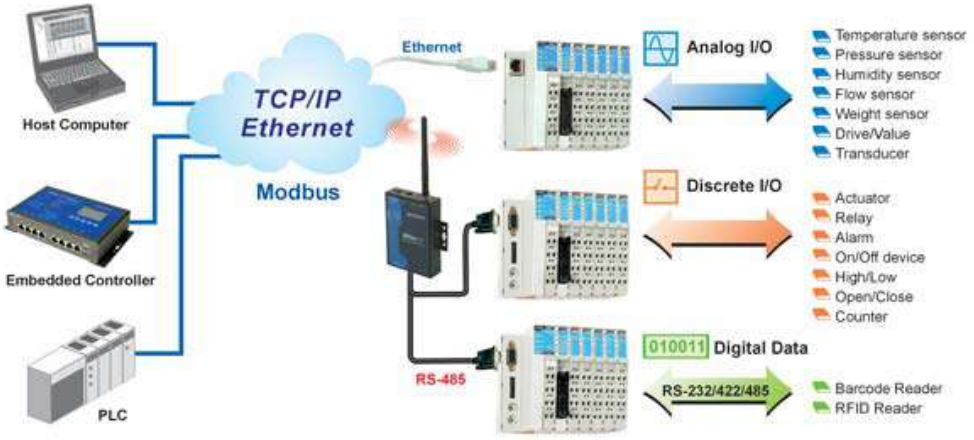
Format de l'APDU de MODBUS-TCP



Identifiant de transaction : Valeur positionnée par le client afin de repérer les réponses retournées par le serveur qui recopie la valeur

Identifiant de Protocole : Valeur positionnée par le client (Modbus = 0)

Les E/S de terrain sur Ethernet



http://www.moxa.com/product/iologik-4000.htm#top

Modbus sur Internet TCP-IP

Scénario d'une requête MODBUS routée sur Internet

L'APDU Modbus-TCP est encapsulée dans une trame Ethernet-TCP-IP.

Trame Ethernet

6	6	2	>20	>20	n	4
@Dest	@Source	0800	Entête IP	Entête TCP	APDU Modbus TCP	CRC

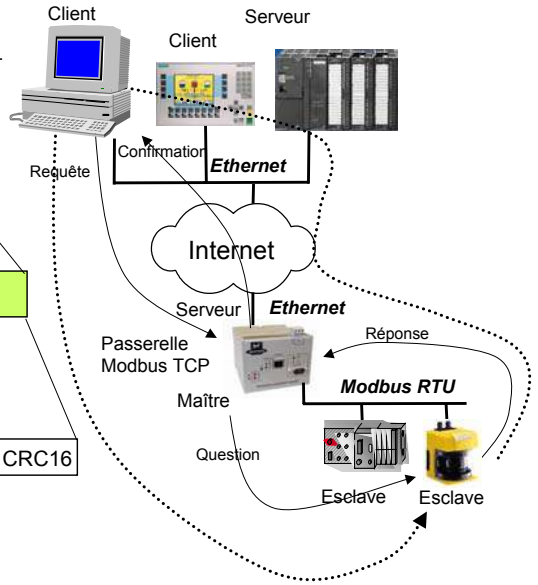
La passerelle (Serveur) Modbus-TCP extrait l'APDU

Entête Modbus-TCP	Adr	Code	Data

La passerelle (Serveur) Modbus-TCP envoie une question Modbus à l'esclave adressé.

Trame Modbus

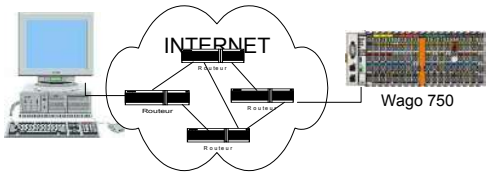
Adr	Code	Data	CRC16



Scénario d'échange MODBUS-TCP (Port 502)

Une application Serveur Modbus TCP est en écoute sur le port 502.

Une application cliente ouvre une connexion sur un port local > 1024 avec le serveur identifié par sa socket (@IP, 502)



Station @IP 1

Client

Port TCP **n** > 1024

Serveur

*Port TCP = **502***

Connexion (@IP1,n, @IP2, 502)

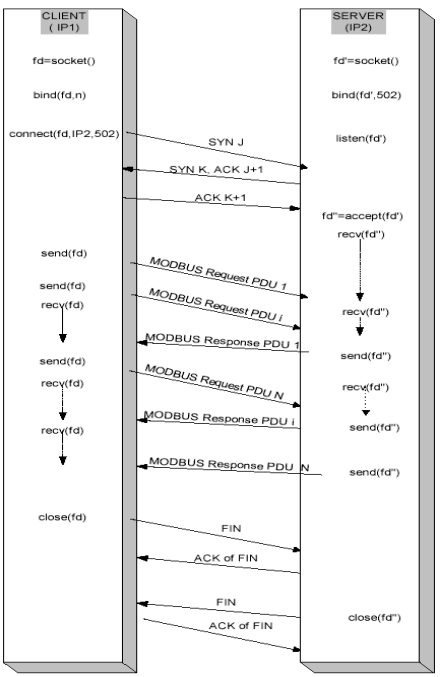
Station @IP 2

Client

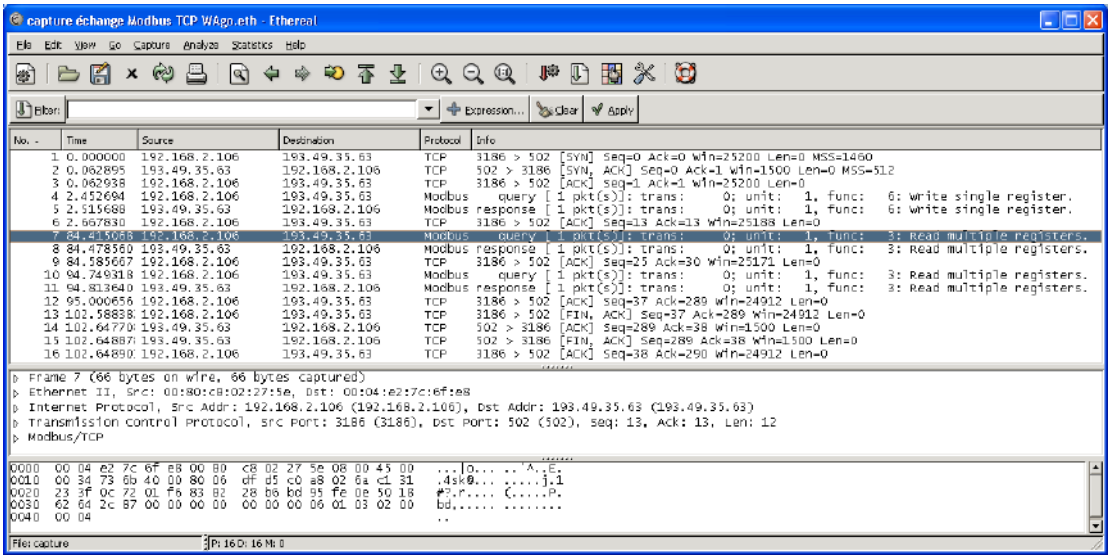
Port TCP m > 1024

Serveur

Port TCP = **502**

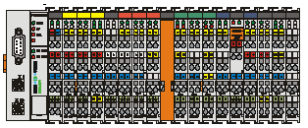


Capture Ethernet : Connexion Modbus TCP

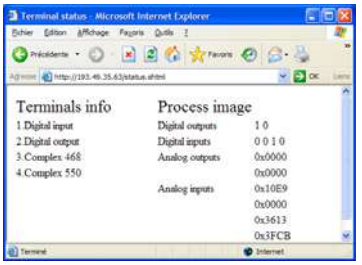


Module E/S WAGO sur Modbus TCP

- Module d'Entrées /Sorties déportées sur Ethernet
- Autoconfiguration de paramètre IP par BootP (serveur BootP Wago freeware)
- Serveur Modbus TCP (3 connexions maximum simultanées)
- Serveur Web Statique intégré



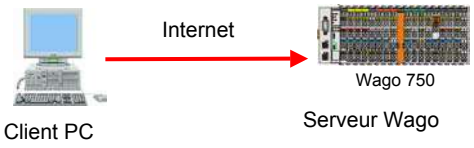
Wago 750



Stack TCP-IP du module WAGO 750 -342

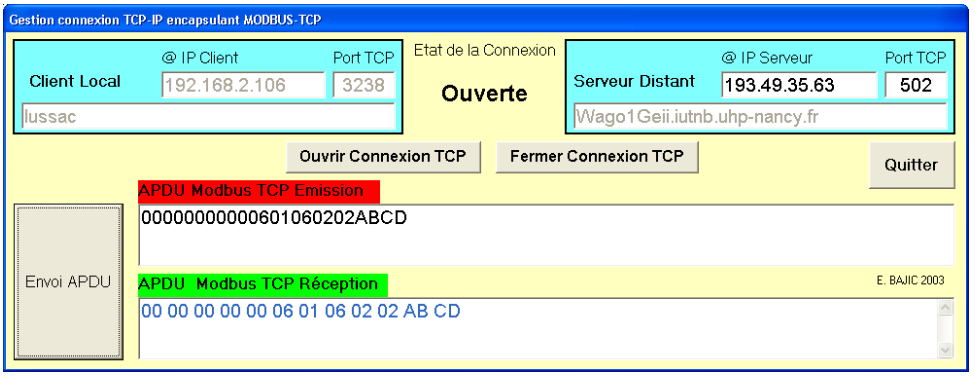
BootP	HTTP	Modbus-TCP
TCP		
IP		
Ethernet		

Application VB – Modbus TCP



Adresse E/S WAgo

E/S	Adresse Modbus	Bits/Mots	
Digital Outputs	**0x0200**	bits 0 et 1	Leds face avant de la carte
Digital Inputs	0x0000	bits 0, 1, 2, 3	
Analog Outputs	**0x0200**	Mot	Cablé sur Canal 1 sortie ANA
	0x0201	Mot	
Analog Inputs	0x0000	Mot	Potentiomètre linéaire local
	0x0001	Mot	Recopie du canal 0 sortie ANA
	0x0002	Mot	Température (-10 °C .. 40 °C)
	0x0003	Mot	Hygrométrie en 0 .. 100%



Programmation VB – Modbus TCP

```
Private Sub BtnClose_Click()
Winsock.Close ' puis on ferme la connection
lblConnexion.Caption = "Fermée"
btnSendData.Enabled = False
End Sub

Private Sub BtnConnect_Click()

Winsock.RemoteHost = txtIPServeur.Text ' Paramétrage de la
connexion
Winsock.RemotePort = txtPortTCP.Text
lblConnexion.Caption = "en cours..."
Winsock.connect ' puis on tente une connexion
'Attente d'acquittement d'ouverture de la cnx
Timer1.Enabled = True: timeout = 0
While (timeout <> 1)
    DoEvents ' on laisse le PC travailler ...
    If Winsock.State = sckConnected Then
        lblConnexion.Caption = "Ouverte"
        txtServeurHostName.Text =
GetHostNameFromIP(txtIPServeur.Text)
        txtPortClient = Winsock.LocalPort
        txtIPClient = Winsock.LocalIP
        txtClientHostName = Winsock.LocalHostName
        Timer1.Enabled = False: timeout = 1 ' on sort de la
boucle dès qu'on est connecté
    End If
Wend
' test si cnx OK
If Not Winsock.State = sckConnected Then
    lblConnexion.Caption = "Refusée"
End If
End Sub

Private Sub btnQuit_Click()
Winsock.Close ' on ferme la connexion
End
End Sub
```

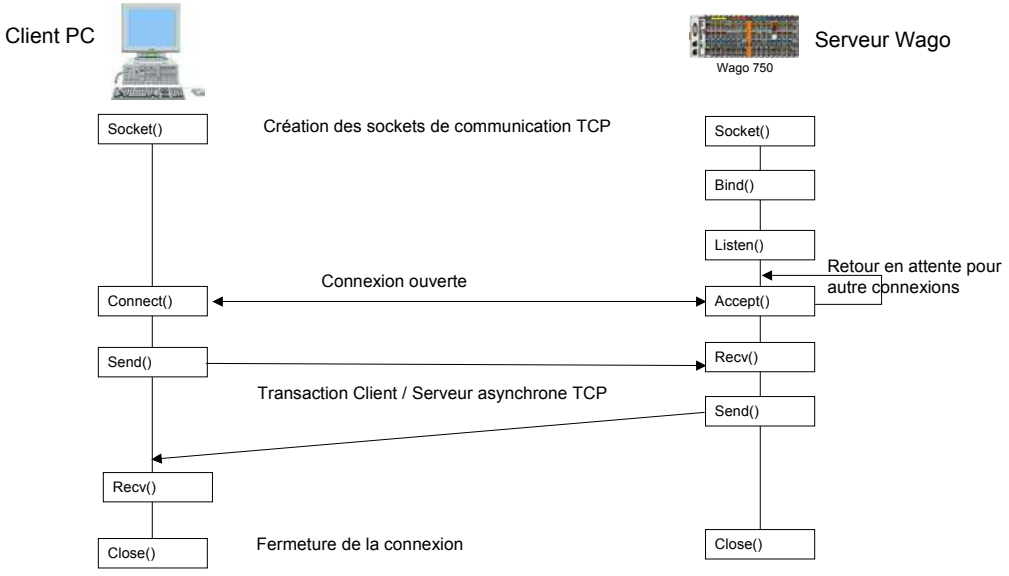
```
Private Sub btnSendData_Click()
'Bouton envoi de données sur la cnx
Dim trame As String

If Not Winsock.State = sckConnected Then
    MsgBox ("Communication impossible, Pas de Connexion ouverte !")
    Else
    trame = Str2hex(txtEnvoiDonnees)
    Winsock.SendData trame
    End If
End Sub

Private Sub Form_Load()
txtIPClient = Winsock.LocalIP
txtClientHostName = Winsock.LocalHostName
End Sub

Private Sub Timer1_Timer()
timeout = 1: Timer1.Enabled = False
End Sub

Private Sub Winsock_Connect()
' Activé lorsque la connexion est effective

lblConnexion.Caption = "Ouverte"
btnSendData.Enabled = True
End Sub

Private Sub WinSock_DataArrival(ByVal bytesTotal As Long)
Dim Mess, msg As String

Winsock.GetData Mess, vbString ' on capture ce qui est arrivé sur la cnx ..
msg = Hex2str(Mess)
txtAPDUReception = msg
End Sub

Private Sub Winsock2_Error(ByVal number As Integer, Description
As String, ByVal Scode As Long, ByVal Source As String, ByVal
HelpFile As String, ByVal HelpContext As Long, CancelDisplay As
Boolean)
End Sub
```

Interface API Socket TCP – Langage C

Primitives de programmation en C :

Socket ()	: Création d'un objet socket pour gérer une connexion TCP
Bind ()	: Associer l'objet socket à une connexion TCP
Listen ()	: Mise en écoute sur une socket pour une application serveur
Accept ()	: Accepter une connexion d'un client sur un port TCP pour une application serveur
Connect ()	: Demande de connexion effectuée par une application client
Send ()	: Envoi d'une APDU à TCP pour émission sur la connexion socket
Recv ()	: Lire les messages reçus dans la socket
Close ()	: Fermer une connexion TCP

Gestion Socket TCP – Langage C



Passerelle ModBus série RTU / Modbus Ethernet TCP, alimentation interne

Passerelle industrielle

- **Compatible Ethernet 10Base-T (802.3)**
- **Fonction serveur TCP/IP (HTTP, TFTP, SMTP, SNMP, etc...)**
- **Supporte les protocoles Modbus RTU-ASCII**
- **E/S série V24, V11 ou RS485 avec isolation optique**
- **Logiciel re-routage port COM vers Ethernet en option**

Le serveur de périphériques SP1xxxRMB ou passerelle Modbus/TCP permet la gestion de périphériques asynchrones Modbus, via un réseau Ethernet , à partir du protocole TCP/IP.
Le logiciel applicatif pilotant l'équipement asynchrone peut utiliser soit un port COM traditionnel avec un logiciel de re-routage soit un port Ethernet 10Base-T (Modbus/TCP).
Le protocole de communication Modbus, basé sur le principe Maître - Esclaves, est supporté de différentes façons par le serveur de périphérique SP1xxxRMB:
- Connexion d'équipements Modbus esclaves vers un poste maître Modbus/TCP
- Connexion d'un poste maître Modbus RTU vers des équipements esclaves Modbus/TCP
- Connexion traditionnelle Modbus maître / esclaves via un support Ethernet TCP/IP

Le serveur de périphérique ou passerelle Modbus SP1xxxRMB supporte ces différentes configurations, qui permettent la migration progressive vers Modbus/TCP. Différents paramètres doivent être sélectionnés lors de la phase de configuration :
- Adresse "slave" unique ou multiple (pour une adresse IP)
- Time-out de réception caractères
- Time-out de réception réponse de l'esclave

Le serveur de périphériques ou passerelle Modbus SP1xxxRMB comporte une E/S série asynchrone de type V24, V11 ou RS485 avec isolation optique et un débit maximal de 115,2 Kbps avec/sans gestion de flux.
L'accès Ethernet 10 Mbps (IEEE802.3) de type 10Base-T (RJ45) permet le raccordement sur le réseau local Ethernet via un port de hub Ethernet ou par l'intermédiaire d'un convertisseur de média cuivre / fibre optique.
Le serveur de périphériques SP1 est proposé sous forme de boîtier plastique autonome ou de boîtier compact pour fixation sur rail Din.
De nombreuses applications sont envisageables grâce au serveur de périphériques, de la simple gestion de capteurs à des processus industriels, en passant par des applications de télécontrôle et de télésalarme, etc...
Le serveur de périphériques peut également supporter localement des applicatifs spécifiques sur demande.
La passerelle Modbus/TCP inclut de nombreux protocoles en plus de TCP/IP, UDP, Telnet, DHCP, SNMP, TFTP, HTTP, etc...
Les applications autorisées par ces différents services sont à prendre en compte par l'utilisateur.
GMI-DATABOX propose plusieurs types de prestations de service, du transfert de compétences à la réalisation complète de vos applications spécifiques.
De nombreux autres modèles sont disponibles pour des environnements ModBus et industriels avec :
- entrées et/ou sorties numériques (T.O.R)
- entrées et sorties analogiques, etc...

Contactez notre service commercial.

CARACTERISTIQUES

: Fonction : Convertisseur Modbus / Ethernet TCP-IP
E/S série : V24, V11 ou RS485
Mode de transmission : Asynchrone
Code de transmission : transparent
Protocole de transmission : Modbus RTU ou ASCII
Débit : jusqu'à 115,2 Kbps
Gestion de flux : XON/XOFF, CTS/DTR ou sans.
Isolation : par coupleurs optiques
Accès Ethernet :
Conforme IEEE 802.3 (CSMA/CD)
Interface : RJ45 (10 Base-T)
Adresse IP : programmable
Protocoles : TCP, IP, ICMP, ARP
Services TCP : SNMP, DHCP, TFTP, HTTP, SMTP, etc...
Dimensions :
- Boîtier plastique : 110 x 50 x 170 mm (l x h x p)
- Boîtier rail Din : 75 x 100 x 110 mm

Connecteurs :
- LAN : RJ45
- E/S de contrôle : DB9 femelle
- E/S utilisateur : DB9 mâle

Voyants :
- Lan : Activité, TD, RD et Collision
- E/S : TD, RD, ON

Alimentation : 230 VAC 50/60 Hz
Option alimentation : 24 ou 48VDC
Environnement :
- Température d'utilisation : 0° à 50°C
- Température de stockage : -10° à +70°C
- Humidité : 10 % à 90 % sans condensation.

Normes : CEM 89/336
EN60950
Fabrication : France

http://www.gmidatabox.fr/gmidatabox/site/produit/produit_fiche.php?id_produit=345&type=reference&let=S