

Bus de terrain

Mise en œuvre d'Ethernet et TCP/IP

Thèmes abordés

- Les protocoles
 - Support physique d'Ethernet
 - Liaison
 - Principes d'accès au média.
 - Le rôle des Hub, Switch, et Routeur
 - Réseau
 - Comment effectuer le routage des messages ?
 - IP, ARP, DHCP, DNS.
 - Comment configurer 2 cartes ethernet sur un PC ?
 - Transport
 - TCP, UDP
 - Application
- La programmation
 - Comment programmer un échange entre deux stations Ethernet ?
 - L'API socket

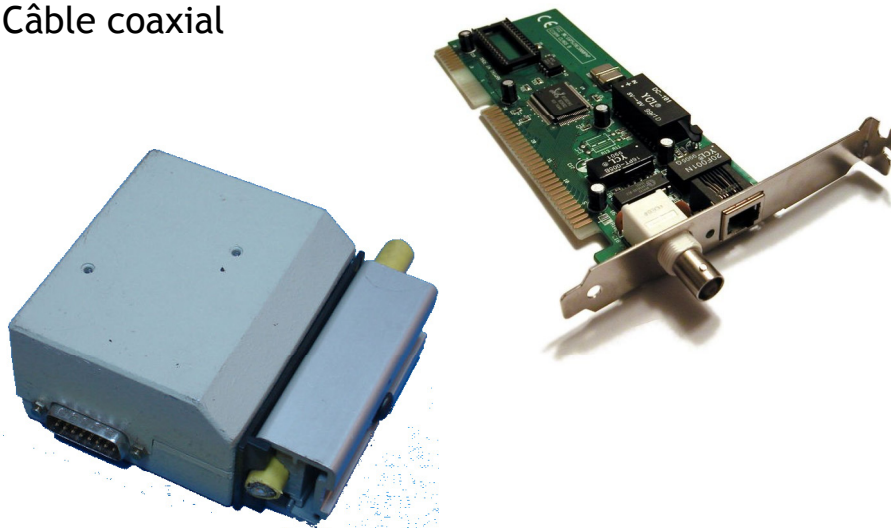
Ethernet - Support physique

Les origines - Ethernet sur câble coaxial

- Inventé en 1982
- Éther
 - substance impalpable, imaginée au XVIIIe siècle pour expliquer la propagation des ondes électromagnétiques
- Normalisation IEEE 802.3
- Câble
 - Coaxial
 - Limité à 10 MBps
 - Toutes les stations se branchaient sur le même câble.
 - Connexion par prise « vampire »
 - Topologie en bus.
- Interface électrique
 - Isolée par transformateur
- Codage
 - Utilise le codage de Manchester.
 - Transmission des bits LSB en premier.
 - Transmission des octets MSB en premier.









| | |
|---|--------------|
| 7 | Application |
| 6 | Présentation |
| 5 | Session |
| 4 | Transport |
| 3 | Réseau |
| 2 | Liaison |
| 1 | Physique |

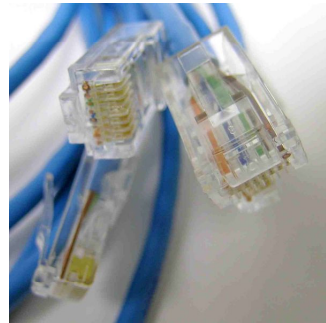
Ethernet 10 Mbits Câble coaxial



Ethernet - Support physique

FastEthernet 100 Mb sur paire torsadée - le standard actuel

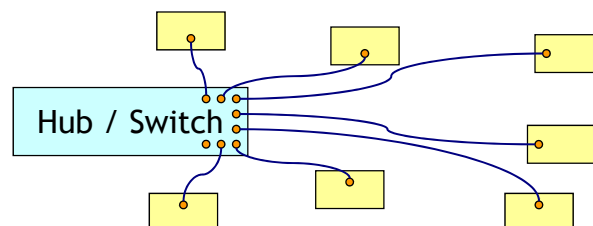
| Broche | Paire | Fil | Couleur | Hub / Switch | PC |
|--------|-------|-----|--|--------------|-----|
| 1 | 2 | 1 | Blanc / orange  | RD+ | TD+ |
| 2 | 2 | 2 | Orange  | RD- | TD- |
| 3 | 3 | 1 | Blanc / vert  | TD+ | RD+ |
| 4 | 1 | 2 | Bleu  | | |
| 5 | 1 | 1 | Blanc / bleu  | | |
| 6 | 3 | 2 | Vert  | TD- | RD- |
| 7 | 4 | 1 | Blanc / brun  | | |
| 8 | 4 | 2 | Brun  | | |



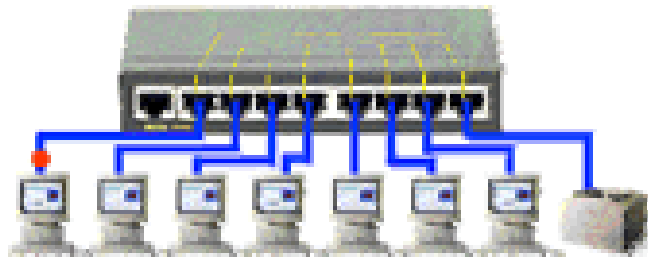
Ethernet - Support physique

FastEthernet 100 Mb sur paire torsadée - le standard actuel

- Topologie physique en étoile
- Topologie logique en bus
- Hub
 - Transmet le signal d'un port vers tous les autres
- Switch
 - Réceptionne un paquet
 - Le retransmet vers le port du destinataire



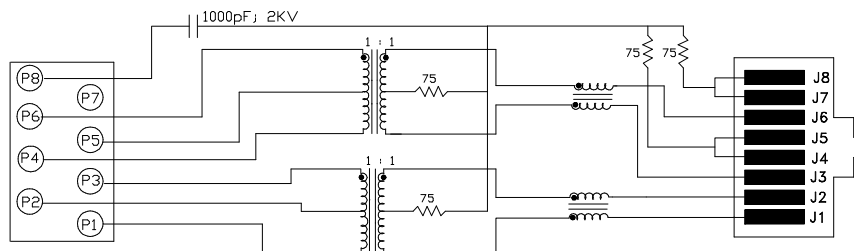
Hub - Illustration du fonctionnement



Ethernet - Support physique

Isolation par transformateur

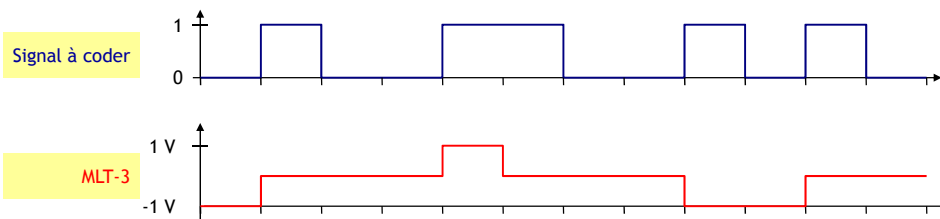
- Transformation et adaptation d'impédance en bout de ligne.
- On trouve des connecteurs incluant directement le transformateur et les résistances de terminaison.



Ethernet - Support physique

FastEthernet 100 Mb sur paire torsadée - Modulation MLT3

- Caractéristique des câbles à paire torsadée catégorie 5
 - Bande passante limitée à 100 MHz.
 - Ne convient pas au codage Manchester de 100 MBps.
- Codage MLT3
 - Codage électrique sur 3 niveaux, permettant de diviser par 3 la bande passante requise.
 - Passe à travers les états -1, 0, +1, 0, -1 pour chaque bit 1.
 - Pour les bits 0, reste à son état.



Ethernet - Support physique

FastEthernet 100 Mb sur paire torsadée - Codage 4B5B

- Le codage MLT3 seul
 - N'est pas équilibré en valeur moyenne.
 - Il n'y a pas de signal d'horloge avec des longues séries de 0.
- Le codage 4B5B est appliqué avant l'émission des données
 - Les paquets de 4 bits sont transformés en 5 bits
 - A chaque combinaison de 4 bits, on associe un paquet de 5 bits n'ayant pas plus de 3 bits consécutifs à 0.
 - Le récepteur procède au codage inverse
 - Augmente la fréquence de bit à 125 MHz.
 - Avec la modulation MLT-3, la fondamentale sur le câble est à 31,25 MHz

| 4 bits | 5 bits |
|--------|--------|
| 0000 | 11110 |
| 0001 | 01001 |
| 0010 | 10100 |
| 0011 | 10101 |
| 0100 | 01010 |
| 0101 | 01011 |
| 0110 | 01110 |
| 0111 | 01111 |
| 1000 | 10010 |
| 1001 | 10011 |
| 1010 | 10110 |
| 1011 | 10111 |
| 1100 | 11010 |
| 1101 | 11011 |
| 1110 | 11100 |
| 1111 | 11101 |

Ethernet - Support physique

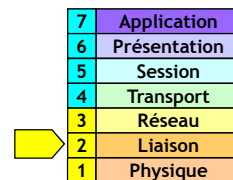
Les perspectives d'avenir

- Aujourd'hui, avec FastEthernet 100 Mb
 - Le débit est déjà élevé par rapport aux capacités de traitement des ordinateurs actuels.
 - La génération et le traitement du signal sont complexes.
 - Il existe des circuits intégrés spécifiques très efficaces.
- Tendances futures
 - Evolution vers 1GB et plus.
 - L'électronique sera encore plus compliquée.
 - Problème résolu grâce à des circuits intégrés spéciaux.

Ethernet - Liaison

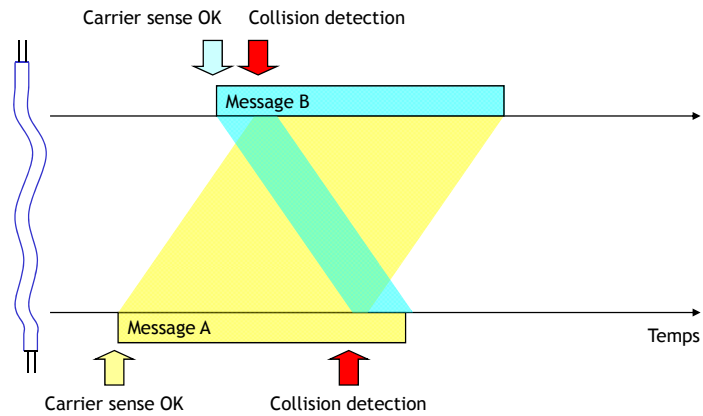
Accès au média - CSMA/CD - Principe

- CSMA : Carrier Sense Multiple Access
 - La station « écoute » si le média est libre avant d'émettre
- CD : Collision detection
 - Si une autre station a commencé à émettre en même temps, il y a collision des messages.
 - Dans ce cas, chaque station arrête d'émettre.
 - Chacune attend un temps aléatoire
 - Le temps d'attente est doublé à chaque ré-essai, jusqu'à un maximum.
- Analyse
 - Mécanisme de partage du média simple.
 - Non prédictible.
 - L'efficacité diminue avec le nombre de participants.



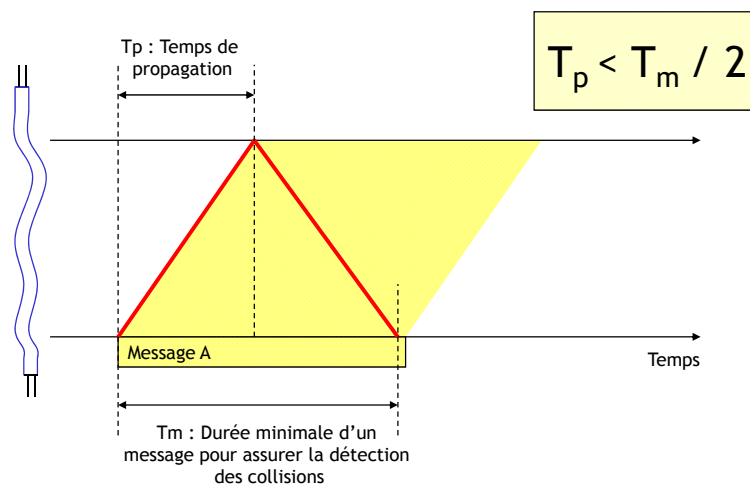
Ethernet - Liaison

Accès au média - CSMA/CD - illustration d'une collision



Ethernet - Liaison

Accès au média - CSMA/CD - Durée minimale d'un message



Ethernet - Liaison

Accès au média - CSMA/CD - Contraintes

- Le plus petit message sur Ethernet est long de 72 octets

- 8 octets de préambule
- 64 octets de message
- 173'000 télégrammes par seconde.

$$T_m \geq \frac{72 \times 8}{100'000'000} = 5.76 \mu\text{s}$$

- Pour que CSMA / CD fonctionne bien

- Le temps de propagation doit être inférieur à la moitié de la durée du plus petit message.
- Ce temps se traduit par une longueur maximale de câble.
- Pour tenir compte des délais de l'électronique et des Hubs
 - Un facteur de sécurité de 4 est appliqué

$$L_{\max} < \frac{5.76 \cdot 10^{-6} \times 200'000'000}{2 \times 4} = 144 \text{ m}$$

Ethernet - Liaison

Accès au média - CSMA/CD - Contraintes

- Dans la pratique, pour Ethernet 100 Mbits :

- On recommande de limiter la longueur des liaisons à 100 m.
- Le diamètre d'un domaine de collision est limité à 200 m.

Ethernet - Liaison

Accès au média - CSMA/CD - Faible efficacité comme bus de terrain

- Efficacité du protocole
 - Peu efficace pour la transmission de petits messages.
 - Particulièrement défavorable pour des périphériques comportant un petit nombre d'entrées / sorties
- Exemple : station avec 32 bits = 4 octets
 - Il faut émettre 72 octets pour 4 octets utiles.
 - L'efficacité est de 5,5 % !
- Mais Ethernet reste cependant très rapide !
 - 5.76 μ s pour un message est un très bon résultat.

Ethernet - Liaison

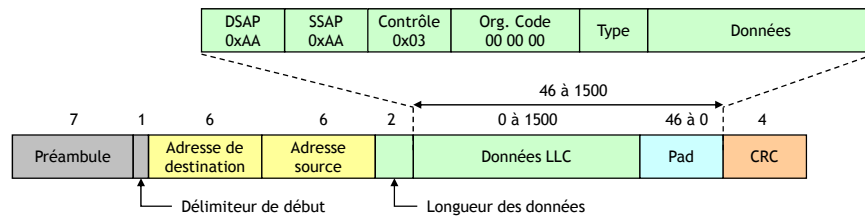
Accès au média - CSMA/CD - Analyse

- Avantages
 - Très bien adapté à la topologie en bus.
 - Facilité de mise en œuvre et d'extension.
 - Arbitration indépendante du nombre et de l'adresse des stations
 - Plus efficace que le « polling »
 - les stations ne doivent pas attendre pour communiquer un événement
- Inconvénients
 - Délais de transmission non déterministes (imprévisibles)
 - Risque de saturation si la ligne est chargée à >35 %
 - Mauvais rendement pour les messages courts
 - Pour envoyer l'état de quelques entrées et sorties tout ou rien
 - On doit quand même envoyer au moins 72 octets.
- Utilisations
 - Réseaux d'ordinateurs
 - De plus en plus comme bus de terrain (non temps réel).

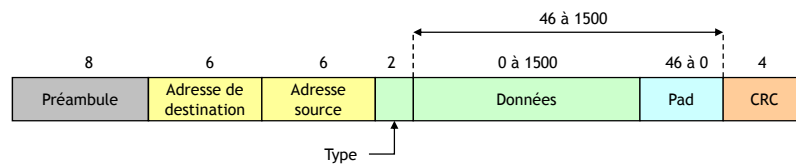
Ethernet - Liaison

Format des trames Ethernet / 802.3

- Format IEEE 802.3



- Format Ethernet II (le plus courant)



Ethernet - Liaison

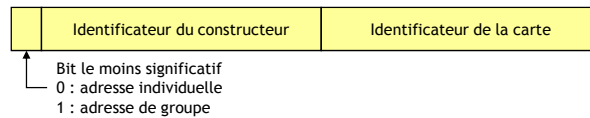
IP Sur Ethernet

- La plupart des cartes réseau utilise le format Ethernet II pour la transmission des datagrammes IP.
- Le champ 'Type' de la trame contient
 - la valeur hexadécimale 0800 pour indiquer un datagramme IP
 - la valeur hexadécimale 0806 pour indiquer un datagramme ARP
 - la valeur hexadécimale 8035 pour indiquer un datagramme RARP

Ethernet - Liaison

Adresse MAC

- Adresse physique d'une carte réseau
 - **Unique** : Toutes les cartes Ethernet ont une adresse différente
 - **Fixe** : configurée dans la mémoire ROM de la carte
- Longueur : 6 octets (48 bits)
 - Les 3 premiers octets : identifient le constructeur (géré par IEEE)
 - 00-00-0C-xx-xx-xx : Cisco
 - 08-00-20-xx-xx-xx : Sun
 - 08-00-09-xx-xx-xx : HP
 - Les 3 derniers octets : identifient la carte (gérés par le constructeur)
 - Bit le moins significatif du premier octet : indique une adresse de groupe.



Ethernet - Liaison

Adresse MAC de groupe

- Diffusion (Broadcast)
 - Adresse FF-FF-FF-FF-FF-FF
 - Toutes les cartes réseau reçoivent ces trames et les traitent.
 - Hub et switch : transmettent ces trames sur tous les ports.
 - Routeurs : filtrent ces trames.
 - Exemple d'utilisation : protocole ARP
- Multicast
 - 1^{er} bit transmis de l'adresse vaut 1
 - Exemple : 09-00-2B-00-00-0E
 - Les stations configurées pour écouter l'adresse multicast recevront les messages.

heig-vd

Ethernet - Liaison

Détection d'erreur par CRC

- CRC : Control de Redondance Cyclique
- Mécanisme de vérification de la validité des trames
 - Très puissant.
 - Permet de détecter de façon fiable une grande variété d'erreurs.
 - Détecte aussi les erreurs en série affectant plusieurs bits.
- Principe
 - Le CRC est le reste de la division entière du message par un motif de bits prédéfini.
 - L'émetteur calcule le CRC et le place dans le message envoyé.
 - Le récepteur recalcule le CRC et le compare avec le CRC reçu.

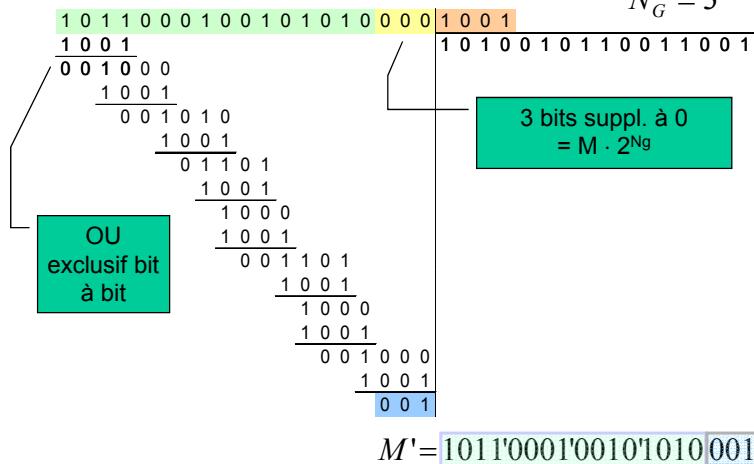
heig-vd

Ethernet - Liaison

Détection d'erreur par CRC

$$M = 1011'0001'0010'1010 \quad G = 1001$$

$$N_G = 3$$



heig-vd

Ethernet - Liaison

Les concentrateurs (Hub)

- **Concentrateur (Hub) : un répéteur multi port**
 - Câblage en étoile.
 - Topologie logique correspondant au bus.
- **Travaille au niveau de la couche physique**
 - Reçoit et régénère les signaux reçus sur chaque port
 - De moins en moins employé



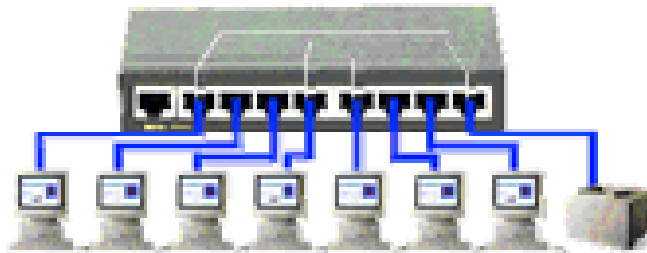
heig-vd

Ethernet - Liaison

Les commutateurs (Switch)

- **Même apparence extérieure qu'un hub**
- **Fonctionnement**
 - Full duplex sur chaque port.
 - Reçoit et mémorise les trames.
 - Les transmet sur le bon port de destination seulement.
 - Séquence les trames destinées à un même port.
 - Dans les limites de ses capacités de mémorisation.
 - Très rapide
- **Conséquences**
 - Plus de risque de collision.
 - Peut adapter des réseaux avec des vitesses différentes : 10, 100 MBps.
 - Auto-sensing : détecte automatiquement les vitesses de transmission.
 - Pas besoin de réémettre une trame, donc pas de délai aléatoire.
 - Risque de perte de trames en cas de saturation du débit.
 - Latence typique de quelques micro secondes.
- **On parle d'Ethernet commuté.**
 - Caractéristiques temps réel nettement améliorées.

Switch - Illustration du fonctionnement



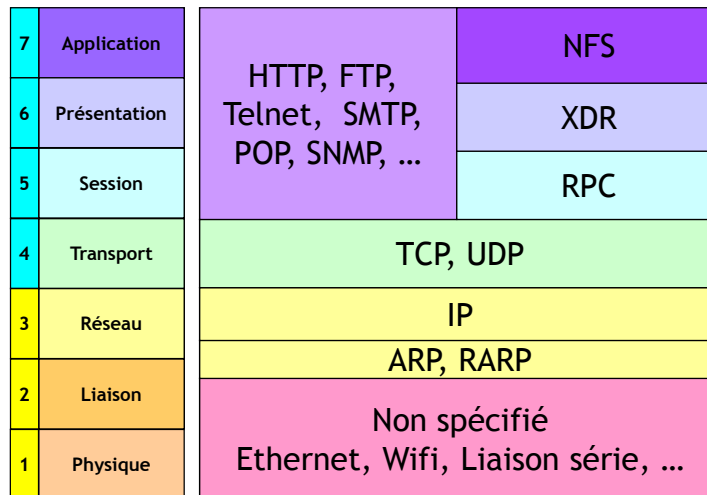
TCP/IP - Réseau

Le protocole d'Internet

- IETF
 - Internet Engineering Task Force
 - www.ietf.org
- IETF Gere notamment les RFC
 - Request For Comment.
 - Documents contenant les spécifications d'IP et des protocoles des couches supérieures.

TCP/IP - Réseau

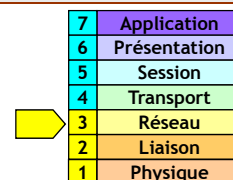
Le modèle TCP/IP



TCP/IP - Réseau

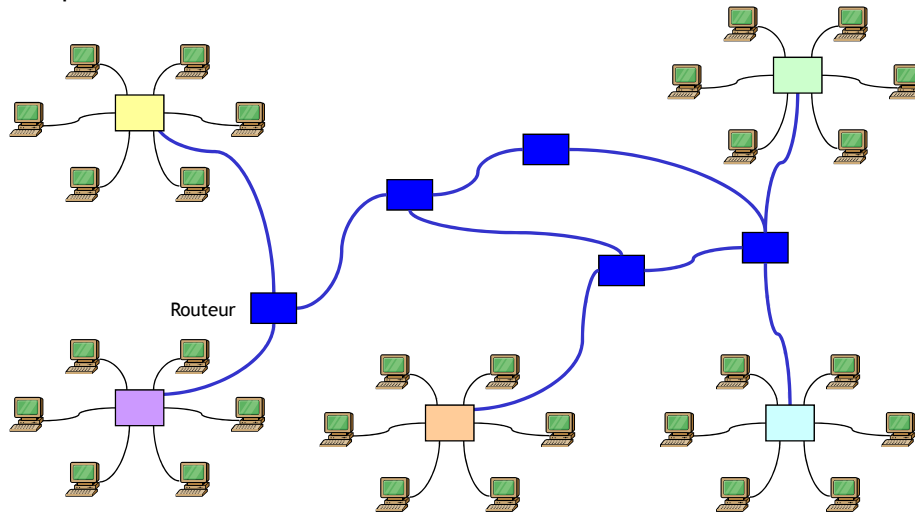
Le protocole IP

- IP : Inter-network Protocol
 - Protocole pour l'acheminement de datagrammes entre réseaux.
 - Adressage (adresse IP).
- Objectif lors de la conception
 - Créer un protocole utilisable sur toutes les technologies sous-jacentes.
 - Ex : Ethernet, WiFi, ADSL, liaison série / modem.
- Modèle de service
 - Sans connexion
 - Service non fiable - les datagrammes peuvent
 - Être perdus
 - Arriver dans le désordre
 - Être dupliqués
 - Être retardés



TCP/IP - Réseau

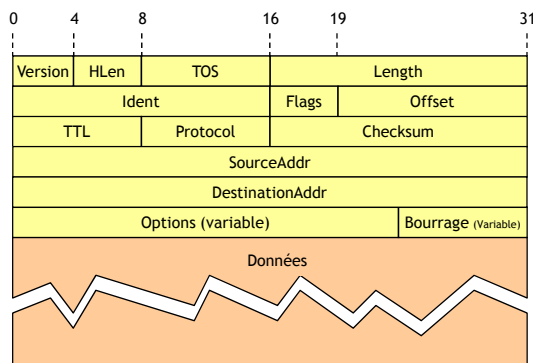
Le protocole IP - Architecture d'un réseau



TCP/IP - Réseau

Le protocole IP - Format des datagrammes

- Version : 4 (IP V4)
- HLen : longueur de l'en tête
- TOS : Type of Service
 - Minimize delay, costs,...
- Length : longueur du datagramme complet.
- Ident : numéro du datagramme
- Offset : position des données dans le datagramme
- TTL : Time To Live
- Protocol
 - TCP, UDP, ...
- Checksum : somme de contrôle de l'en tête IP
- Adresse source et destination : adresse IP sur 4 octets (32 bits).
- Options : peu utilisé.



heig-vd

TCP/IP - Réseau

Le protocole IP - Fragmentation des datagrammes

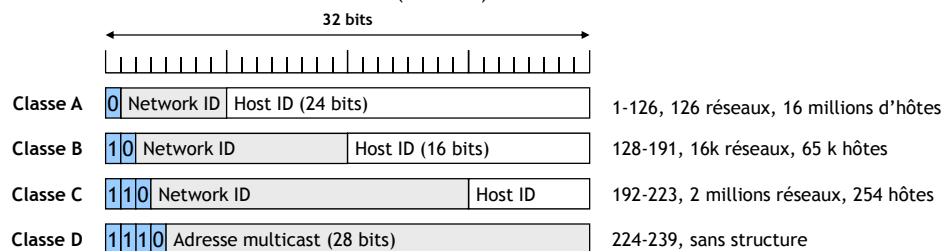
- IP permet d'envoyer des datagrammes jusqu'à 65535 octets (en-tête compris)
- La couche liaison ne peut en général pas envoyer des trames aussi longues.
 - MTU : Max Transmission Unit.
 - MTU doit être au moins égal à 576 octets.
- Les datagrammes IP sont alors fragmentés
 - Envoi en plusieurs trames, pouvant arriver dans le désordre.
 - Réassemblage à l'arrivée.
 - En cas de perte d'un fragment, tout le datagramme est perdu.
 - Pas de mécanisme de retransmission dans IP.

heig-vd

TCP/IP - Réseau

Le protocole IP - Adresse IP

- Chaque interface réseau à une adresse IP unique.
- Constituée de 4 octets.
 - Ex : 193.134.217.17 (www.heig-vd.ch)
- Contient deux parties
 1. Identificateur de réseau (Network ID)
 - Assignée par une autorité (p.ex. ISP)
 2. Identificateur de machine (Host ID)



heig-vd

TCP/IP - Réseau

Le protocole IP - Adresses IP particulières

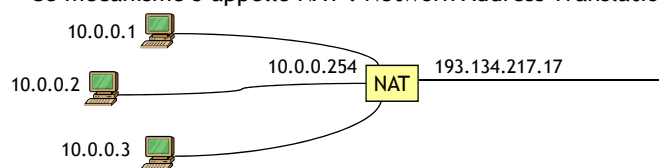
- Adresses 240 à 254
 - Elles sont réservées pour un usage futur.
- Voir RFC 3330 pour obtenir la liste des adresse réservées.
- Adresses de réseau privées (réseaux internes)
 - 10.0.0.X : 8 bits à disposition.
 - 172.16.0X.X : 12 bits à disposition.
 - 192.168.X.X : 16 bits à disposition
 - X : Eviter les valeurs suivantes pour l'adresse d'un hôte
 - 0 : désigne le réseau.
 - 255 : diffusion à tout le réseau.
- Diffusion à tout le réseau local
 - 255.255.255.255
- Diffusion à un réseau distant
 - Adresse du sous réseau, numéro d'hôte avec tous les bits à 1

heig-vd

TCP/IP - Réseau

Le protocole IP - Réseaux privés et routage

- Problème du routage
 - Il n'y a pas assez de réseaux de classe A, B et C pour toutes les organisations reliées aujourd'hui au réseau.
- Solution
 - Pour une organisation, on donne une adresse IP unique pour un routeur.
 - A l'intérieur de l'organisation, on utilise une plage d'adresses privées. Comme 10.0.0.X
 - Pour les requêtes entrant et sortant du réseau de l'organisation, les adresses privées sont remplacées par l'adresse du routeur.
 - Ce mécanisme s'appelle NAT : Network Address Translation



heig-vd

TCP/IP - Réseau

Résolution d'adresse - ARP

- Problème à résoudre
 - On veut émettre un message vers une station du même réseau local.
 - On connaît son adresse IP.
 - On doit trouver son adresse MAC pour envoyer la trame vers la bonne carte réseau.
 - Comment trouver l'adresse MAC ?
- Le protocole ARP : Address Resolution Protocol
 - Envoie un message ARP en diffusion sur le réseau local:
 - Qui possède l'adresse IP recherchée ?
 - La station concernée, si elle existe, envoie une trame ARP contenant
 - Son adresse IP.
 - Son adresse MAC.

heig-vd

TCP/IP - Réseau

Résolution d'adresse - Le cache ARP

- Ce processus prend du temps.
- Il ne peut donc pas être répété à chaque message.
- Chaque station gère une table, appelée « cache » ARP
 - Cette table mémorise les associations MAC - IP.
 - L'entrée du cache est effacée au bout de 20 minutes
 - Si elle n'a pas été rafraîchie

heig-vd

TCP/IP - Réseau

Le protocole IP - Réseau, sous réseau et passerelle

- Notion de sous réseau
 - Permet de définir l'ensemble des machines reliées au même réseau physique.
- Masque de sous réseau
 - Toutes les machines qui donnent le même résultat lors de la combinaison Adresse et Masque sont considérées comme étant reliées sur le même réseau physique.

| | | | | | |
|------------|---|-----|-----|-----|-----|
| Adresse IP | & | 10 | 0 | 0 | 183 |
| Masque | | 255 | 255 | 255 | 0 |
| | | 10 | 0 | 0 | 0 |

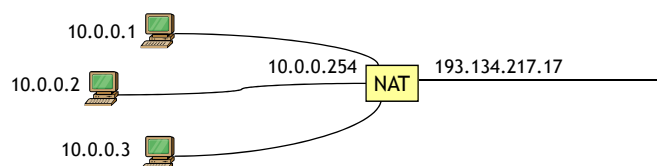
- IP essaiera donc de les atteindre directement.
- Pour les autres, les trames sont envoyées vers la passerelle par défaut.

heig-vd

TCP/IP - Réseau

Le protocole IP - Réseau, sous réseau et passerelle - illustration

- 10.0.0.1 envoie un datagramme à 10.0.0.3
 - C'est le même sous réseau (masque 255.255.255.0)
 - Recherche de l'adresse MAC par ARP ou dans le cache ARP
 - Envoie directement vers la carte réseau de 10.0.0.3
- 10.0.0.1 envoie un datagramme vers 209.85.135.104
 - Ce n'est pas le même sous réseau.
 - Ce message est envoyé vers la carte réseau de la passerelle.
 - La passerelle se charge de relayer plus loin ce datagramme.

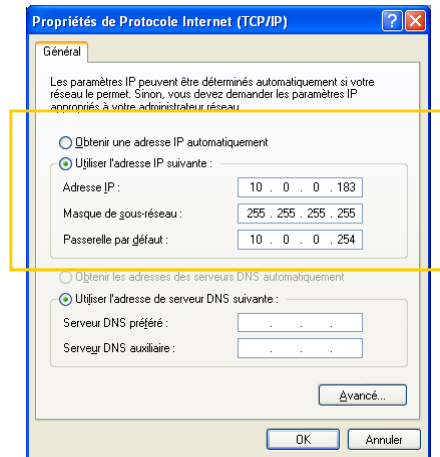


heig-vd

TCP/IP - Réseau

Attribution d'adresse fixe

- L'adresse IP d'une carte réseau peut être configurée de façon fixe.
- Exemple : sous Windows, propriétés TCP/IP

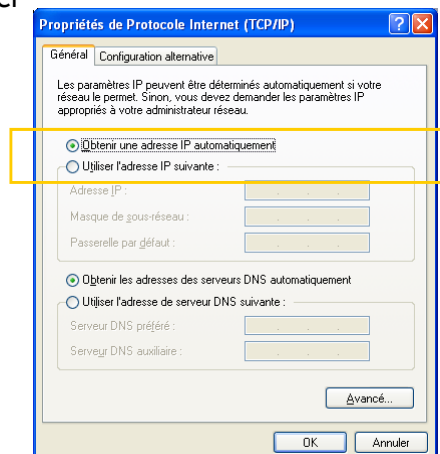


heig-vd

TCP/IP - Réseau

Attribution d'adresse dynamique - DHCP

- L'adresse IP peut aussi être définie automatiquement.
- DHCP : Dynamic Host Configuration Protocol
 - La station qui a besoin d'une adresse IP envoie une requête DHCP sur le réseau (diffusion).
 - Un serveur DHCP doit être présent sur le réseau
 - Il renvoie l'adresse IP assignée, le masque de sous-réseau la passerelle ainsi que l'adresse des DNS à utiliser.



TCP/IP - Réseau

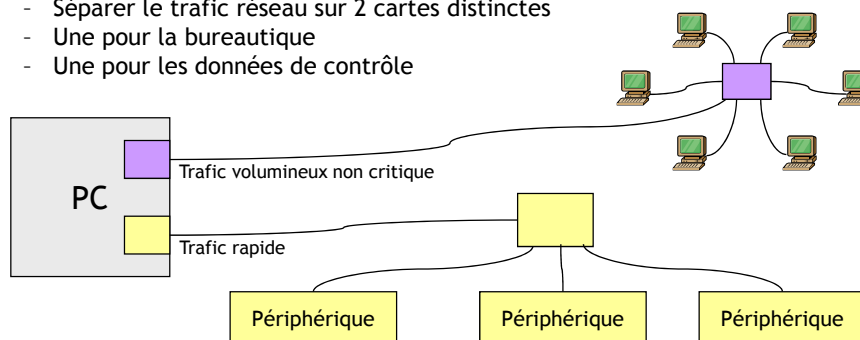
Résolution de nom - DNS

- Défaut des adresses IP dans le contexte d'Internet
 - Elles sont difficiles à mémoriser pour nous.
 - Pas parlantes du tout.
- Utilisation de noms de domaines plus parlants
 - www.vacances.ch
 - A quelle adresse IP correspond ce nom de domaine ?
- DNS : Domain Name Server
 - La station qui veut connaître l'adresse IP correspondant à un nom de domaine envoie un message au DNS contenant le nom de domaine.
 - Le DNS possède une table de conversion.
 - Le DNS renvoie un message contenant l'adresse IP.

TCP/IP - Réseau

Séparation du trafic - Utilisation industrielle d'Ethernet

- Comment garantir un trafic rapide vers les systèmes d'entrée sortie connectés sur Ethernet ?
 - Risque de latences élevées pendant le transfert de données « bureautiques ».
- Solution
 - Séparer le trafic réseau sur 2 cartes distinctes
 - Une pour la bureautique
 - Une pour les données de contrôle



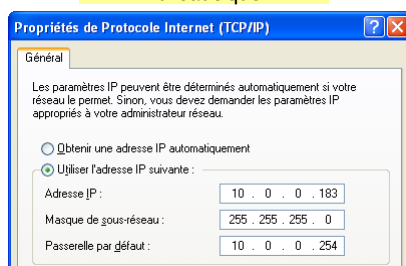
TCP/IP - Réseau

Séparation du trafic - Configurer 2 adaptateurs Ethernet sur un PC

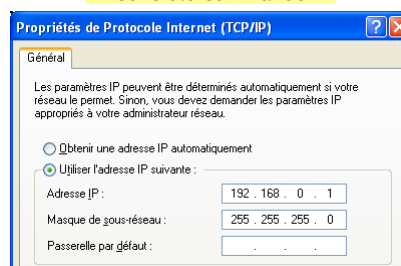
- Principe

- Créer 2 sous réseaux distincts.
- Utiliser une adresse privée pour le réseau de commande !
- Attention, une seule passerelle pour les adresses hors du sous réseaux !

Bureautique



Contrôle commande



TCP/IP - Réseau

Quelques commandes pratiques sous Windows

- PING

- Envoie un message PING vers une station.
- Affiche les réponses reçues.
- Permet donc de tester la liaison

- Exemples

- PING 10.0.0.25
- PING www.heig-vd.ch

- IPCONFIG

- Permet d'afficher la configuration des adaptateurs réseau

- NETSTAT

- Affichage des connexions réseau

- ROUTE

- Gestion des tables de routage

heig-vd

TCP/IP - réseau

Perspectives d'avenir

- Evolution d'IP vers une version 6
- Adresses de 128 bits
 - Organisés en 8 paquets de 16 bits.
 - Exemple :
1fff:0000:0a88:85a3:0000:0000:ac1f:8001
- Une adresse IP v6 peut contenir une adresse IP v4.

heig-vd

TCP/IP - Transport

Introduction

- 2 protocoles principaux
 - UDP : très simple, transport de messages.
 - TCP : protocole complet, transmission d'un flot de données.

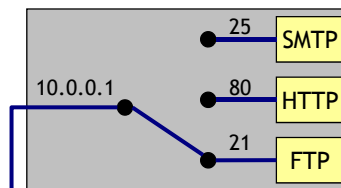
| | |
|---|--------------|
| 7 | Application |
| 6 | Présentation |
| 5 | Session |
| 4 | Transport |
| 3 | Réseau |
| 2 | Liaison |
| 1 | Physique |

heig-vd

TCP/IP - Transport

L'adressage dans TCP/IP - Notion de port

- Adressage de l'interface de destination
 - Réalisé en utilisant l'adresse IP
- Adresse du service ou du client
 - Utilisation d'un numéro de « port ».
 - Permet de démultiplexer ou « aiguiller » les paquets reçus sur une interface.
- Les numéros de port
 - Entiers 16 bits, 0 à 65535
 - Espace de numéros UDP et TCP indépendants. Le port TCP 21 est différent du port UDP 21



heig-vd

TCP/IP - Transport

L'adressage dans TCP/IP - Attribution des numéros de port

- « Ports bien connus » 0 à 1023
 - Définis dans des RFC
 - ftp (données) 20 TCP
 - ftp (contrôle) 21 TCP
 - telnet 23 TCP
 - smtp 25 TCP
 - snmp 161 UDP
 - portmap 111 TCP
 - http 80 TCP
- « Ports éphémères »
 - Assignés dynamiquement par le protocole PORTMAP
 - Un serveur s'enregistre auprès de PORTMAP de sa machine
 - Un client contacte le PORTMAP la machine éloignée pour demander le no de port correspondant à un nom d'une application
- Ports assignés en dur
 - Approche utilisée par un très grand nombre d'applications.

heig-vd

TCP/IP - Transport

Envoi de messages simples - UDP

- UDP : User Datagram Protocol
 - Permet d'envoyer un paquet de données.
 - Il n'y a pas de mécanisme d'acquittement prévu dans UDP.
 - Il peut être programmé dans l'application.

| | |
|----------------------------|---------------------------------|
| 16 bits source port number | 16 bits destination port number |
| 16 bits UDP Length | 16 bits UDP Checksum |
| Data | |

- Taille maximale
 - En théorie, un message UDP doit comporter moins de 65535 octets (limitation de IP)
 - En pratique, la limitation peut être plus basse, dépend de l'implémentation.
 - Si le paquet est plus petit que le MTU (576 octets), transmission sur un seul datagramme IP.

heig-vd

TCP/IP - Transport

La diffusion de messages UDP (broadcast)

- UDP permet d'envoyer un message à plusieurs hôtes simultanément.
- Utilise la diffusion permise par IP et Ethernet.
- Diffusion limitée
 - Envoi à toutes les stations du réseau local.
 - Utiliser l'adresse 255.255.255.255
 - Ne traverse pas les routeurs.
- Diffusion vers un réseau
 - Mettre tous les bits du sous réseau à 1.
 - Exemple :
 - 10.0.0.255
 - Envoie à toutes les stations du sous réseau 10.0.0.X

TCP/IP - Transport

Analyse du protocole UDP

- **Simplicité**
 - Protocole non connecté.
 - Un message peut être envoyé sans initialisation préalable.
- **Performances**
 - Ethernet à 100 Mbits peut véhiculer 170'000 messages par seconde.
 - Le facteur limitant est en général la capacité de traitement des ordinateurs.
 - Temps d'envoi + réception observé en pratique
 - 250 à 300 µs entre PC
 - Aller retour d'un message à partir de 250 µs.
- **Risques**
 - Les paquets peuvent être perdus.
 - Ils peuvent aussi arriver dans un ordre différent de l'émission !

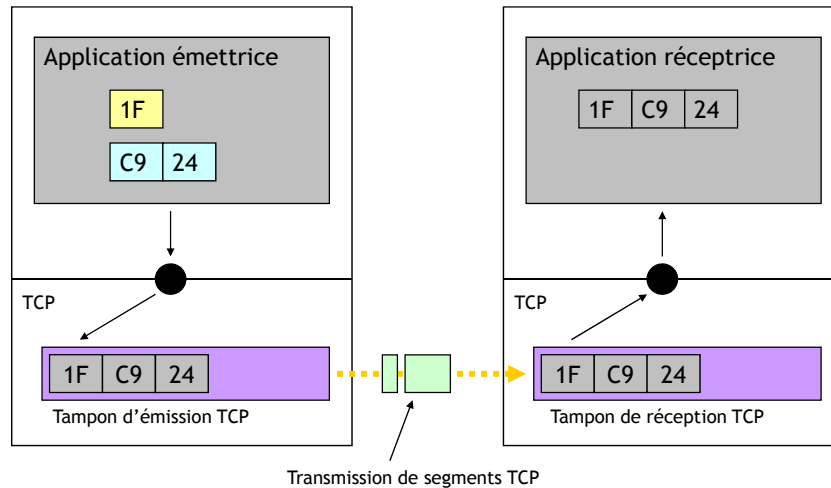
TCP/IP - Transport

Le protocole TCP - Principe

- **TCP : Transmission Control Protocol**
- **Fonctionnalité offerte**
 - Connexion
 - Une connexion logique doit être ouverte pour utiliser TCP.
 - Transmission point à point seulement
 - Pas de diffusion avec TCP.
 - Bidirectionnel : les deux stations engagées dans un échange TCP peuvent envoyer et recevoir des données.
 - Transmission fiable de bout en bout d'un flot d'informations.
 - Flot d'information : suite d'octets. Le découpage en bloc des données est perdu.
 - Il doit être reconstitué par l'application à la réception.
 - TCP réalise une transmission fiable sur IP qui n'est pas fiable !

TCP/IP - Transport

Le protocole TCP - Flot d'octets



TCP/IP - Transport

Le protocole TCP - Transmission fiable

- Comment réaliser une transmission fiable ?
 - La couche IP n'est pas fiable !
 - Elle peut perdre et mélanger les messages.
- TCP met en place un mécanisme d'acquittement.
 - Pour chaque message envoyé, TCP attend un acquittement.
 - En l'absence d'acquittement, il y a retransmission.
 - Le délai avant retransmission est exponentiel.
 - Peut donc envoyer les données des secondes plus tard.
- TCP numérote les messages
 - Permet de reconstruire le flot de données même si les messages arrivent dans le désordre.
 - La numérotation des messages est initialisée lors de l'ouverture de la connexion

heig-vd

TCP/IP - Transport

Le protocole TCP - Mécanismes d'optimisation

- Flot d'octets
 - Certaines applications envoient le texte caractère par caractère.
 - TCP est optimisé pour éviter d'envoyer un message par caractère.
 - But : regrouper les données pour réduire le nombre de messages IP.
- Algorithme 1 : Acquittement retardé
 - Lorsqu'un petit paquet de données est reçu, TCP attend jusqu'à 200 ms avant d'envoyer l'acquittement.
 - Cela donne une chance de grouper l'acquittement avec un message de retour.
- Algorithme 2 : Algorithme de Nagle
 - Un petit paquet de données n'est pas envoyé tant qu'un précédent petit paquet de données n'a pas été acquitté.

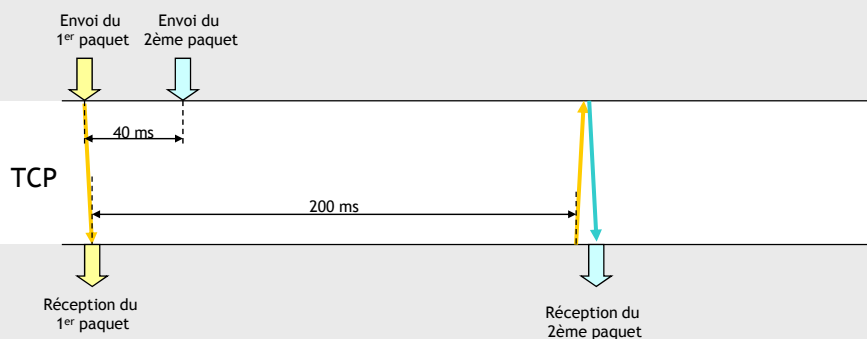
heig-vd

TCP/IP - Transport

Le protocole TCP - Mécanismes d'optimisation - illustration

Envoi de 2 paquets pour basculer des sorties, séparés de 40 ms

Application contrôlant une machine



Entrées sorties déportées sur Ethernet

heig-vd

TCP/IP - Transport

Le protocole TCP - Mécanismes d'optimisation - Conséquences

- Les mécanismes d'optimisation de TCP
 - Ils sont prévus pour améliorer l'efficacité moyenne de TCP.
 - Réduction du nombre de petits messages envoyés.
- Totalemment inadaptés au contexte du contrôle commande
- Il est possible de les désactiver
 - Fonction setsockopt
 - Paramètre TCP_NO_DELAY et TCP_NO_NAGLE

heig-vd

TCP/IP - Transport

Comparaison UDP et TCP dans la perspective des bus de terrain

- Bus de terrain
 - Transmettre de petits paquets de données délimités
 - Ex : Valeurs d'un groupe de 40 sorties tout ou rien - 40 bits.
 - Maîtriser le mieux possible les aspects temporels
- UDP
 - Orienté message
 - Envoi les données dès que possible.
 - Permet d'implémenter un acquittement et un réessai « personnalisés ».
- TCP
 - Orienté flot d'octets, ne délimite pas les messages.
 - Décide du moment de l'envoi des données.
 - Latences difficilement contrôlables
 - Peut envoyer les données plusieurs secondes plus tard
 - Effets catastrophiques et dangereux dans le domaine de la commande.

Programmation TCP/IP

L'API socket - Un bref aperçu

- Une application communiquant avec TCP peut être
 - « serveur » : elle attend les connexions de clients.
 - « client » : elle se connecte à un serveur qui attend.
- Pour communiquer en utilisant TCP ou UDP
 - Créer un socket : point de connexion logique.
 - Configurer le socket : options, ...
- Comme serveur
 - Il faut ensuite attendre les connexions de clients.
 - Lire les messages reçus.
 - Envoyer les réponses.
- Comme client
 - Il faut lier le socket à une adresse.
 - Envoyer les données.
 - Lire les réponses.
- Pour terminer un échange
 - Fermer le socket.
- Les fonctions sockets
 - Se trouvent dans une bibliothèque <sys/socket.h> ou <winsock2.h>

Programmation TCP/IP

L'API socket - Notion de socket

- Socket
 - Un socket est un objet logiciel.
 - Il représente un point de connexion avec une application.
 - S'utilise aussi bien avec TCP et UDP.
- Principe d'utilisation
 - Créer un socket en précisant le protocole souhaité.
 - Définir les options associées, adresse, port, etc.
 - Utiliser le socket pour
 - Envoyer des données.
 - Recevoir des données.
 - Etablir une connexion avec un serveur (TCP seulement)
 - Attendre des connexions d'un client (TCP seulement)

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP serveur par l'exemple - Initialisation

```
#include <winsock2.h>
#include <stdio.h>
#include <string.h>

int main()
{
    WSADATA info; // Spécifique à Windows
    SOCKET socket_handle;
    struct sockaddr_in socket_address;
    struct sockaddr_in sender_address;
    int sender_address_length;
    char message_data[1024];
    int data_size, status;

    // Initialisation Winsock 2 - spécifique à Windows
    WSASStartup(MAKEWORD(2, 2), &info);
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP serveur par l'exemple - Création

```
// Créer le socket UDP
socket_handle = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

// Préparer la structure avec l'adresse et le port
memset(&socket_address, sizeof socket_address, 0);
socket_address.sin_family = AF_INET;
socket_address.sin_port = htons(27015); // Port in network byte order
socket_address.sin_addr.s_addr = htonl(INADDR_ANY);

// Lier le socket au port (pour recevoir les datagrammes)
bind(socket_handle,
      (SOCKADDR *)&socket_address, sizeof socket_address);
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP serveur par l'exemple - Communication

```
// Gérer la communication
do
{
    // recevoir les données et l'adresse de l'émetteur
    sender_address_length = sizeof sender_address;
    data_size = recvfrom(socket_handle,
        message_data, sizeof message_data, 0,
        (struct sockaddr *)&sender_address, &sender_address_length);
    printf("Message reçu:%s", message_data);
   strupr(message_data); // convertit le message reçu en majuscules
    printf("Message renvoyé:%s", message_data);
    // retourner une réponse à l'émetteur
    status = sendto(socket_handle,
        message_data, strlen(message_data) +1, 0,
        (struct sockaddr *)&sender_address, sender_address_length);
}
while (strcmp(message_data, "QUIT\n") != 0);
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP serveur par l'exemple - Terminaison

```
// ferme le socket (libérer les ressources)
closesocket(socket_handle);

// Finaliser Winsock 2 - spécifique Windows
WSACleanup();
}
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP client par l'exemple - Initialisation

```
#include <winsock2.h>
#include <stdio.h>
#include <string.h>

int main()
{
    WSADATA info; // Spécifique à Windows
    SOCKET socket_handle;
    struct sockaddr_in destination_address;
    struct sockaddr_in sender_address;
    int sender_address_length;
    char message_data[1024];
    int data_size, status;

    // Initialisation Winsock 2 - spécifique à Windows
    WSStartup(MAKEWORD(2, 2), &info);
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP client par l'exemple - Création

```
// Créer le socket UDP
socket_handle = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
// Préparer la structure avec l'adresse et le port de destination
memset(&destination_address, sizeof destination_address, 0);
destination_address.sin_family = AF_INET;
destination_address.sin_port = htons(27015); // Port net. byte order
destination_address.sin_addr.s_addr = inet_addr("127.0.0.1");

// PAS DE BIND !
// La structure destination_address sera utilisée à l'envoi
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP client par l'exemple - Création

```
// Gérer la communication
do
{
    printf("Message a envoyer:");
    fgets(message_data, sizeof message_data, stdin);
    // Envoyer les données
    status = sendto(socket_handle,
        message_data, strlen(message_data) + 1, 0,
        (struct sockaddr *)&destination_address,
        sizeof destination_address);
    // recevoir les données et l'adresse de l'émetteur
    sender_address_length = sizeof sender_address;
    data_size = recvfrom(socket_handle,
        message_data, sizeof message_data, 0,
        (struct sockaddr *)&sender_address, &sender_address_length);
    printf("Reponse recue:%s", message_data);
}
while (strcmp(message_data, "QUIT\n") != 0);
```

heig-vd

Programmation TCP/IP

L'API socket - Un socket UDP client par l'exemple - Terminaison

```
// ferme le socket (libérer les ressources)
closesocket(socket_handle);

// Finaliser Winsock 2 - spécifique Windows
WSACleanup();
}
```

heig-vd

Programmation TCP/IP

L'API socket - Echange de données TCP

- Principe similaire, mais avec connexion préalable
 - Le serveur doit se mettre à l'écoute des connexions
`listen(socket_server, BACK_LOG);`
 - Le serveur doit ensuite attendre et accepter les connexions
`new_socket = accept(socket_server,
(SOCKADDR*)&host_address, &host_address_length);`
 - Le client doit se connecter
`connect(socket_client,
(SOCKADDR *)&server_address, server_address_length);`
- La communication peut ensuite être gérée, mais
 - On utilise plutôt `send` pour envoyer
 - `send(new_socket, data, data_size, 0);`
 - Et `recv` pour recevoir
 - `recv(new_socket, buffer, buffer_size, 0);`

heig-vd

Programmation TCP/IP

L'API socket - Mode bloquant et non bloquant

- Par défaut, l'API socket est en mode bloquant
 - Un programme qui appelle `recv` ou `recvfrom` est bloqué jusqu'à ce que les données soient arrivées.
- Certaines options permettent de rendre l'API non bloquant
 - Dans ce cas, `recv` et `recvfrom` ne bloquent jamais.
 - S'il n'y a pas de données arrivées, elles renvoient la valeur spéciale `SOCKET_ERROR`.

heig-vd

Intégration électronique

Les solutions basées sur des ordinateurs embarqués

- Une large gamme d'ordinateurs et ordinateurs embarqués
 - Dispose des ports Ethernet
 - Supporte un système d'exploitation comportant la pile TCP/IP
 - Linux, Windows CE
- Exemples
 - Embedded PC
 - Exemple : Format ETX
 - Cartes embarquées spéciales

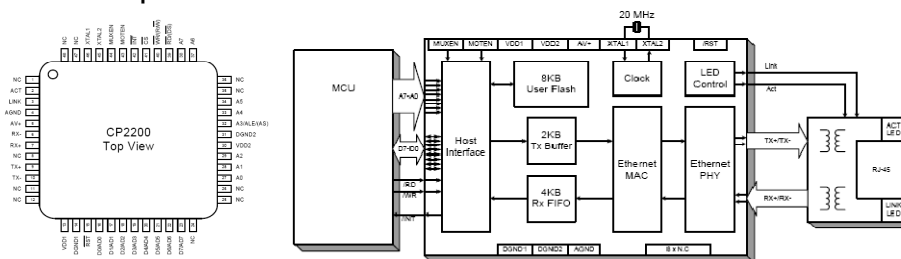


heig-vd

Intégration électronique

Les solutions à micro contrôleur

- De nombreux fabricants proposent des solutions Ethernet
 - Cœur micro contrôleur.
 - Circuit périphérique (parfois intégré) pour Ethernet.
 - Logiciel : pile de protocoles TCP prêts à l'emploi.
 - API de type socket.
- Exemple - Silabs CP 2200



Qu'avons-nous appris ?

- La couche physique d'Ethernet
 - Isolation galvanique, longue distance, performance.
- Les principes de base d'IP
 - Créer un sous réseau dédié au trafic contrôle commande.
- Les protocoles TCP et UDP
 - Préférer UDP pour un échange contrôle commande.
- L'API socket
 - Des fonctions de haut niveau prêtes à l'emploi.
- Quelques solutions pour l'intégration électronique

Vos questions

