Introduction aux Réseaux

Aurélien Esnard
Université de Bordeaux
MIAGE L3

aurelien.esnard@labri.fr
http://www.labri.fr/~esnard

Plan

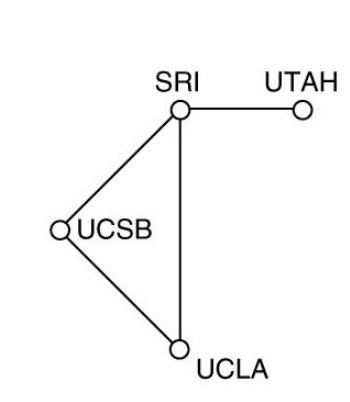
- 6 cours (2h)
  - Introduction
  - Couche Réseaux (IP)
  - Couche Transport (TCP)
  - Couche Liaison de données (Ethernet)
- 6 Tps (2h, CREMI)
  - Administration réseau avec QemuNet (IP, Routage, Firewall, ...)
- Evaluation
  - CC (mini DS de 1h)
  - Examen

Cours 1

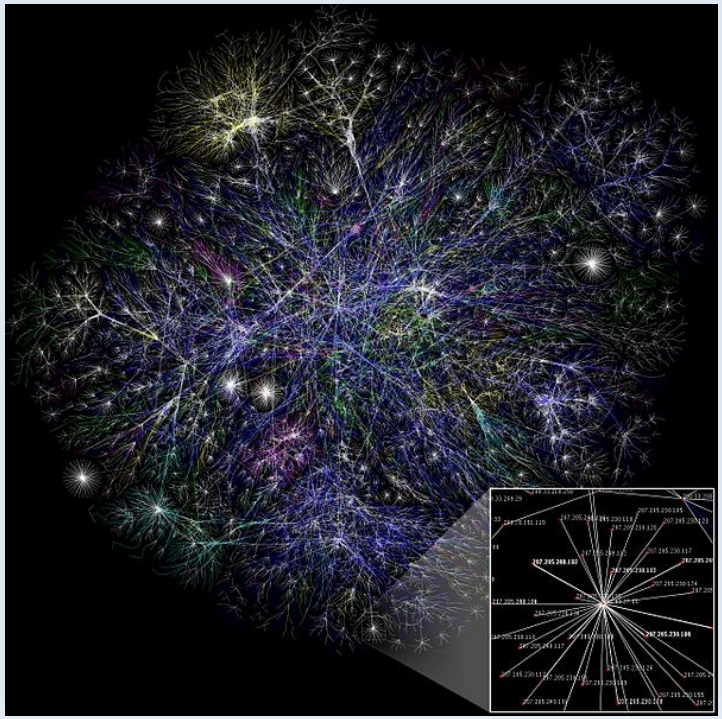
Introduction

Internet

- Internet : réseau informatique mondial, résultant de l'interconnexion d'une multitude de réseaux informatiques à travers la planète, unifiées grâce au protocole IP. [1983]
- Protocole réseau : un protocole définit de manière formelle et interopérable l'échange des informations entre ordinateurs.



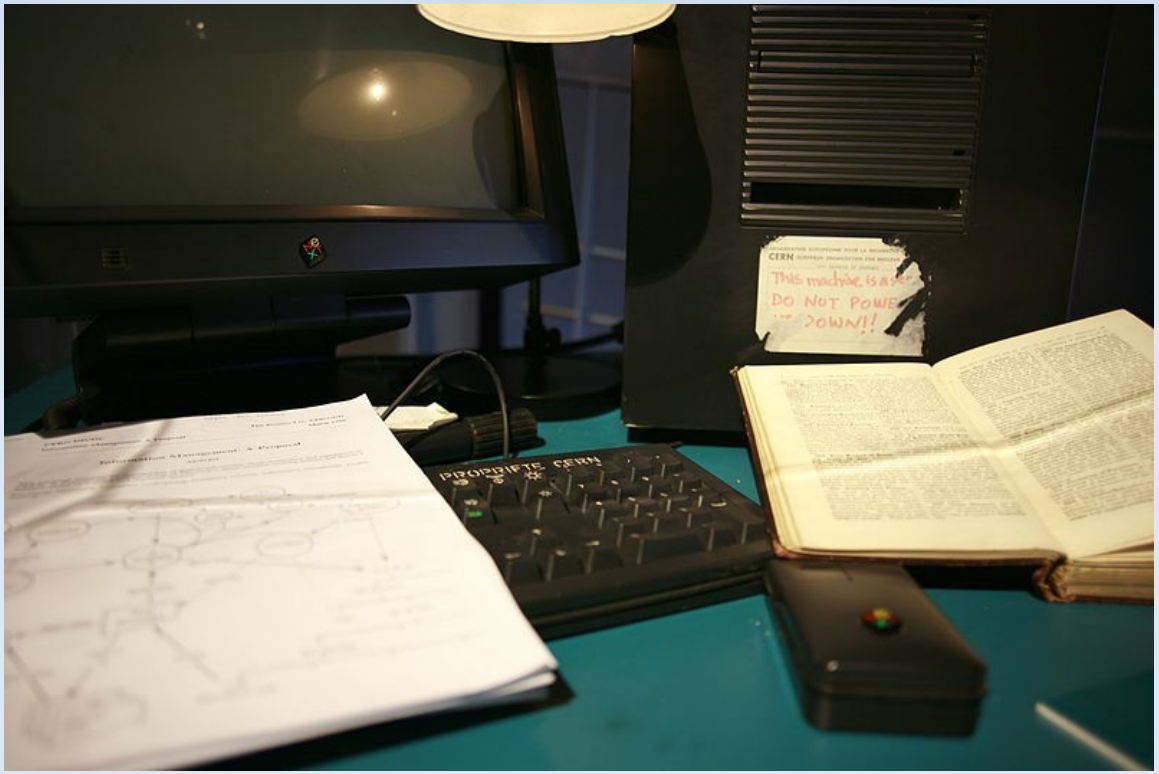
ARPANET. Source : Tannenbaum



Source : Wikipedia

Web

- Web (ou la toile) : l'ensemble des hyperliens (ou liens hypertextes) qui relient les pages web entre elles. [1990]



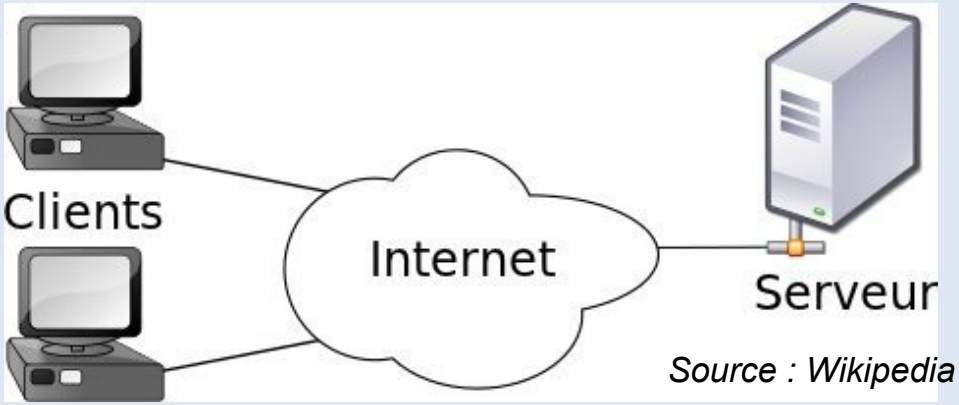
L'ordinateur utilisé au CERN par Tim Berners-Lee pour inventer le Web. Source : Wikipedia.

→ Ne pas confondre Internet et le Web, qui est un des nombreux services Internet !

Web

- Serveur Web : ordinateur qui contient les ressources du Web (pages, media, …) et les met à disposition sur Internet.
  - Ex. : www.google.com, fr.wikipedia.org, …

- Navigateur Web : logiciel (client du serveur Web) permettant de consulter les ressources du Web.
  - Ex. : Internet Explorer, Firefox, Chromium, …



Source : Wikipedia

Web

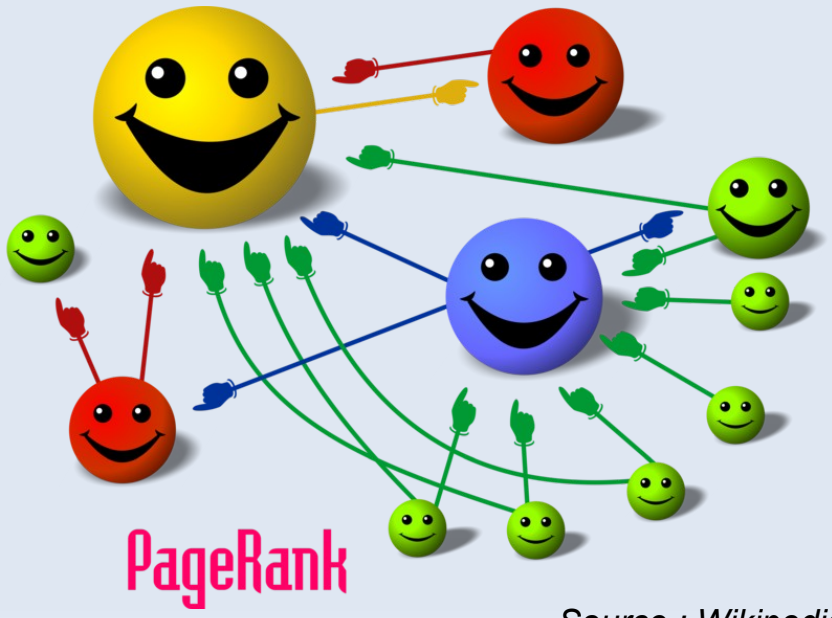
- HTTP (HyperText Transfert Protocol) : protocole de transfert des pages HTML permettant de naviguer sur le Web (HTTPS pour la version sécurisée).
- HTML (Hypertext Markup Language) : langage à balise pour représenter les pages Web (mise en forme, liens hypertextes, ressources multimédias, …).

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
 <head>
  <title>
   Exemple de HTML
  </title>
 </head>
 <body>
  Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
  <p>
   Ceci est un paragraphe où il n'y a pas d'hyperlien.
  </p>
 </body>
</html>
```

Source : Wikipedia

Moteur de Recherche

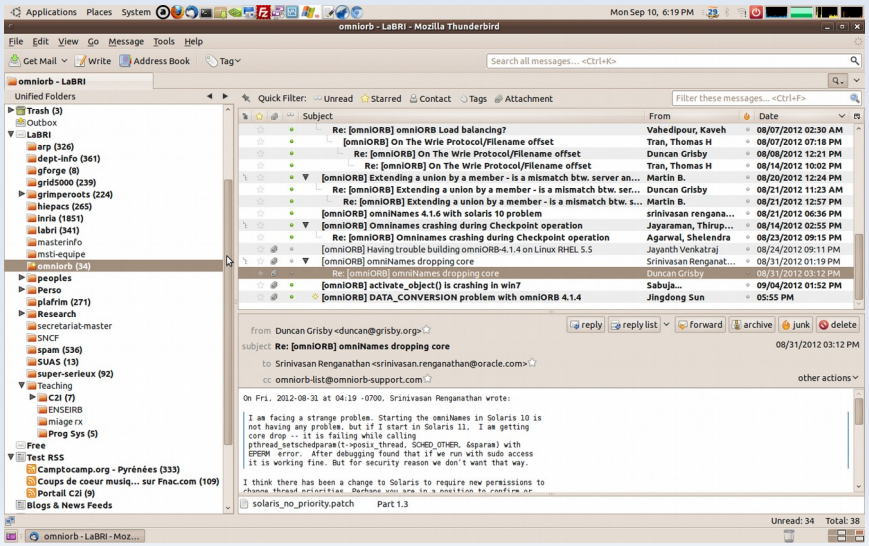
- Moteur de recherche : outil permettant de retrouver des pages Web à partir d'une requête
  - Ex. : Google, Yahoo Search, Bing, …
- Indexation automatique : les pages du Web sont parcourues automatiquement par un « robot » et analysées pour en extraire des mots-clés significatifs.
- Ordre des réponses : il dépend de l'adéquation des mot-clefs et de la « popularité » de la page web
  - nombre de liens vers la page (PageRank de Google)
  - les clics des utilisateurs sur la page de réponse



Source : Wikipedia

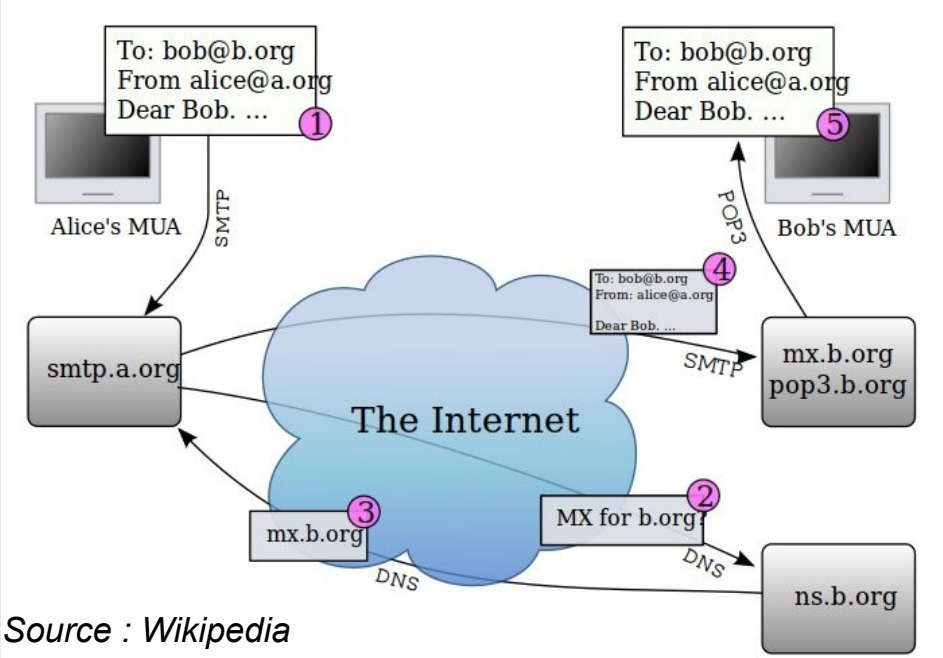
Messagerie Electronique

- Messagerie électronique : outil permettant d'échanger des messages (courriel ou *mail*) de manière asynchrone par l'intermédiaire d'une boîte à lettres électronique identifiée par une adresse électronique.
- Adresse életronique : prenom.nom@etu.u-bordeaux.fr
- Client de messagerie local ou application webmail
  - Ex. : Thunderbird, Outlook, … vs Gmail, Yahoo!, ...



Messagerie Electronique

- Principe d'acheminement d'un courriel
  - Envoi : lorsqu'un expéditeur envoie un courriel, son ordinateur soumet une reqûete au serveur sortant (SMTP), qui l'achemine vers le serveur entrant du destinataire
  - Réception : lorsqu'un destinataire relève ses courriels, ils sont téléchargés sur son ordinateur depuis le serveur entrant (POP3 ou IMAP)



Source : Wikipedia

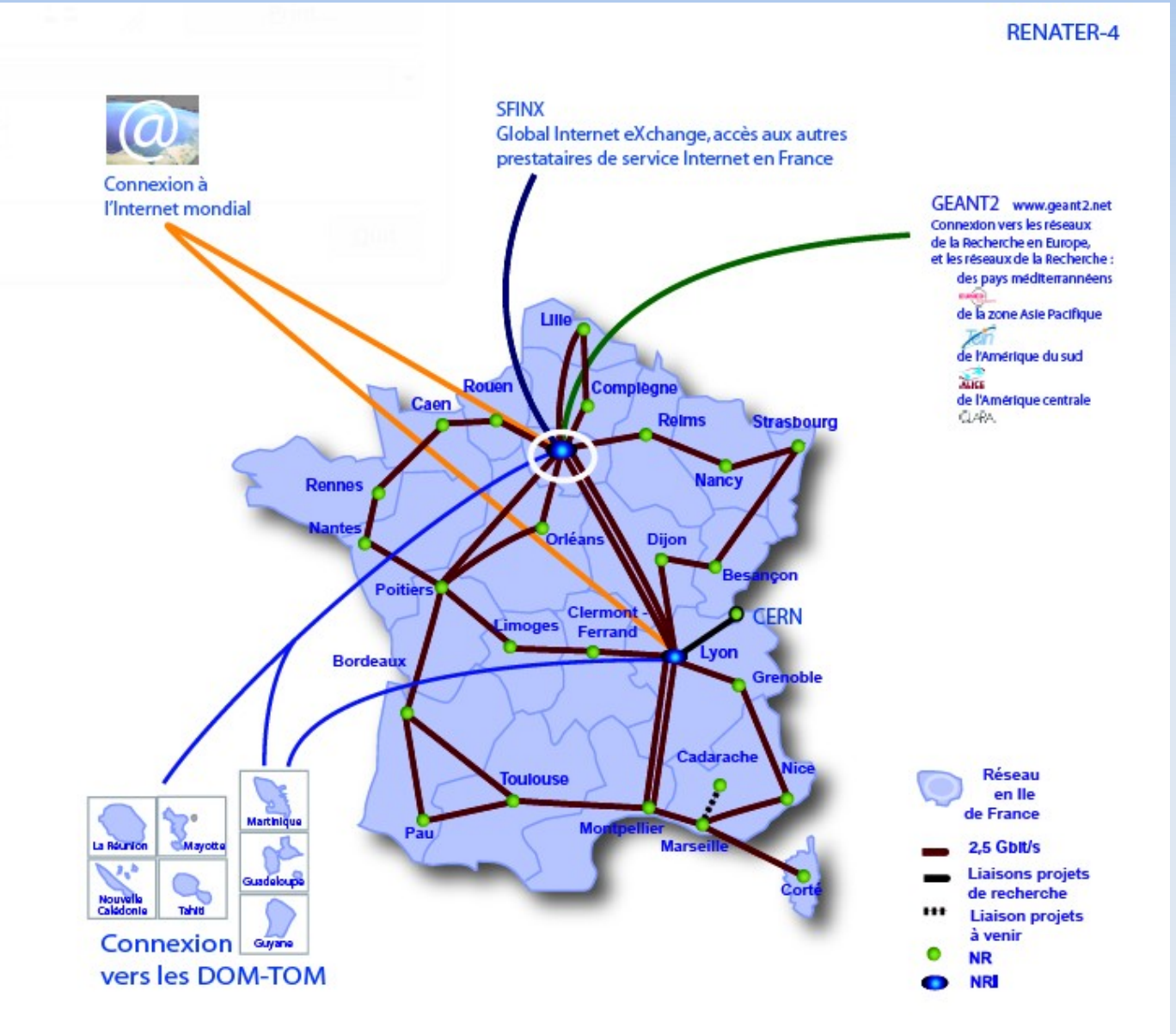
Performance des réseaux

- Débit
  - nombre de bits que le réseau peut transporter par seconde
- Latence
  - nombre de secondes que met le premier bit pou aller de la source à la destination
- Quelques exemples de débits (en bit/s)
  - modem RTC 56K, ADSL (1M à 8M)
  - Ethernet (10M, 100M, 1G, 10G), ATM (155M), FDDI (100M), ...
  - sans-fil : IEEE 802.11 (11M à 54M)
  - GSM : 3G (144K-1,9M), EDGE (64k-384k), 3G+ (3,6M, 14,4M), 4G (100M-1G), ...

Carte Internet

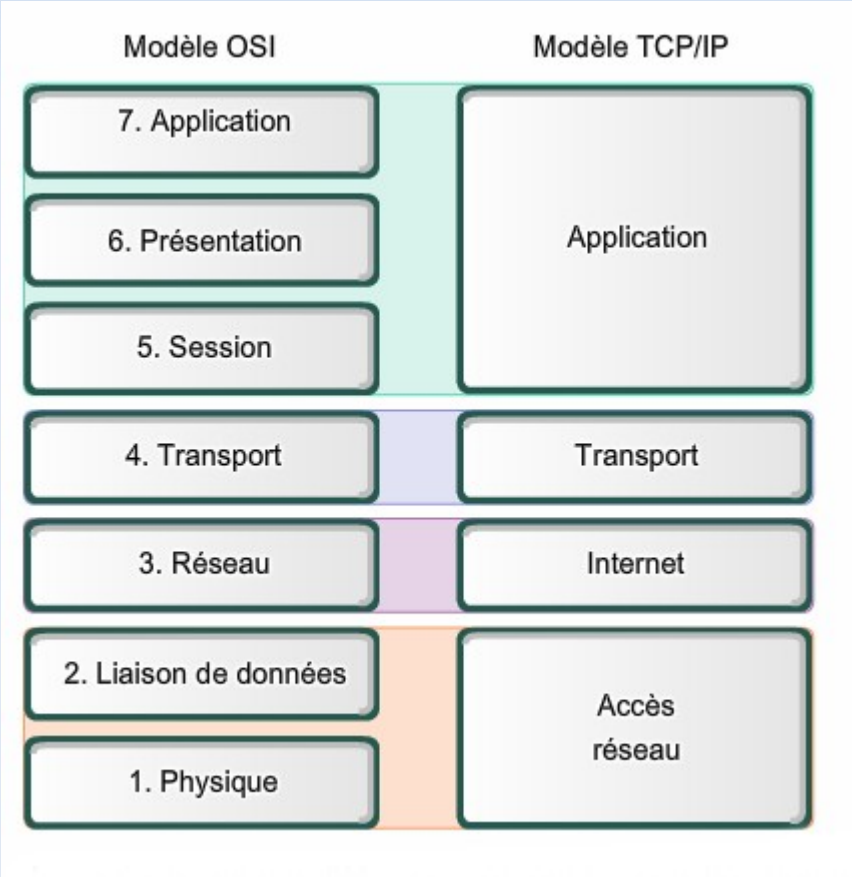
- todo

Exemple de Renater



Modèle OSI & TCP/IP

- Protocole
  - spécification de plusieurs règles pour communiquer sur une même couche d'abstraction entre deux machines différentes



Les couches OSI

(1) Couche physique (physical layer)

- transmission effective des signaux entre les interlocuteurs
- service typiquement limité à l'émission et la réception d'un bit ou d'un train de bit continu

(2) Couche liaison de données (datalink layer)

- communications entre 2 machines adjacentes, i.e. directement reliés entre elle par un support physique

(3) Couche réseaux (network layer)

- communications de bout en bout, généralement entre machines : adressage logique et routage des paquets

(4) Couche transport (transport layer)

- communications de bout en bout entre processus

Les couches OSI

(5) Couche session (session layer)

- synchronisation des échanges et transaction, permet l'ouverture et la fermeture de session

(6) Couche présentation

- codage des données applicatives, et plus précisément conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises

(7) Couche application

- point d'accès aux services réseaux
- elle n'a pas de service propre spécifiable et entrant dans la portée de la norme

Piles de protocoles

- À remplacer...

Cours 2

Couche Réseaux
(IP)

Introduction

- Internet Protocol (IP)
  - communication de bout en bout entre des machines qui ne sont pas connectés directements, c'est-à-dire situées dans des réseaux locaux différents (géographie, technologie)
  - adressage logique : identifier les machines indépendament de l'adressage physique (Ethernet, ...)
  - routage : acheminement des données entre les réseaux via des routeurs/passerelles intermédiaires
- Versions
  - IPv4, RFC 791, sept. 1981 (2^{32} adresses)
  - IPv6, le successeur de IPv4, RFC 2460, déc. 1998 (2^{128} adresses)

IPv6

- Adresse IPv6 (8 groupe de 2 octets, noté en hexa)
  - Exemple : 2001:0db8:0000:85a3:0000:0000:ac1f:0001
  - Forme canonique : 2001:db8::85a3:::ac1f:1

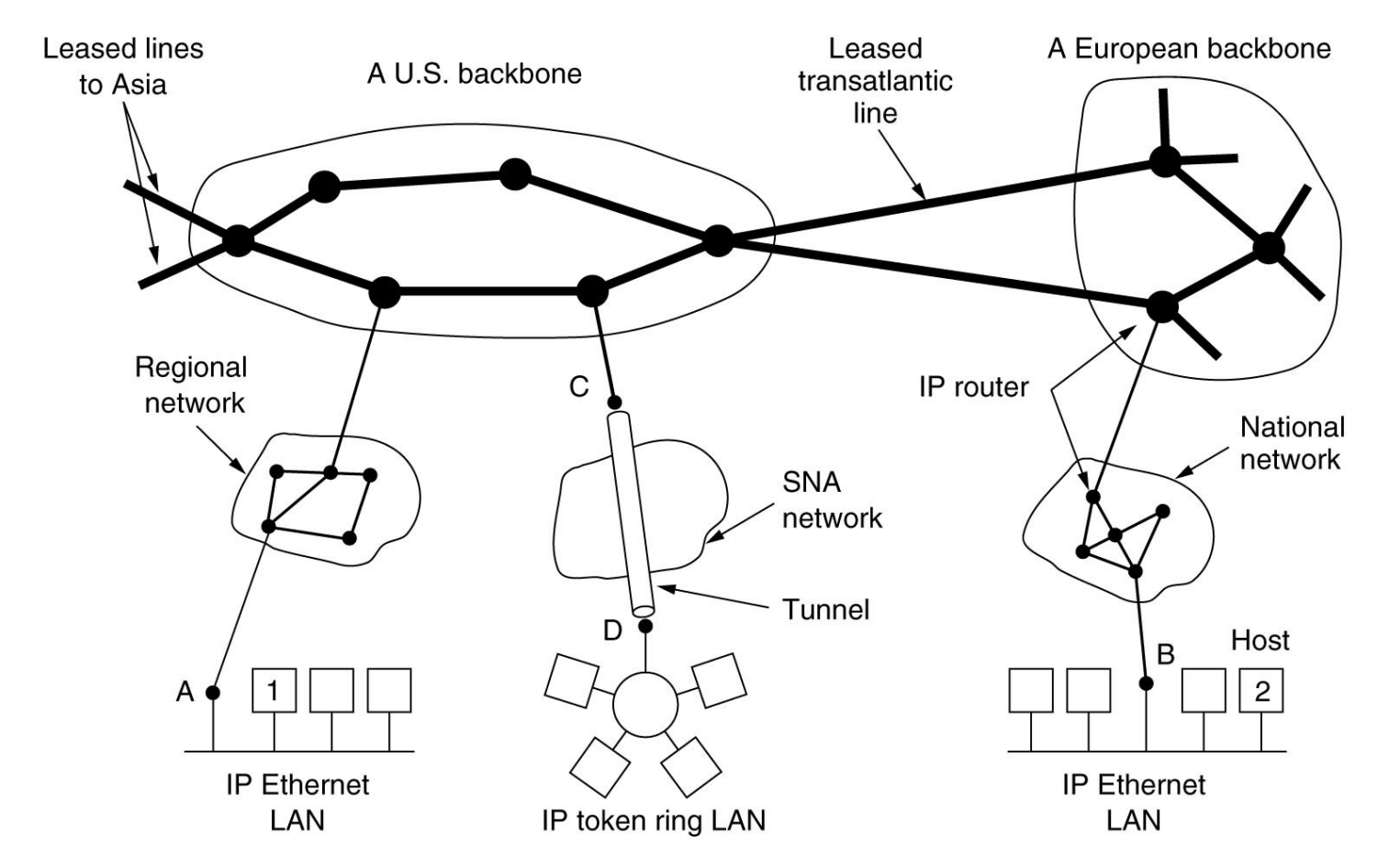
- Epuisement des adresses IPv4
  - En février 2011, IANA annonce qu'il n'y a plus de bloc d'adresse libre !

- Combien d'adresse IPv6 par mm^2 de surface terrestre ?
  - 667 millions de milliards d'appareils connectés sur chaque millimètre carré de la surface de la Terre !

- A compléter : https://fr.wikipedia.org/wiki/IPv6

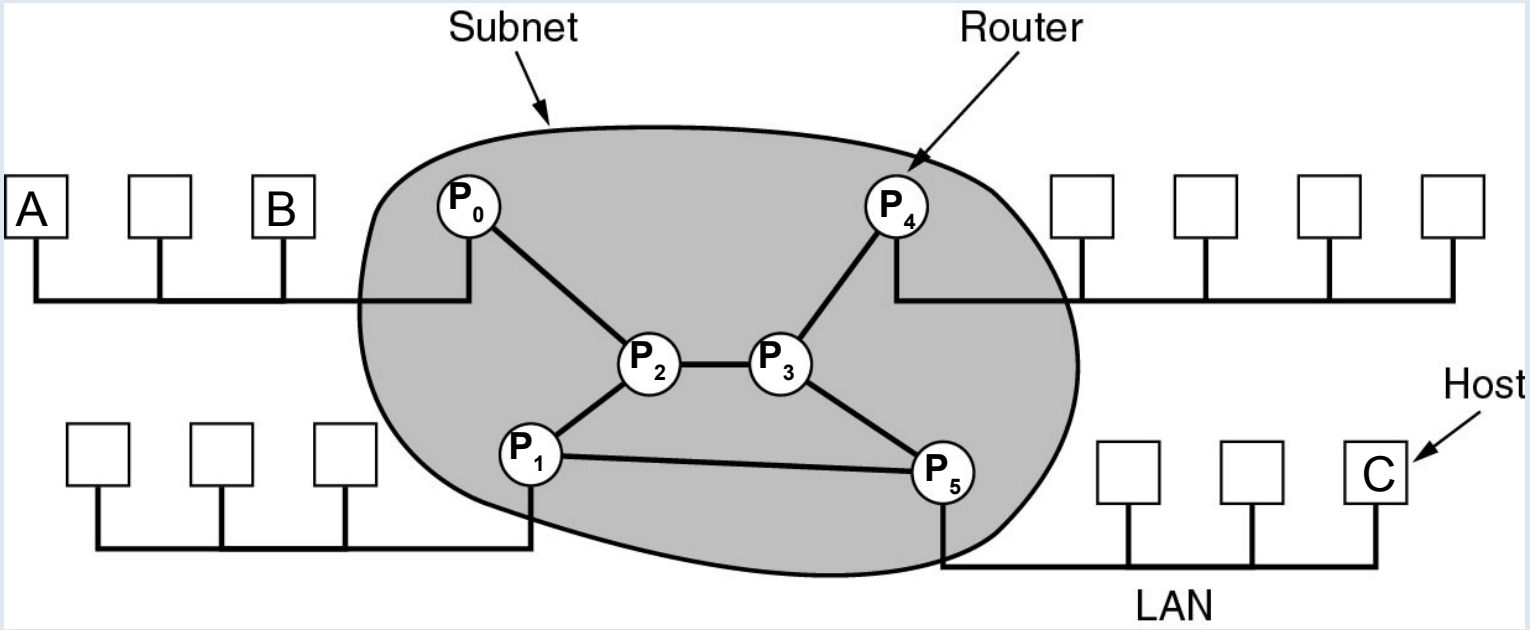
Le réseau Internet

- Interconnexion de multiples réseaux hétérogènes et distants...



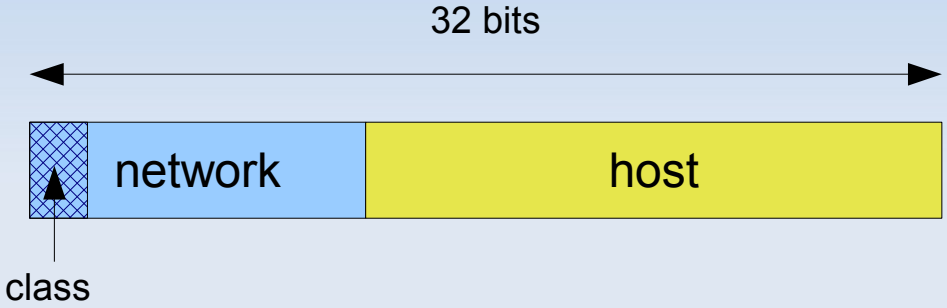
Introduction

- Communication directe de A vers B
- Communication de A vers C via P_0, puis P_2, ...
  - la passerelle permet de passer d'un réseau à un autre ; elle possède donc deux interfaces réseau



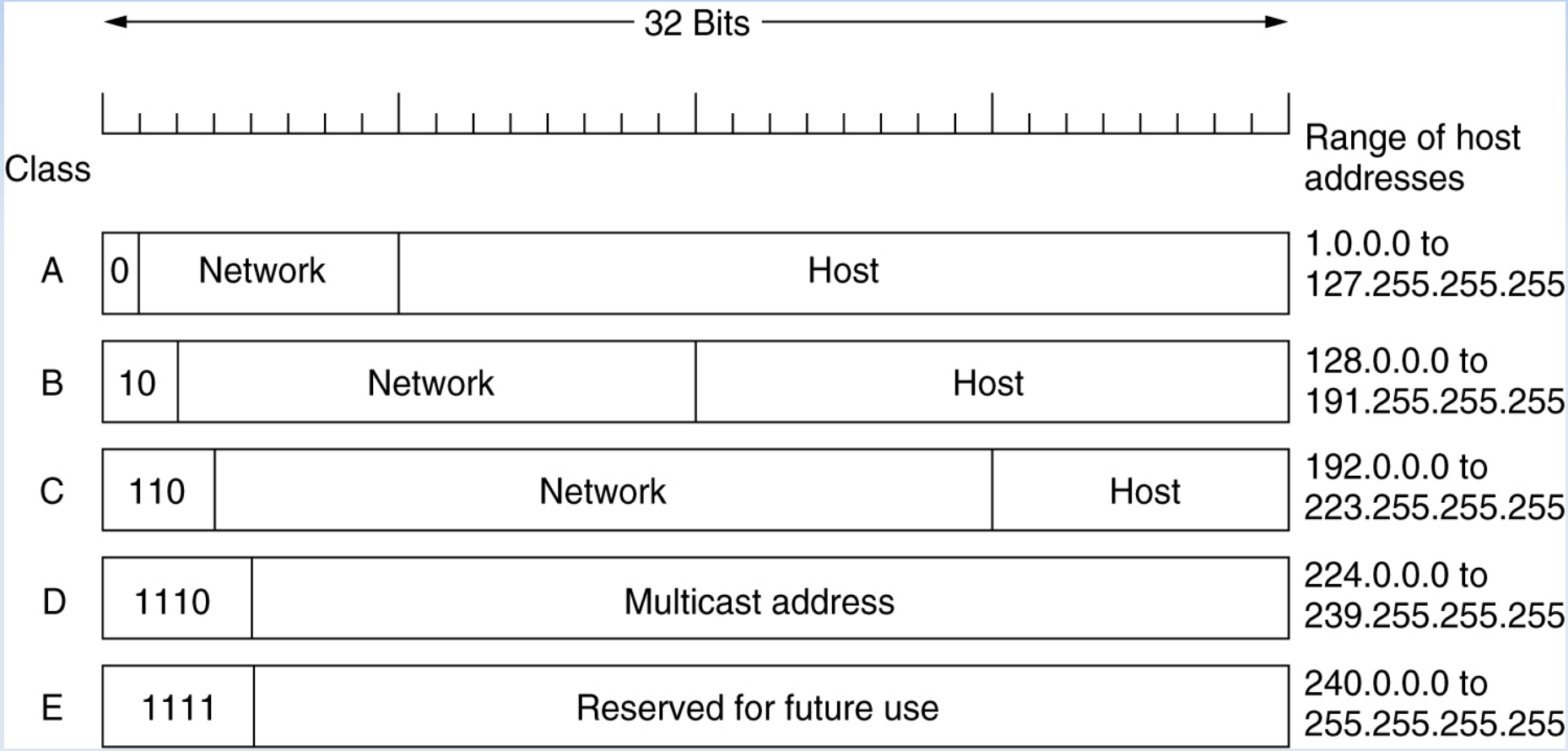
Adressage IP

- Format des adresses IP (32 bits)
  - 2^{32} adresses, environ 4 milliards d'adresses



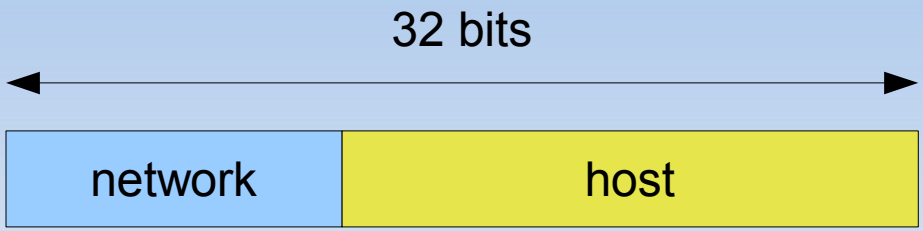
- Les 5 classes d'adresse IP
  - classes générales A, B, C (unicast)
    - classe A : 8 bits network, 24 bits host (grands réseaux)
    - classe B : 16 bits network, 16 bits host (moyens réseaux)
    - classe C : 24 bits network, 8 bits host (petits réseaux)
  - classe D (multicast)
  - classe E (réservé pour un usage futur)

Adressage IP



Adressage IP

- Format des adresses IP



- Les adresses spéciales
  - Adresse de la boucle locale (loopback) : 127.0.0.1 ou *localhost*
  - Adresse d'un réseau : tous les bits de l'adresse hôte à 0

    192.168.10.0 / 24
  - Adresse de diffusion d'un réseau : tous les bits de l'adresse hôte à 1

    192.168.10.255
  - Adresse du routeur (par convention) : adresse de diffusion – 1
- Les masques
  - 255.255.255.0 ↔ /24

Adressage IP

- Exercice 2.1
  - Compléter le tableau suivant...

Adresse IP Hôte	Classe d'adresses	Adresse Réseau	Adresse Hôte	Adresse de broadcast réseau	Masque de sous-réseau
216.14.55.137					
123.1.1.15					
175.12.239.244					

Adressage IP

- Exercice 2.1 (correction)

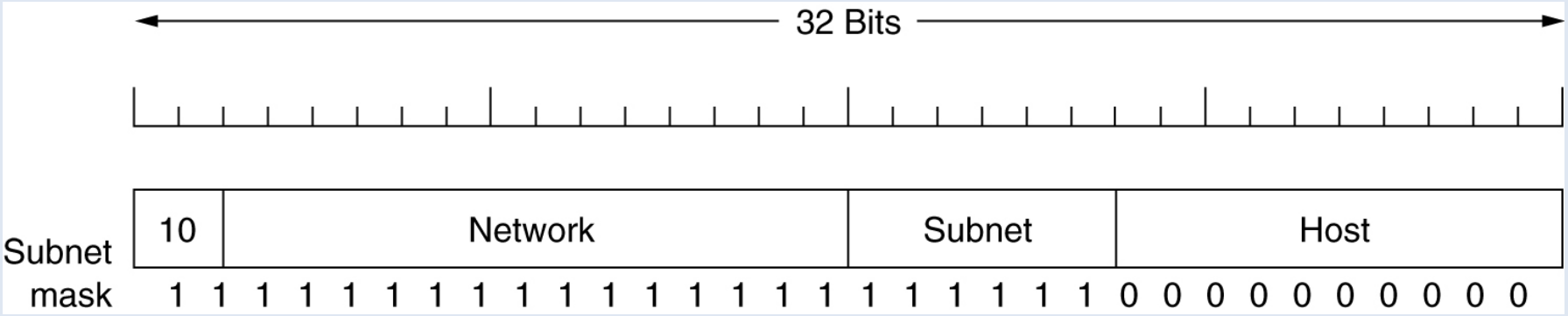
Adresse IP Hôte	Classe d'adresses	Adresse Réseau	Adresse Hôte	Adresse de broadcast réseau	Masque de sous-réseau
216.14.55.137	C	216.14.55.0	.137	216.14.55.255	255.255.255.0
123.1.1.15	A	123.0.0.0	1.1.15	123.255.255.255	255.0.0.0
175.12.239.244	B	175.12.0.0	239.244	175.12.255.255	255.255.0.0

Réseau privé

- Besoin de réseaux privés
  - sécurité : réseau inaccessible depuis l'extérieur (Internet)
  - palier le manque d'adresse dans IPv4
    - seulement 256 adresses publiques disponibles pour un réseau acheté de classe C
- Adresses privées (utilisables uniquement en interne)
  - classe A : 10.0.0.0 – 10.255.255.255 (1 réseau)
  - classe B : 172.16.0.0 – 172.31.255.255 (16 réseaux)
  - classe C : 192.168.0.0 – 192.168.255.255 (256 réseaux)
- Seule la passerelle d'un réseau privé nécessite de posséder une adresse Internet publique !
  - NAT (Network Address Translation)

Sous-réseaux

- Délimitation de plusieurs sous-réseaux dans un réseau
  - Adresse IP découpée en trois parties (network, subnet, host)
  - On utilise une partie des bits de l'hôte pour identifier le sous-réseau (subnet).



- Masque de sous-réseau
  - Le masque du sous-réseau s'obtient en mettant à 1 tous les bits du réseau et du sous-réseau, puis le reste à 0.
  - (adresse IP) AND (masque) = (adresse sous-réseau)

Sous-Réseaux

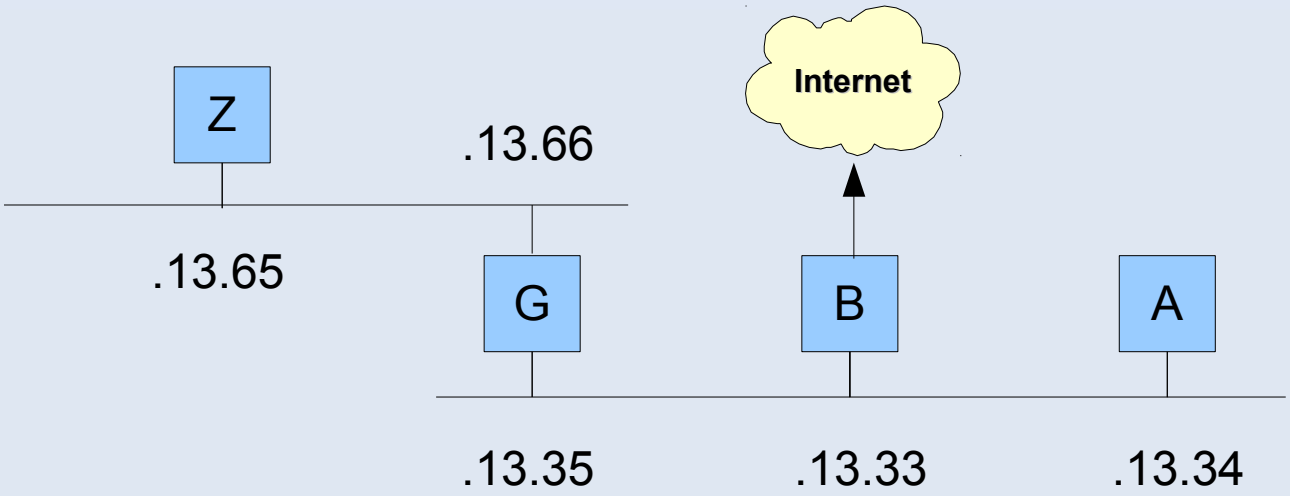
- Exercice 2.2
  - Dans un réseau de classe C d'adresse IP 193.51.199.0, on souhaite constituer 5 sous-réseaux.
  - Combien de bits sont nécessaires pour coder ces sous-réseaux ?
  - Combien de machines trouve-t-on dans chaque sous réseau ?
  - Quel est le masque de réseau et de sous-réseau ?
  - A quel adresse de sous-réseau appartient la machine 193.51.199.67 ?
  - Donner l'adresse de diffusion correspondant à ce sous-réseau ?
  - Quel sont les adresses des autres sous-réseaux ?

Sous-Réseaux

- Exercice 2.2 (correction)
  - Il faut 3 bits pour coder la partie sous-réseau à choisir parmi 000, 001, 010, 011, 100, 101, 110 et 111 (nb max de sous-réseaux avec n bits = 2^n)
  - Le masque du réseau de classe C est 255.255.255.0.
  - Le masque du sous-réseau est 255.255.255.224 car 224 = $(1110\ 0000)_{binaire}$
  - adresse du réseau
    - 193.51.199.67 AND 255.255.255.0 = 193.51.199.0
  - adresse sous-réseau
    - 193.51.199.67 AND 255.255.255.224 = 193.51.199.X
    - X = 67 AND 224 = $(010\ 00011)_{binaire}$ AND $(111\ 00000)_{binaire}$
    - X = 010 0000 = 64 ; adresse sous-réseau = 193.51.199.64

Sous-Réseaux

- Exercice 2.3
  - On considère le réseau de classe B d'adresse IP 140.252.0.0. Ce réseau local est composé de deux sous-réseaux d'adresses 140.252.13.32 et 140.252.13.64.



  - Quel est le masque de chaque sous-réseau et son adresse de broadcast ?

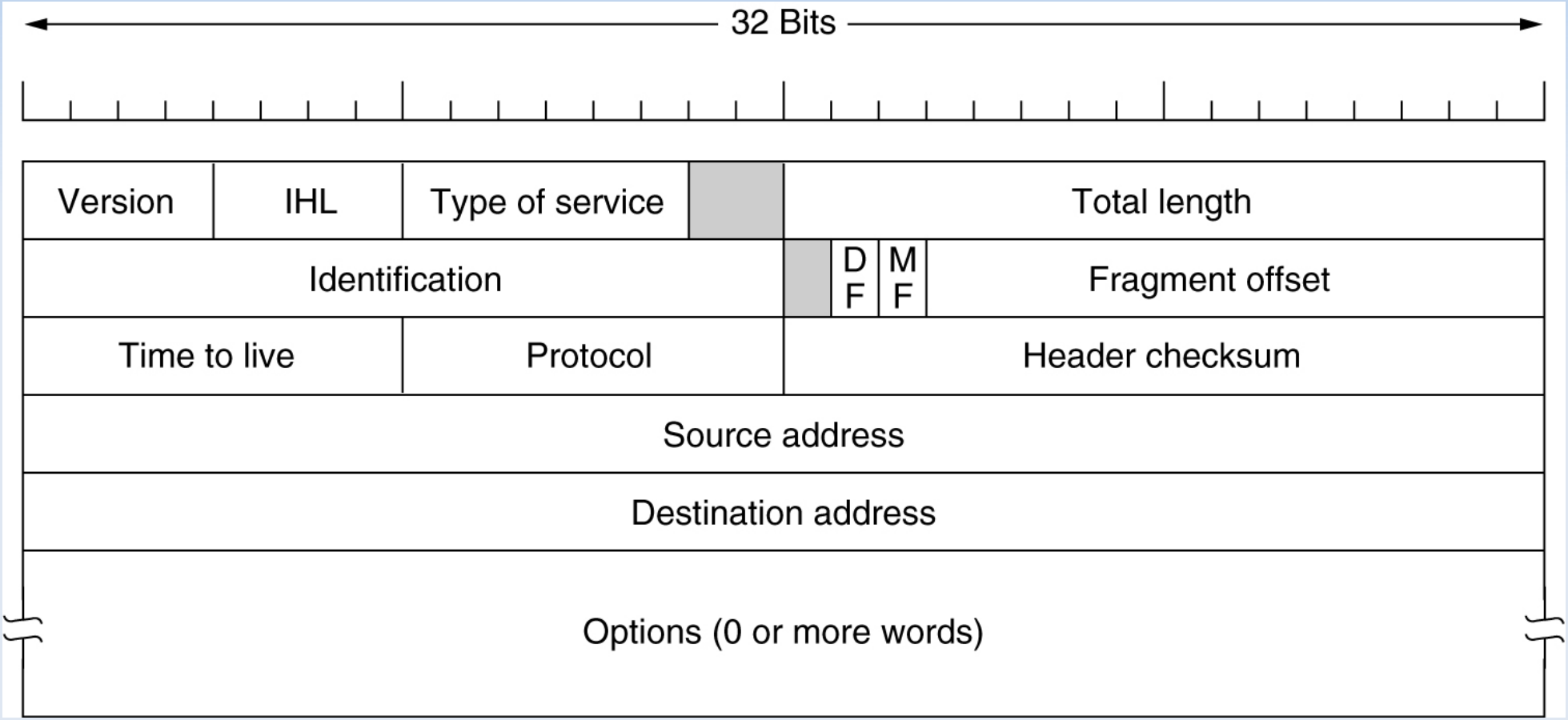
Sous-Réseaux

- Exercice 2.3 (correction)
  - Calcul du masque
    - sous-réseaux 140.252.13.32 et 140.252.13.64
    - on a 32_{10} = 0010 0000_2 et 64_{10} = 0100 0000_2
    - donc masque commun de 27 bits (255.255.255.224), car 1110 0000_2 = 128 + 64 + 32 = 224_{10}
  - Calcul de l'adresse de broadcast (adresse maxi du sous-réseau)
    - @Broadcast(Rx/27) = @Rx AND 0000 ... 0001 1111_2
    - Il suffit d'ajouter 0001 1111_2 = 31_{10} à @Rx
    - @Broadcast(140.252.13.32/27) = 140.252.13.63
    - @Broadcast(140.252.13.64/27) = 140.252.13.95

Sous-Réseaux : VLSM

- Sous-réseaux de tailles variables
  - VLSM = Variable-Length Subnet Masking...
  - http://www.iprezo.org/index.php?page=vlsm
  - A compléter...

En-tête du paquet IP (v4)



En-tête du paquet IP (v4)

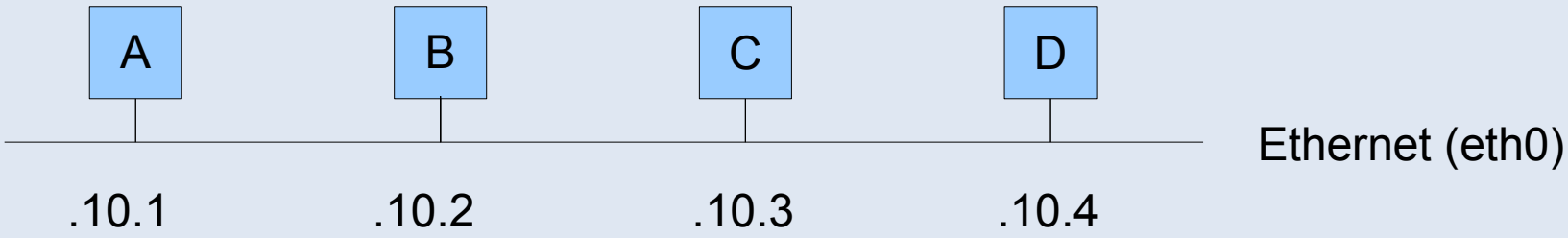
- Version : v4
- IHL (Internet Header Length) : longueur de l'en-tête en mot de 32 bits
- Type of Service : qualité de service (minimal cost: 0x02, reliability: 0x04, throughput: 0x08, low delay: 0x10)
- Identification : identifiant d'un ensemble de fragments pour leur rassemblage
- Flags : DF (Don't Fragment) / MF (More Fragment)
- Fragment Offset : position du fragment dans le message
- Time To Live (TTL) : temps de vie maximal en sec.
- Protocol : protocole de la couche supérieur encapsulé dans le paquet (ICMP, UDP, TCP, etc.)
- Header Checksum : contrôle d'erreurs de l'en-tête
- Adresses IP source et destination

Administration : réseau IP

- Configuration du réseau 192.168.10.0/24
  - Configuration de la machine A (masque de 24 bits)

    $ ifconfig eth0 192.168.10.1 netmask 255.255.255.0
  - De même pour toutes les machines B, C et D
  - On peut ensuite effectuer des tests avec 'ping'

    $ ping 192.168.10.2



Administration : réseau IP

- Configuration du réseau 192.168.10.0/24
  - Configuration des interfaces réseaux de la machine A ?

$ Ifconfig -a

eth0 Link encap:**Ethernet** HWaddr **00:15:c5:3d:52:b6**
inet addr:**192.168.10.1** Bcast:**192.168.10.255** Mask:**255.255.255.0**
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:66 errors:0 dropped:0 overruns:0 frame:0
TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7571 (7.3 KB) TX bytes:9560 (9.3 KB)
Interrupt:18

eth1 Link encap:**Ethernet** HWaddr **00:13:02:dc:2a:fd**
...

lo Link encap:**Local Loopback**
inet addr:**127.0.0.1** Mask:**255.0.0.0**
...

Adresse Ethernet

Adresse IP de la machine

Masque du Réseau IP

Routage

- Principe
  - Mécanisme par lequel le message d'un expéditeur est acheminé jusqu'à son destinataire, même si aucun des deux ne connaît le chemin complet que le message doit suivre...

- Deux types logiques d'ordinateur dans le WAN
  - les hôtes (hosts) ou stations, qui sont reliés à un seul réseau et qui ont par conséquent une table de routage simple
  - les routeurs/passerelles (gateway), qui relient au moins deux réseaux et possèdent une table de routage plus complexe

Routage

- Routage statique et dynamique
  - statique (pour les stations)
  - dynamique (pour les routeurs) : adaptation de la table de routage pour optimiser les chemins et réagir aux pannes

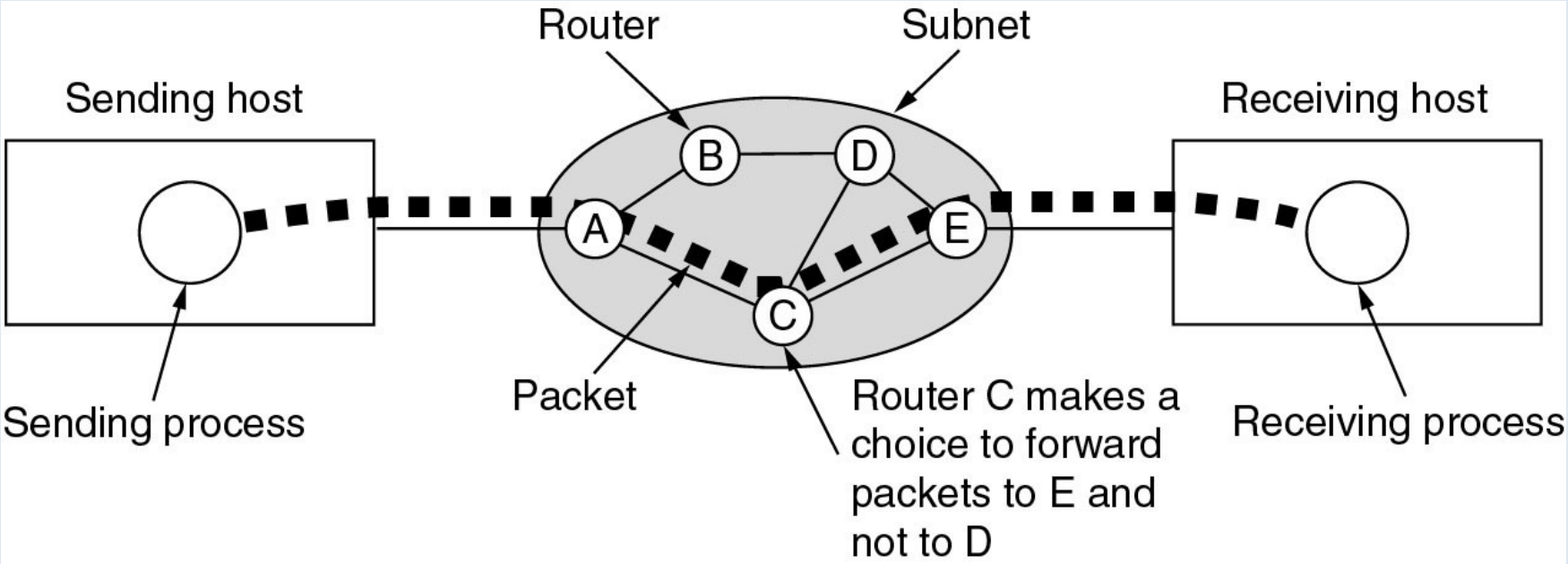


Table de routage

- Principe
  - U : la route est active (Up)
  - G : route indirecte qui passe par un routeur (Gateway)
    - sinon route directe (pas G)
  - H : l'adresse destination est une adresse de machine (Host)
    - sinon l'adresse destination est celle d'un réseau (pas H)

$ route -n

Destination	*Gateway*	*Genmask*	*Flags*	*Interface*
140.252.13.64	140.252.13.35	255.255.255.224	UG	eth0
127.0.0.1	*	0.0.0.0	UH	lo
140.252.13.32	*	255.255.255.224	U	eth0
default	140.252.13.33	0.0.0.0	UG	eth0

- Exercice 2.4
  - Donner la signification de chaque ligne de la table de routage...

Exemple...

- Exercice 2.4 (correction)
  - Ligne 1 : Indique que pour atteindre les machines du réseau 140.252.13.64 (machines G et Z), il faut passer par le routeur G d'adresse 140.252.13.35.
  - Ligne 2 : Indique que pour envoyer un message à soi-même, il suffit d'envoyer ce message à l'adresse 127.0.0.1 (loopback)
  - Ligne 3 : Indique que toutes les machines du réseau 140.252.13.32 (pas de flag H) peuvent être atteintes directement.
  - Ligne 4 : Indique que lorsque l'on ne sait pas comment atteindre une destination (une machine Internet par exemple), il faut envoyer le message au routeur 140.252.13.33 (default).

Algorithme de routage statique

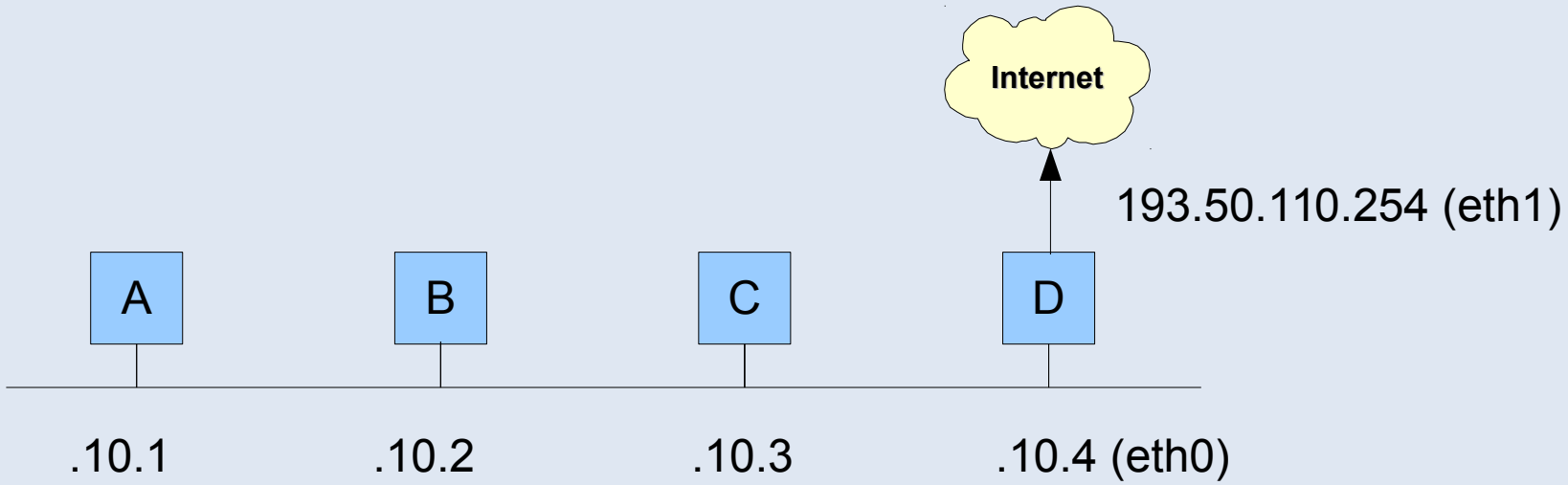
- Principe
  - On regarde les routes “Up” telles que :
    - l'adresse dans la table de routage soit la même que l'adresse destination, si l'adresse dans la table est l'adresse d'une machine (flag H).
    - l'adresse dans la table de routage soit la même que “adresse destination” AND “masque réseau”, si l'adresse dans la table est l'adresse d'un réseau (pas de flag H)
  - Si la route est directe (pas de flag G), le paquet est envoyé directement au destinataire. Sinon, le paquet est envoyé au routeur pour atteindre l'adresse du destinataire.

Administration : routage

- Configuration d'une passerelle D pour le réseau 192.168.10.0/24 permettant d'accéder à Internet
  - Activer la machine comme passerelle (IP Forward)

    $ echo 1 > /proc/sys/net/ipv4/ip_forward
  - Configuration d'une route par défaut vers l'extérieur...

    $ route add default gw 192.168.10.4



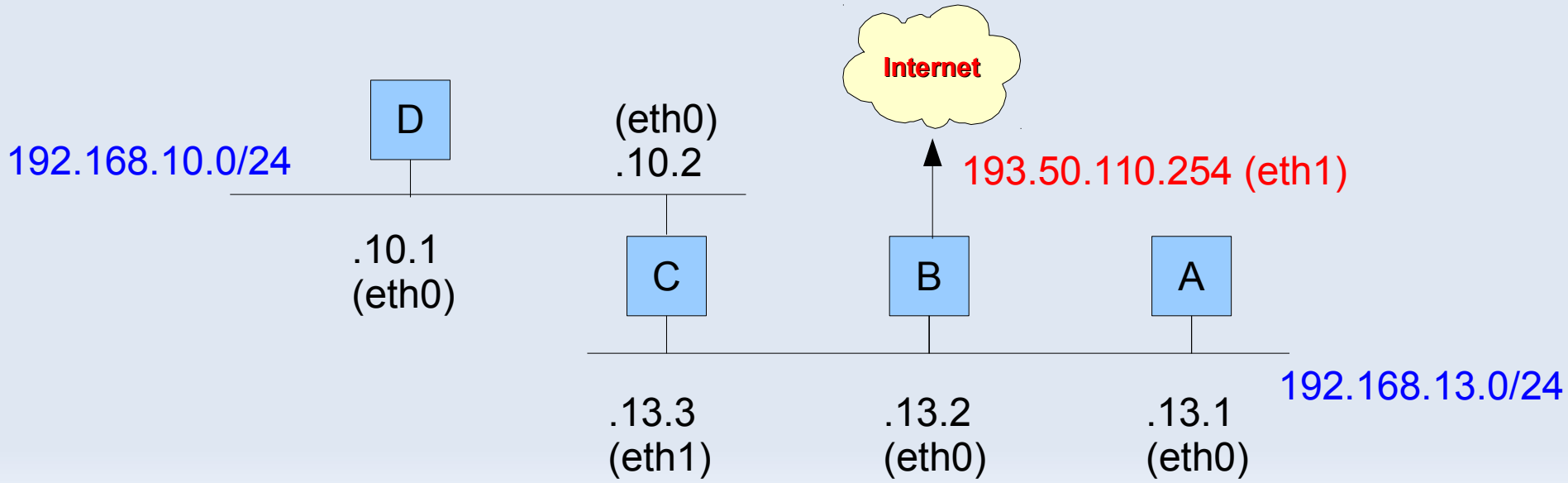
Administration : routage

- Pour les machines de 192.168.10.0/24, C joue le rôle de passerelle par défaut

```
root@D$ route add default gw 192.168.10.2
```

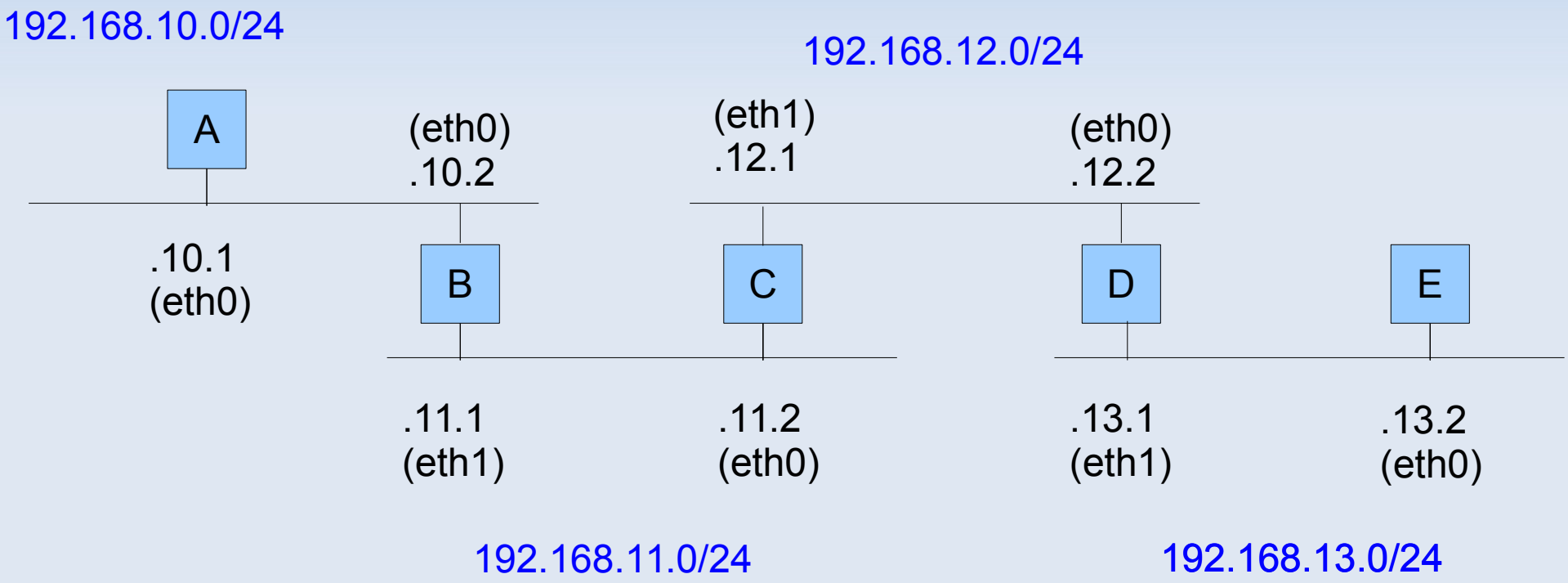
- Pour 192.168.13.0/24, C joue le rôle de passerelle vers 192.168.10.0/24 et B joue le rôle de passerelle par défaut

```
root@A$ route add -net 192.168.10.0 netmask 255.255.255.0 gw 192.168.13.3
root@A$ route add default gw 192.168.13.2
...
```



Exercice

- Ecrire les règles de routage pour toutes les machines



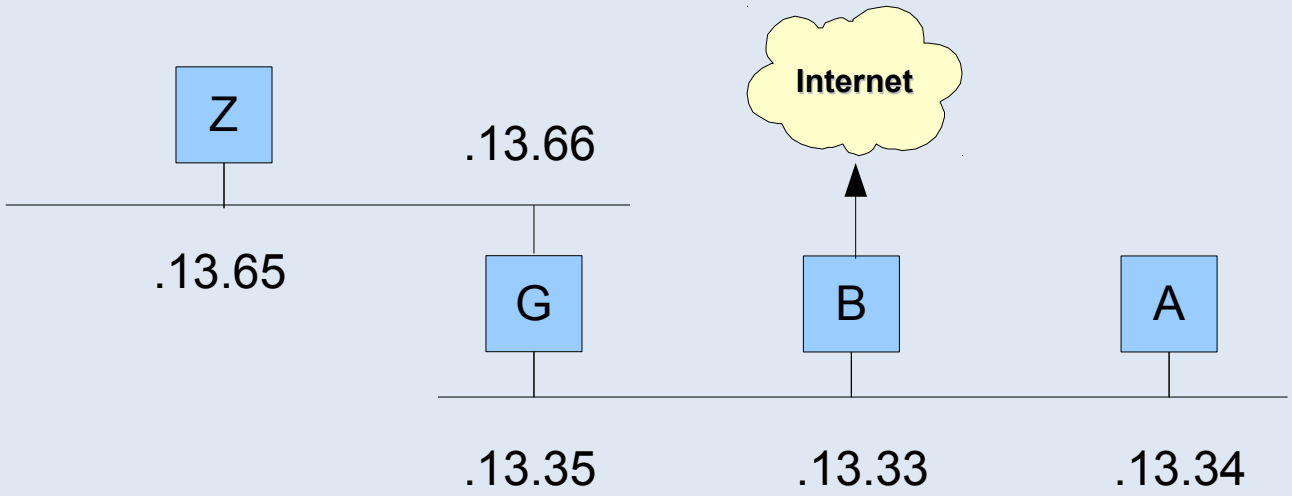
ARP

- Address Resolution Protocol (ARP), RFC 826
  - Récupérer l'adresse Ethernet correspondant à une adresse IP lorsque celle-ci n'est pas connue par une station (broadcast de la requête “arp who has $@IP_{dest}$; answer $@E_{source}$”)
  - Mise à jour du cache ARP dans le kernel de la station
    - association @IP / @E

- Attaque ARP-spoofing
  - L'attaquant X répond à la requête “arp who has $@IP_Y$” par l'adresse $@E_X$ le plus rapidement possible (avant Y) afin de falsifier les caches ARPs des stations sur le réseau...
  - Ainsi l'attaquant X recevra les messages destinés à $@IP_Y$

Routage statique et ARP

- Notation
  - On note @E(@IP) l'adresse Ethernet correspondant à l'adresse Internet @IP.
  - On note ($@E_{source}$,$@E_{dest}$,$@IP_{source}$,$@IP_{dest}$) un paquet IP est émis de *source* vers *dest*.



- Exercice 2.5
  - Représentez avec cette notation la trame envoyée de A vers G.
  - Même chose de A vers Z.

Routage statique et ARP

- Exercice 2.5 (correction)
  - Cas de A vers G
    - récupérer l'adresse Ethernet de G : arp “who has @IP_G”, réponse “@E(@IP_G)”
    - envoi de la trame : “@E(@IP_A),@E(@IP_G),@IP_A,@IP_G”
  - Cas de A vers Z (route indirecte par G)
    - pas besoin de récupérer l'adresse Ethernet de G, déjà dans la cache ARP
    - envoi de la trame : “@E(@IP_A),@E(@IP_G),@IP_A,@IP_Z”
    - mise à jour de la trame par le routeur (sans toucher au paquet IP) : “@E(@IP_G),@E(@IP_Z),@IP_A,@IP_Z”

ICMP

- Internet Control Message Protocol (ICMP), RFC 792
  - accompagne IP pour gérer les erreurs et propager des informations de routage

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

  - Exemple du ping : envoi d'une requête ICMP *'echo request'* et attente de la réponse *'echo reply'*

Cours 3

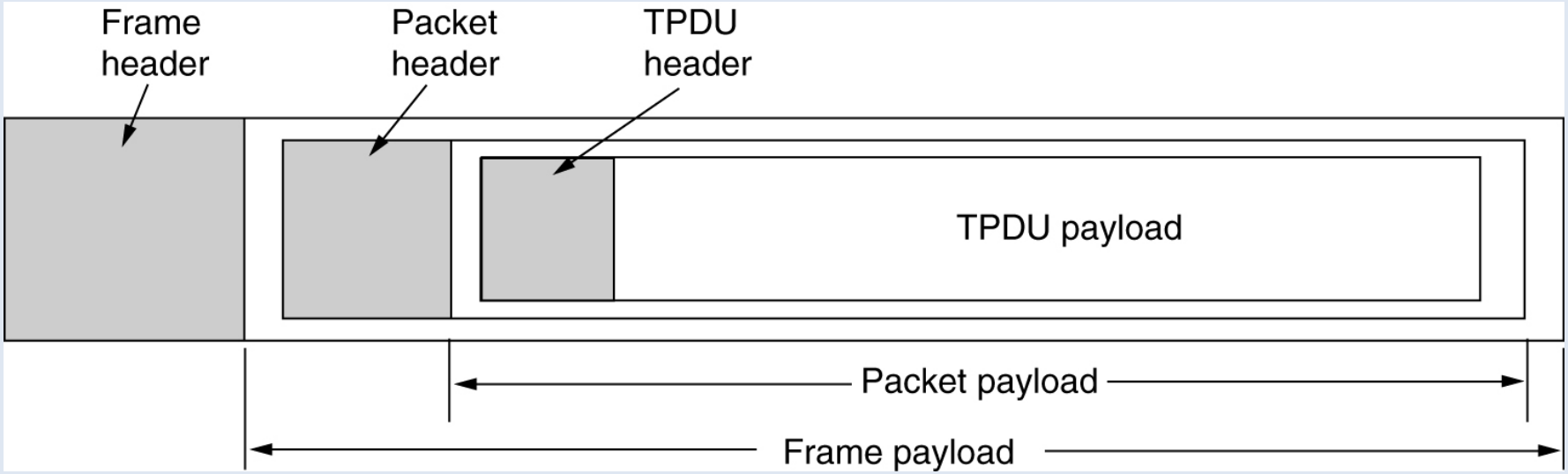
Couche Transport
(TCP)

Introduction

- La couche réseau (IP)
  - Communication de bout-en-bout ntre machines
  - Transfert de paquet en “best-effort” (non fiable)
- La couche transport
  - TCP : communication de bout-en-bout entre processus, orienté connexion et fiable

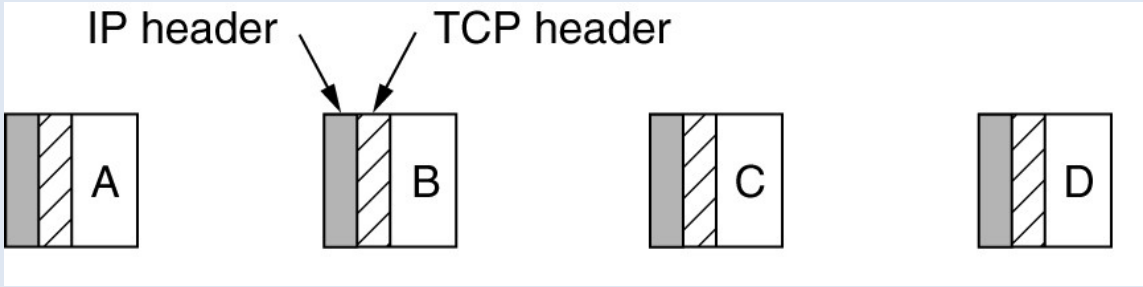
TCP

- Transmission Control Protocol (TCP)
  - service en mode connecté
  - connexion bidirectionnelle et point-à-point
    - (IP_{source} ; $Port_{source}$; $IP_{destination}$; $Port_{destination}$)
  - le numéro de port désigne un processus et un seul

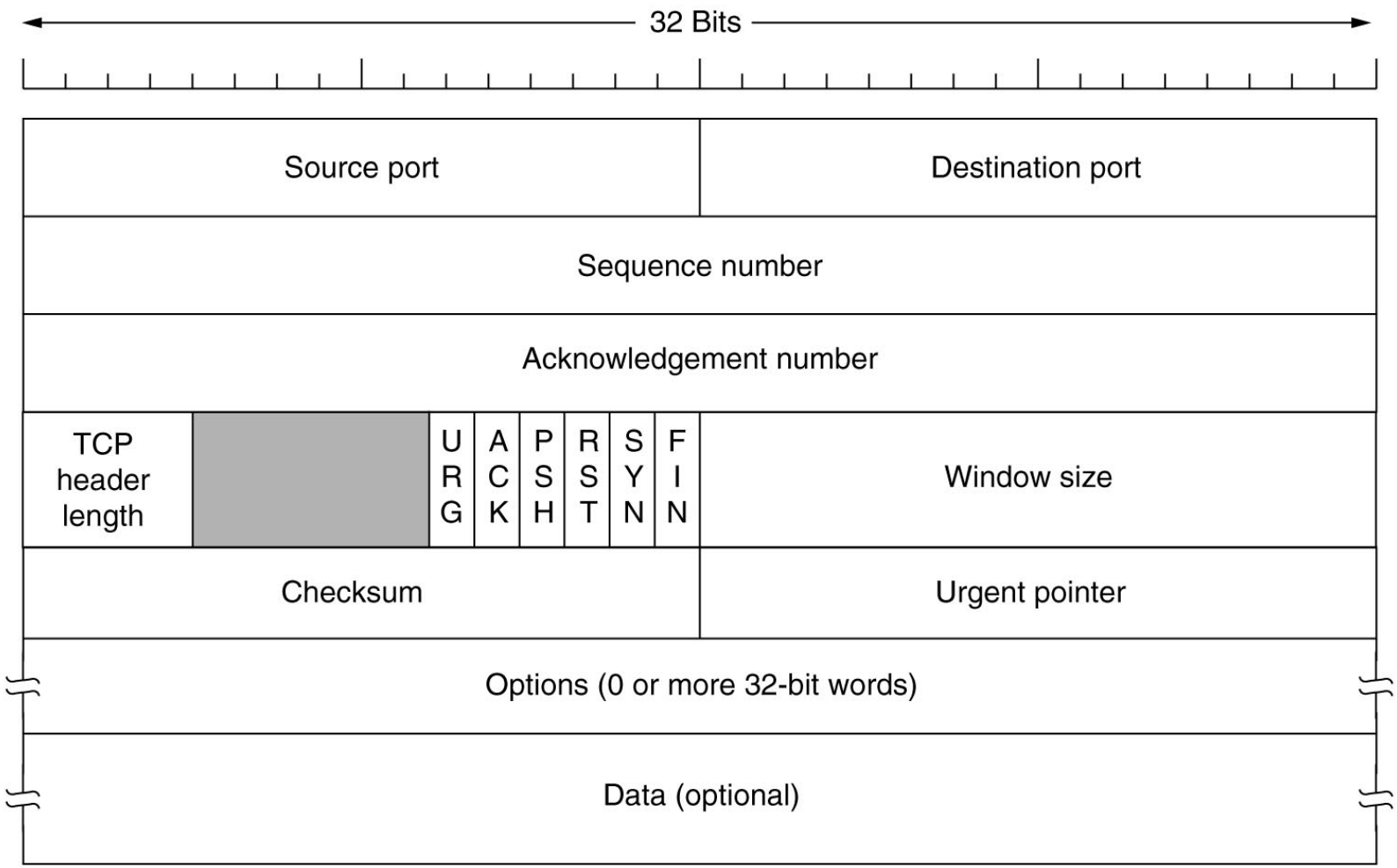


TCP

- MTU (Maximum Transfert Unit)
  - Taille maximale des paquets IP (1500 octets sur Ethernet)
- Fragmentation des messages en plusieurs segments
  - Numérotation des segments composant le message
- Exemple de fragmentation en 4 segments du message ABCD



En-tête TCP

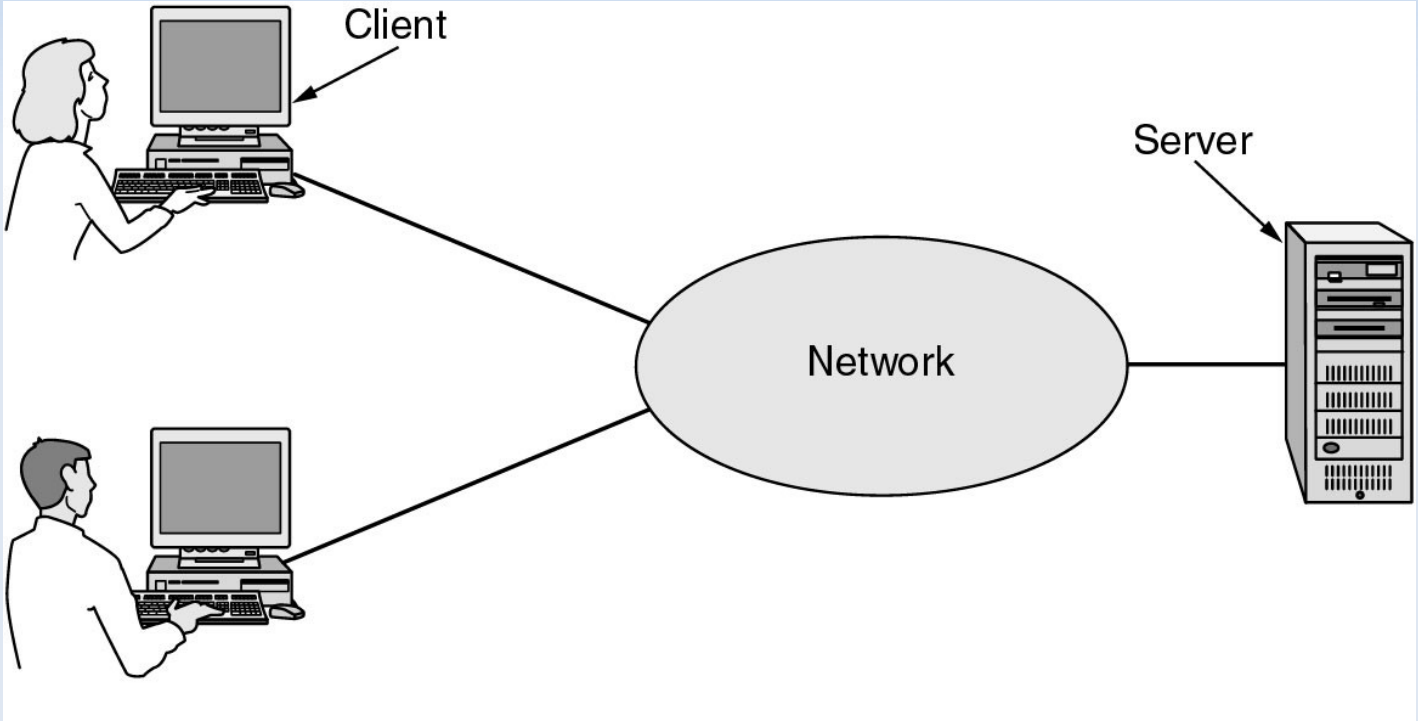

32 Bits
Source port
Destination port
Sequence number
Acknowledgement number
TCP header length
URG
ACK
PSH
RST
SYN
FIN
Window size
Checksum
Urgent pointer
Options (0 or more 32-bit words)
Data (optional)

En-tête TCP

- Source Port et Destination Port
- Numéro de séquence : le numéro du segment TCP
- Numéro d'accusé de réception : numéro du prochain octet attendu
- 6 flags binaires :
  - ACK : indique si le numéro d'accusé de réception est valide
  - SYN : demande d'établissement d'une connexion
  - FIN : libération de la connexion
  - RST : réinitialisation d'une connexion (reset) ; rejet d'une connexion
  - Autres : PSH, URG
- Window size : nombre d'octets souhaités pour la réception ; si 0, stoppe temporairement la transmission

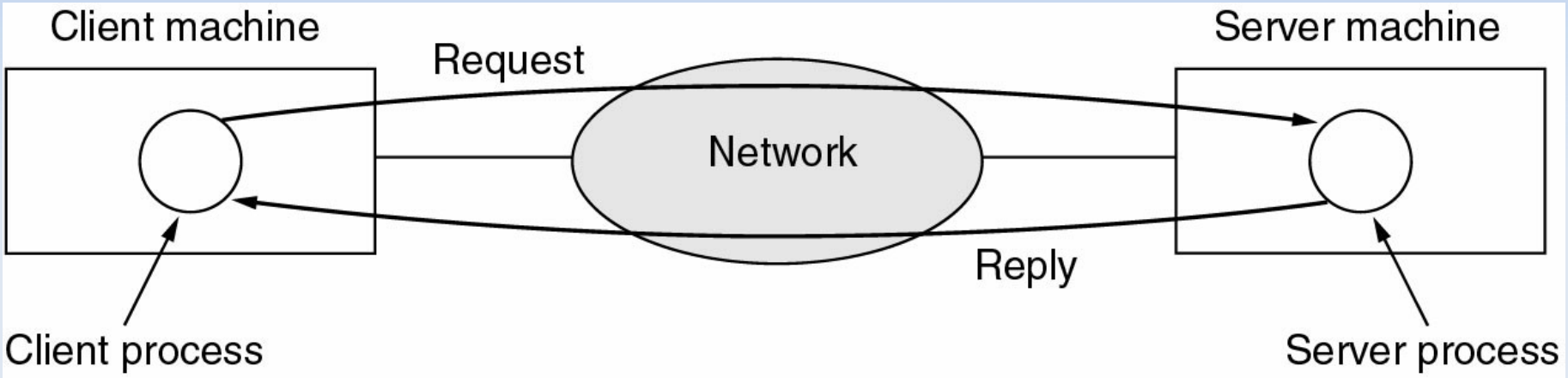
Modèle client-serveur de TCP/IP

- Plusieurs clients et un serveur



Modèle client-serveur de TCP/IP

- Schéma de communication requête-réponse...



- Notion de connexion
  - adresse IP source, numéro de port source
  - adresse IP destination, numéro de port destination

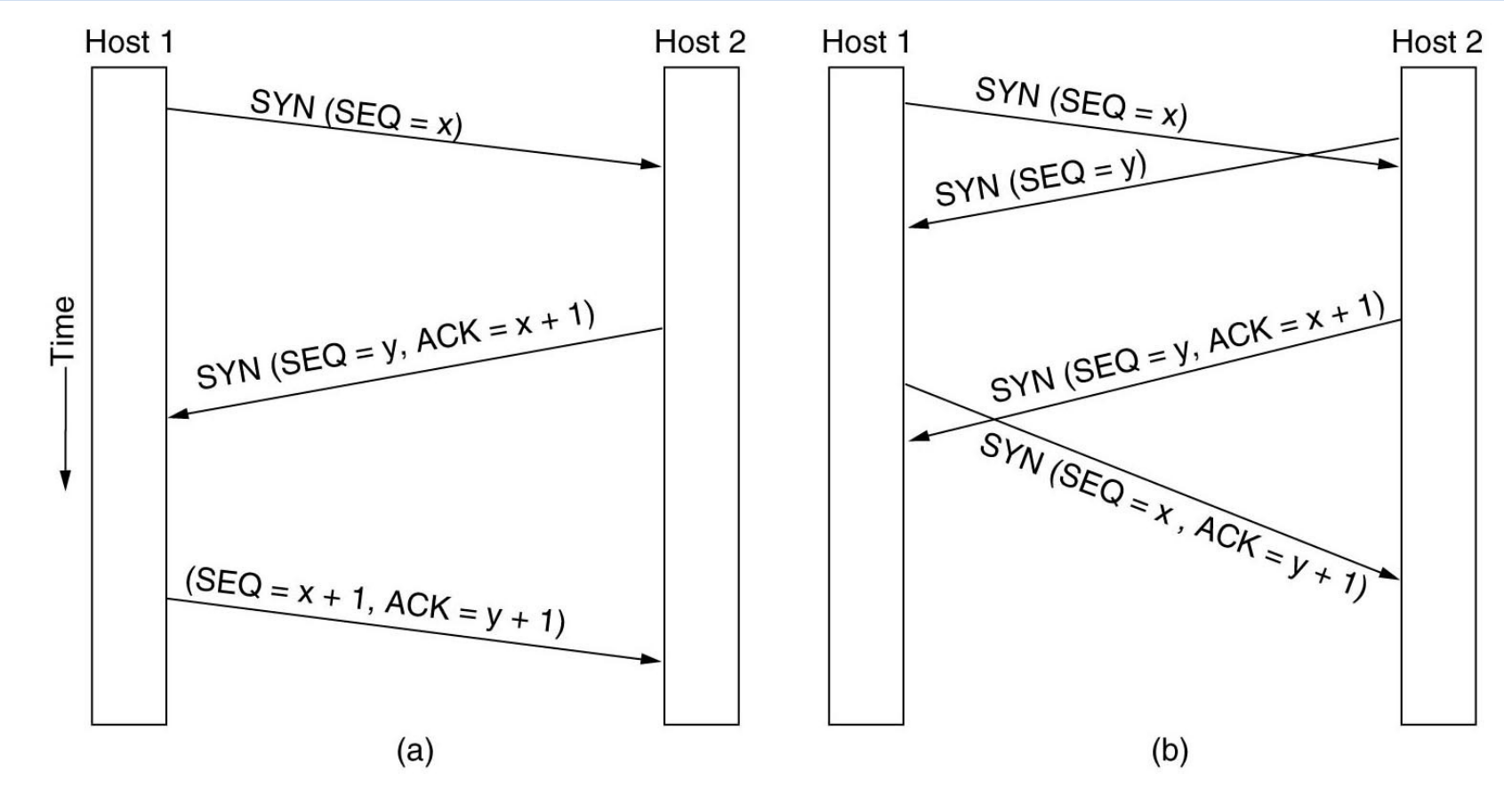
TCP

- Exemples de service TCP standards (ports < 1024)

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Connexion TCP

- La “poignée de main” en 3 étapes
  - Synchronisation des numéros de séquence



Connexion standard

Connexion simultanée

Connexion TCP

- Programme Scapy

  → *todo*

Netstat (à remplacer)

- netstat -tnap (liste des connexions TCP/IP)

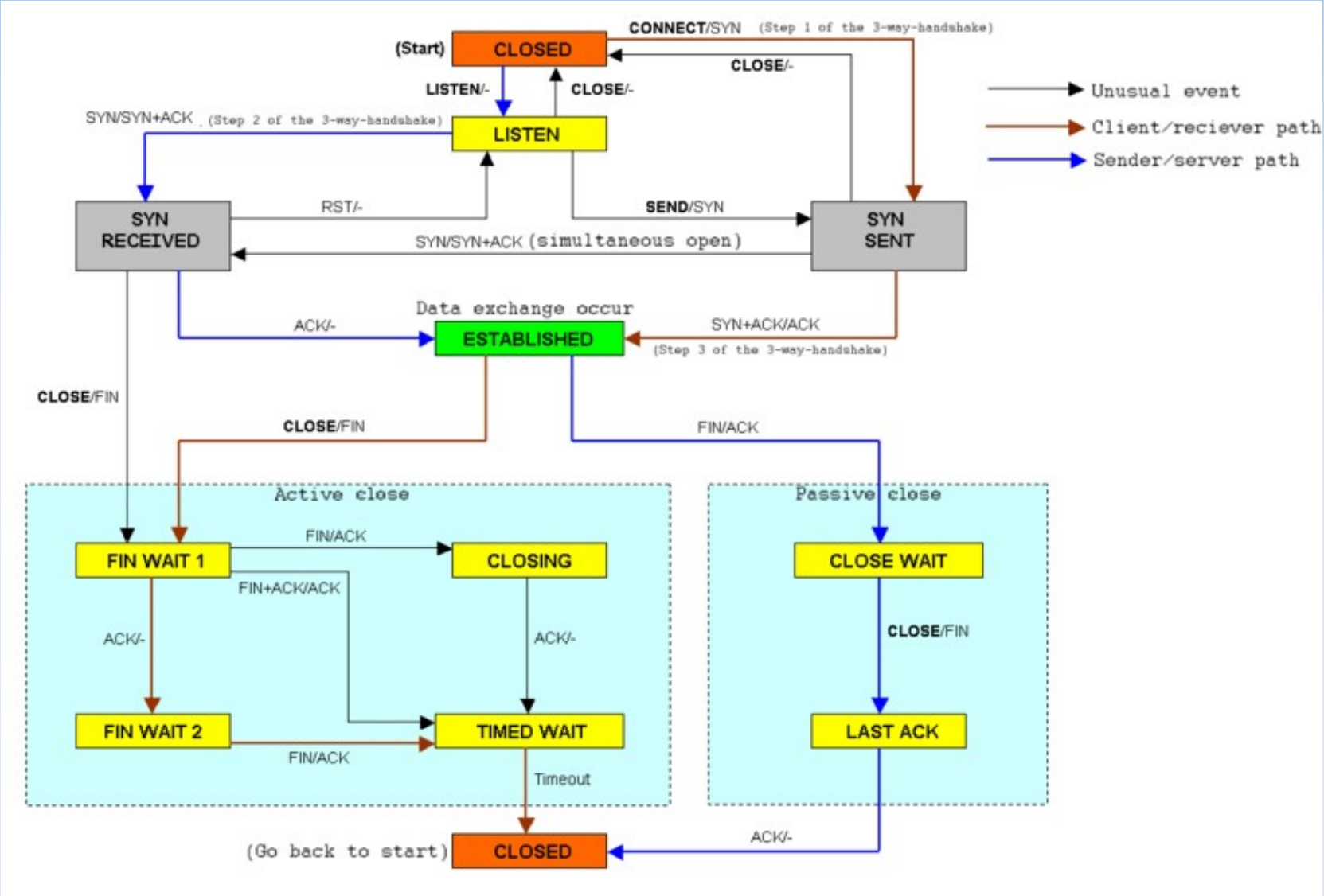
Proto	R-Q	S-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:2208	*:*	LISTEN	3266/hpiod
tcp	0	0	127.0.0.1:34818	*:*	LISTEN	3275/python
tcp	0	0	127.0.0.1:3306	*:*	LISTEN	3642/mysqld
tcp	0	0	0.0.0.0:25	*:*	LISTEN	3525/exim4
tcp	0	0	82.225.96.37:35551	147.210.8.143:993	ESTABLISHED	10503/mozilla
tcp	0	0	82.225.96.37:39243	147.210.13.65:22	ESTABLISHED	13758/ssh
tcp	0	0	82.225.96.37:35750	147.210.9.15:22	ESTABLISHED	13763/ssh
tcp6	0	0	*:80	*:*	LISTEN	3979/apache2
tcp6	0	0	*:22	*:*	LISTEN	3746/sshd
tcp6	0	0	*:25	*:*	LISTEN	3525/exim4

TCP

- Les états d'une connexion TCP

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP



E/M →

Lorsque l'evènement E se produit, envoyé le message M ou ne rien faire si M='-'.

Administration : Firewall

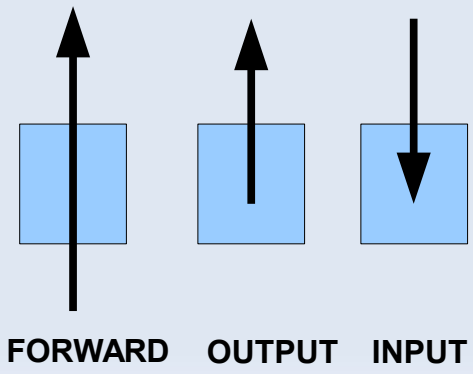
- Configurer le firewall avec iptables...
  - Lister les règles

    $ iptables -t filter -L - v
  - Ajouter une nouvelle règle

    $ iptables -t filter -A <CHAIN> <SRC> <DST> <...> -j <ACTION>
  - Politique par défaut (si aucune règle ne s'applique avant)

    $ iptables -t filter -P <CHAIN> <ACTION> # <ACTION> = ACCEPT | DROP
  - Memento

    <CHAIN> = FORWARD | INPUT | OUTPUT
    <ACTION> = ACCEPT | REJECT | DROP
    <SRC> = -i eth0 | -s 192.168.0.1 | -s 192.168.0.0/24
    <DST> = -o eth0 | -d 192.168.0.1 | -d 192.168.0.0/24
    <...> = -p icmp | -p tcp --dport 80 | -m state --state <STATE>
    <STATE> = NEW | ESTABLISHED
     * NEW : établissement d'une nouvelle connexion
     * ESTABLISED : une connexion déjà établie



Administration : Firewall

- Protéger une machine connectée directement sur Internet...
  - On configure le firewall de A pour les chains INPUT / OUPUT
- Exemple
  - On interdit tout par défaut...

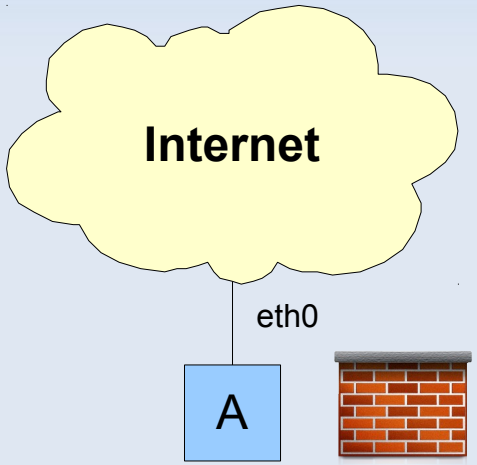
```
$ iptables -t filter -P INPUT DROP
$ iptables -t filter -P OUTPUT DROP
```

  - On autorise le ping !

```
$ iptables -t filter -A INPUT -p icmp -j ACCEPT
$ iptables -t filter -A OUTPUT -p icmp -j ACCEPT
```

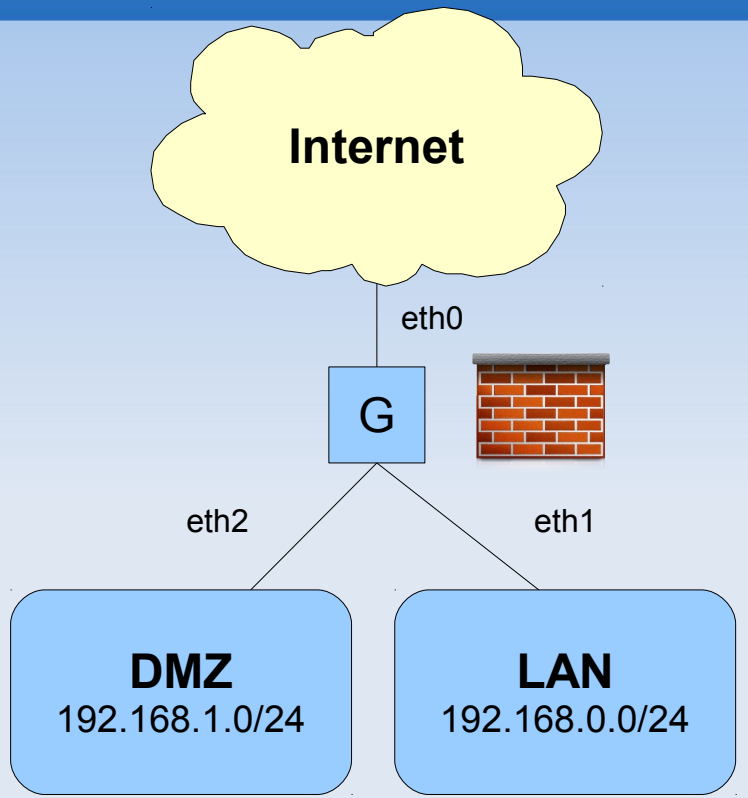
  - On autorise uniquement l'accès de A au web...

```
$ iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
$ iptables -t filter -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```



Administration : Firewall

- Protéger un réseau...
  - On configure le firewall uniquement sur la passerelle G pour la chain FORWARD
  - On déporte les services sensibles (web, ...) dans un sous-réseaux, appelé DMZ
- Exemple
  - On interdit tout par défaut




```
$ iptables -t filter -P FORWARD DROP
```

  - On autorise l'accès aux serveurs web dans la DMZ

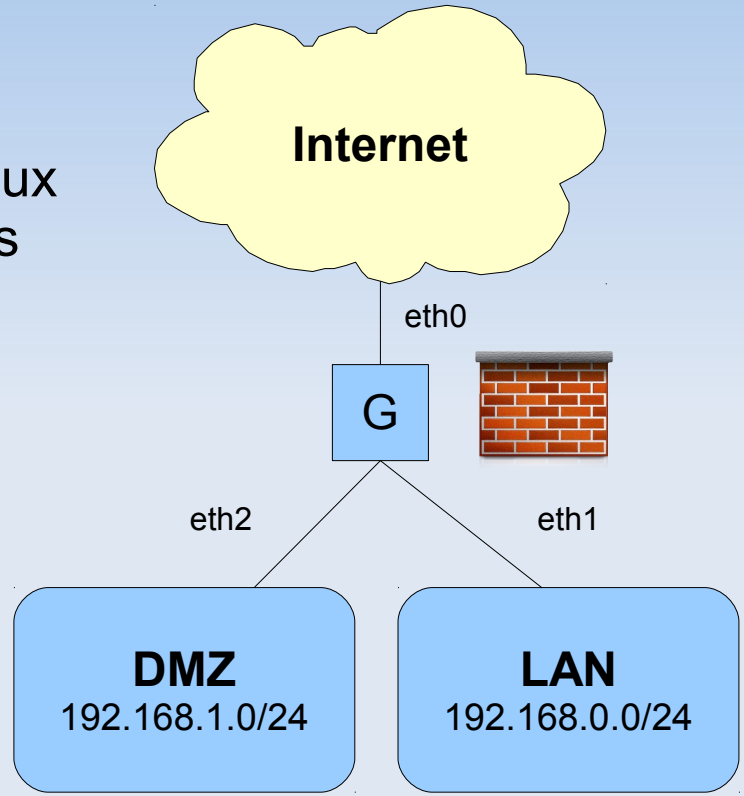
```
$ iptables -t filter -A FORWARD -d 192.168.1.0/24 -p tcp --dport 80 -j ACCEPT
$ iptables -t filter -A FORWARD -s 192.168.1.0/24 -p tcp --sport 80
        -m state --state ESTABLISHED -j ACCEPT
```

  - On autorise tout le traffic sortant du LAN (et le retour...)

```
$ iptables -t filter -A FORWARD -s 192.168.0.0/24 -j ACCEPT
$ iptables -t filter -A FORWARD -d 192.168.0.0/24 -m state --state ESTABLISHED -j ACCEPT
```

Administration : Firewall

- Exercice
  - Ajouter une régle iptables pour permettre aux utilisateurs du LAN de se connecter dans la DMZ par SSH (tcp, port 22).



- Correction

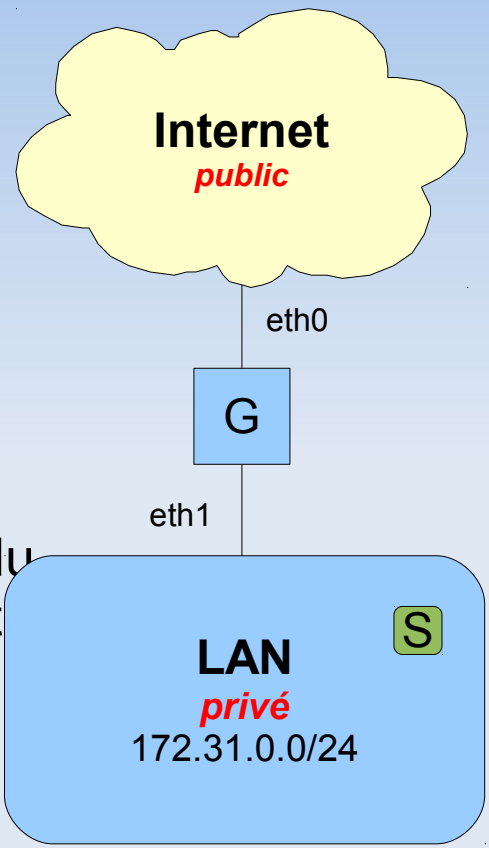
```
$ iptables -A FORWARD -s 192.168.0.0/24
-d 192.168.1.0/24 -p tcp --dport 22
-j ACCEPT
$ iptables -A FORWARD -d 192.168.0.0/24
-s 192.168.1.0/24 -p tcp –sport 22
-m state --state ESTABLISHED
-j ACCEPT
```

Administration : NAT

- NAT (Network Address Translation)
  - Un réseau privé ne peut pas accéder et n'est pas accessible depuis Internet (adresses IP non routables)
  - Mais possibilité d'utiliser une passerelle NAT !
- Exemples
  - Exemple de masquage des IPs du LAN : les machines du LAN peuvent communiquer avec Internet en empruntant l'adresse publique de G

    root@G$ iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
  - Exemple de Port-Forwarding de G:80 vers S:80 : le serveur web S (port 80) est ainsi accessible depuis l'extérieur

    root@G$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to <S>:80



Cours 4

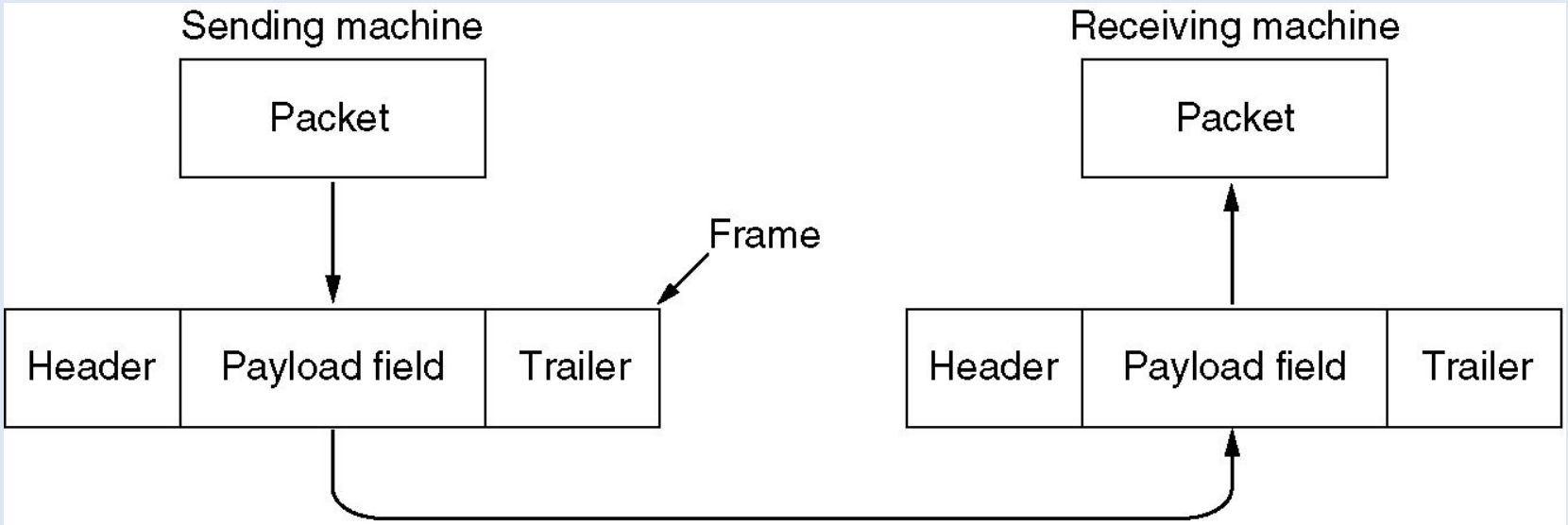
Couche Liaison de Données (Ethernet)

Introduction

- La couche liaison de données
  - Le rôle de la couche liaison est de fournir à la couche réseau une transmission fiable en s'appuyant sur des supports de transmission qui ne sont pas parfaits !
- Plan
  - Notion de Trame
  - Contrôle d'erreurs (parité, Hamming, CRC)
  - Ethernet et CSMA/CD
  - Switch, Hub
  - Autres exemples de protocoles (PPP, ATM, ...)

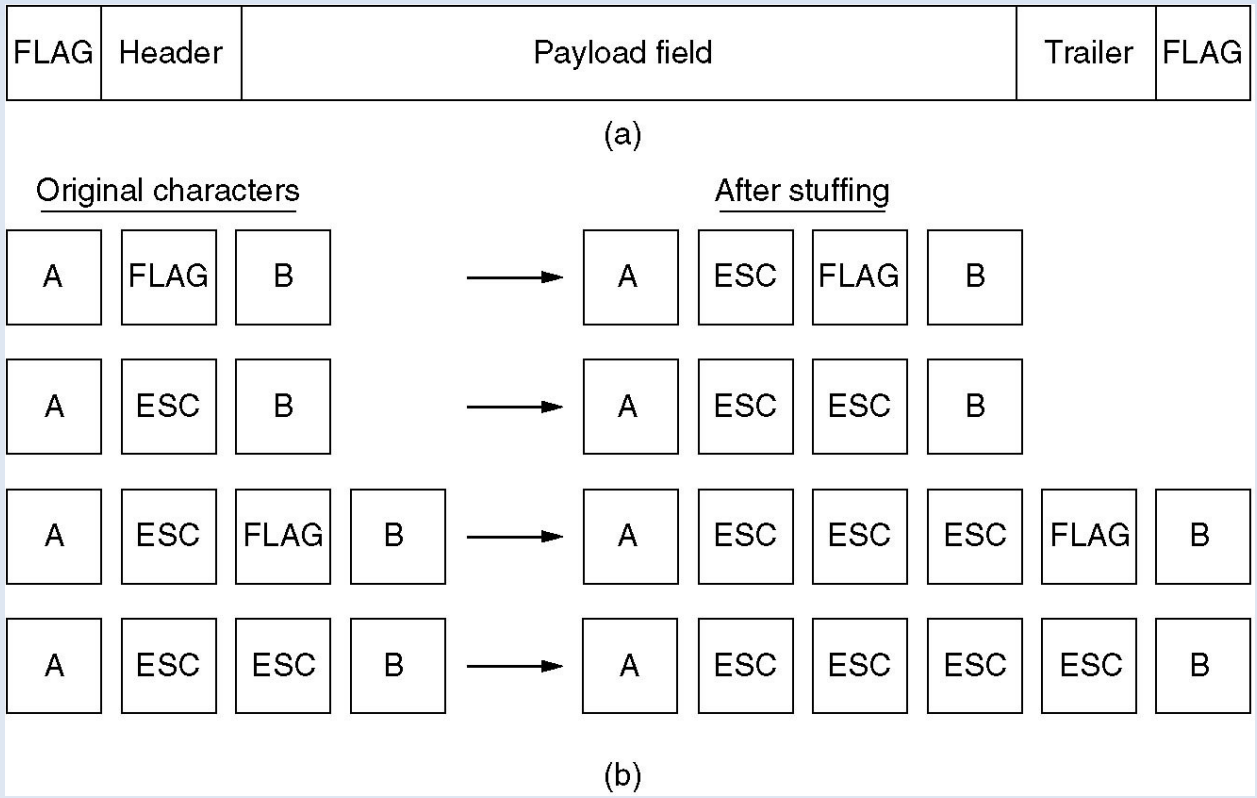
Notion de trame

- Composition d'une trame (*frame*)
  - en-tête (*header*)
  - paquet fourni par la couche supérieure (la couche réseau)
  - en-queue (*trailer*)



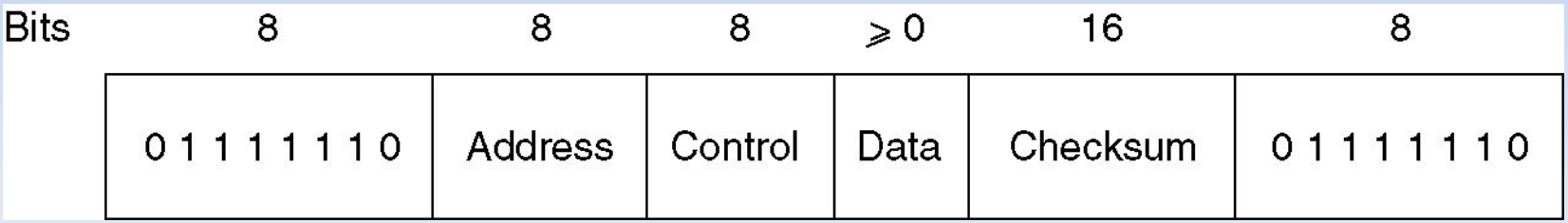
Trames

- Délimitation des trames (flag)
  - utilisation d'un caractère d'échappement (esc) si les caractères flag ou esc apparaissent dans le message initial

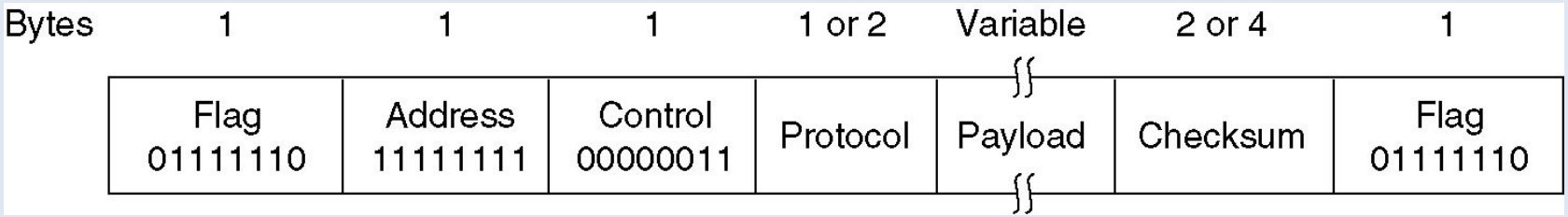


Format des trames

- Exemple HDLC

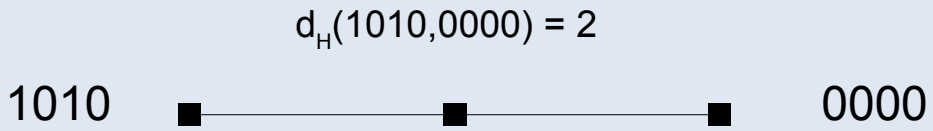


- Exemple du protocole PPP



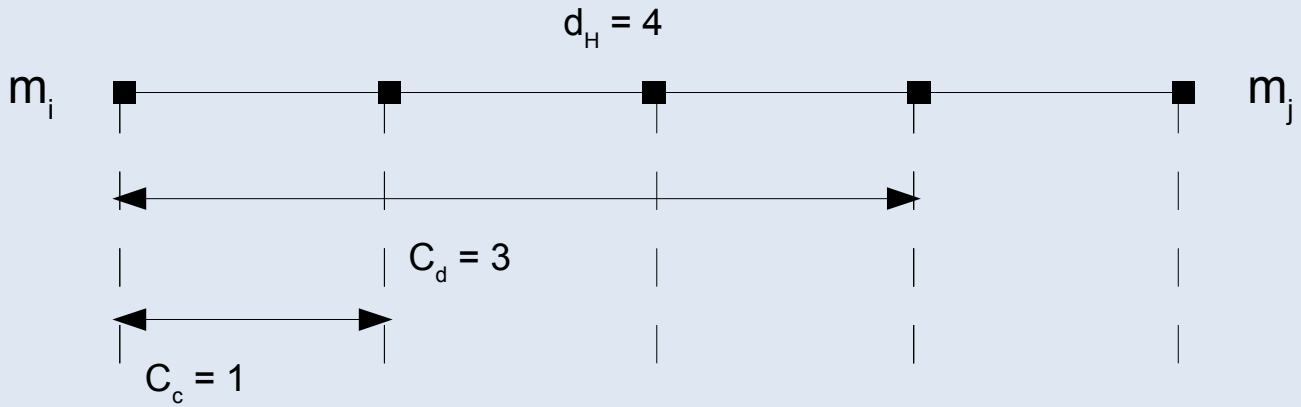
Contrôle d'erreurs

- Etant donné un code C = {m_0,m_1,m_2, ...} avec m_i un mot binaire du code de longueur noté $|m_i|$
- Distance de Hamming entre deux mots m_i et m_j
  - $d_H(m_i,m_j)$ = nombre de bits qui différent entre m_i et m_j
- Distance de Hamming du code C
  - $d_H(C)$ = min des distances de Hamming entre les mots du code
  - Tous les mots du code sont au moins à une distance $d_H(C)$
- Exemple



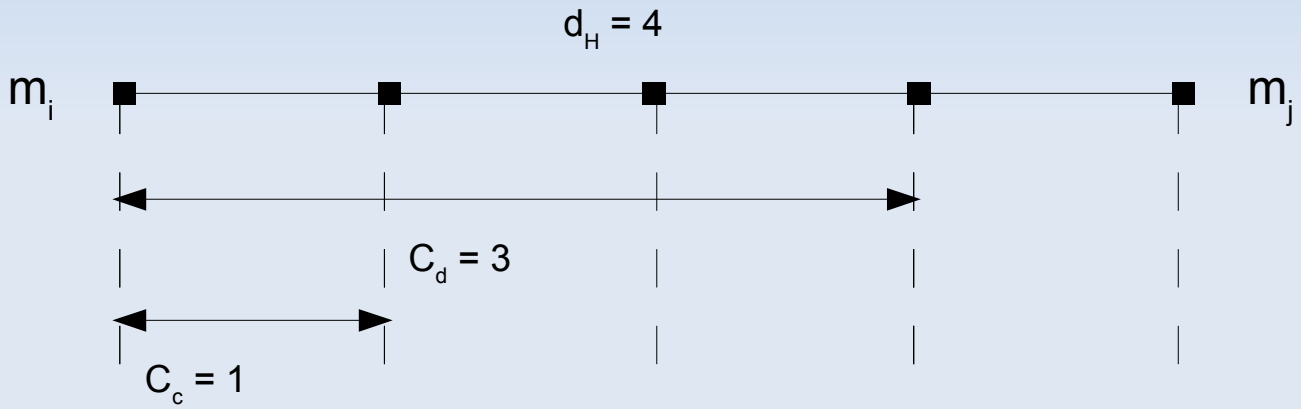
Contrôle d'erreurs

- Capacité de détection d'un code C
  - nombre maximum d'erreurs pouvant être détectés
  - $C_d(C) = d_H(C) - 1$
- Capacité de correction d'un code C
  - nombre maximum d'erreurs pouvant être corrigés
  - $C_c(C) = d_H(C) / 2 - 1$ si pair ; $(d_H(C) - 1) / 2$ sinon
- Exemple d'un code C avec $d_H = 4$



Contrôle d'erreurs

- Exemple d'un code C avec $d_H = 4$
- Transmission de m_i



- En cas d'erreurs de transmission, plusieurs bits ont pu changer...
  - si 1 bit d'erreur, détection et correction
  - si 2 ou 3 bits d'erreur, détection sans correction
  - si 4 bits d'erreur, reconnaissance d'un autre mot !!!

Contrôle d'erreurs

- Exercice 4.1 (code barre postal)
  - Quel est le code postal suivant ?

    I.I.II - ..IIII- I..III - .III.I - .III.I
  - Réponse 33405
  - Quel est la distance de Hamming de ce code ?
  - Combien d'erreurs peut-on détecter et corriger ?

(0) . . I I I I

(1) . I . I I I

(2) . I I . I I

(3) . I I I . I

(4) I . . I I I

(5) I . I . I I

(6) I . I I . I

(7) I I . . I I

(8) I I . I . I

(9) I I I . . I

Lecture de droite à gauche.

Bit de parité

- Principe
  - Considérons un mot binaire m=(m_1,m_2,m_3,...,m_n) de taille *n*
  - Ajout à la fin de m du bit m_{n+1} tel que la somme de tous les bits du mot soit paire
- Exemple
  - m = 0101001 et m' = 01010011

Code de Hamming (m,n)

- Considérons un mot binaire de données de taille n

$$d=(d_1,d_2,d_3,...,d_n)$$

- Insertion dans d de k bits de contrôle p_i aux positions

2^0, 2^1, 2^2, 2^3, ... c'est-à-dire 1, 2, 4, 8, ...

- Soit s le mot à transmettre de taille m=n+k

$$s = (s_1,s_2,s_3,...,s_j,..., s_{n+k}) = (p_1,p_2,d_1,p_3,d_2,d_3,...,p_i,...,d_n)$$

- Le nombre de bits de contrôle est le premier entier k supérieur à $\log_2(m)$. Ex. : $\log_2(11) < 4$ donc k=4 et n=7

Code de Hamming

- Calcul des bits de contrôle
  - Le bit de donnée s_j est contrôlé par les bits dont les positions sont les coefficients de la décomposition binaire de j.
  - Le bit de contrôle p_i (en position i) est choisi de telle sorte que la somme des bits qu'il contrôle (ainsi que lui-même) fasse 0 modulo 2 (contrôle de parité).
- Détection et correction d'erreur
  - à la réception d'un message, on effectue le contrôle de parité sur le bits de contrôle
  - si p_a et p_b sont faux, alors il y a une erreur sur le bit s_{a+b} qui peut être corrigée !
  - Capacité de détection et de correction d'une seule erreur !

Exemple

- Exemple de calcul des bits de parités dans le code de Hamming (11,7)

	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7
Data word (without parity):			**0**		**1**	**1**	**0**		**1**	**0**	**1**
p_1	**1**		0		1		0		1		1
p_2		**0**	0			1	0			0	1
p_3				**0**	1	1	0				
p_4								**0**	1	0	1
Data word (with parity):	**1**	**0**	0	**0**	1	1	0	**0**	1	0	1

Calculation of Hamming code parity bits

Exercice

- Exercice 4.2 : Code de Hamming (11,7)
  - Quels sont les bits qui contrôlent d_7 en position 11 ?
  - Quels sont les bits contrôlés par p_1, par p_4 ?
  - Calculer le code de Hamming du mot 1101011 ?
  - Quel est le message correspondant au code 11111011100 ?
    - Y-a-til une erreur ? Si oui, corriger cette erreur !

Code de Hamming

- Correction
  - le bit $s_{11} = d_7$ est contrôlé par p_1, p_2 et p_8 (11 = 1 + 2 + 8)
  - p_1 contrôle $s_3+s_5+s_7+s_9+s_{11}$; p_4 contrôle $s_8+s_9+s_{10}+s_{11}$
  - d = 1101011 ; s = p_1.p_2.1.$p_3$1.0.1.p_4.0.1.1 = 00101010011
  - s' = 11111011100 (après réception)
    - p_1 faux, p_2 faux, p_3 faux et p_4 ok
    - erreur sur le bit d'indice 1+2+4=7
    - s = 11111001100 (après correction)
    - d = 1100100 (mot reçu)

CRC

- CRC (Cyclic Redundancy Check)
  - Calcul d'un checksum basé sur l'arithmétique polynomiale modulo 2
  - On considère le mot binaire suivant de taille *n*

$$b=(b_{n-1},b_{n-2},...,b_1,b_0)$$

  - Ce mot s'exprime sous la forme d'un polynôme de degrés n-1, à coefficient binaire :

$$B(X)=b_{n-1}.X^{n-1} + b_{n-2}.X^{n-2} + ... + b_1.X + b_0$$

  - La clé C(X) associée à un tel mot est définie comme étant le reste de la division de $B(X).X^k$ par un polynôme générateur G(X) de degré k.
  - Le mot à transmettre est alors $M(X) = B(X).X^k + C(X)$.

CRC

- Exemple d'utilisation des CRCs
  - CRC-1 (bit de parité) : $G(X) = X + 1$
  - CRC-8 (ATM) : $G(X) = X^8 + X^2 + X + 1$
  - CRC-16 (USB, PPP, Bluetooth, ...)
  - CRC-32 (Ethernet) : $G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - CRC-160 (MD5 checksum)

Exemple

- Quel est la clé associée au mot 110111 avec $G(X) = x^2+x+1$?
  - Mot = 110111
  - $B(X) = X^5 + X^4 + X^2 + X + 1$
  - $B(X).X^2 = X^7 + X^6 + X^4 + X^3 + X^2$
  - Calcul : $B(X).X^2 / G(X) = ...$
  - Le reste est $C(X) = X+1$
  - Donc la clé est 11 (coefficients de C(X))
  - Le mot à envoyer sera 11011111
  - Vérifier que M(X) est divisible par G(X) (reste nul)
    - M(X) est le polynôme correspondant au mot transmis...

CRC

- Correction...

$B(X).X^2 = X^7 + X^6 + X^4 + X^3 + X^2$ | $G(X) = X^2 + X + 1$

$-(X^7 + X^6 + X^5)$ | $P(X) = X^5 + X^3 + 1$

$X^5 + X^4 + X^3 + X^2$

$-(X^5 + X^4 + X^3)$

X^2

$-(X^2 + X + 1)$

$C(X) = X + 1$

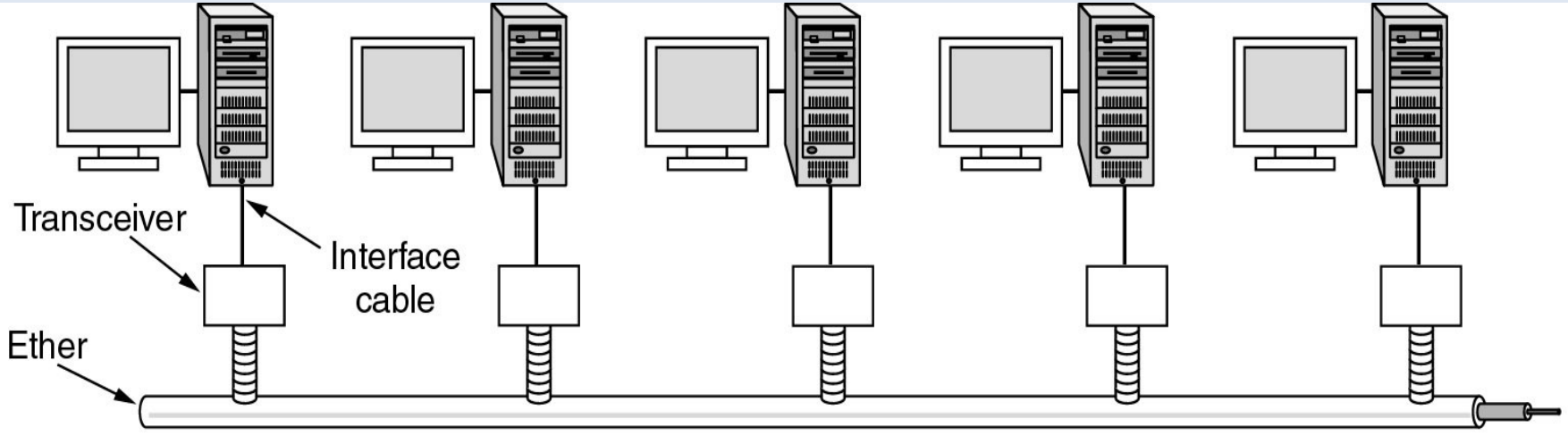
CRC

- Détection d'erreur
  - M(X) doit être divisible par G(X).
  - On peut le vérifier en effectuant la division de M(X) par G(X) ; le reste R(X) doit être nul.
  - Si ce n'est pas le cas, une erreur est détectée !

- Quelle condition doit vérifier B(X), C(X) et G(X) ?
  - $B(X).X^k = P(X).G(X) + C(X)$ avec C(X) de d° < k
  - $M(X) = B(X).X^k + C(X) = P(X).G(X) + C(X) + C(X) = P(X).G(X)$
  - En effet, en algèbre binaire (modulo 2), on a 1+1 = 0 ou encore 1 = -1, par conséquent ajouter est identique à soustraire !

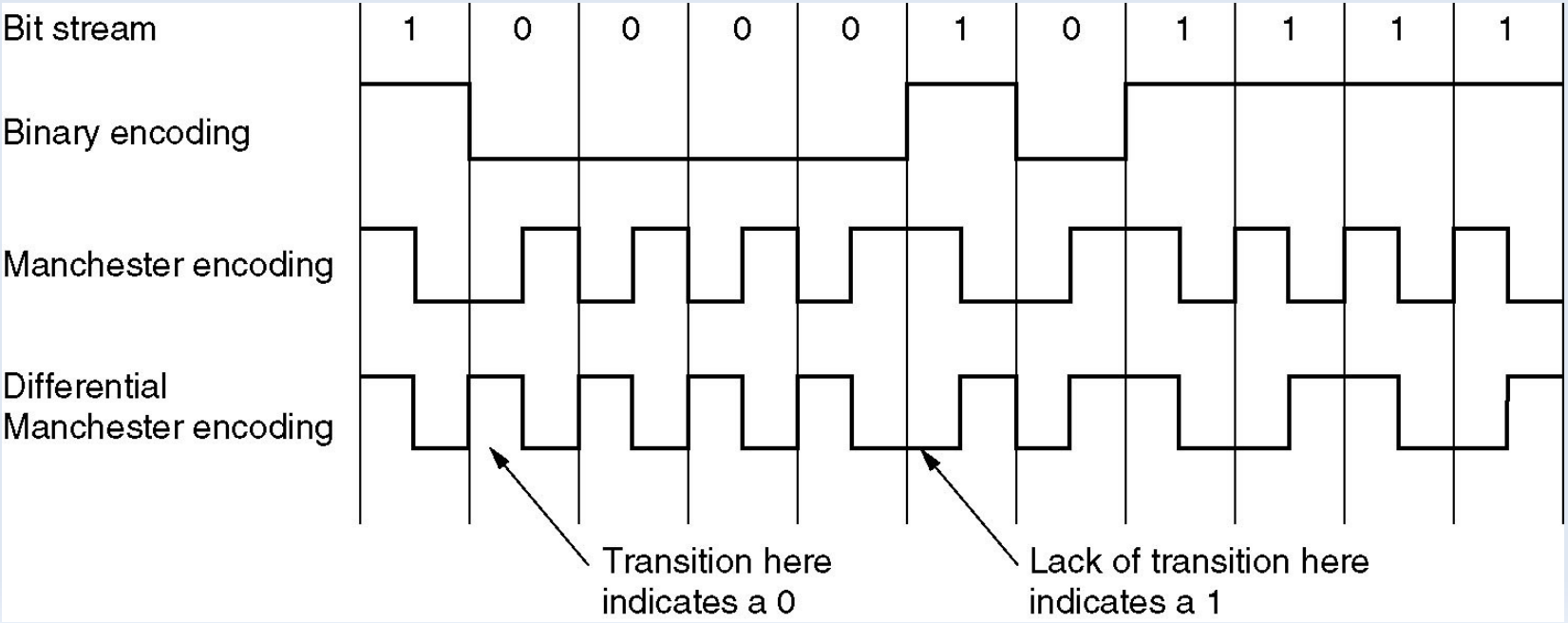
Ethernet (IEEE 802.3)

- Première technologie LAN haut-débit grand-public
  - un standard qui existe depuis plus de 20 ans...
  - permet d'échanger des trames sur divers supports physiques
  - adresse MAC des cartes Ethernet (ex. 00:15:C5:3D:52:B6)



Ethernet

- Codage du flux binaire
  - Ethernet est basé sur le codage Manchester (simple)
  - tensions -0.85 et +0.85 volts
  - approche robuste utilisant une transition pour chaque bit, ce qui facilite la synchonisation ainsi que la détection du début de l'émission

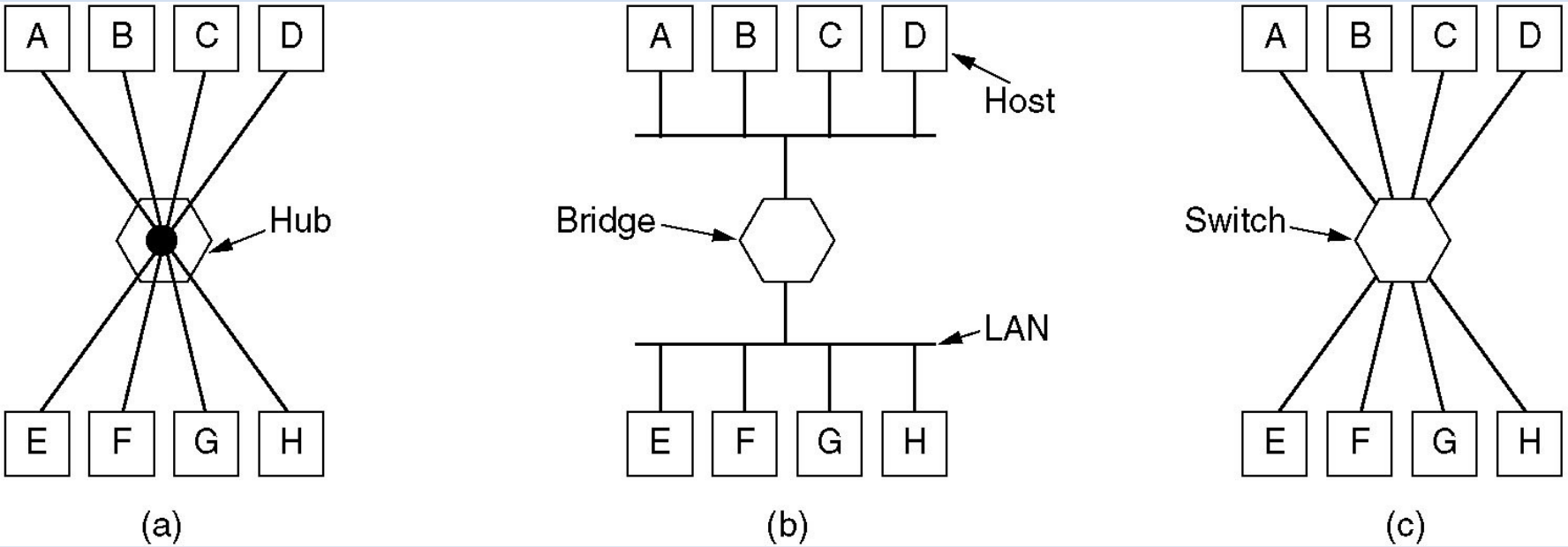


Ethernet

- Répétiteur
  - amplification du signal pour limiter l'atténuation
- Hub (concentrateur)
  - diffusion à tout le monde (équivalent bus)
  - mode semi-duplex, CSMA/CD
- Switch (commutateur)
  - diffusion des trames uniquement au destinataire choisi
  - ligne dédiée entre la station et le commutateur (mode full-duplex) donc pas de collision possible, donc pas besoin de CSMA/CD
- Bridge (pont)
  - généralisation du commutateur lorsqu'on interconnecte plusieurs technologies différentes...

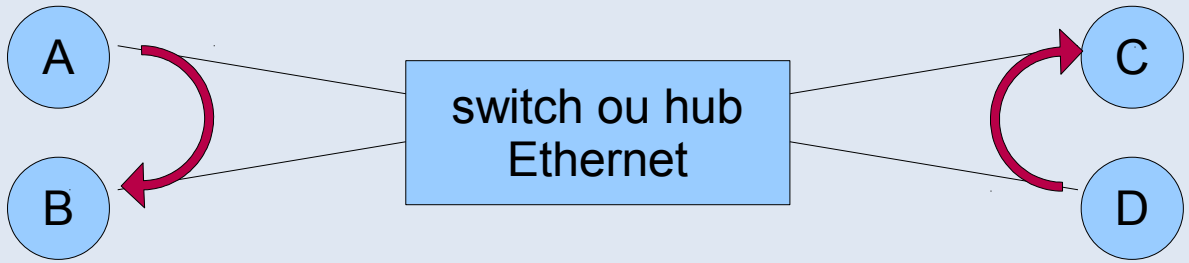
Ethernet

- Hub, Bridge, Switch



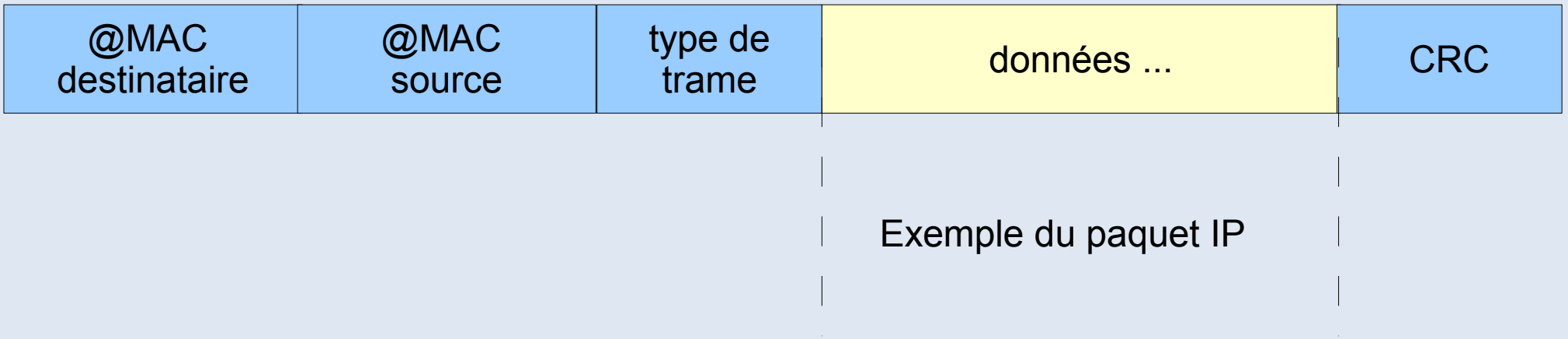
Exercice

- Au même moment, A communique avec B et D avec C en saturant un réseau Ethernet 100 Mbit/s
  - quel est le débit maximal atteint entre A et B dans le cas d'un hub ?
  - même question dans le cas d'un switch



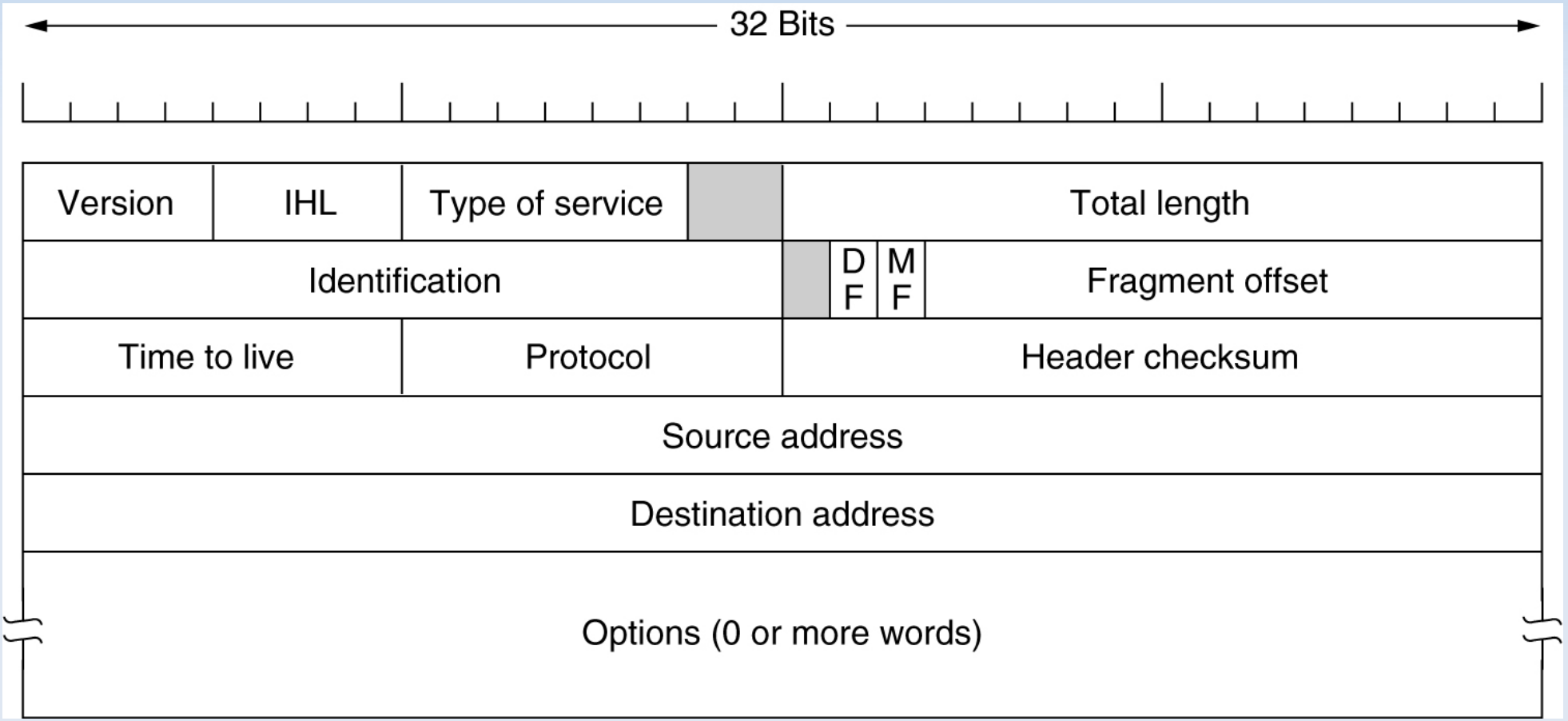
La trame Ethernet

- Format des trames (frames)
  - Adresse MAC du destinataire (6o) et du source (6o)
  - Type de protocole : 0800 = IP ; 0806 = ARP ; ... (2o)
  - Code CRC-32 (4o)
  - Données : au minimum 46o, jusqu'à 1500o
    - caractères de bourrage si données < 46o



Exemple du paquet IP (v4)

- Le paquet IP de la couche réseau est encapsulé dans la trame



Exercice

- Exemple de trame Ethernet

  00 40 07 03 04 2b 02 60 8c e8 02 91 08 00 45 00

  00 2c 14 ee 00 00 3c 06 85 7a 93 d2 5e 63 93 d2

  5e 5c 10 a4 09 e7 42 0c 56 01 00 00 00 00 60 02

  40 00 c1 29 00 00 02 04 05 b4 02 80 xx xx xx xx

- Questions
  - adresses MAC du destinataire et de l'émetteur ?
  - que repésente les 4o de la fin ?
  - protocole encapsulé dans la trame ?
  - bits de bourrage ?

Correction

- Exemple de trame Ethernet

  00 40 07 03 04 2b 02 60 8c e8 02 91 08 00 [45 00

  00 2c 14 ee 00 00 3c 06 85 7a 93 d2 5e 63 93 d2

  5e 5c 10 a4 09 e7 42 0c 56 01 00 00 00 00 60 02

  40 00 c1 29 00 00 02 04 05 b4 ~~02 80~~] xx xx xx xx

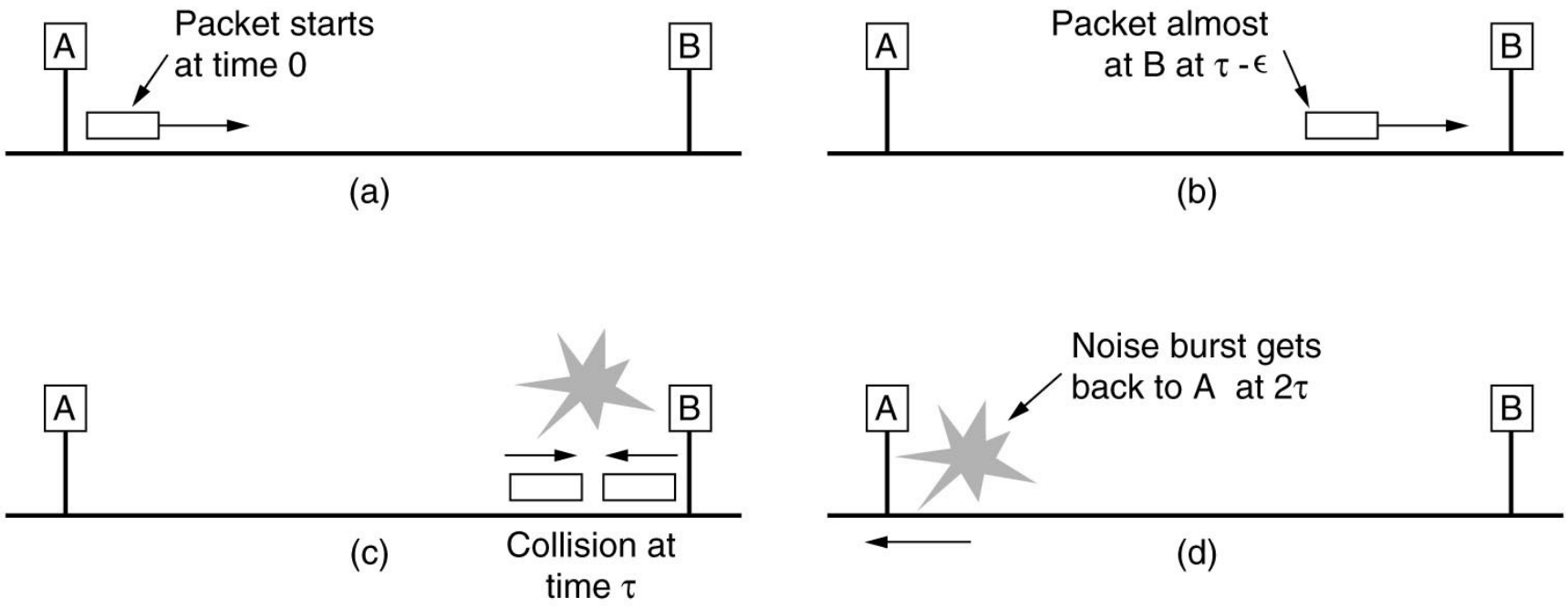
- Analyse
  - @MAC destinataire : 00 40 07 03 04 2b
  - @MAC source : 02 60 8c e8 02 91
  - Type de protocole : IP (0800)
  - Taille du paquet IP : 00 2c, soit 44 o dont 2o de bourrage
  - xx xx xx xx : CRC-32

Ethernet (CSMA/CD)

- CSMA/CD : Carrier Sense Multiple Access / Collision Detection
  - un seul émetteur à la fois qui monopolise le canal
  - pas de multiplexage, donc débit max pour chaque émetteur
  - écoute de porteuse : permet de sonder si le canal est libre

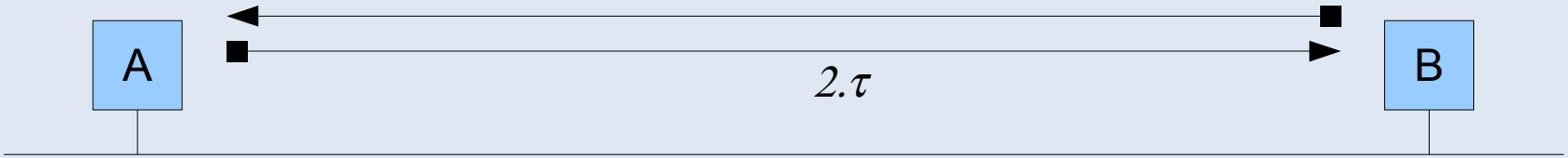
- Principe détection de collision sur le bus Ethernet
  - la détection doit se produire lors de l'emission qui doit donc durer au moins $2.\tau$
  - en cas de collision, réémission avec un délai aléatoire supplémentaire

Ethernet (CSMA/CD)



Ethernet (CSMA/CD)

- Exercice 3.5
  - Calcul de la taille minimale S_{min} d'une trame Ethernet en fonction du débit D de la carte Ethernet, de la distance maximale d_{max} séparant deux stations (A et B) et de la vitesse v de propagation du signal ?



- La détection de collision doit avoir lieu pendant l'émission...
  - donc : $T_{\text{emission min}} = T_{\text{détection collision}} = 2.\tau$

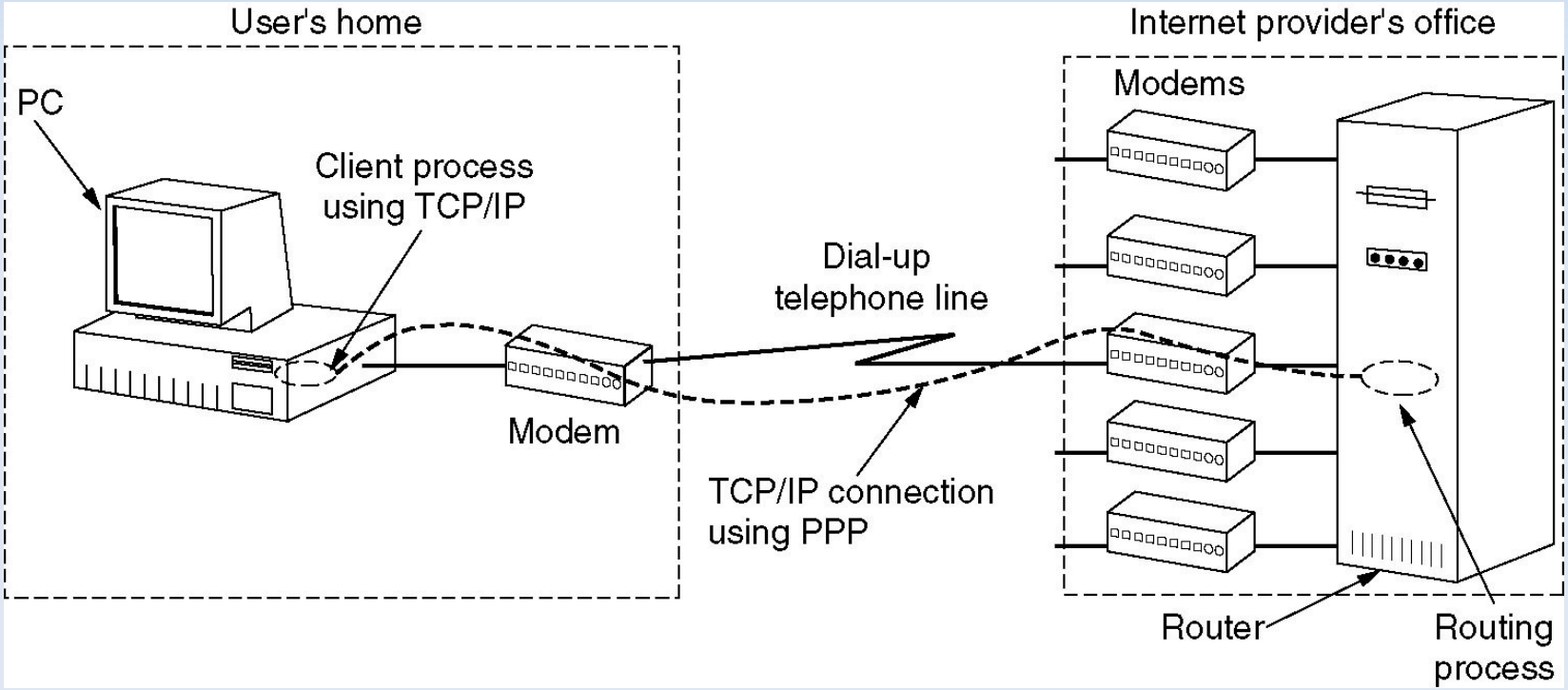
Ethernet (CSMA/CD)

- Exercice 3.5 (correction)
  - $T_{min} = 2.\tau$ avec $\tau = d_{max} / v$
  - $D = S_{min} / T_{min}$ donc $S_{min} = D.T_{min} = 2D.d_{max} / v$

- Cas Ethernet (10 Mbit/s)
  - D = 10 Mbit/s et d_{max} = 5000 m
  - v = 0,70 c = 200 000 km/s (vitesse signal électrique)
  - $T_{min} = 2\tau = 2.5000 / 200\,000\,000 = 50$ µs
  - En fait, T_{min}= 51,2 µs
  - Donc $S_{min} = T_{min} . D = 51{,}2.10^{-6} \times 10.10^{6} = 512$ bits = 64 o

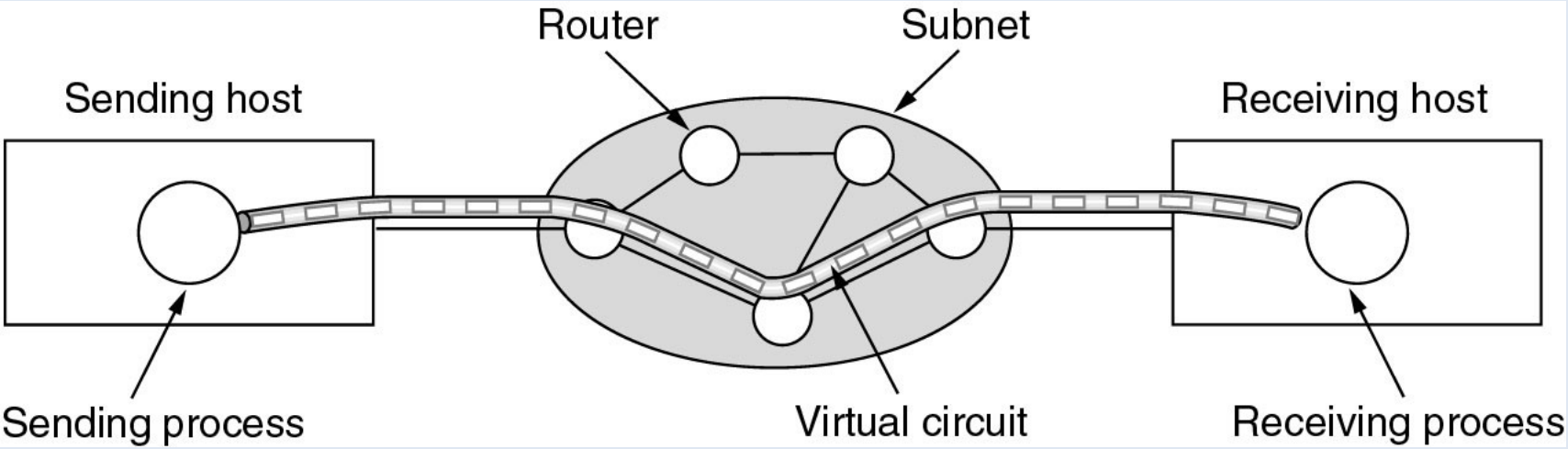
PPP

- PPP (Point-to-Point Protocol)
  - protocole de connexion directe entre une station et un FAI



ATM

- ATM (Asynchronous Transfer Mode)
  - très répandu au coeur des réseaux de télécommunication
    - en particulier, les FAIs ADSL
  - transmission des données par « cellules » de 53 octets plutôt que par trames de longueur variable comme Ethernet



ATM

- Exercice 3.7
  - On considère le lien ATM à fibre optique s'étendant sur 400 km. La vitesse de propagation dans la fibre optique est de 2.5×10^5 km/s. Le débit d'ATM est de 155 Mb/s. Contrairement à Ethernet qui utilise des trames de taille variable, ATM utilise des trames, appelées cellules, de taille fixe égale à 53 octets.
  - Questions
    - Calculer le temps de transmission d'une cellule.
    - Calculer la durée d'un trajet aller-retour.
    - Calculer la taille minimum de la fenêtre (en nombre de cellules) pour que l'émetteur puisse envoyer des cellules de façon continue avant de recevoir le premier acquittement.

ATM

- Exercice 3.7 (correction)
  - Avec un débit de 155 Mb/s, le temps de transmission d'une cellule de 53 octets est de :

$$t = (53 \times 8) / (155 \times 10^6) = 2.73\ \mu s$$

  - La durée du trajet aller-retour s'écrit :

$$t_{AR} = (2 \times 400) / (2.5 \times 10^5) = 3.2\ ms$$

  - L'accusé de réception parviendra à la source après une durée t_{AR}. Pendant ce temps, il faut que la source envoie des cellules de manière continue. En d'autres termes, la taille minimum de la fenêtre doit être :

$$3.2\ X\ 10^{-3} / 2.73 \times 10^{-6} = 1172\ \text{cellules}$$

Annexes

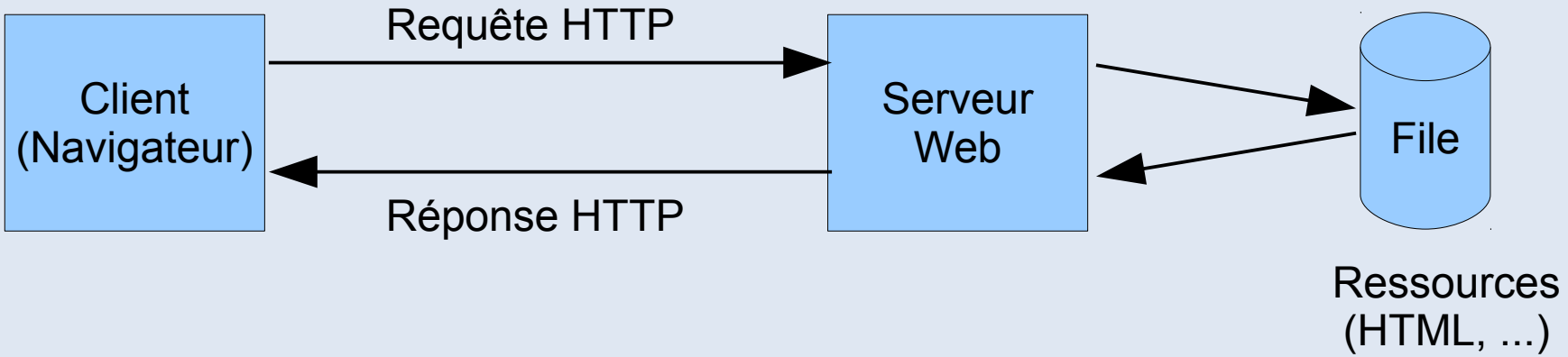
Quelques Notes Diverses

TODO

- Quelques idées pour compléter ce cours :
  - HTTPS, SSH, certificat x509
  - Web Service REST
  - ADSL, WIFI ?

HTTP

- HTTP (HyperText Transfer Protocol)
  - Protocole *stateless* basé sur TCP/IP (port 80)
  - Le navigateur effectue une requête HTTP pour obtenir la ressource URI (Uniform Ressource Identifier)
  - Le serveur traite la requête puis retourne une réponse HTTP, typiquement une page HTML



HTTP

- Requêtes
  - GET : demander une ressource (la plus courante)
  - POST : ajouter une nouvelle ressource (ex. message de forum)
  - HEAD : demander uniquement l'en-tête HTTP
  - TRACE : echo de la requête
  - CONNECT, PUT, DELETE, ...
- Historique
  - Version 0.9 : requête GET, réponse HTML
  - Version 1.0 : gestion de cache, description du type MIME des ressources (content-type), ...
  - Version 1.1 : connexion persistante (keep-alive), négociation de contenu (accept-*), ...

Un peu de HTML

- Structure classique

```
<html>
  <head>
    <title>Hello World!</title>
     ...
  </head>
  <body>
    <center>
      <h1>Hello World!</h1>
    </center>
    <h2> Subtitle </h2>
    <p> paragraph </p>
    ...
  </body>
</html>
```

- Formulaire HTML
  - Passage de paramètres (POST)

```
...
<form action="/target" method=POST>
  First Name:
  <input type=text size=20 name=firstname>
  <br>
  Last Name:
  <input type=text size=20 name=lastname>
     <br>
  <input type=submit>
</form>
...
```

Les formulaires HTML

- A compléter….
  - http://www.commentcamarche.net/contents/html/htmlform.php3

Exemple HTTP

- Requête

GET /HelloWorld.html HTTP/1.1 ← *commande GET*

Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.15) Gecko/2009102815 Ubuntu/9.04 (jaunty) Firefox/3.0.15
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
If-Modified-Since: Thu, 19 Nov 2009 14:06:01 GMT
If-None-Match: W/"153-1258639561000"
Cache-Control: max-age=0

header

Exemple HTTP

- Réponse

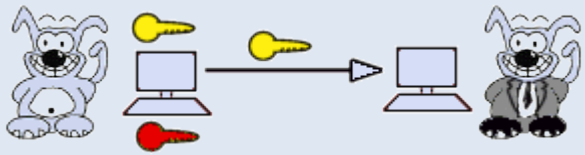
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"153-1258639561000"
Last-Modified: Thu, 19 Nov 2009 14:06:01 GMT
Content-Type: text/html
Content-Length: 153
Date: Tue, 24 Nov 2009 15:48:32 GMT
Connection: close

<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <center>
      <h1>Hello World!</h1>
    </center>
  </body>
</html>
```

type MIME de la ressource

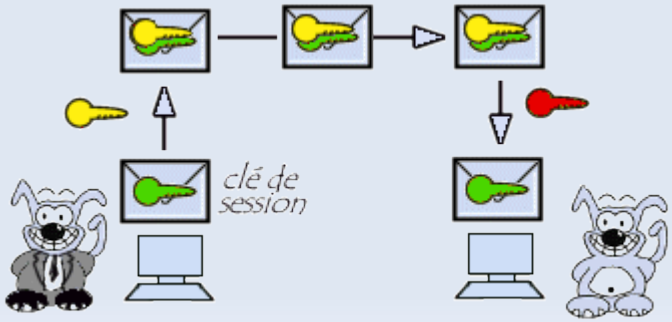
corps de la réponse

Rappel Cryptographie

- Cryptographie à clé publique (asymétrique)
  - Confidentialité : chiffrement par Bob du message avec la clé publique d'Alice et déchiffrement par Alice avec sa clé privée
  - Signature : chiffrement par Alice d'un condensat du message avec sa clé privée et déchiffrement par Bob avec la clé publique d'Alice, qui peut ainsi vérifier le condensat

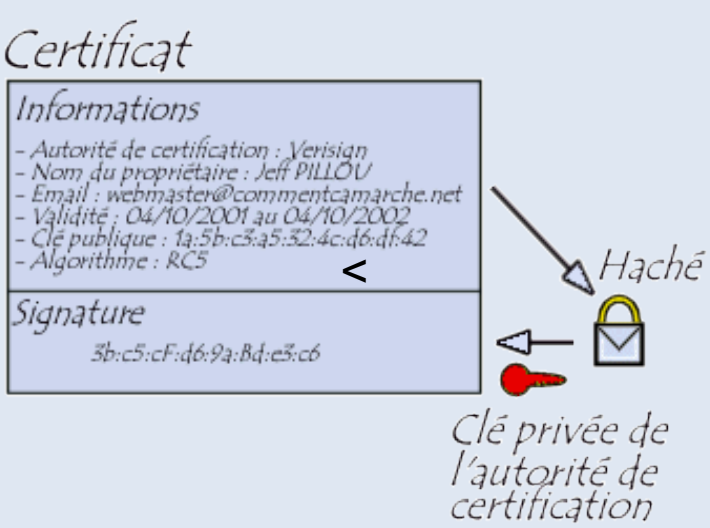


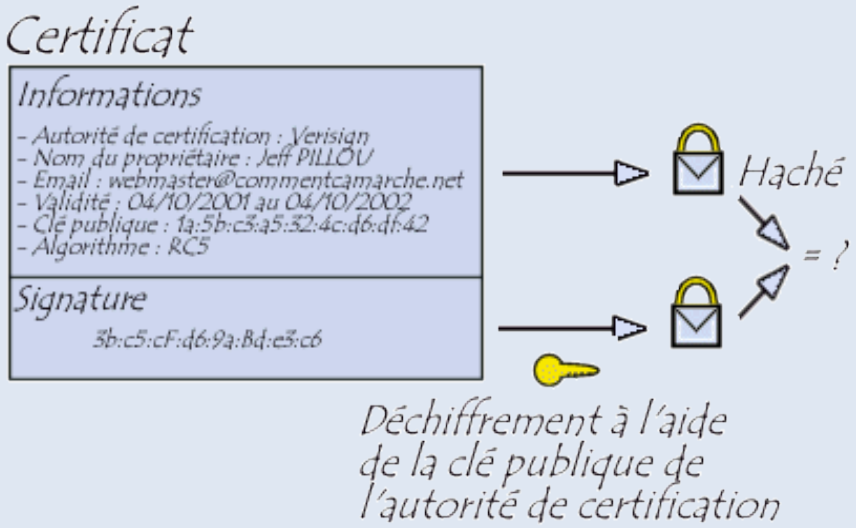
  - Etablissement d'une clé de session symétrique...



HTTPS

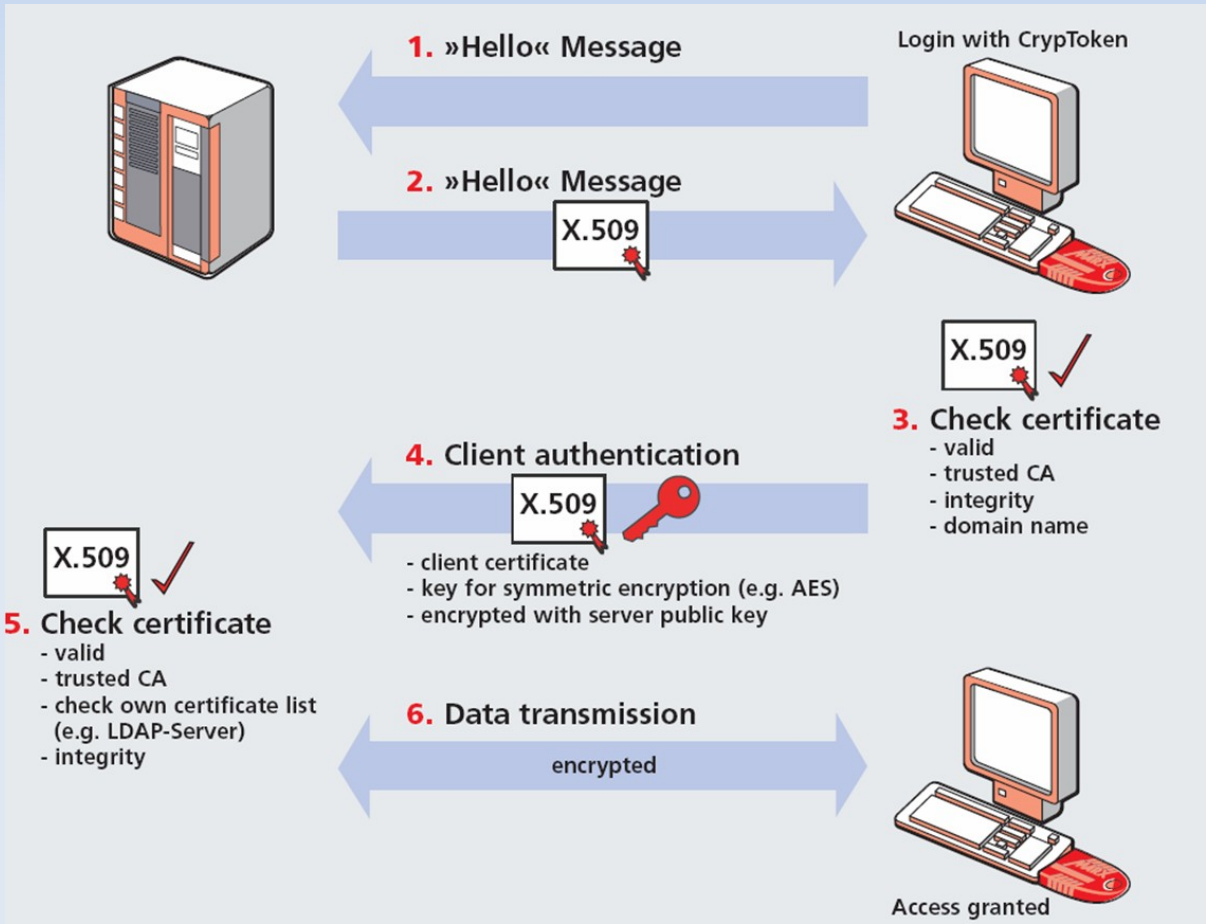
- HTTPS (HTTP Secure)
  - Utilisation transparente du protocole HTTP au-dessus de TLS/SSL (port 443 au lieu de 80)
  - Authentification du serveur web via son certificat (signé du CA)
  - Confidentialité et intégrité des données envoyées au serveur
  - Authentification du client facultative





HTTPS

- Authentification du serveur et du client avec SSL/TLS



Source : wikipedia

Web Service REST

- Requête HTTP GET d'une URI → Réponse XML ou JSON

```json
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

```xml
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

- Exemples
  - https://developers.facebook.com/tools/explorer
  - http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP?ZIP=90001

REST

- API Google Direction
  - Exemple

    http://maps.googleapis.com/maps/api/directions/json?origin=Bordeaux&destination=Toulouse&sensor=false
  - Réponse JSON

    (...)
  - API https://developers.google.com/maps/documentation/directions/?hl=fr-FR&csw=1