Communications numériques et protocoles

Builders FOR SOCIETY
ÉCOLE D'INGÉNIEURS

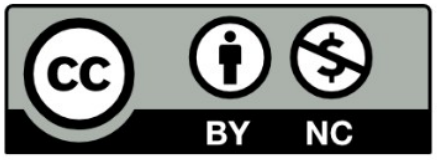
ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

L'École des INGÉNIEURS Scientifiques

Licence du document

Auteurs

- FOUREY Sébastien
- LEVEE Freddy
- LEFEBVRE Philippe



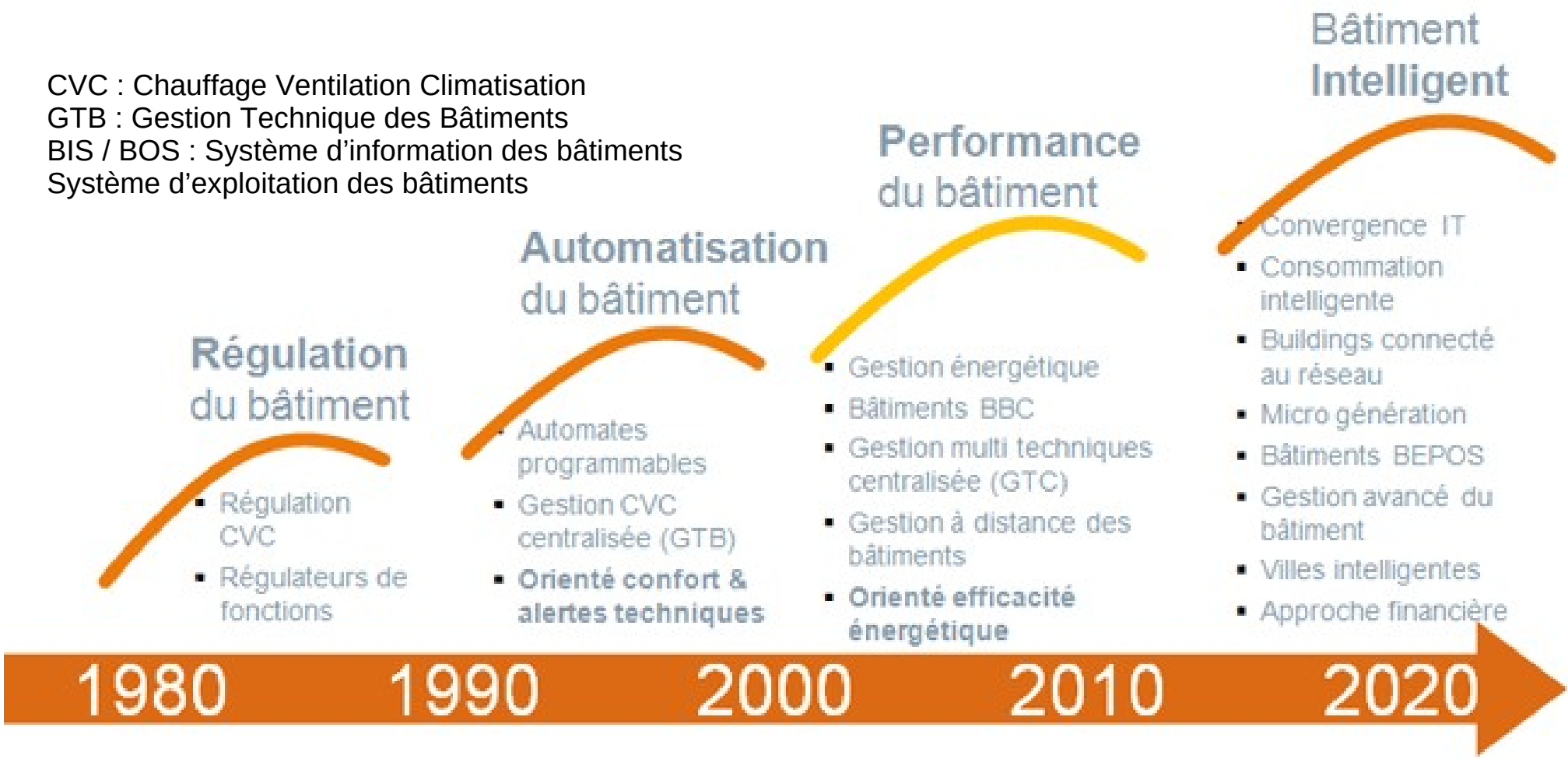
Organisation

4,5 h de cours

- Introduction, Modèle OSI
- Couches basses
- Interconnexion / Sécurité

Un bâtiment numérique intelligent c’est quoi ?

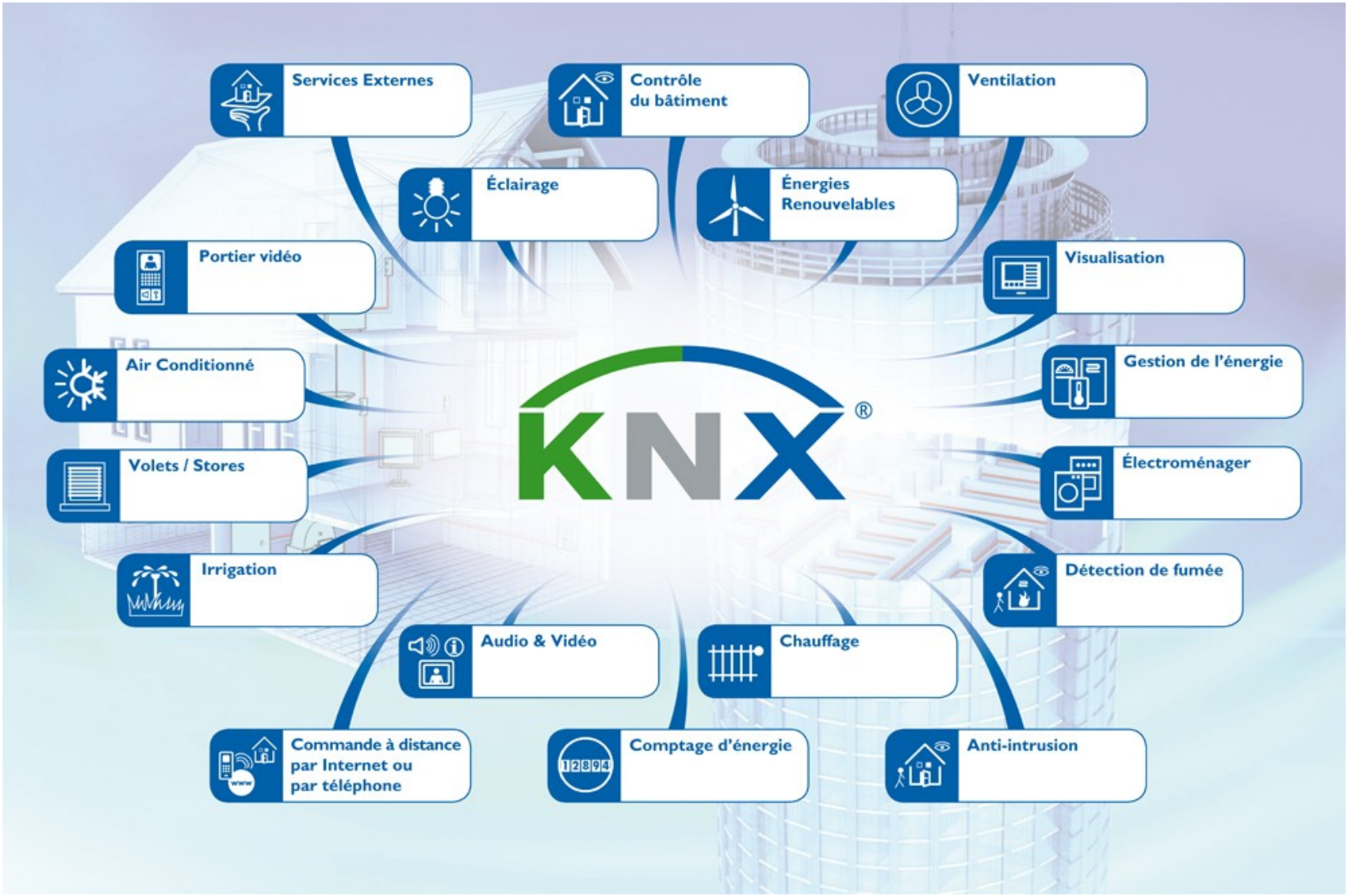
CVC : Chauffage Ventilation Climatisation
GTB : Gestion Technique des Bâtiments
BIS / BOS : Système d’information des bâtiments
Système d’exploitation des bâtiments



Illustrations : conseils.xpair.com

Des bâtiments intelligents pour quoi faire ?

Gestion du confort
Sécurité incendie
Contrôle d'accès
Détection intrusion
Vidéo-surveillance
Gestion énergétique
Gestion électrique
Smart Building



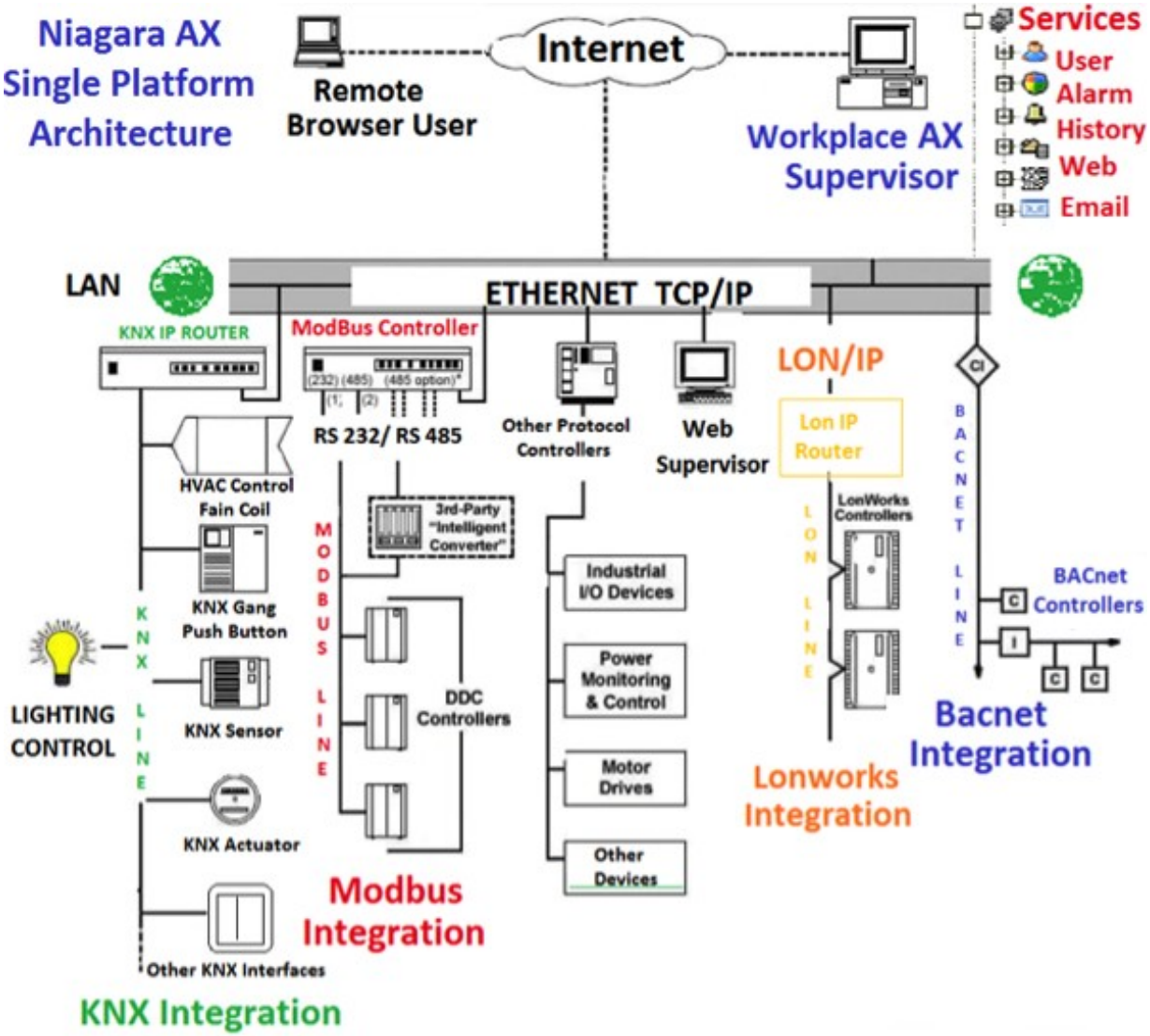
Des maisons connectées pour quoi faire ?

S’inscrire sur le cours de France Université Numérique : Maison connectée

https://www.fun-mooc.fr/courses/course-v1:CNAM+01058+session01/about

- Service vidéo : télévision en direct, Video On Demand...
- E-santé : téléconsultation, télésurveillance...
- Gestion de l’énergie : smart-grid, télé-collecte...
- Services de confort : domotique
- Services de sécurité : alarme anti-intrusion...

Des bâtiments intelligents
De nombreux sigles



ESITC CAEN
Ecole Supérieure d'Ingénieurs des Travaux de la Construction

Plan du cours

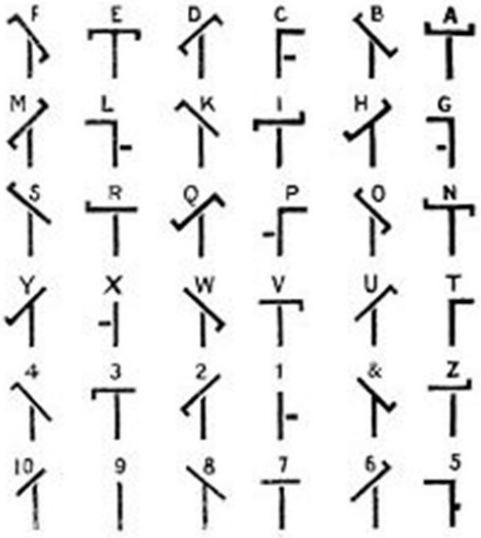
- Le modèle OSI
- Les méthodes d'accès : ethernet, wifi, GSM/GPRS/UMTS, fibre optique, liens série, bus de terrain
- La commutation de paquets / la commuation de circuits virtuels
- La couche réseau : IP
- La couche transport : UDP, TCP
- Quelques services applicatifs : Web, DNS, courrier,
- La sécurité

Historique

Muraille de chine, 400 av. JC. : Tour de signaux

Claude Chappe et son télégraphe – 1793 (Paris-Lille),

utilisé jusqu’en 1854 par l’armée.



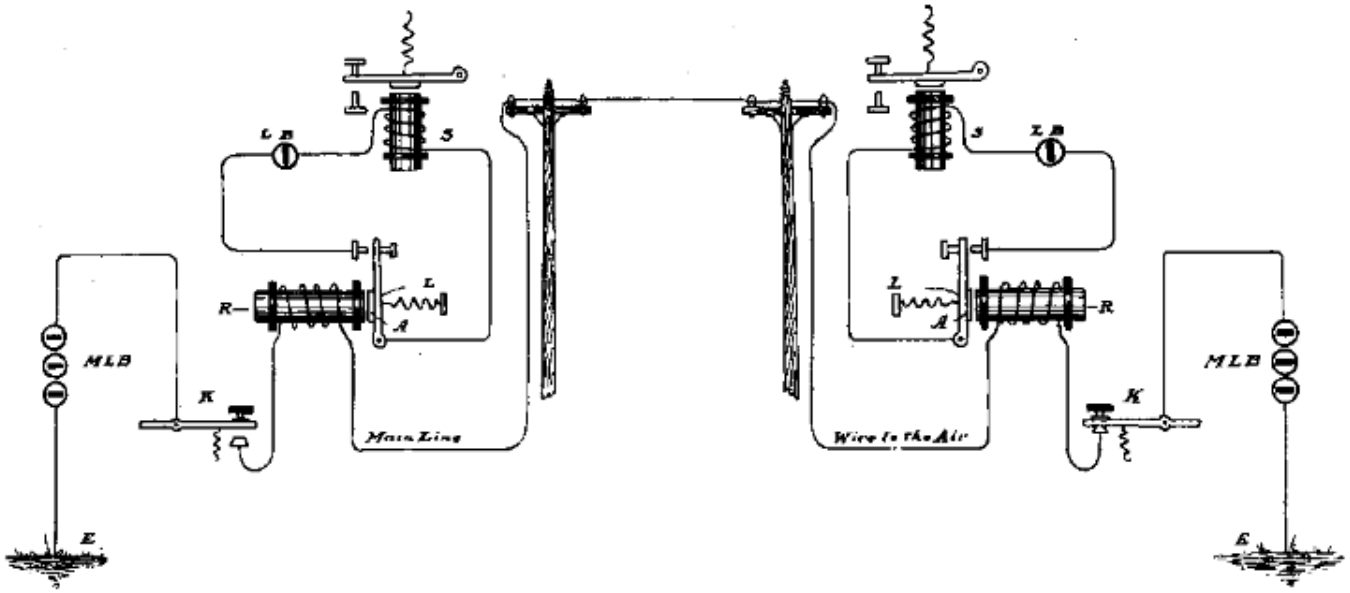
Historique

Télégraphe électrique

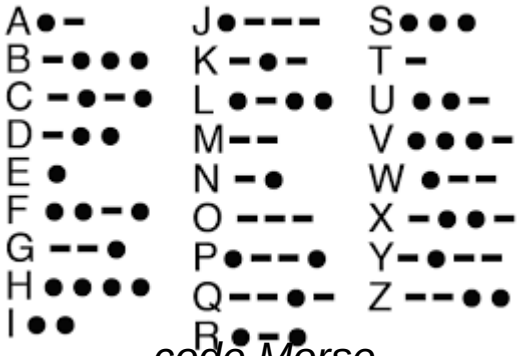
Gauss and Weber, 1833 (Göttingen Observatory)

Cooke and Wheatstone, 1837 (2 fils, Great Western Railway)

Samuel Morse, 1844 (1 fil)



The Main Line Circuit.



code Morse

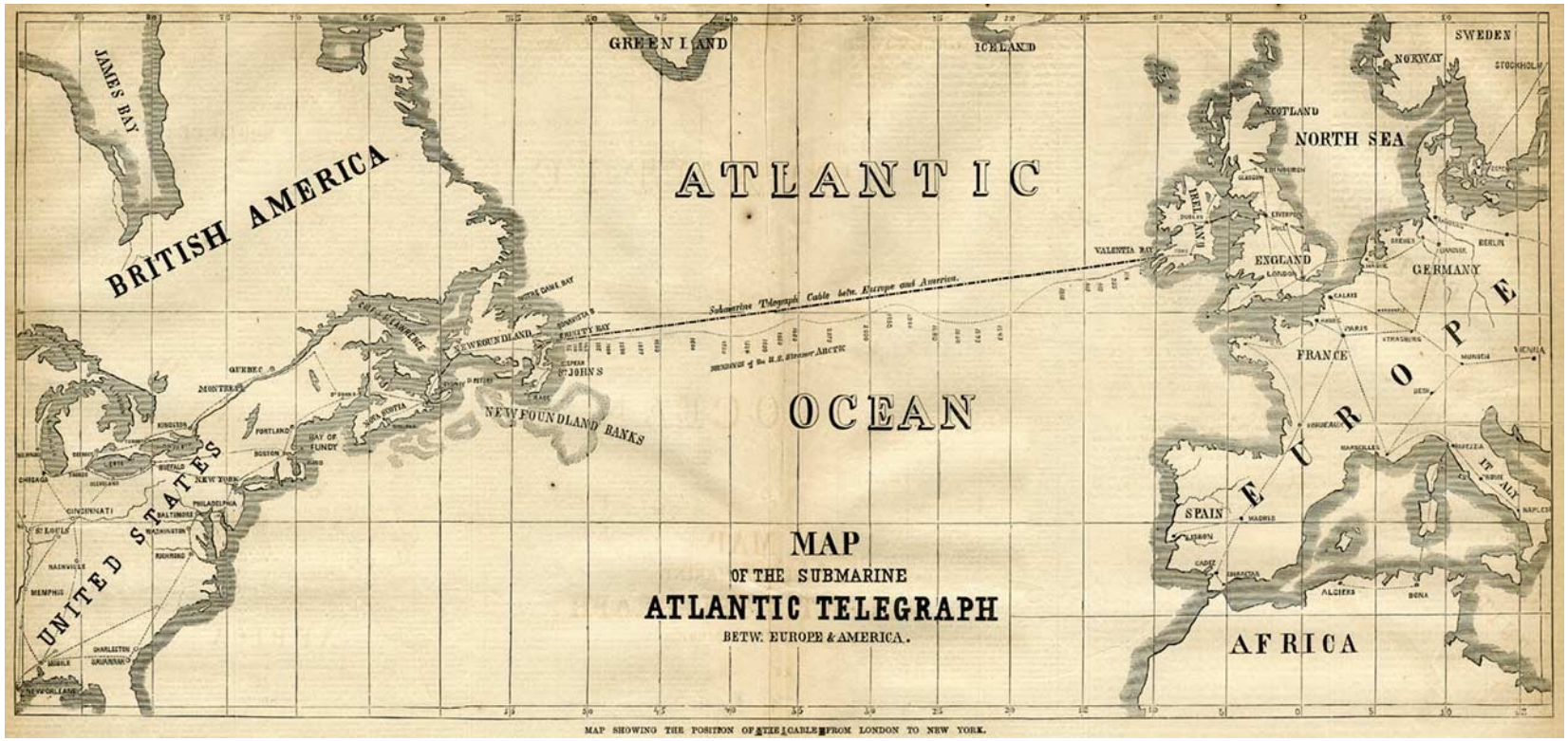
Historique

Premier câble de télégraphe transatlantique

opérationnel en 1866

sans répéteur

navire cablier (wikipedia)



Historique

Bell et le téléphone - 1870

Graham Bell

Table de commutation – city of Vancouver (1940)
Ici on peut réellement parler de connexion entre deux abonnés !

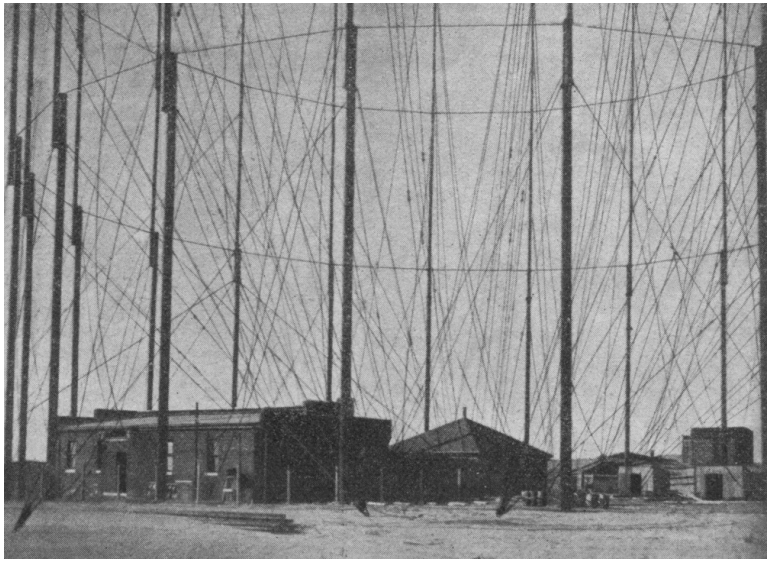
Historique

Dès 1895, Hertz, Popov, Marconi et Tesla font des expérimentations sur les ondes électromagnétiques.

1901, Guglielmo Marconi réalise la première transmission radio transatlantique

1957 Spoutnik

spoutnik

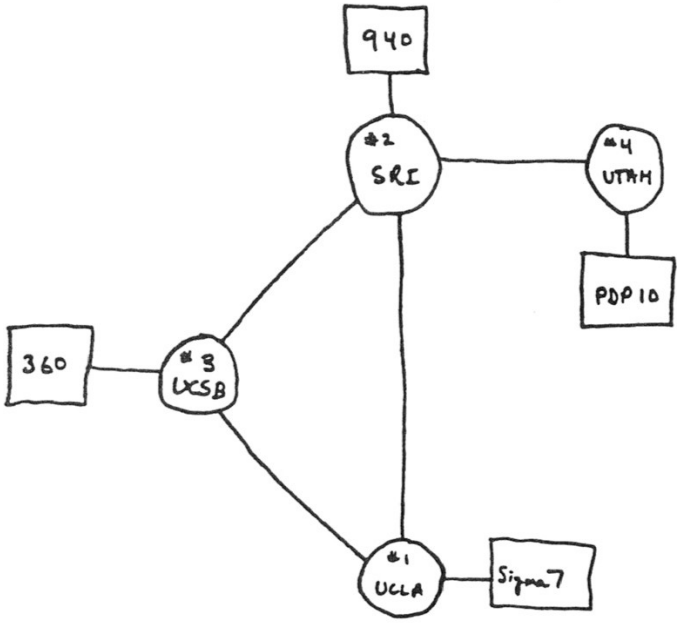


Antenne TSF en 1901 (wikipedia)

Historique

1969, 40 terminaux peuvent dialoguer entre eux : ARPANET, la naissance du réseau Internet.

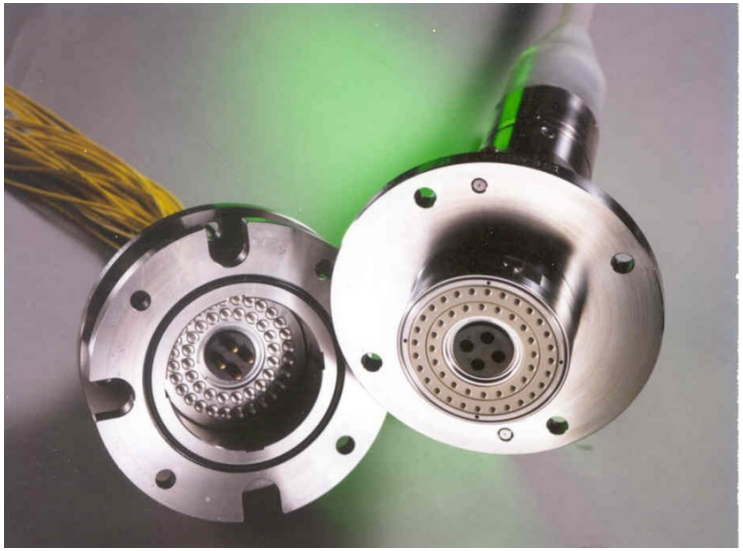
DEC PDP10 (UTAH)



Historique

1950, Alfred Kastler invente le laser

1977, premier système de communication par fibre optique installé à Chicago



connecteur de câble sous-marin (Antares) 48 fibres.
Au centre, câbles électriques pour alimentation des répéteurs

Record mondial de transmission optique

38,4 Tbits/s

(térabits par seconde, millions de mégabits par secondes) réalisé sur le réseau fibre opérationnel d'Orange sur les 762 km de la liaison Lyon-Marseille-Lyon, en 2015

Record de capacité de transmission à

1,5 Tbits/s

entre Varsovie et Wroclaw (CP Orange Pologne / Nokia du 21 juillet 2016)

2016 - record de transmission par fibre optique Orange/Nokia (1,5 Tb/s sur 1 seule fibre de 870 km)

2022 – 1,2 Pb/s répartis sur 4 fibres de 52 km (NICCT)

Prévisions (Cisco)

Prévision 2022 à partir des données 2017 par Cisco	2017	2022
données échangées	$1{,}5 \times 10^{21}$ octets	4×10^{21} octets
données échangées par tête	16 GB	50 GB
nombre d'appareils connectés	18×10^{9} (2,4 / hab.)	28×10^{9} (3,6 / hab.)
Part du trafic vidéo / total	75 %	82 %
Part du trafic dû aux PC	41 %	19 %
Part du trafic dû aux mobiles	18 %	44 %

Au total, le numérique consomme 10 à 15 % de l'électricité mondiale, soit l'équivalent de 100 réacteurs nucléaires. **Et cette consommation double tous les 4 ans !**

Une tranche de centrale $1{,}3 \times 10^{9}$ W → soit 10 TWh par tranche

(la France compte 58 tranches qui produisent 75 % de l'électricité française).

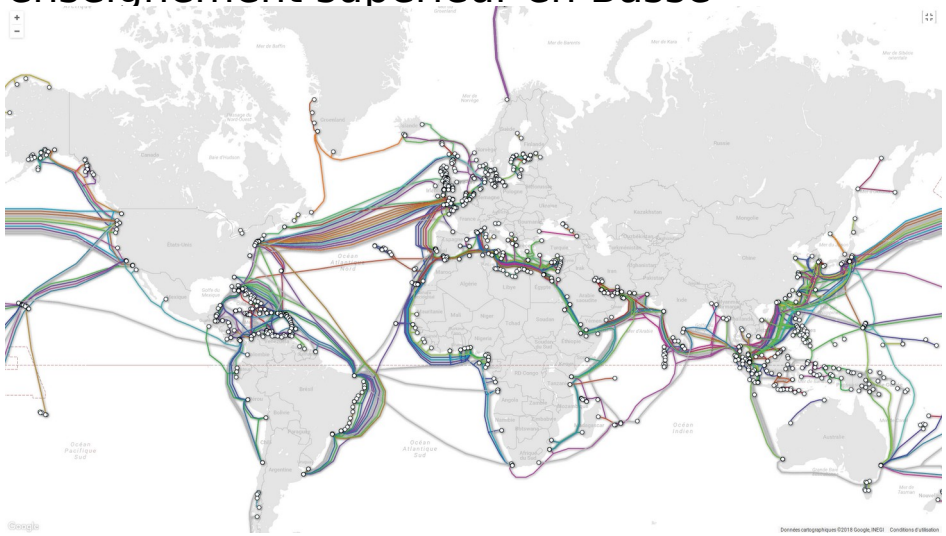
ESITC CAEN
Ecole Supérieure d'Ingénieurs des Travaux de la Construction

Les réseaux sont classés en catégories en fonction de leur grandeur :

- PAN (Personnal Area Network) : ~ qq m
  - Bluetooth, réseau sans fil entre un téléphone mobile et son oreillette
  - Zigbee réseau de capteurs sans fil (commande de volets...)

- LAN (Local Area Network) : ~ qq 100 m
  - Ethernet pour le réseau d'une entreprise.
  - Wifi

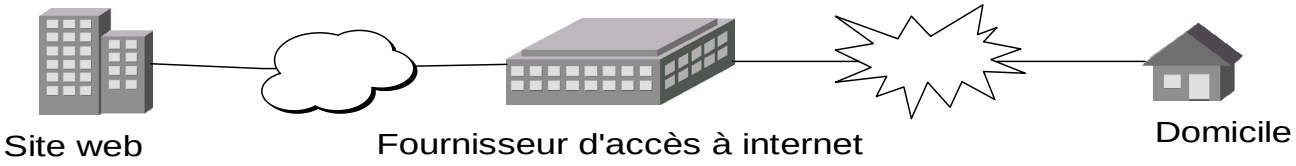
- MAN (Metropolitan Area Network) : qq km
  - Vickman, le réseau d'interconnexion de l'enseignement supérieur en Basse-Normandie.
- WAN (Wide Area Network) : le monde
  - comme le réseau Internet, bien sûr...
  - Le réseau téléphonique



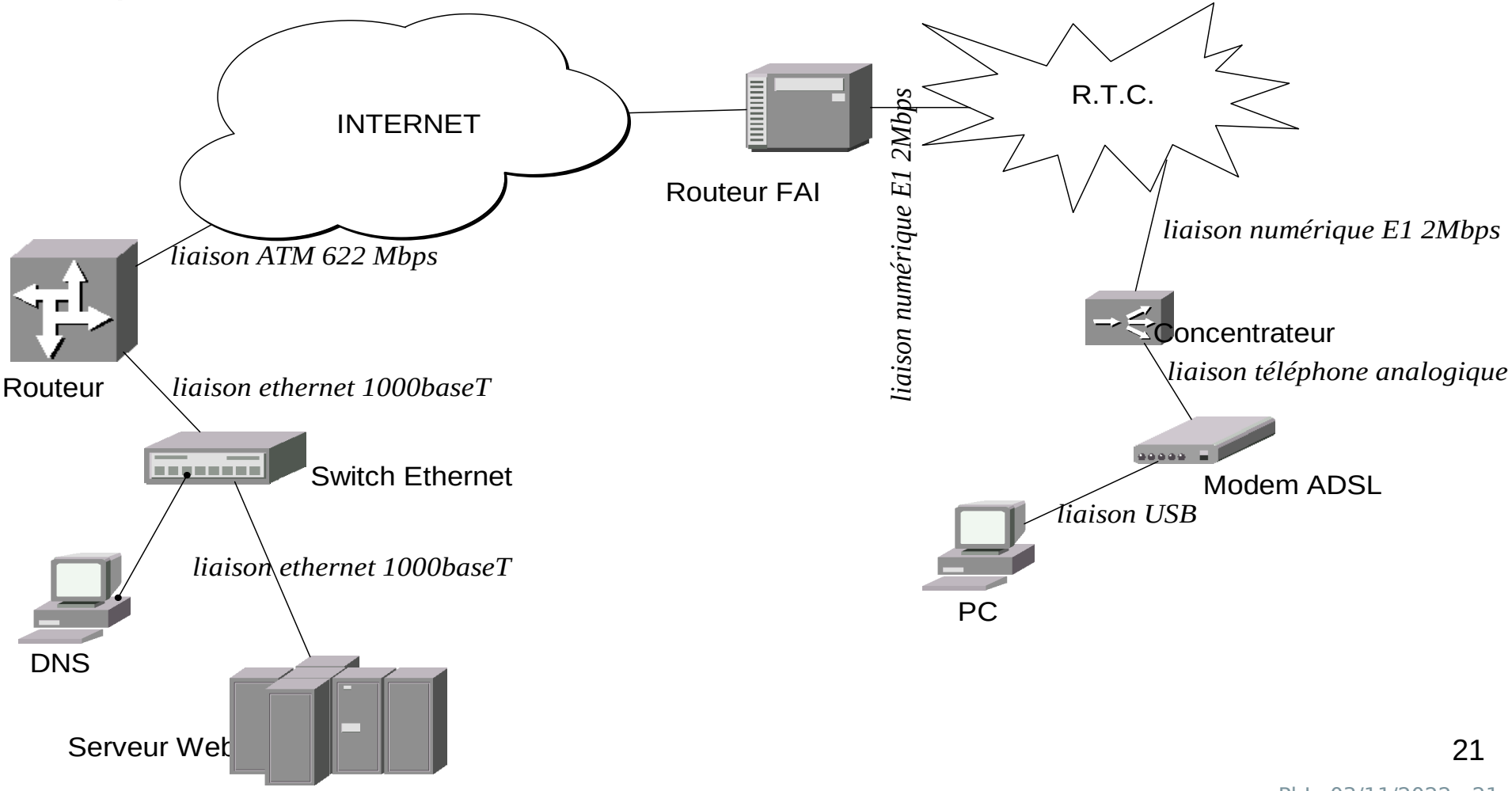
Quelques débits nécessaires à la réception en temps réel :

- voix codée en RPE-LTP (GSM) : 13 kbps
- voix non codée (PCM) : 64 kbps (débits des vieux modem)
- HI-FI codée en MPEG3 : 128 kbps
- vidéo au format UltraHD 4K : 25 Mbps

Délai maximum d'acheminement garantissant un bon confort d'interactivité perceptible par l'homme : **50 ms**



Un exemple de communication : consultation d'un site web



ESITC CAEN
Ecole Supérieure d'Ingénieurs des Travaux de la Construction

Exemple de problèmes à résoudre : (découpage selon le modèle OSI)

Couche 1 (physique)

- Quels niveaux de tension sur la ligne ?
- Codage NRZ ou Manchester différentiel ?
- Quelle cadence ?
- Quelle forme de connecteur ?

Couche 2 (liaison)

- Comment s'assurer du début d'un envoi d'information ?
- Comment se synchroniser ?
- Comment être sûr qu'il n'y a pas eu d'erreurs de transmission ?
- Le WIFI émet à 300 Mbps, l'ADSL à 2 Mbps. Comment éviter l'engorgement au niveau du modem ?
- Le PC et la TV peuvent vouloir parler en même temps. Comment gérer le tour de parole ?
- Comment faire pour envoyer les informations du PC à la Box et non à la TV ?
- Puis-je multiplexer mes données sur plusieurs liaisons physiques ?

Exemple de problèmes à résoudre (suite) :

Couche 3 (réseau)

- Comment aller du PC au routeur du FAI ? Par quel chemin ?
- Je peux joindre tout le monde. Comment gérer l'unicité des adresses ?
- Y a-t-il un centre de tri qui connaît toutes les adresses de tout l'internet ?
- Et si un second message arrive avant un premier, comment les remettre dans l'ordre ?
- Et si mon message pendant son voyage arrive sur un support qui n'accepte que des petits messages ! Comment segmenter ?
- Et si un message ne trouve pas son destinataire ?

Couche 4 (transport)

- Le message arrive enfin sur ma machine ! Pourquoi est-il acheminé sur mon navigateur et pas sur mon logiciel de peer2peer ?
- Au fait, après avoir traversé autant de supports physiques différents, le message a-t-il été altéré ?
- Comment être sûr que le message soit arrivé ?
- Comment initier et rompre proprement le dialogue ?
- Comment faire patienter le l'émetteur ?

Exemple de problèmes à résoudre (suite) :

Couche 5 (session)

- Comment m'assurer de l'identité de l'utilisateur ?
- Suis-je capable de reprendre correctement une session brutalement interrompue ?

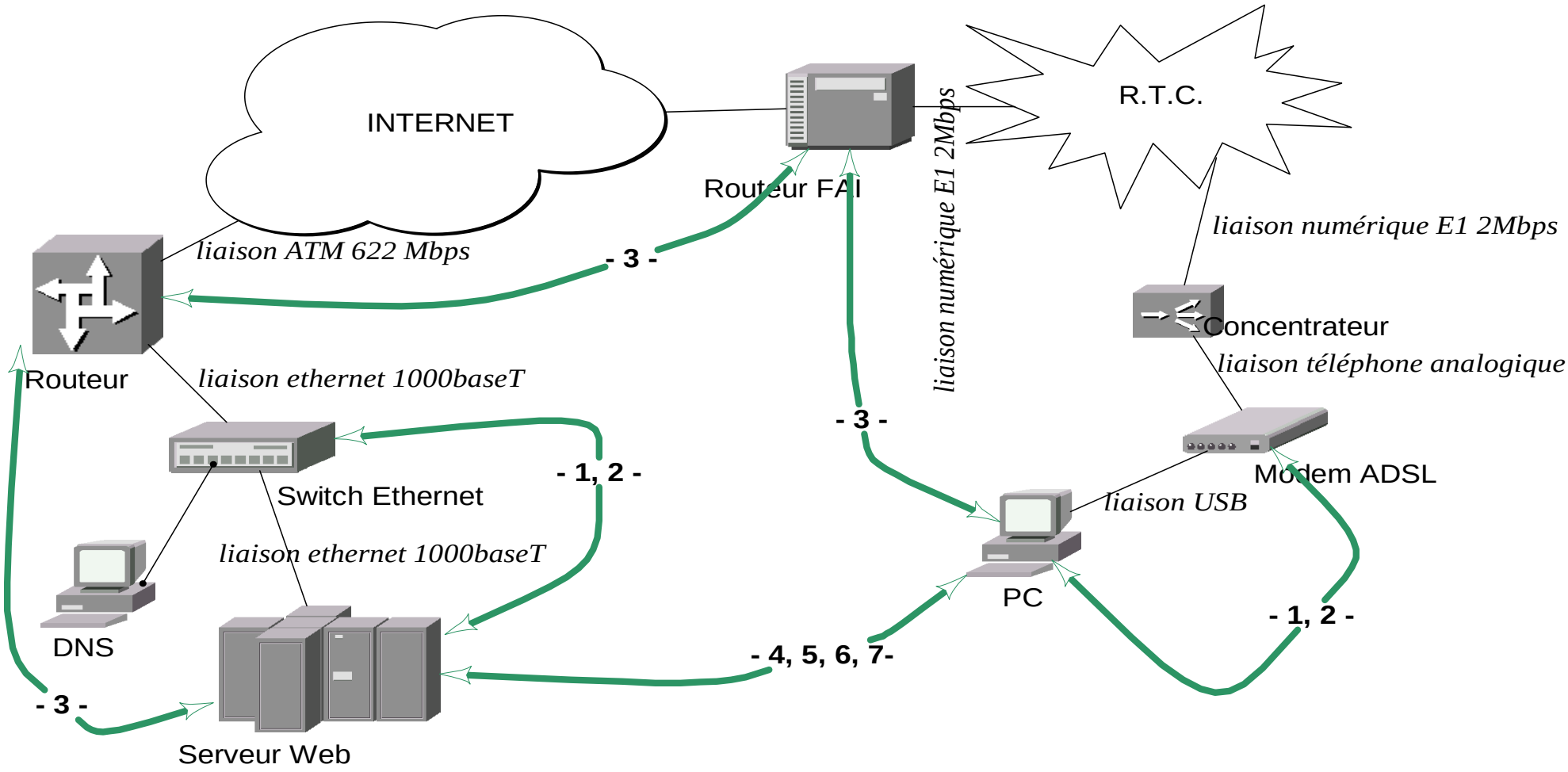
Couche 6 (présentation)

- Et si j'envoie le mot «Été » sera-t-il bien transmis ?
- Et les nombres entiers du destinataire : 16 bits ou 32 bits ?
- Et si je lui envoie une image, quel format ?

Couche 7 (application)

- Le courrier électronique, le partage de fichier, la vidéo-conférence, un jeu, un annuaire...

Les interactions entre couches



Le modèle OSI de l'ISO

Modèle **O**pen **S**ystem **I**nterconnection de l'International Standards Organization

Autres organismes classiques de normalisation:
- American National Standards Institute (ANSI). ex : le langage C, norme ATA
- Electronic Industries Association (EIA). ex : RS 232C
- Institute of Electrical and Electronic Engineers (IEEE). ex : ethernet
- International Telecommunication Union Telecommunication Standardization Sector (ITU-T). ex : G711 (codage de la voix)
- Internet Activities Board (IAB). ex : les RFC sur IP.

Élaboré en 1984 ; idée : **Compatibilité entre tous les systèmes !**

PDU : Protocol Data Unit : bits, octets, trames, fichiers... => modèle en couches.

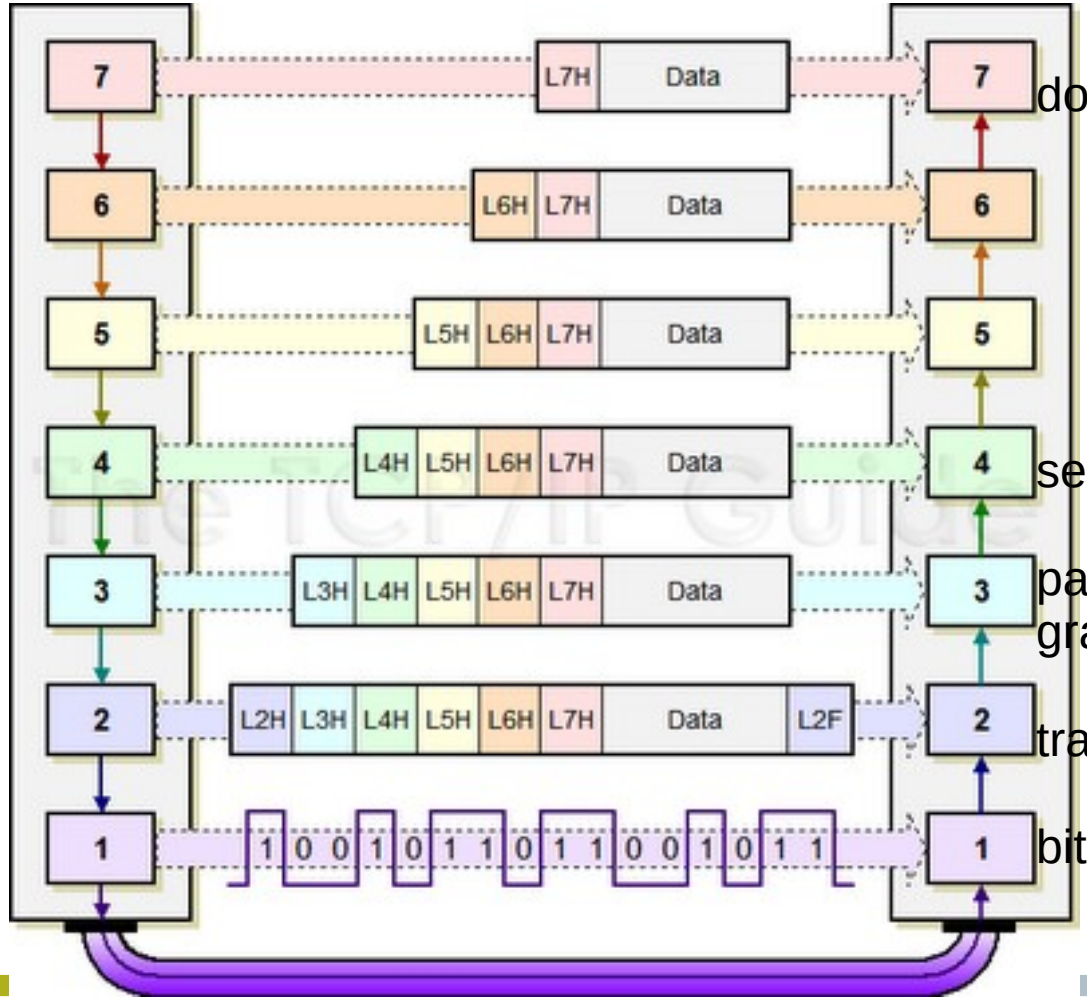
Le modèle OSI de l'ISO

No	nom	Rôle	Exemple de PDU	exemple de protocole
7	Application	Applications s'appuyant sur le réseau : transfert de fichiers, émulation de terminal, messagerie, partage de fichiers, web…	fichier	telnet, HTTP, mail, NFS, FTP
6	Présen-tation	Conversion des données numériques propres au réseau dans leur version finale ou abstraite compréhensible par le programme. Cette couche est souvent associée à un langage possédant des règles lexicales (les mots), syntaxiques (la grammaire) et sémantique (le sens).	chaîne de caractères	ASN1...
5	Session	Gestion d'une session : ouverture, mots de passe, reprise en cas d'erreur, fermeture.		Netbios
4	Transport	Multiplexage/démultiplexage des paquets, segmentation, contrôle de flux, correction des erreurs. Service de bout en bout.	paquet	~TCP
3	Réseau	Service point à point. Assure le routage, l'adressage.	trame	~IP
2	Liaison	Délimitation d'une trame, contrôle et correction d'erreurs, contrôle de flux, règle d'accès au médium. Service point à point.	trame	HDLC
1	Physique	Conversion des signaux électriques en bits. Définition des caractéristiques électriques et mécaniques du support de transmission.	bits	~ethernet

Le modèle OSI de l'ISO

L'information est regroupées coupée en PDU : Protocol data unit. Le nom de la PDU change selon le niveau du modèle.

Encapsulation des PDU : la PDU de couche N est contenue dans la PDU de couche N-1.



PDU

données

segment

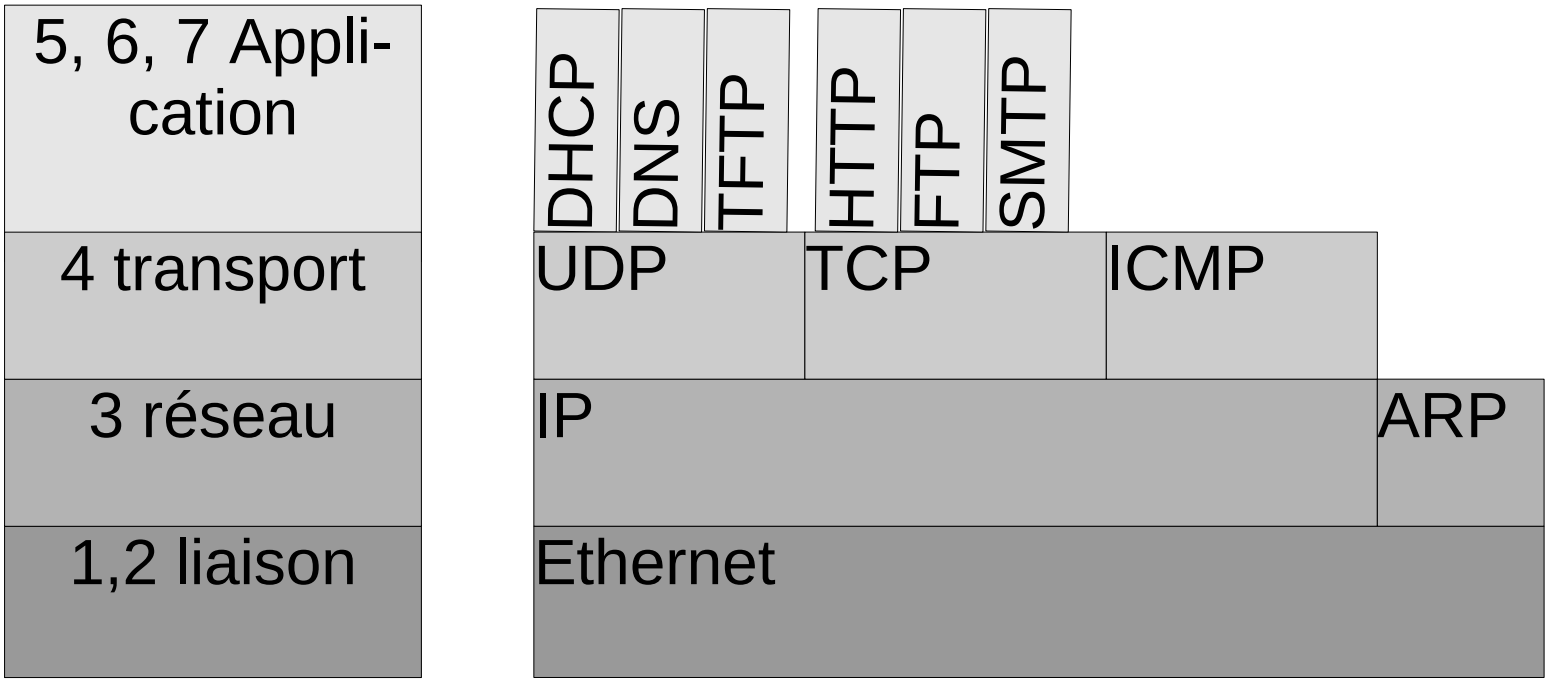
paquet, data-gramme

trame, cellule

bit

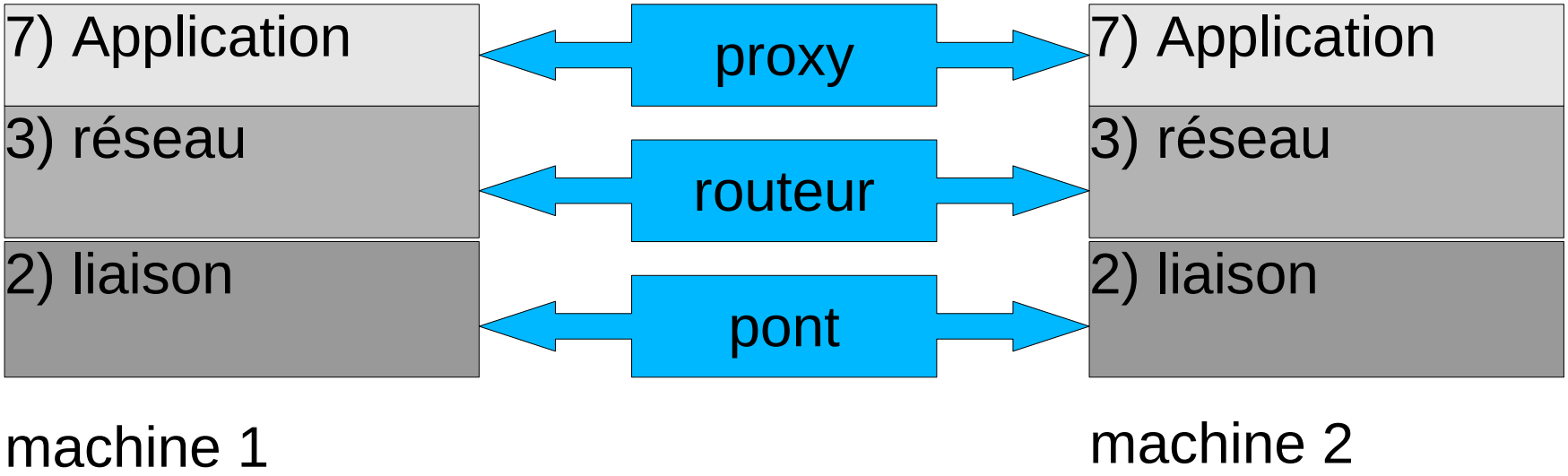
Le modèle « ethernet/TCP/IP »

Presque ISO !

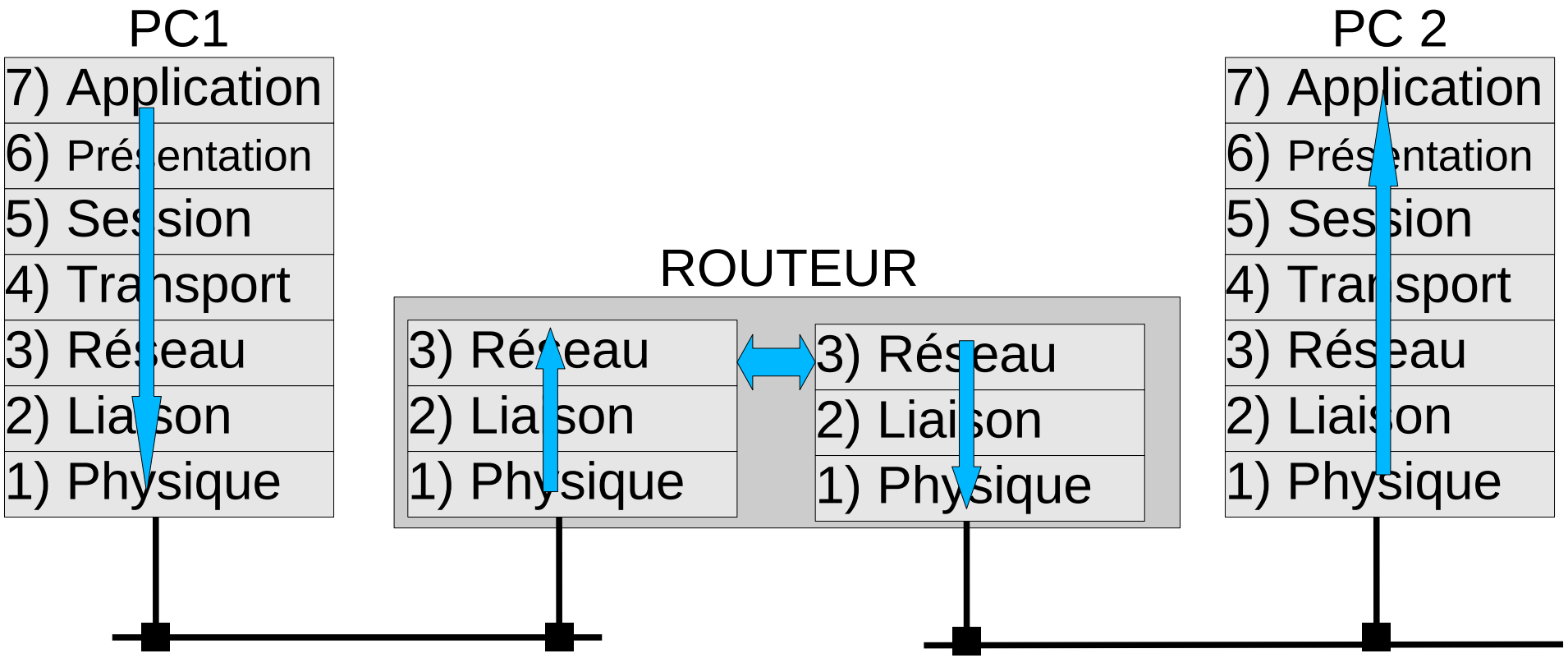


Les passerelles

- Machines qui permettent de relayer les informations lorsque la liaison n'est pas directe.
- Le terme passerelle est générique

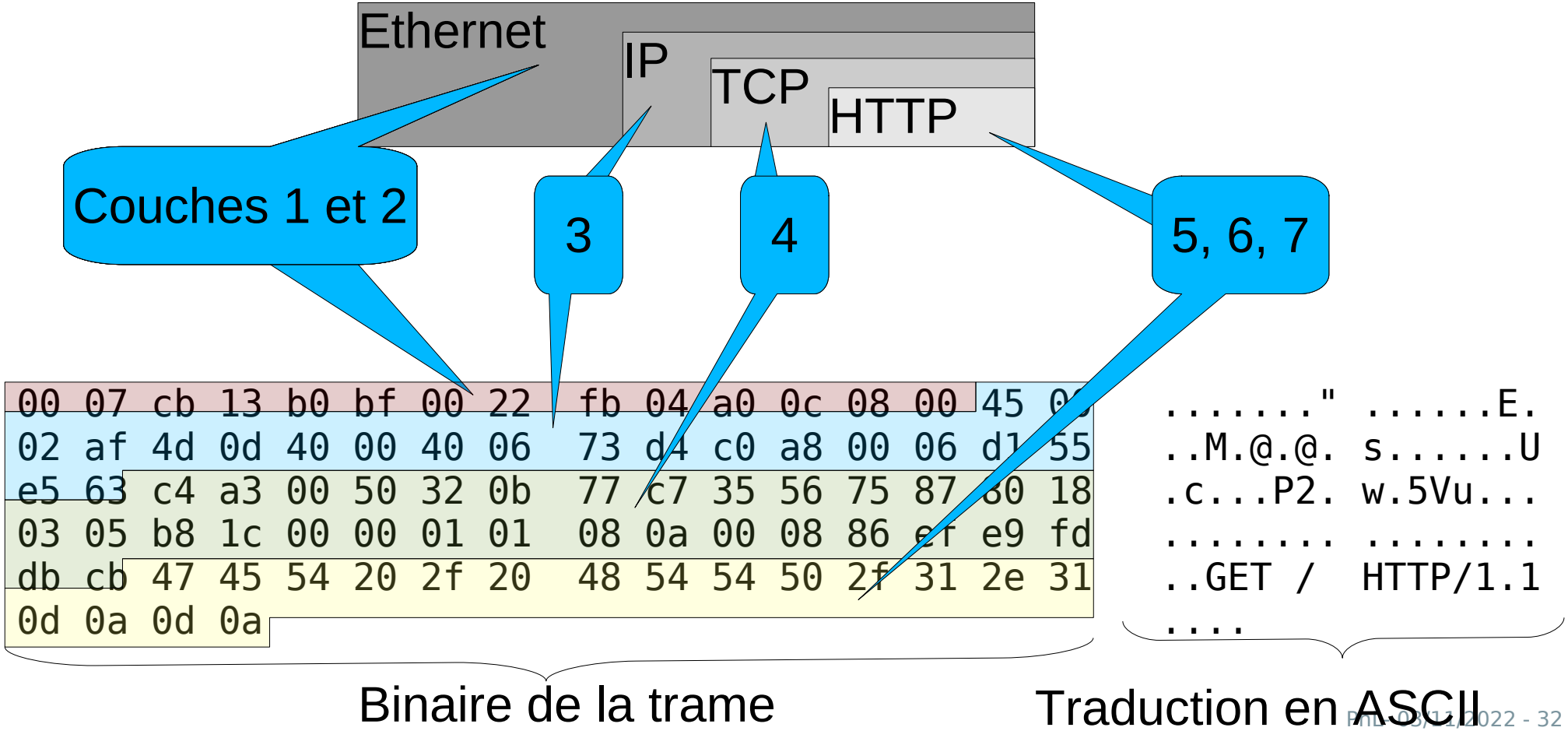


Exemple un routeur



Le modèle Ethernet – IP

Exemple d'une requête envoyée par un navigateur (demande la page d'accueil de Google à partir d'un PC connecté à une Freebox)



ESITC CAEN
Ecole Supérieure d'Ingénieurs des Travaux de la Construction

Les couches basses

PLAN

1.Topologie

2.Codage

3.Les méthodes d'accès au média

4.Cas de la téléphonie mobile

5.Cas des réseaux longue distance sur fibre optique

6.Ethernet

7.Le Wifi

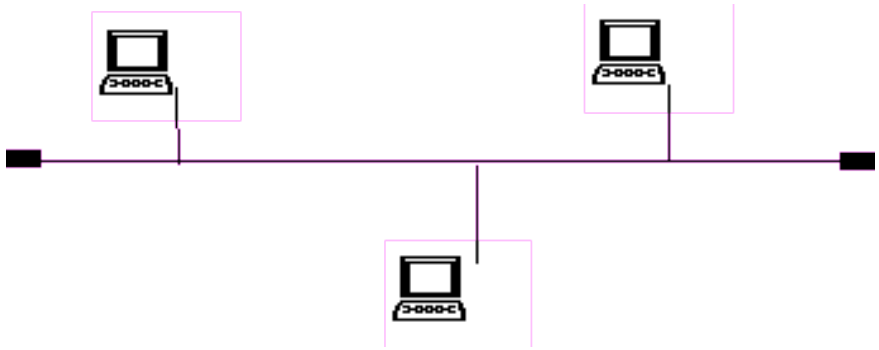
8.Les réseaux de terrain :HPIB, GPIB, CAN , Modbus

Topologie

2 types de topologie:
Physique: la forme du maillage physique (câbles à l'intérieur d'un bâtiment)
Logique: la forme "que voit le protocole".

Topologie en bus
- simple
- diffusion naturelle des trames

mais
- localisation des pannes difficile
- collisions de trames



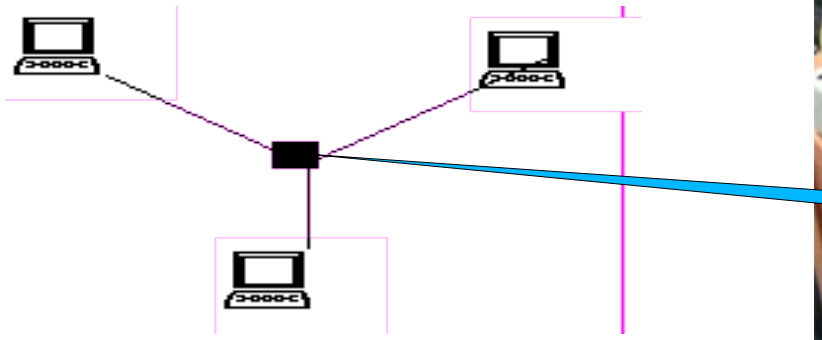
C'est la topologie des réseaux embarqués : CAN bus, LIN bus, 1-wire ®, I2C ®

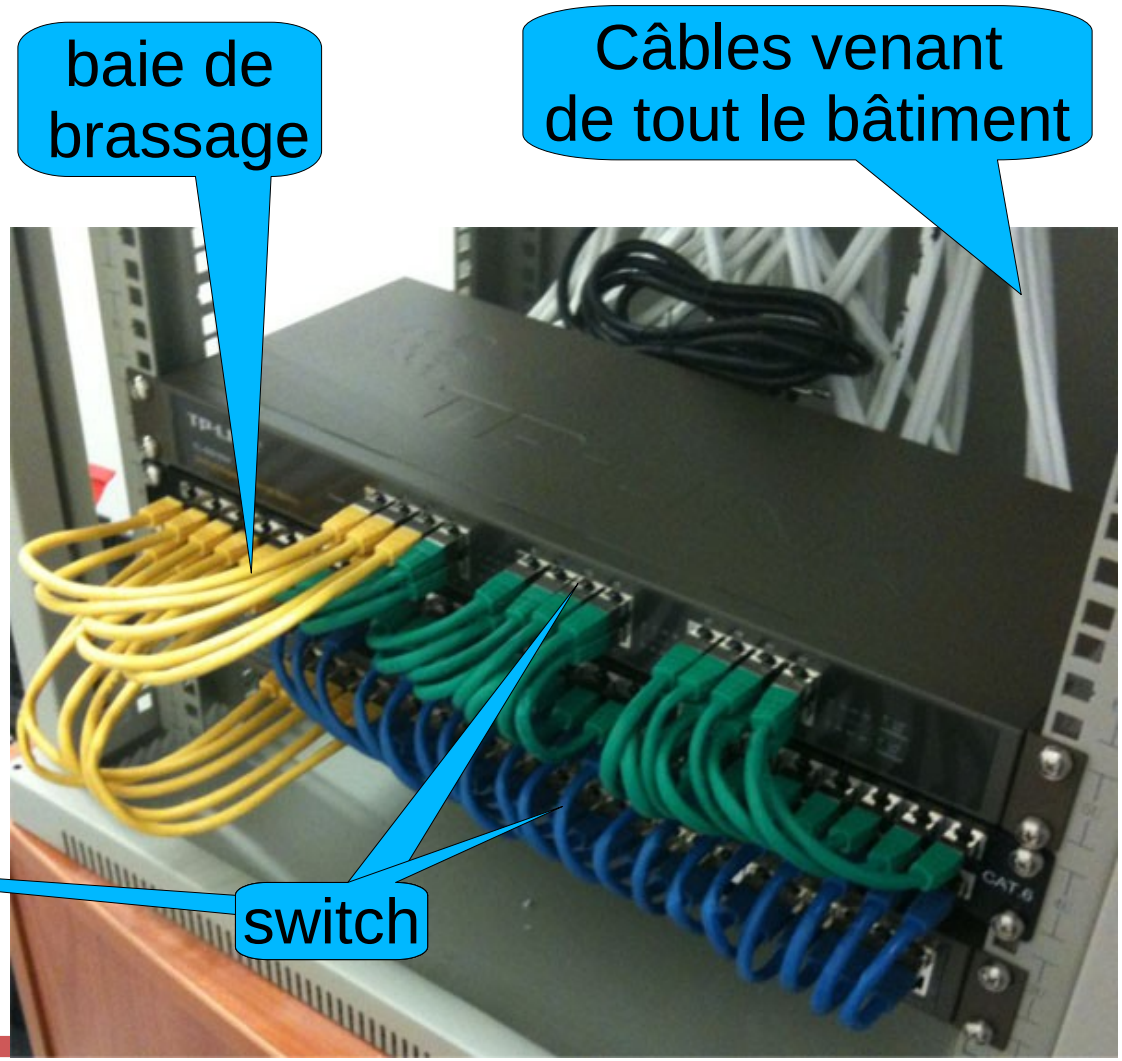
Topologie

Topologie en étoile

– Pannes faciles à détecter
– Administration centralisée
mais
– Nécessite souvent un pré-câblage

→ utilisé dans les réseaux tertiaires.





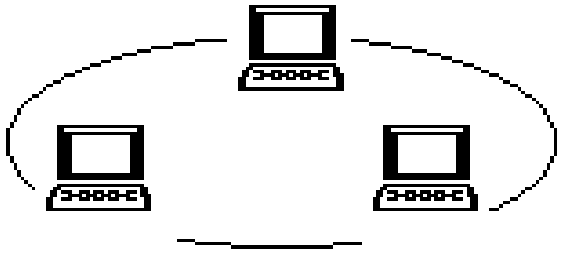
Topologie

Topologie en anneau

- une trame particulière circule sur le réseau toujours dans le même sens
- quand une machine veut émettre :
  - → attendre de recevoir le jeton ;
  - → émission d'une trame de données à la place du jeton ;
  - → le destinataire modifie la trame pour indiquer qu'il l'a bien reçue ;
  - → quand la trame a fait le tour, réémission du jeton.

régénération du signal électrique
protocole simple : pas de collision.

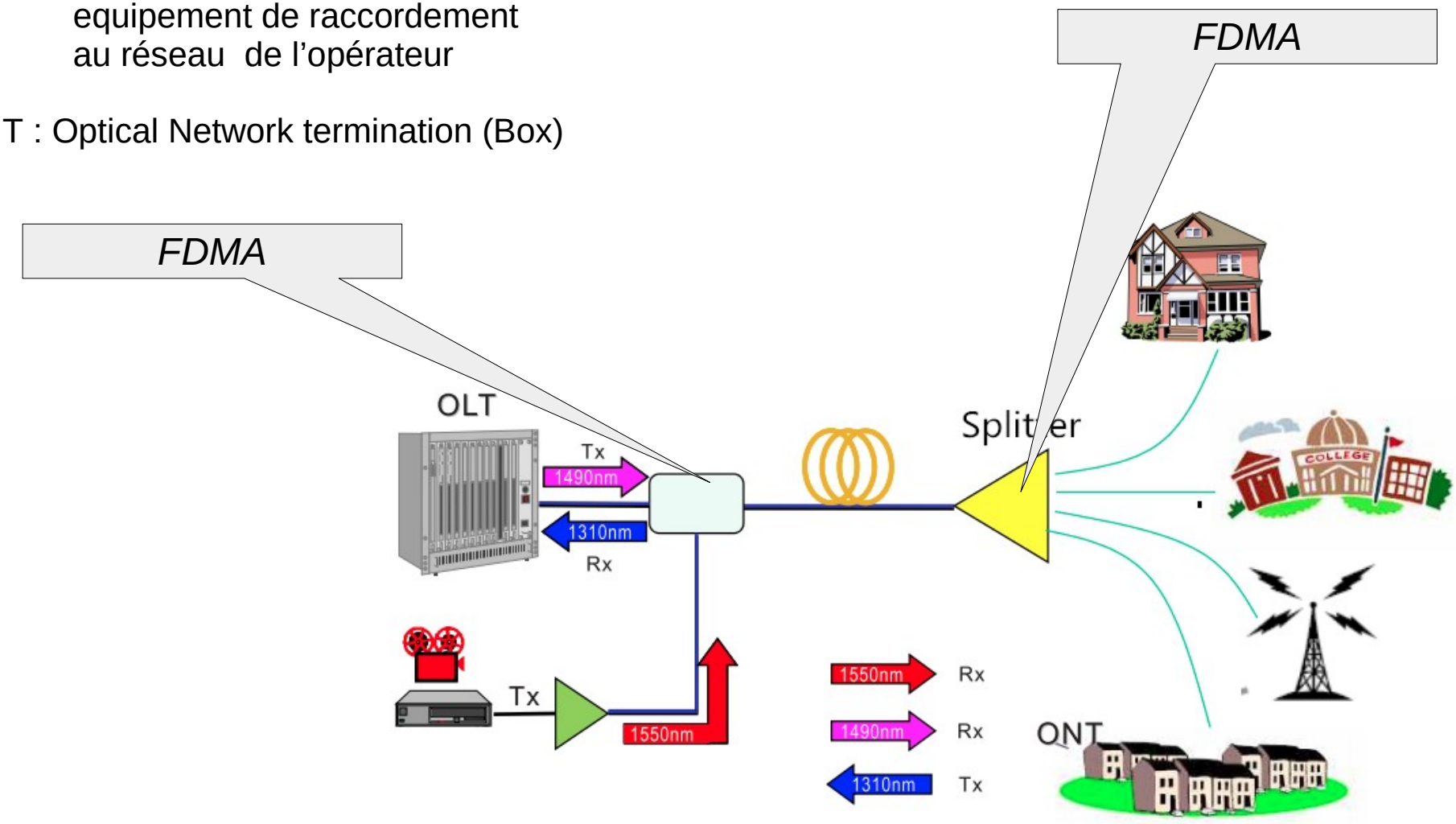
Mais
pb en cas de rupture d'un câble
câble 2 fois plus long qu'en topologie bus



Les fibres optiques

OLT : Optical Line Termination
equipement de raccordement
au réseau de l'opérateur

ONT : Optical Network termination (Box)



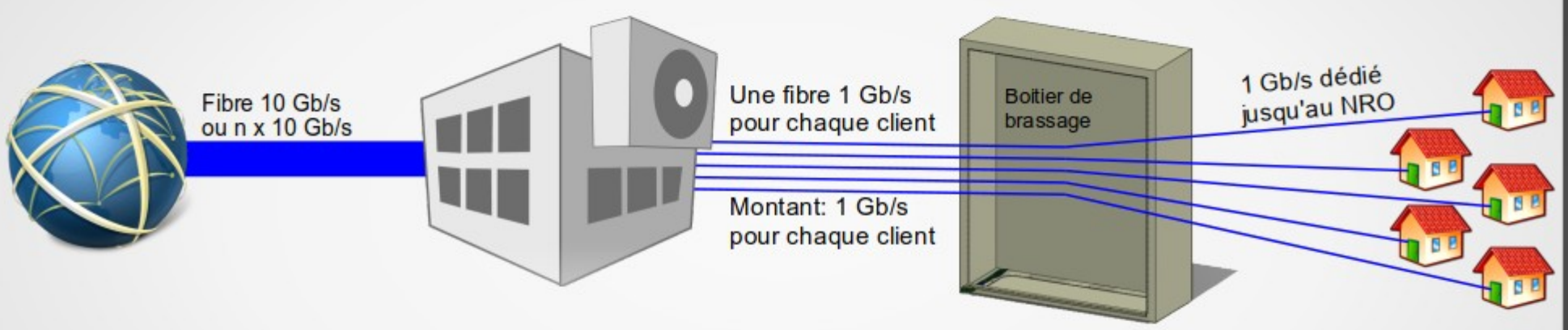
Aller voir https://www.youtube.com/watch?v=X1QJphPLhlM

Les fibres optiques

Technologie utilisé pour des offres grand publc (FTTH), professionnelles (FTTE) ou entreprise (FTTE / FTTO)

FTTH point à point

Utilisé en France par Free (Dans les 'zones très denses' où Free a commencé son déploiment avant 2015) pour des offres à 1Gb/s.



Le schéma est presque identique pour des offres entreprises FTTO (*Fiber To The Office* – Fibre jusqu'au bureau) : On a une fibre dédiée par entreprise, permettant des débits débits de 1 Gb/s, 10 Gb/s voir plus. Sur le terrain, les fibres utilisées par le FTTO sont distincts des fibres FTTH.

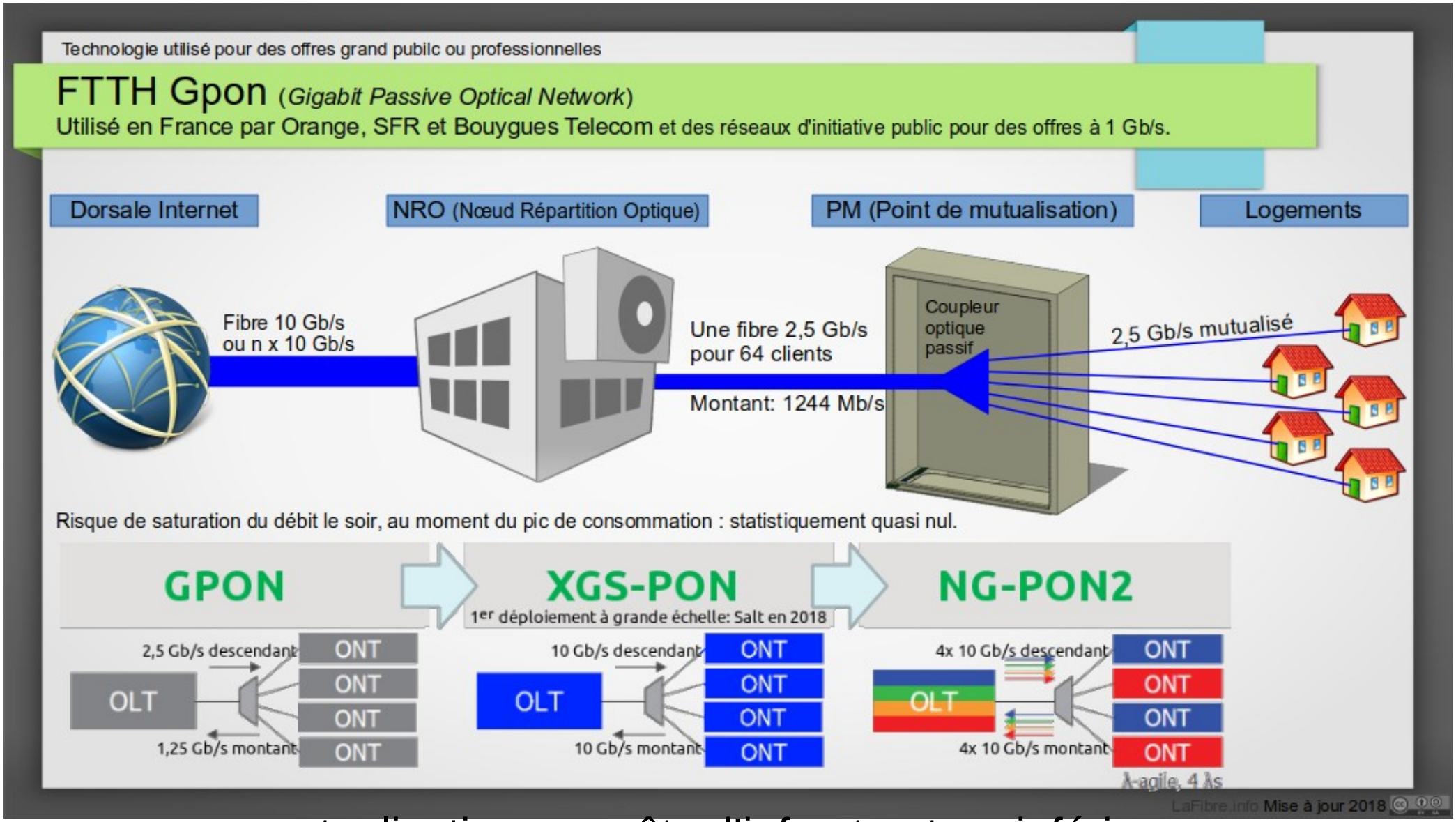
Le FTTE est un animal hybride qui devrait combler le vide technique et tarifaire entre le FTTH pro (entre 50 et 90 € HT par mois en moyenne selon les acteurs) et le FTTO (entrée de gamme nationale à près de 300/400 € HT par mois). Le FTTE offre une fibre dédiée tout en utilisant une partie des infrastrucutes mises en place pour le FTTH Gpon.

Risque de saturation du débit le soir, au moment du pic de consommation : nul, si le NRO a une collecte suffisante.

LaFibre.info Mise à jour 2018

source : lafibre.info

Les fibres optiques

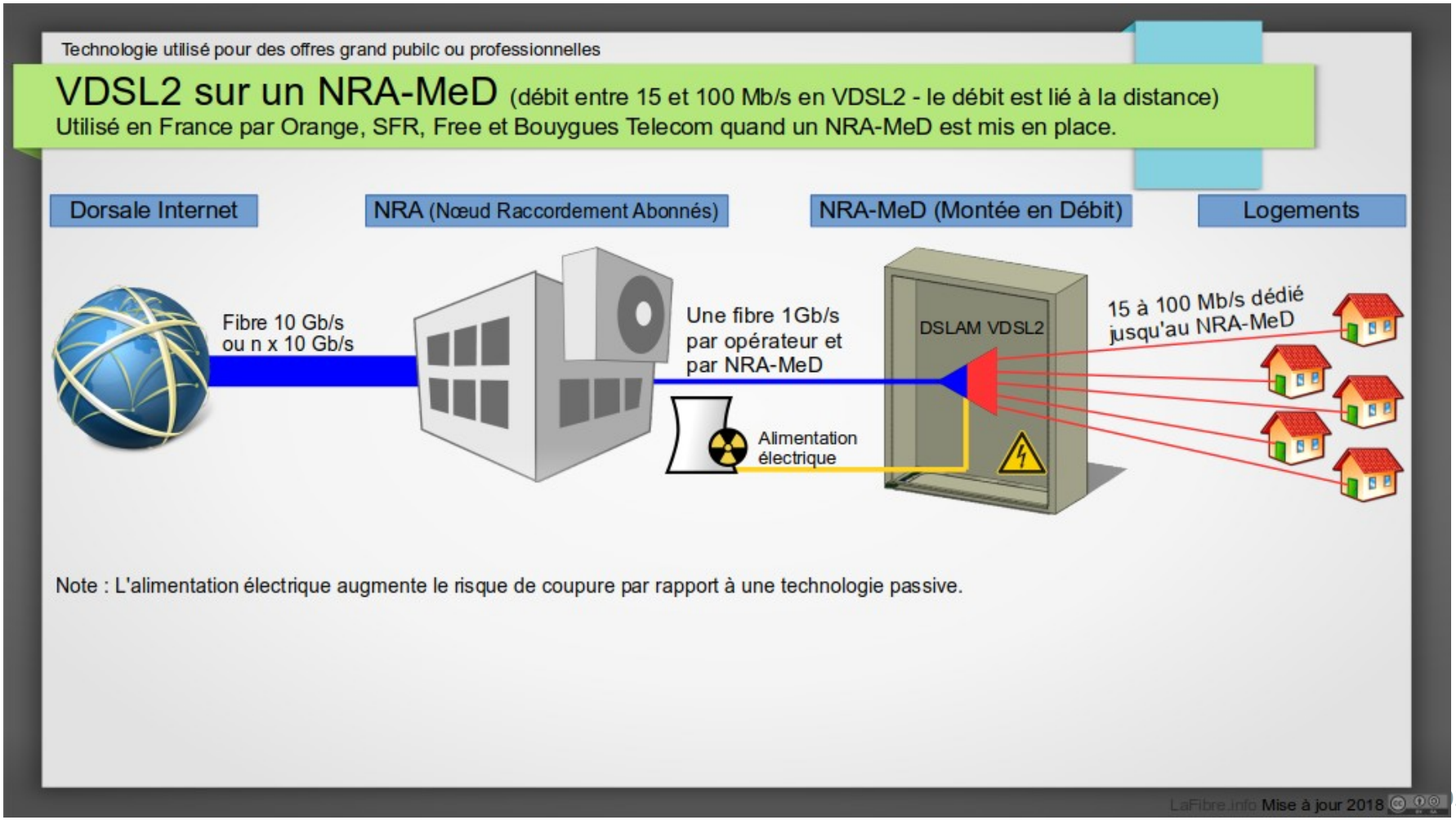


mutualisation → coûts d'infrastructure inférieurs

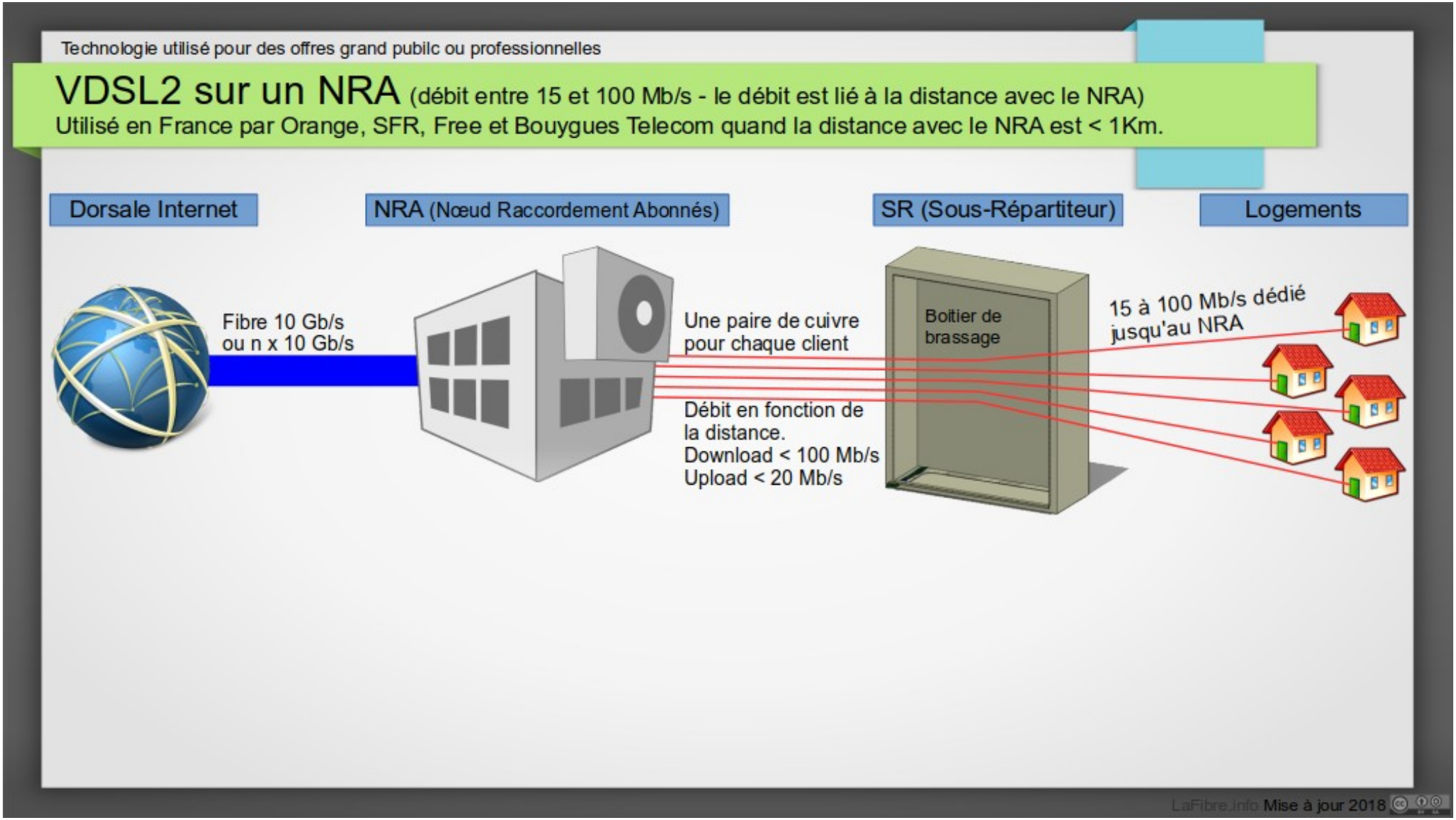
ADSL / VDSL

Asymmetric digital subscriber line / Very-high-bit-rate digital subscriber line

Utilise le « vieux réseau téléphonique ». Débits de qq Mbps à 100 Mbps



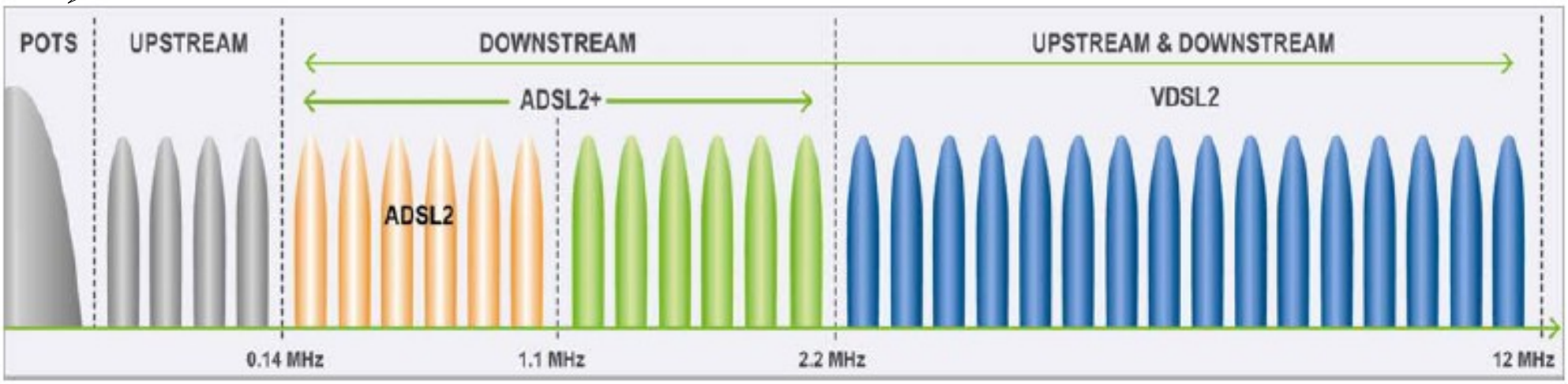
ADSL / VDSL



source : lafibre.info

ADSL / VDSL

Plain Old Telephone Service
< 8 kHz



wikipedia

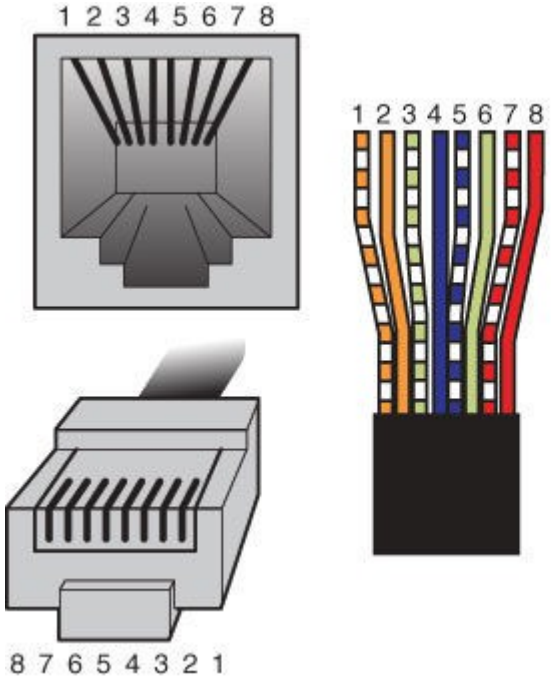
Remarque: Sur une ligne ADSL, l'affaiblissement du signal électrique est de 15 dB/km.
Donc si la bande passante est de 2MHz, 1km plus loin, le débit est réduit de 7 Mbps.

Ethernet

Introduction

- Inventé par Xérox dans les années 70 puis normalisé en 83.
- Norme IEEE 802.3
- Topologie logique : Bus à diffusion naturelle
- Topologie physique : Bus, mais aussi étoile.
- Débit : 10 Mbps théorique, mais plus faible dans la pratique à cause des collisions.
- Codage : 10 MHz, Manchester Différentiel.
- Évoltion vers les hauts débits avec Fast Ethernet et Gigabit Ethernet.

RJ45



Pin	Color	Pair
1	Orange/White	2
2	Orange	2
3	Green/White	3
4	Blue	1
5	Blue/White	1
6	Green	3
7	Brown/White	4
8	Brown	4

Câble droit RJ 45 (photo).
Interconnexion d'équipements de niveaux 3 avec un équipement de niveau 2. Ex : PC-switch

Les câbles RJ 45 croisés : Interconnexion d'équipements de niveaux 2 entre eux ou de niveau 3 entre eux. Ex : PC-PC

Exercice :
10GBaseT : 800 Mbauds, PAM-16 sur une paire, codeur de rendement 32/25.

Quel débit peut-on atteindre ?

RJ45

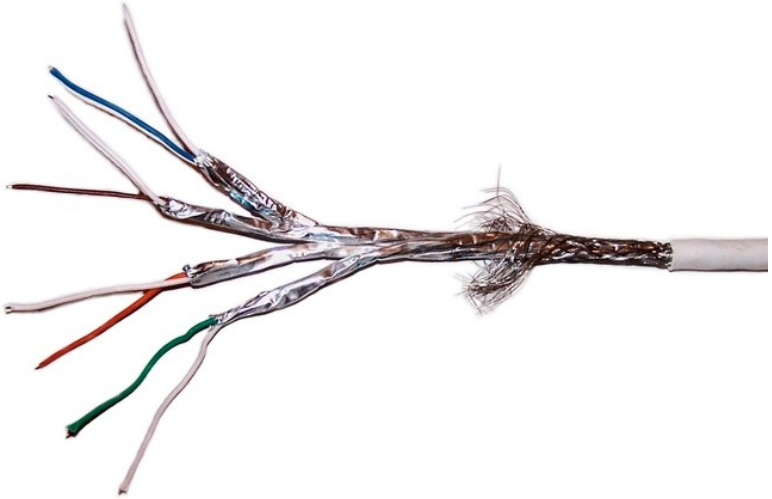
Paire torsadée

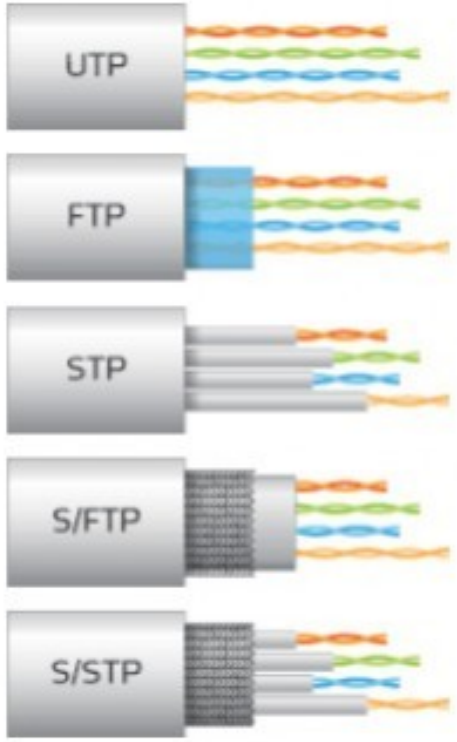
Pour limiter les interférences

Paire blindée

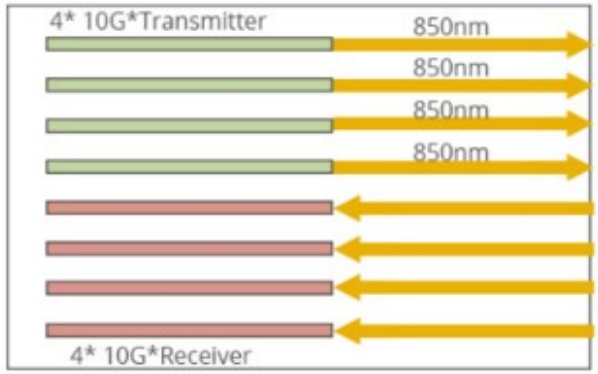
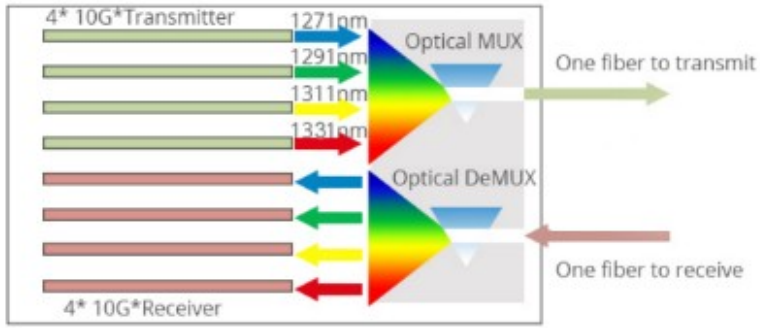
Pour renforcer la lute contre les interférences

Unshielded twisted pair (UTP)
Foiled twisted pair (FTP)
Shielded twisted pair (STP)





Ethernet optique

Physical Transceiver	Electrical and Optical Lanes Diagram	Description
		40GBASE-SR4 transceiver support with a link length up to 100 m on OM3 and 150 m on OM4 over 850nm multimode fiber, MPO Connector
		40GBASE-LR4 transceiver support with a link length up to 10 km over 1310 nm single mode fiber, LC Connector

Switch Ethernet

•Comme un hub mais le switch **ne répète la trame émise que sur les ports concernés**. Plusieurs machines peuvent parler en même temps.

•**Apprend la topologie** du réseau ;

•**Full duplex**, jusqu'à 10 Gbps ;

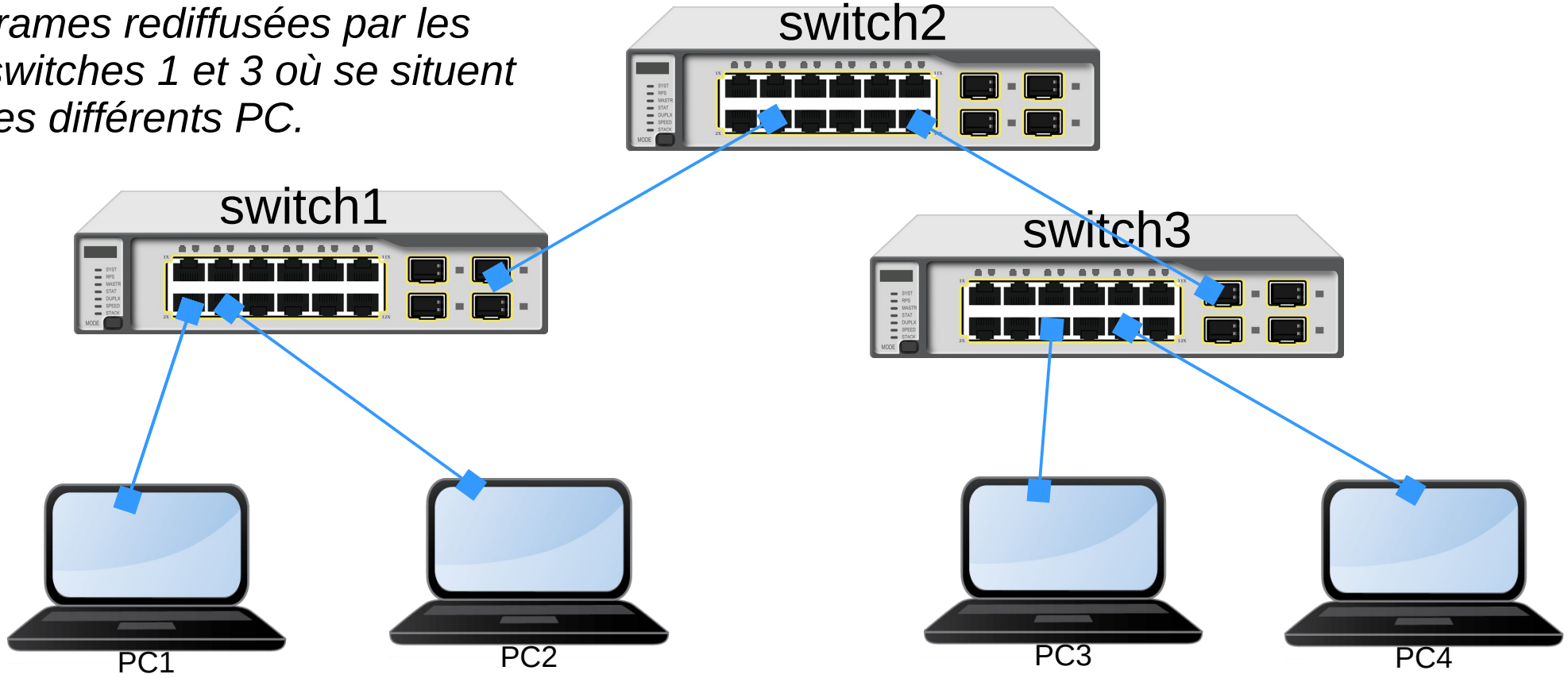
•Protocole STP (**Spanning Tree Protocol**) permettant d'éviter les boucles dans le réseau en cas de maillage de plusieurs switch.

- Il est transporté par ethernet + LLC.
- C'est un protocole permettant d'établir un arbre couvrant à coût minimum entre tous les noeuds du réseau.

Switch Ethernet

Auto apprentissage de la topologie du réseau par adresse MAC

Par les trames de broadcast (ARP par exemple), le switch 2 apprend par les trames rediffusées par les switches 1 et 3 où se situent les différents PC.

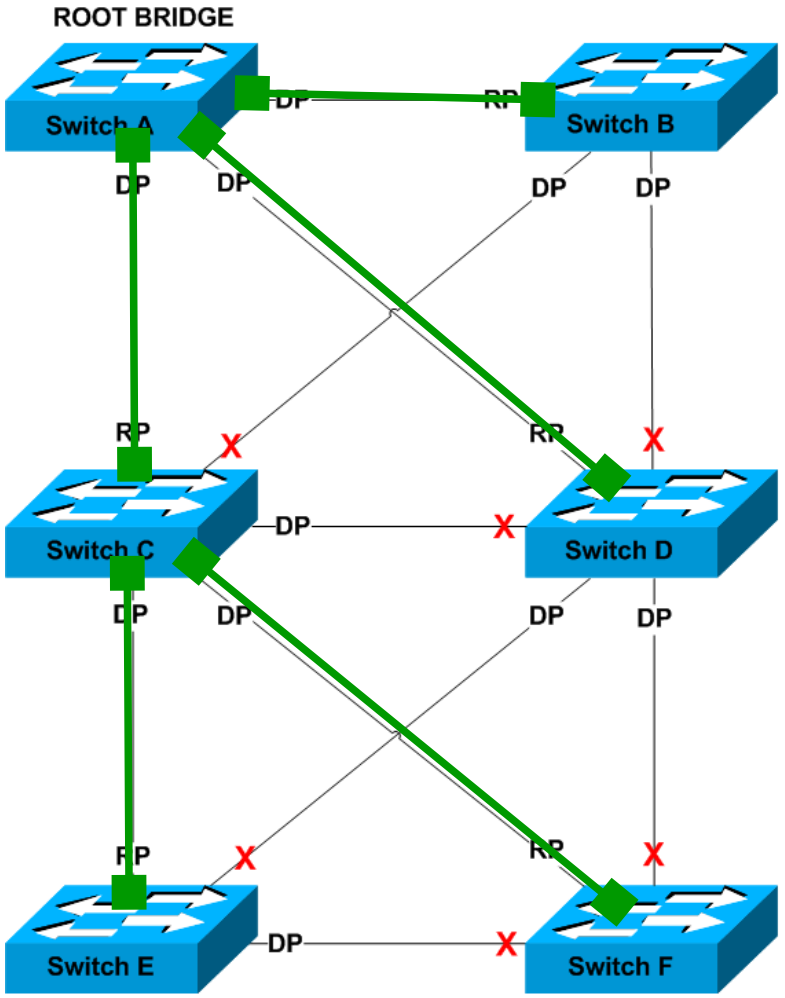


Spanning Tree Protocol

Pour augmenter la robustesse du réseau
→ liaisons redondantes
→ pb des trames vont être dupliquées
→ tempêtes de broadcast

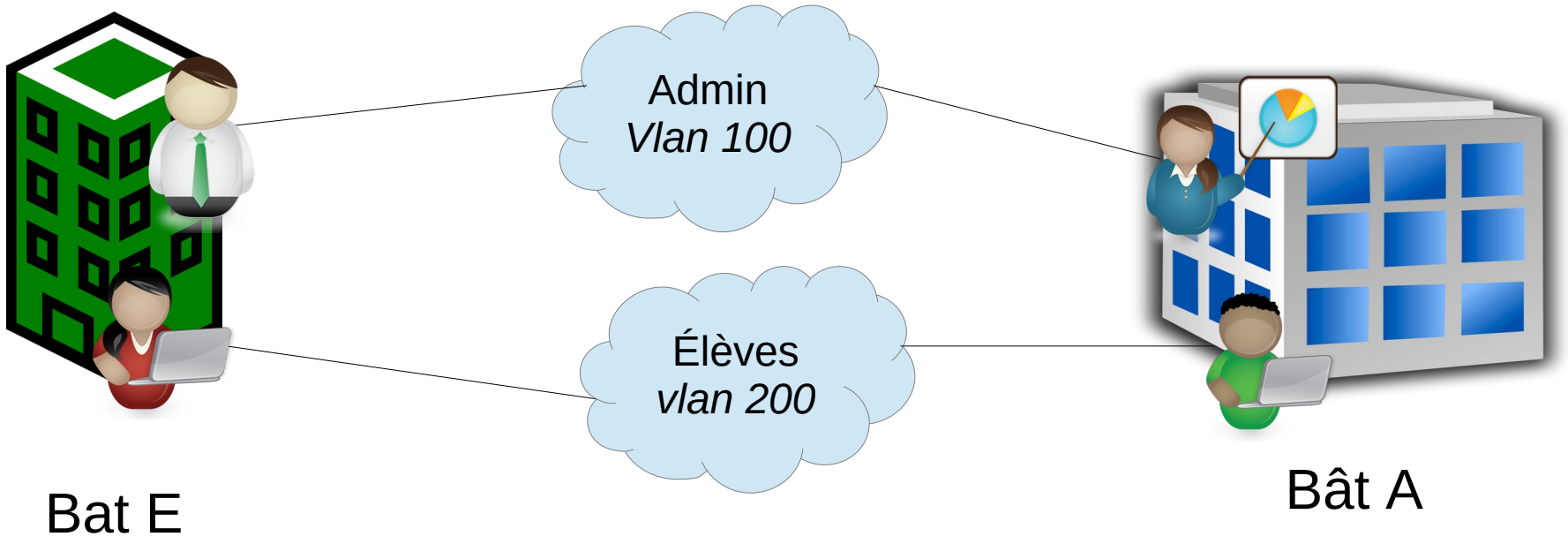
Solution établir un Arbre couvrant (en vert) à coût minimum sur le réseau.

Algorithme de spanning tree.

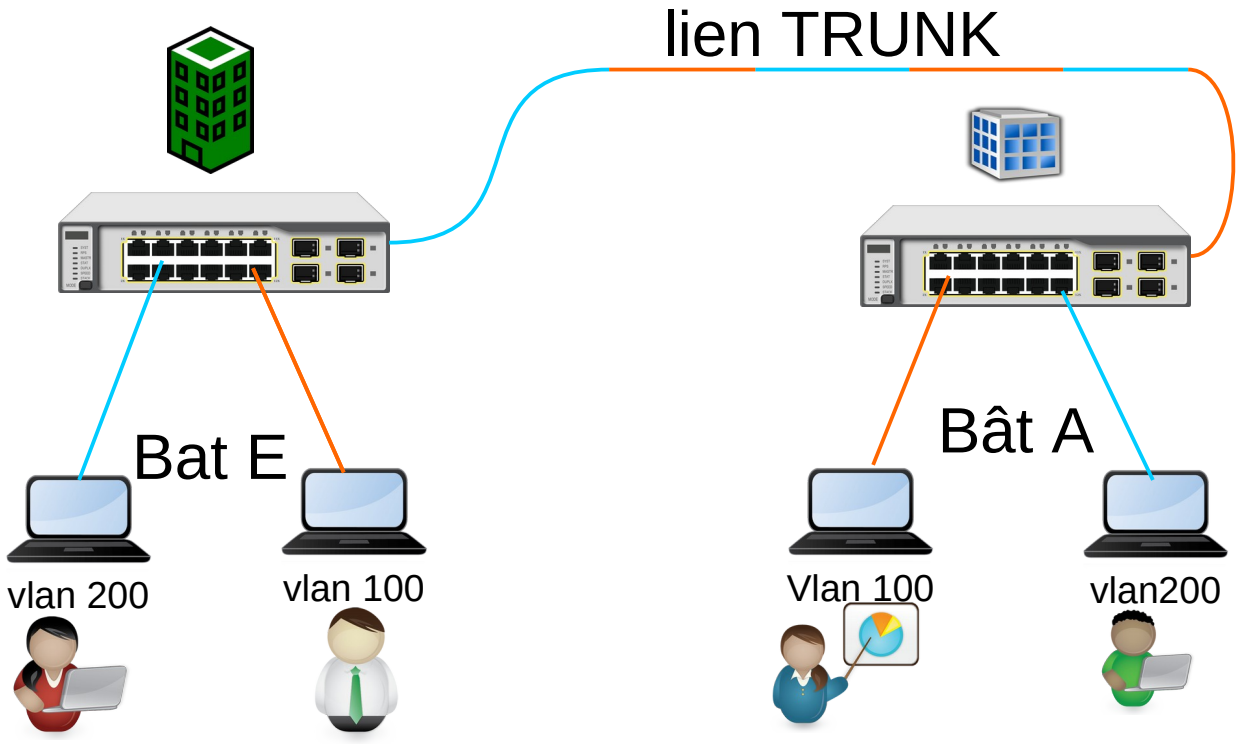


Virtual LAN

utiliser un même switch MAIS créer des réseaux locaux virtuels étanches



VLAN



WIFI ou norme IEEE 802.11

- Propagation du signal sans fil.
  Si rayonnement isotrope :

  $$Puissance\ reçue = S \,.\, Pe \,/\, (4.\pi.d^2)$$

  S : surface équivalente de l'antenne
  Pe : puissance émise
  d : distance émetteur - récepteur

- Réglementation sur la puissance émise
  - Dépend des pays
  - Dépend des différentes normes Wifi
  - En France, voir : http://www.arcep.fr/index.php?id=9272
  - Pour la bande autour des 2,4 GHz :
    Puissance Isotrope Rayonnée Equivalente < 100 mW
    en intérieur et en extérieur.

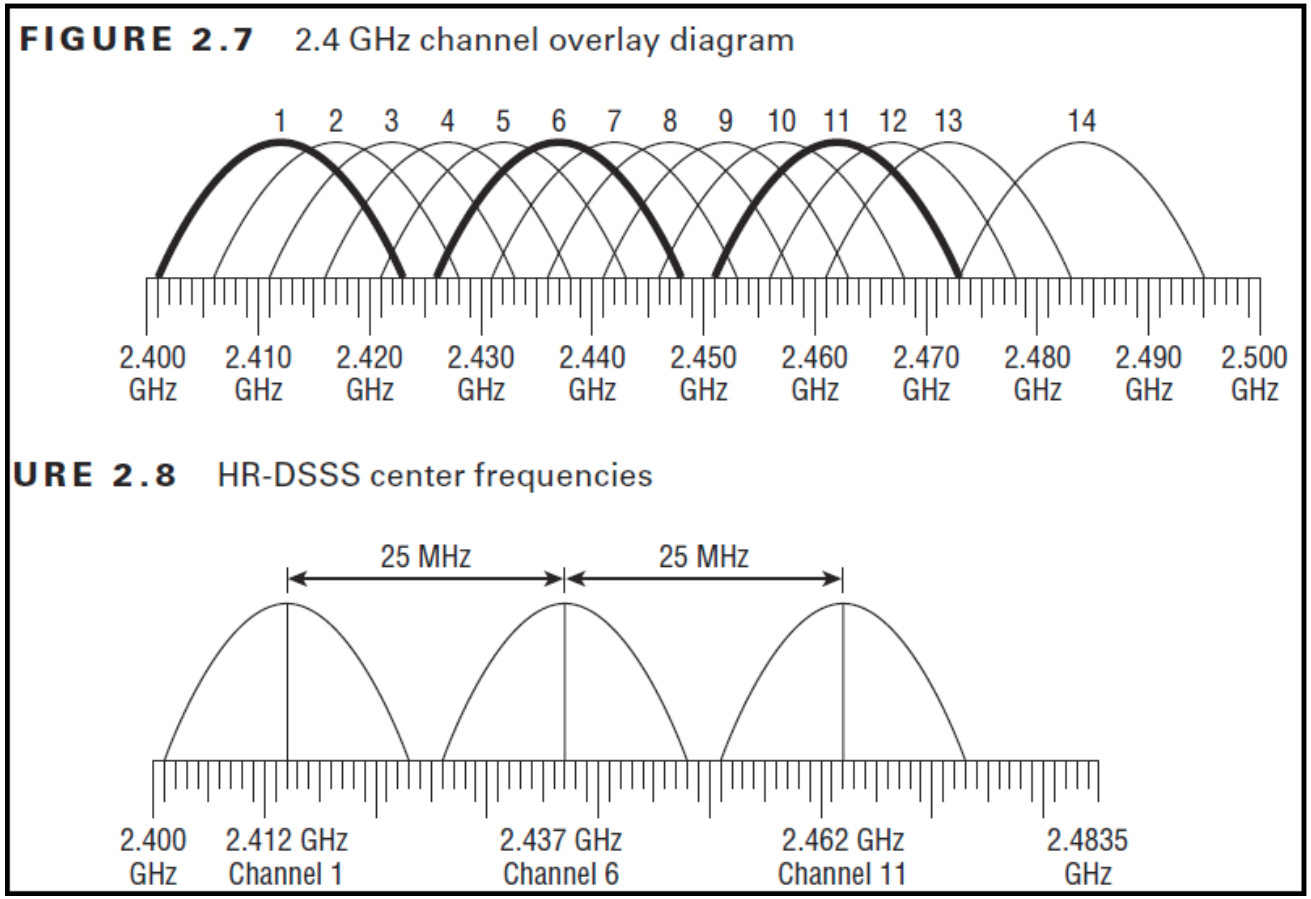
WIFI ou norme IEEE 802.11

Amendement	Principales caractéristiques
802.11a	Bande des 5 GHz Débits max 54 Mbps
802.11b	Bande des 2,4 GHz Débits max 11 Mbps
802.11g	Bande des 2,4 GHz Débits max 54 Mbps
802.11e	Qualité de service
802.11i	Sécurité
802.11n	Bande des 2.4 et 5 GHz Débit typique = 144 Mbps ; Débit max normalisé = 600 Mbps
802.11ac	Bande des 5 GHz Débit typique = 433 Mbps ; Débit max normalisé = 3,466 Gbps

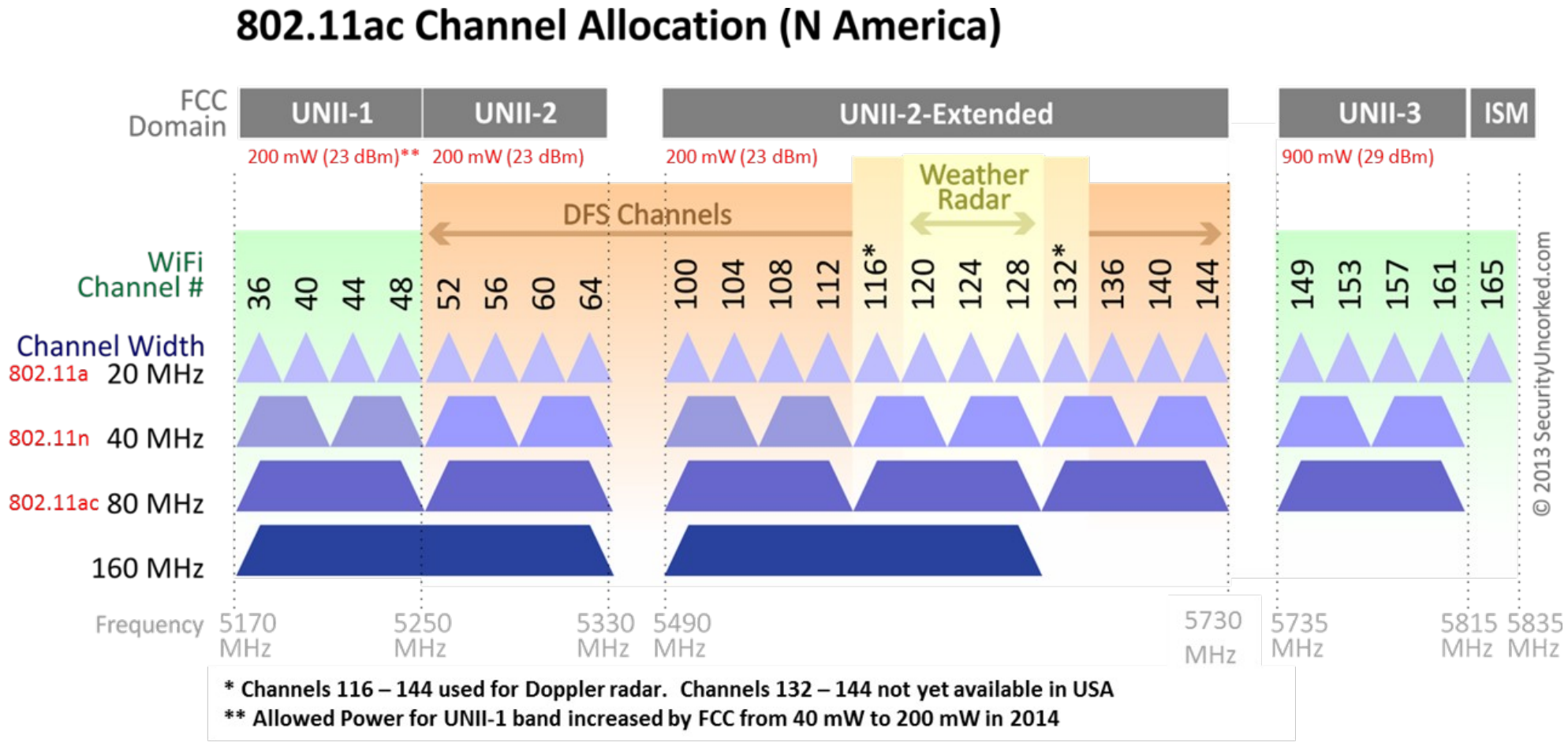
WIFI / 802.11b

Fréquences : 14 canaux de 5 MHz sont disponibles dans la bande des 2400-24835 MHz.

En pratique seuls 3 canaux sont utilisés : 1, 6 et 11, car un canal a besoin d'une bande passante de 25 MHz pour fonctionner. Ceci évite les chevauchements.



WIFI 802.11n 802.11 ac



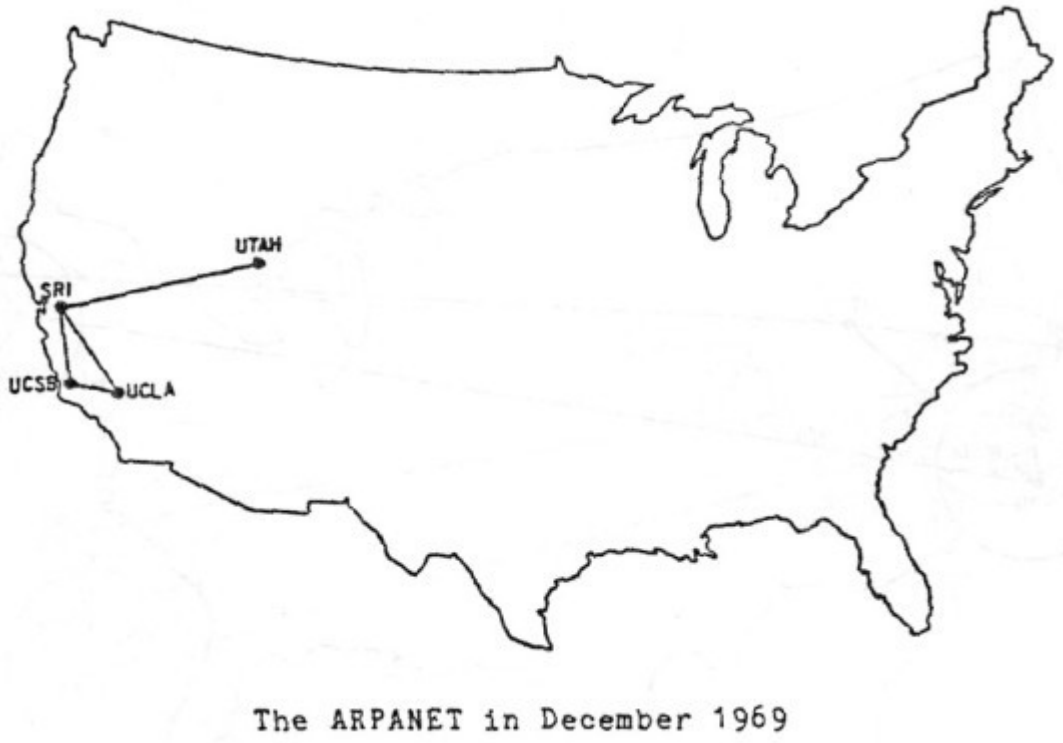
Réseaux de communication

Couches Réseau

L'École des INGÉNIEURS Scientifiques

Normandie Université

Internet Protocol



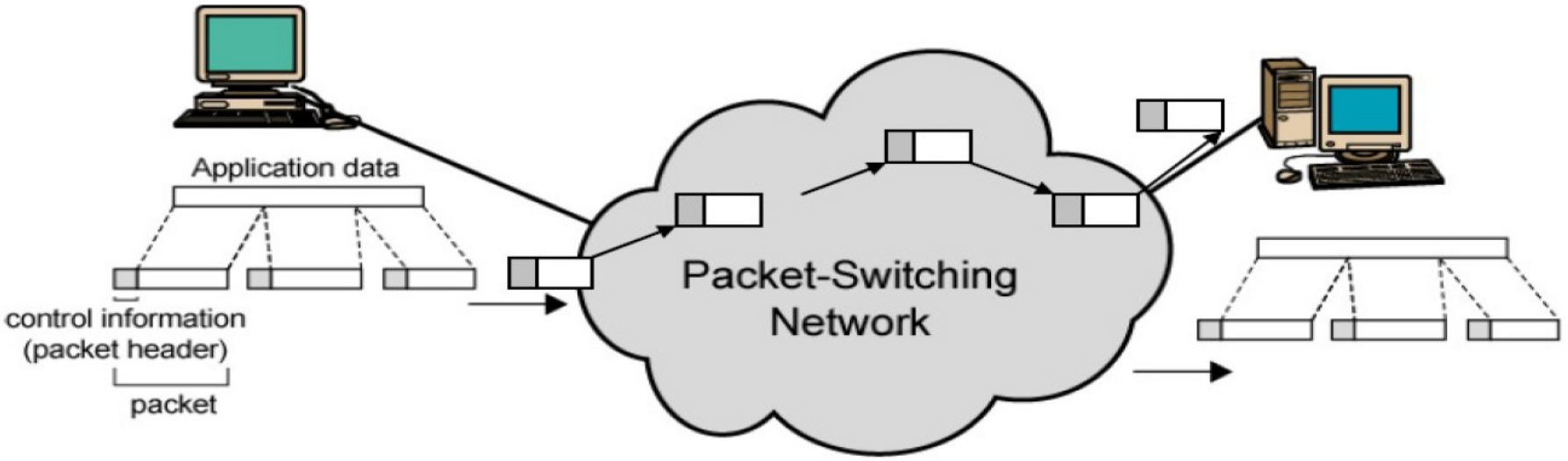
The ARPANET in December 1969

naissance du projet ARPANET
initié par le Department Of Defence aux USA

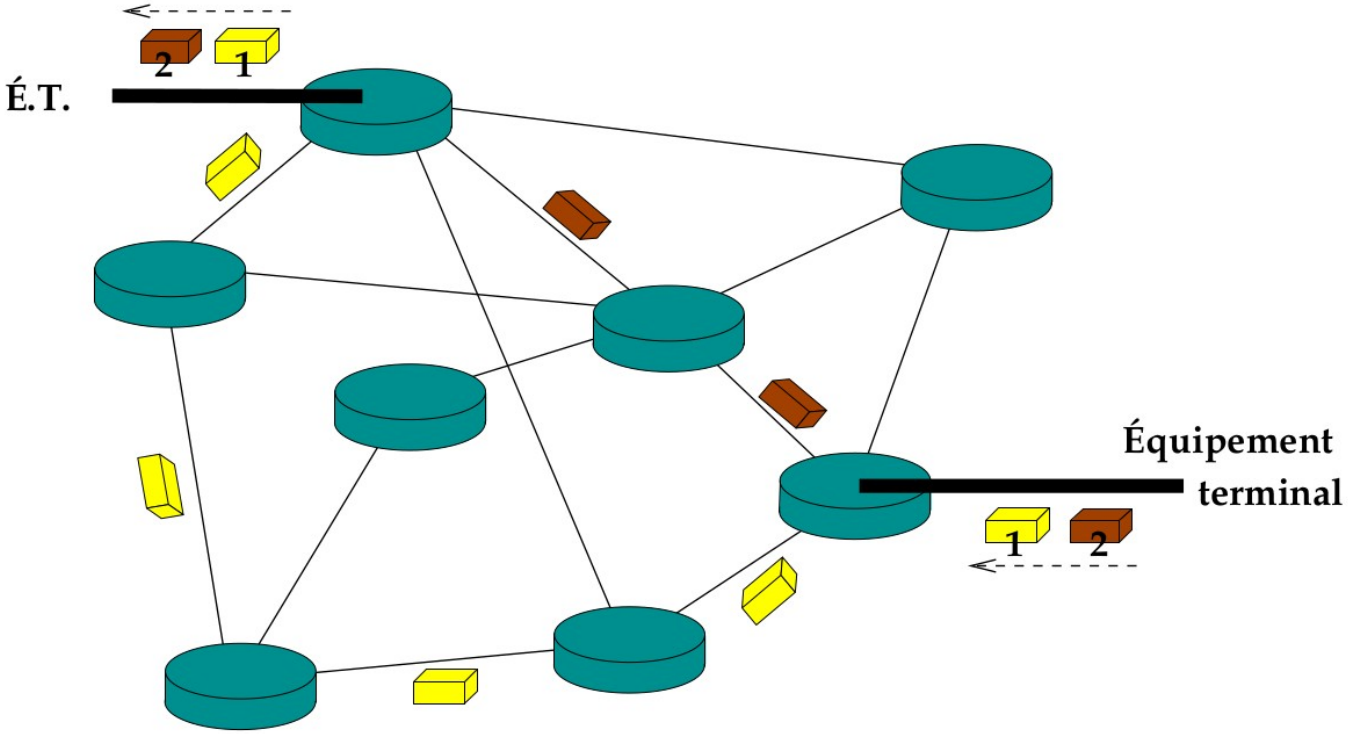
Internet Protocol

Comment acheminer des paquets d'un point à l'autre en traversant de multiples réseaux locaux ?

Une idée simple : des paquets (datagrammes) autonomes !



Ce qu’est Internet Protocol



Commutation de paquets

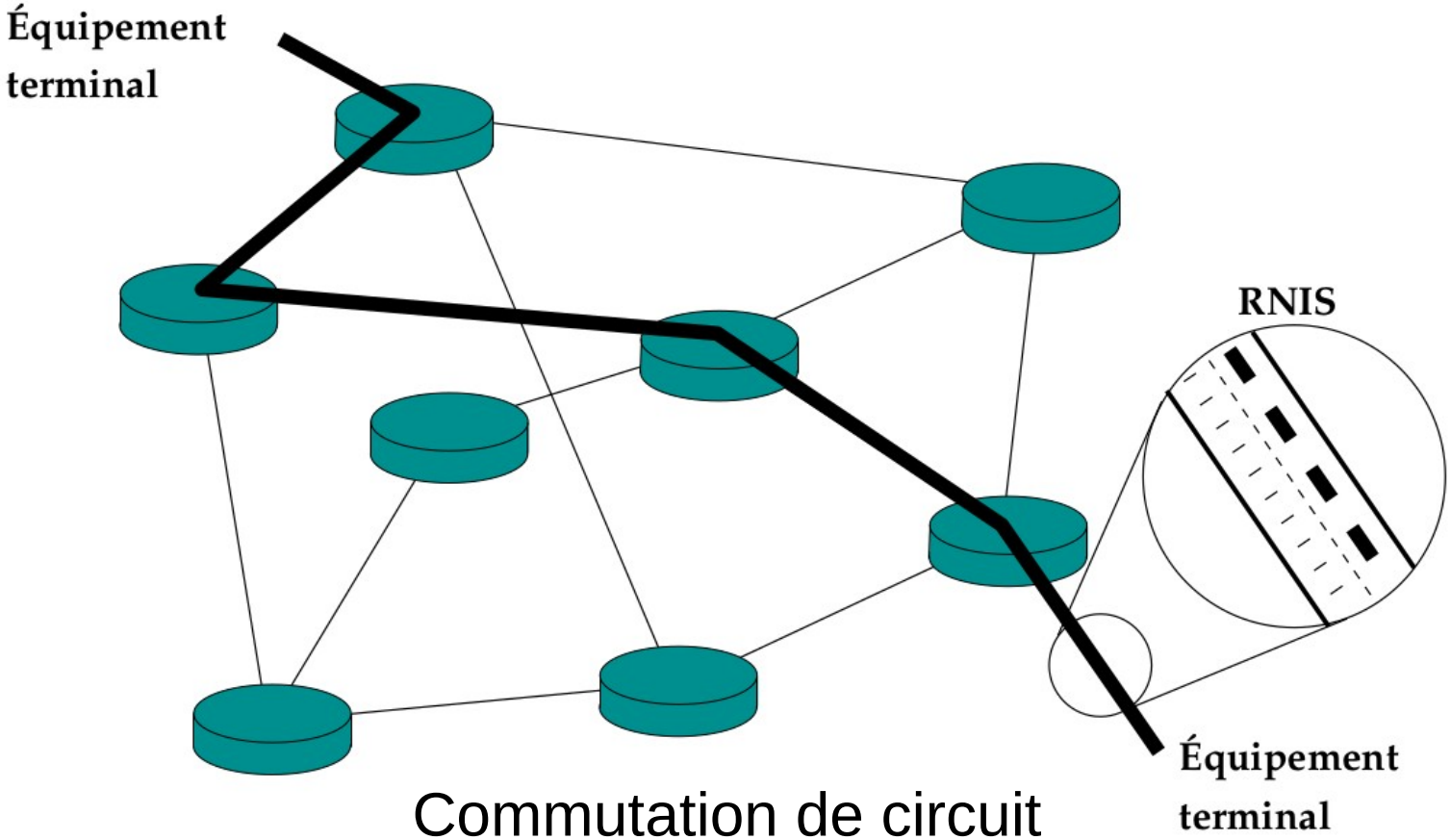
utilisé dans IP et dans feu X25

Chaque paquet est idépendant : @ du destinataire et de l’émetteur dans chaque paquet

Le réseau peut être saturé (sur-booking)

Les chemins pris par les paquets peuvent être différents

ce que n’est pas Internet :



Commutation de circuit

utilisée dans la téléphonie

RNIS = Réseau Numérique à Intégration de Services

wagons de 8bits qui circulent même s’il n’y a pas de données à envoyer !

IP – 3 types de datagramme

- les **datagrammes de données** : transportent l'information utilisateur ;
- les **datagrammes de contrôle** des données : contrôle de flux, acquittement, reprise en cas d'erreurs... ;
- les **datagrammes de supervision** du réseau : gestion du routage, maintenance afin de prévenir la congestion...

IP - Quelles informations dans la trame ?

Quelles informations dans les entêtes des datagrammes:

2 version d'IP : version 4 ou version 6

2 versions **incompatibles** avec des entêtes différentes.

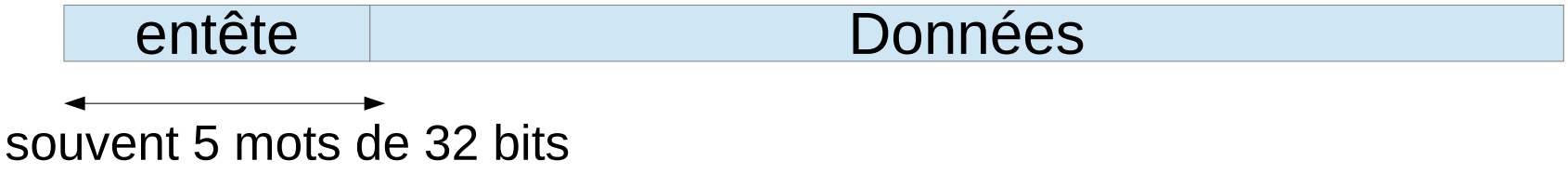
Par exemple :

- Version 4 : adresse sur 32 bits
- Version 6 : adresse sur 128 bits

Le premier champ du datagramme doit permettre de savoir de quelle **version** il s'agit.

IP - Quelles informations dans la trame ?

- Tous les champs ne sont pas de l'entête ne sont pas utiles ;
- Certains sont optionnels ;
- Il faut donc préciser la **longueur de l'entête** pour savoir où commencent les données.
- La longueur est donnée en nombre de mots de 32 bits.



adressage IPv4

- **Une adresse IP** = un numéro sur 32 bits, noté par des octets en base 10 séparés par des point. Ex : 10.7.0.254

- **Un réseau IP** = un ensemble de machines sur un réseau local

- **Un réseau IP** = un espace d'adressage/numérotation **contigu** des machines

  Ex de 192.168.1.0 à 192.168.1.254

adressage IPv4

- **Une adresse IP** = un numéro sur 32 bits, noté par des octets en base 10 séparés par des point. Ex : 10.7.0.254

- **Un réseau IP** = un ensemble de machines sur un réseau local

- **Un réseau IP** = un espace d'adressage/numérotation **contigu** des machines

  Ex de 192.168.1.0 à 192.168.1.254

adressage IPv4

Des machines d'un même réseau ont les **premiers bits de leur adresse identiques.**

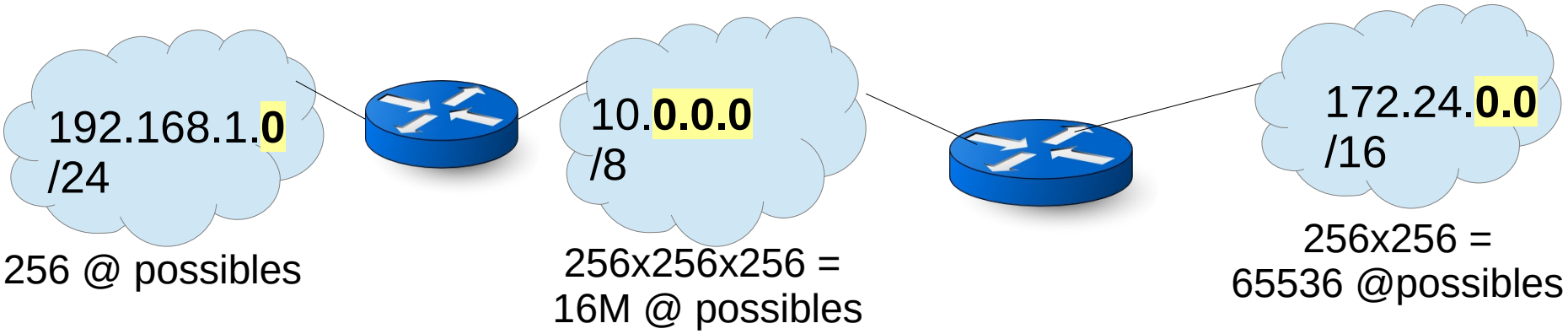
- adresse sur 32 bits
- Limite donnée par netmask
- notation du netmask soit par un « slash » soit en décimal pointé.

Exemple /24 == 255.255.255.0

	Net ID	Host ID
adresse IP :	$n_0n_1n_2n_3$……..	….$n_{31}n_{32}$
netmask :	111….11111	00….000

32 bits

IP - Adressage



Exemple avec 3 réseaux locaux utilisant des adresses en /24, /16 et /8

adressage IPv4

- Dans l'espace d'adressage du réseau :
  - 1ère adresse réservée au « nom du réseau »
  - dernière adresse réservée au broadcast

- Exemple : A quel réseau appartient la machine 3.4.5.6/23 ?

adressage IPv4

Exemple : A quel réseau fait partie la machine 3.4.5.6/23 ?

3.4.5.6 == 0000 0011 . 0000 0100 . 0000 0101 . 0000 0110

/23 == 1111 1111 . 1111 1111 . 1111 1110 . 0000 0000

DONC

@reseau == 0000 0011 . 0000 0100 . 0000 0100 . 0000 0000

@reseau == 3.4.4.0

Nombre de machines dans ce réseau : $2^{(32-23)}$ -2 = 510

@ de broadcast dans ce réseau : 3.4.5.255

adressage IPv4

Des adresses réservées :

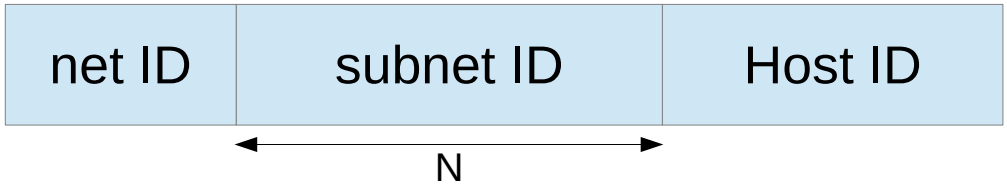
– **127.0.0.0 /8** : communication interne à une machine (entre programmes).
127.0.0.1 est en général utilisé

– **10.0.0.0 /8**
172.16.0.0 /12
192.168.0.0 /16
adresses réservées à un usage interne. Les machines avec ce type d'adresse doivent passer par une passerelle de type proxy ou NAT pour avoir accès à internet.

– **224.0.0.0 /24** : adresses de multicast réservées à la communication de groupe.

Sous réseaux

Permet de segmenter un réseau en plusieurs réseaux locaux



Quelques bits du HostID vont servir à identifier les sous-réseaux.
Ces quelques bits seront le subnetID

Exemple : réseau initial 10.20.0.0 /16 que l'on voudrait segmenter en 3 sous réseaux.
Donc 2 bits sont nécessaires pour identifier ces 3 sous-réseaux.

Un choix possible est : 2 sous-réseaux /18 et un sous-réseau de /17

```
10.20.0.0/18    == 0000 1010 . 0001 0100 . 0000 0000 . 0000 0000
10.20.64.0/18   == 0000 1010 . 0001 0100 . 0100 0000 . 0000 0000
10.20.128.0/17  == 0000 1010 . 0001 0100 . 1000 0000 . 0000 0000
```

Classes de réseaux

Créé initialement pour distribuer les adresses IP

A : 1.0.0.0/8 → 126.0.0.0 /8

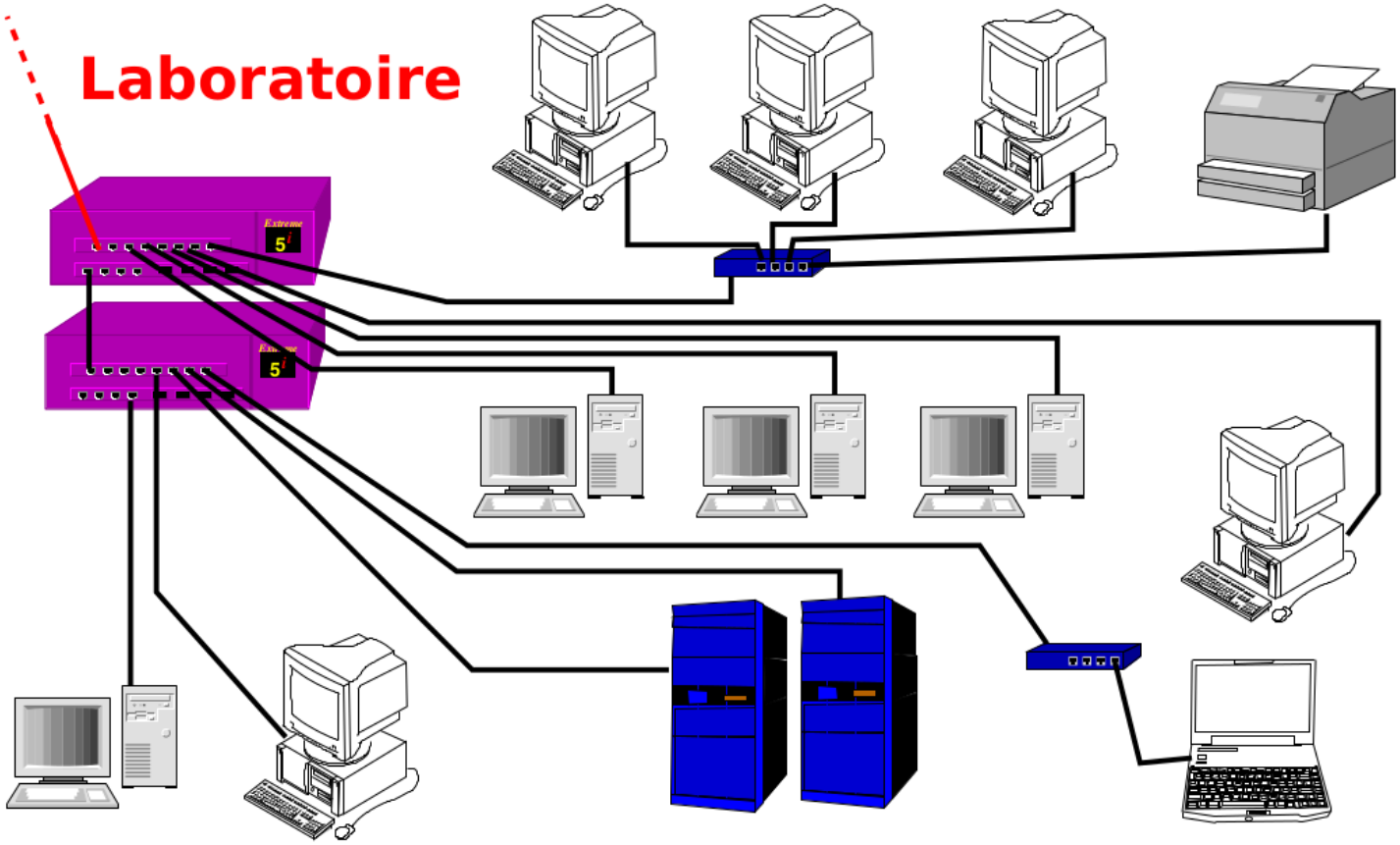
B : 128.0.0.0/16 → 191.0.0.0/16

C : 192.0.0.0/24 → 223.0.0.0/24

D : 224.0.0.0/32 → 239.0.0.0/8 : multicast

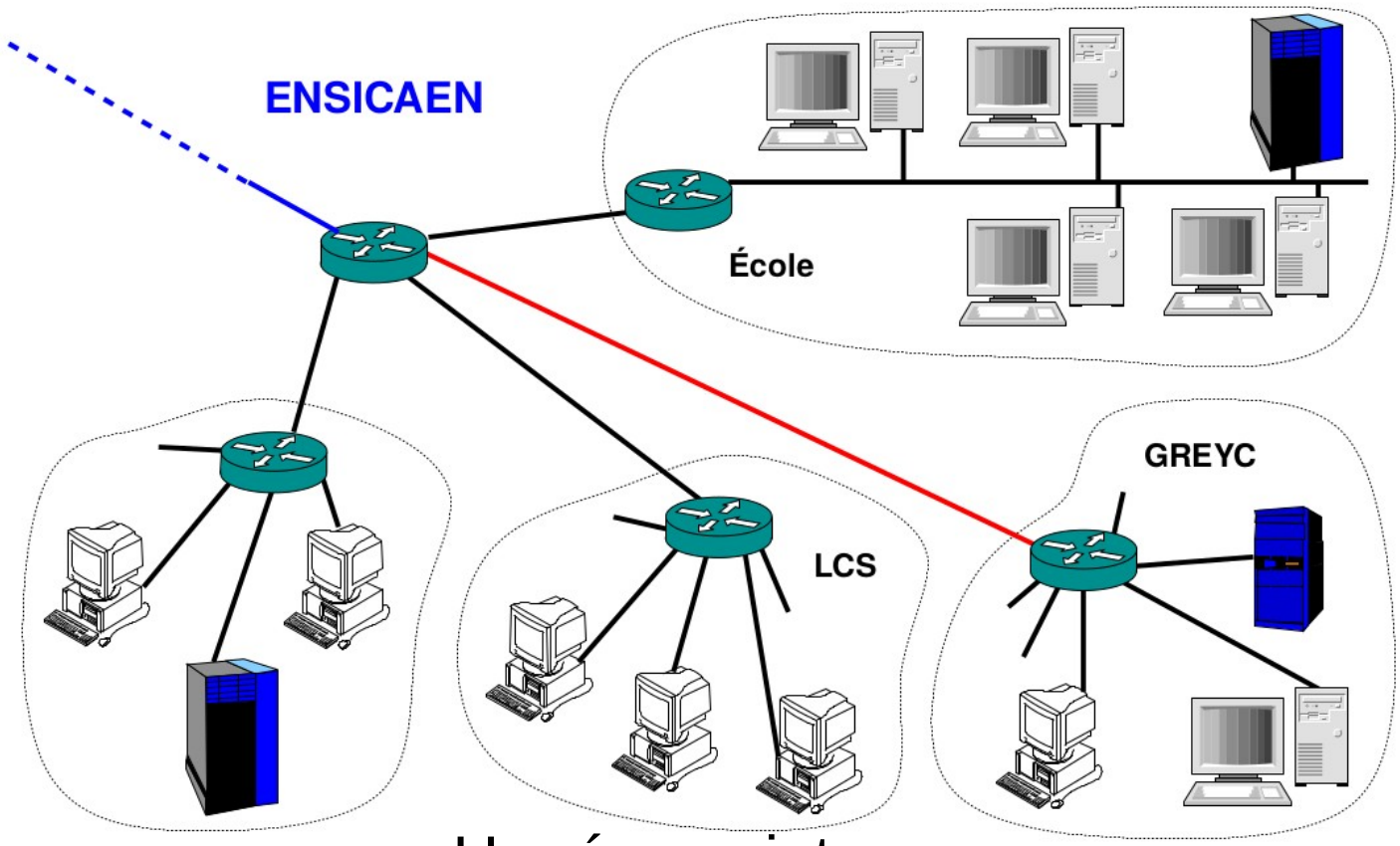
E : 240.0.0.0 → 255.255.255.255 : usages expérimentaux

Routage IP



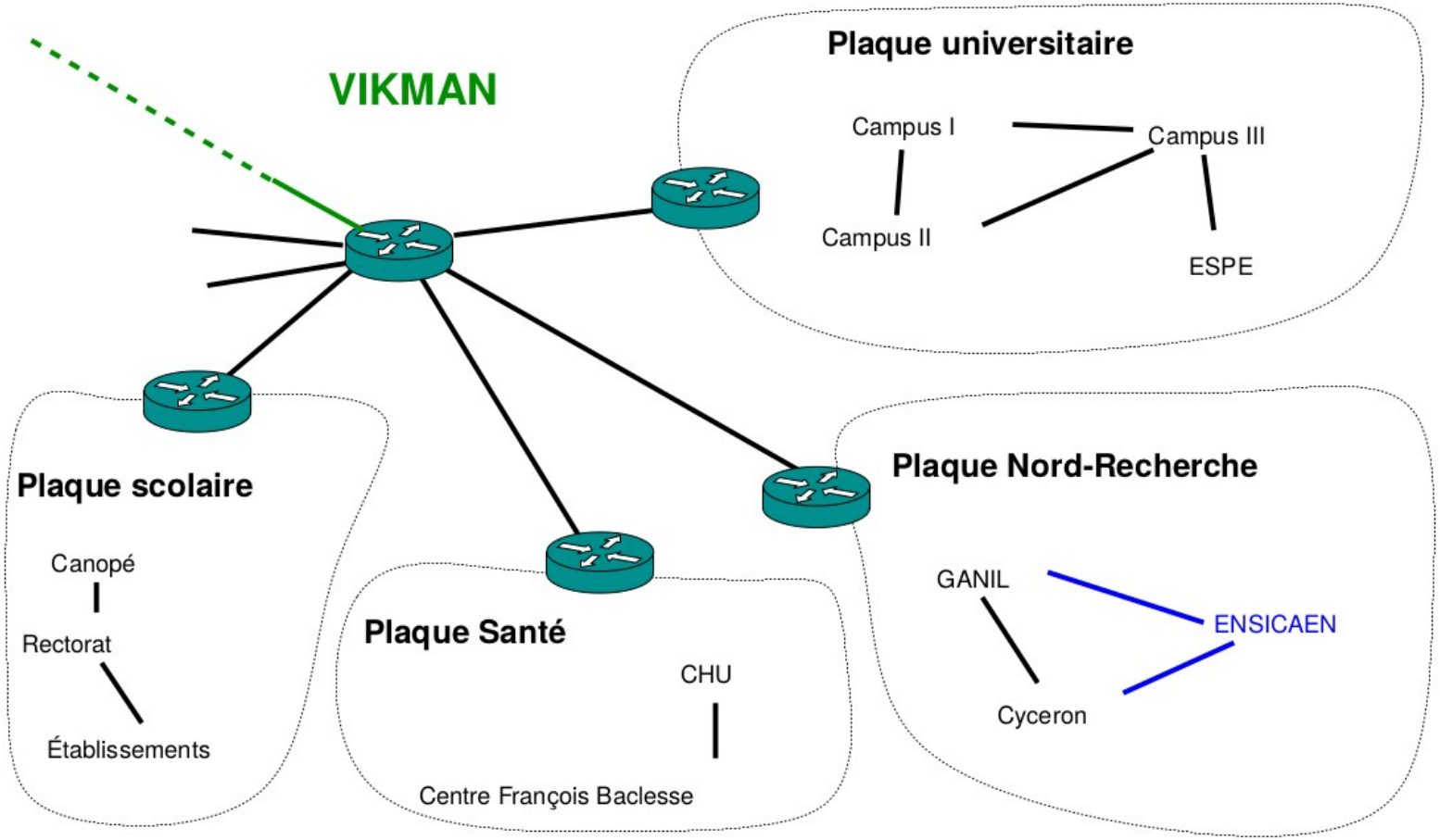
Un réseau local : pas besoin de routage.
diffusion naturelle (ou aidée par un commutateur)
sur tout le réseau local

Routage IP



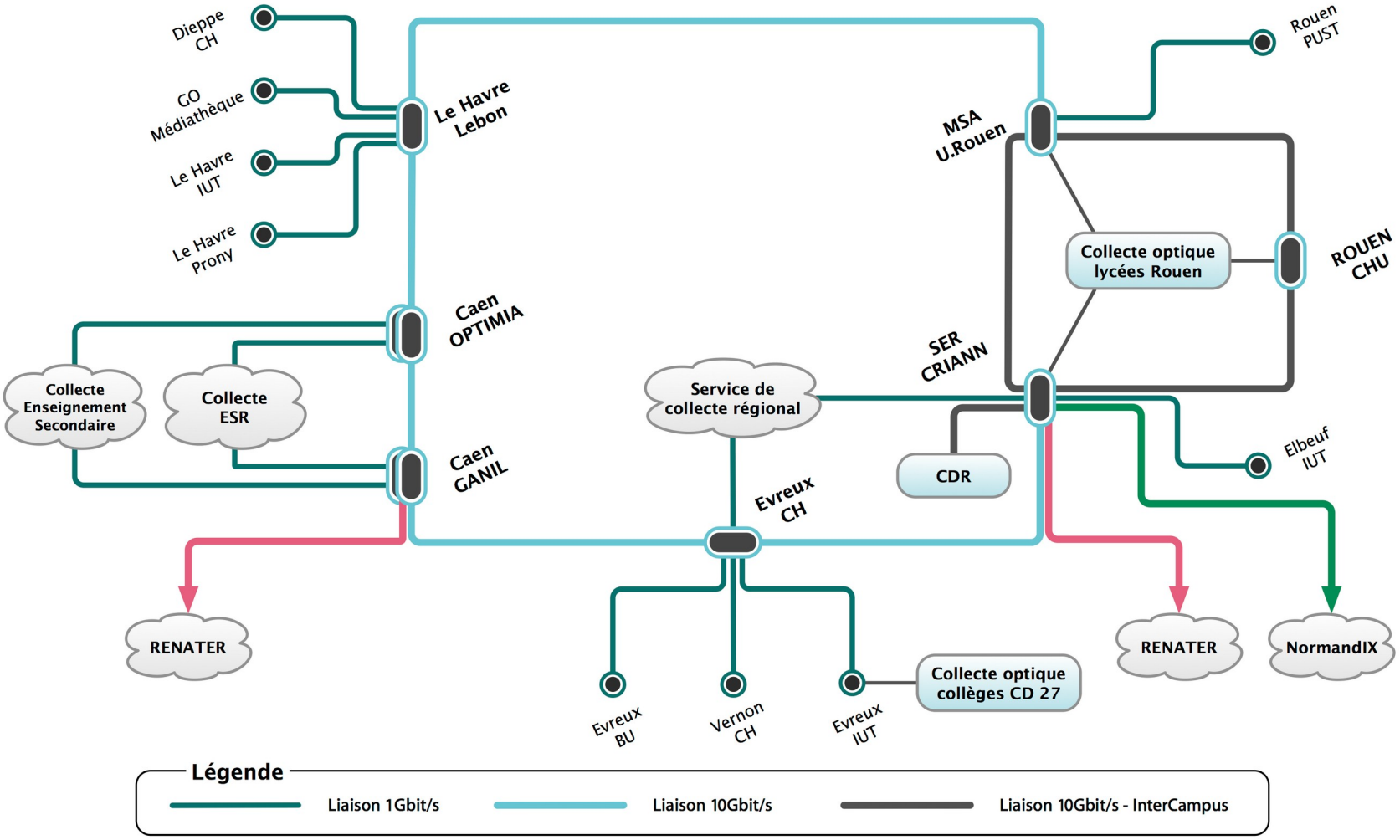
Un réseau interne
Besoin de routage entre les services
Mais l'extérieur n'a pas besoin de connaître les détails

Routage IP



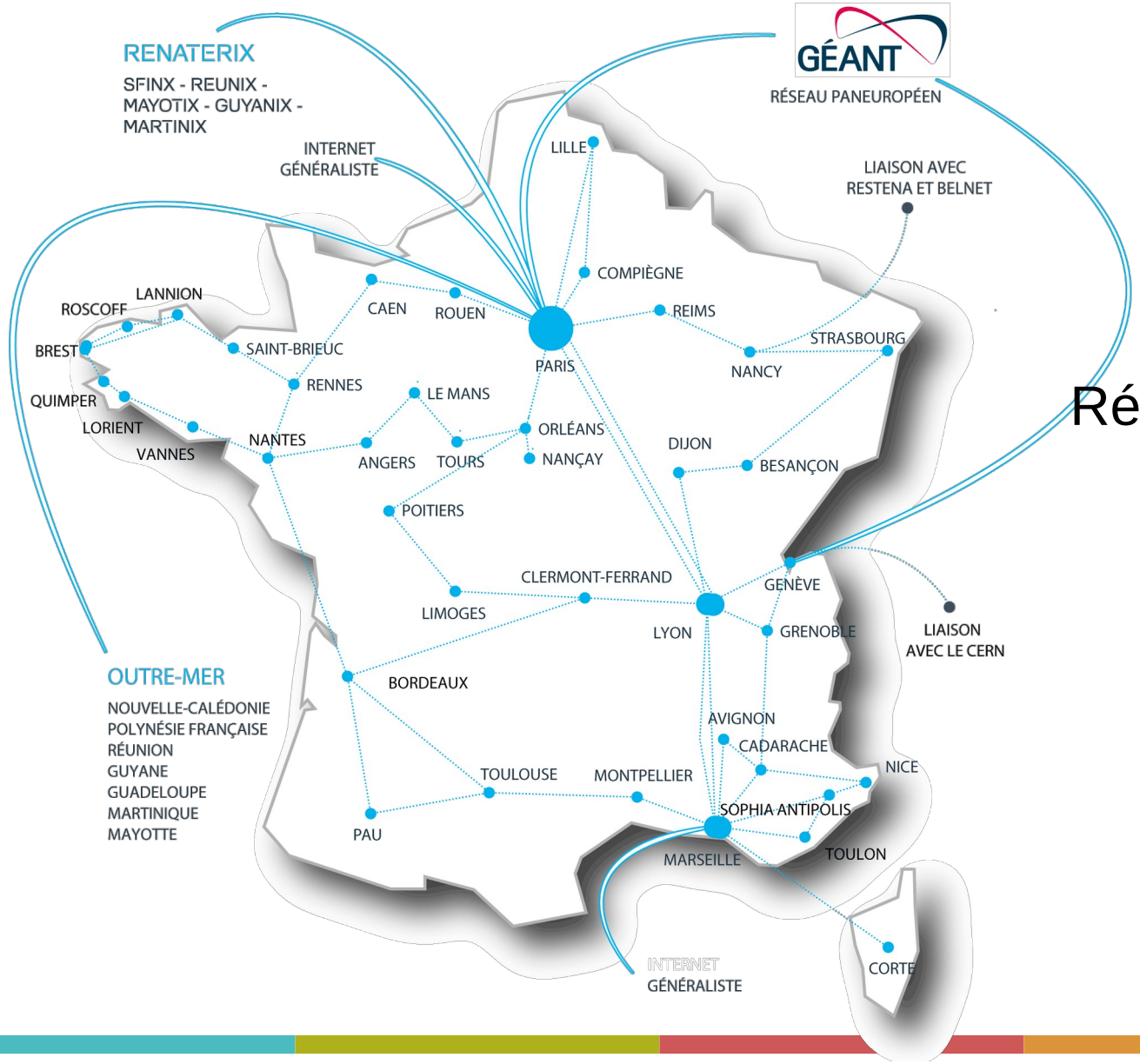
Réseau Métropolitain

Routage IP



Le réseau SYVIK (Normandie)

Routage IP



RENATER
Réseau d'interconnexion
de la recherche
et de l'enseignement

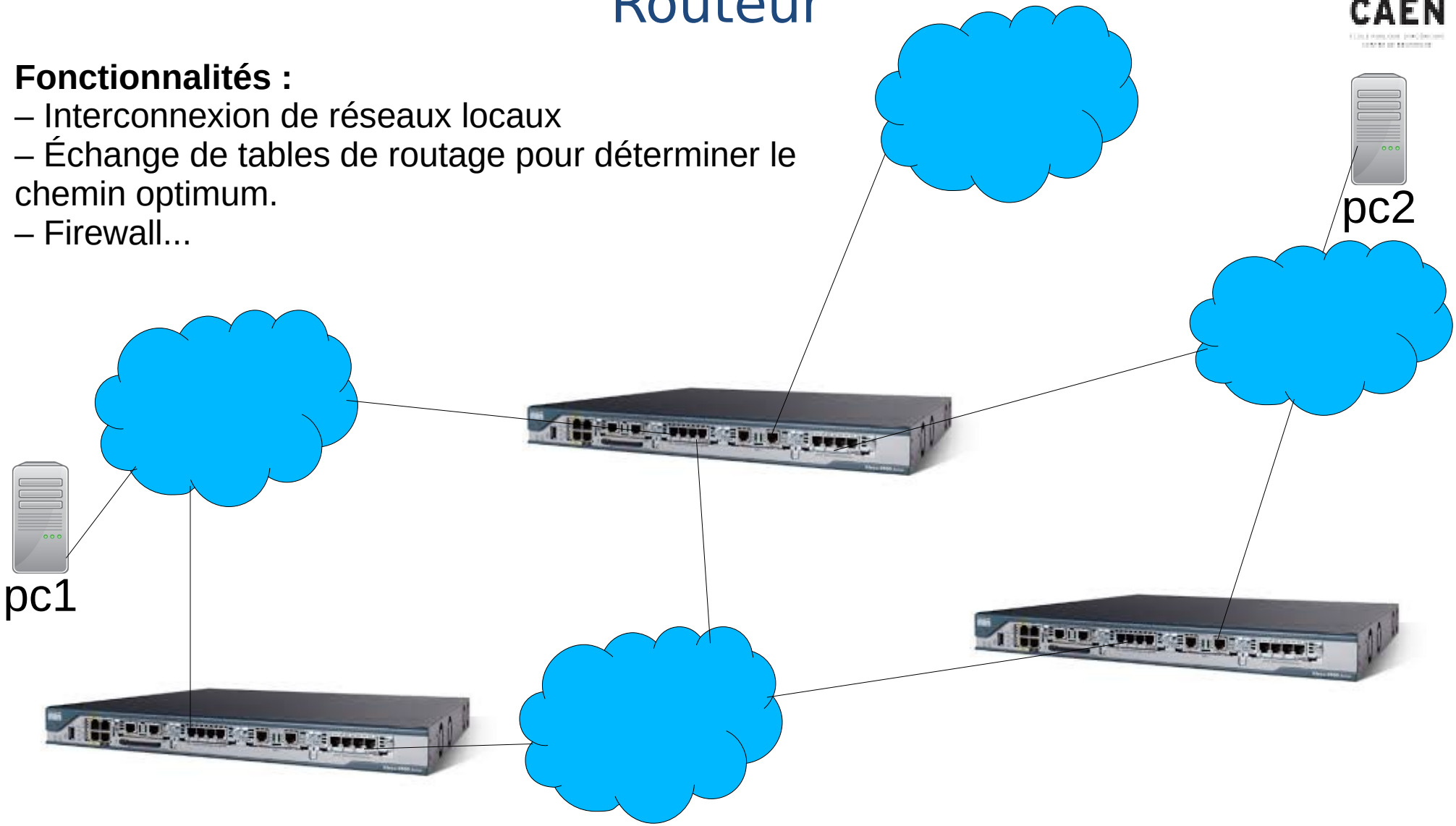
Routeur

Fonctionnalités :

– Interconnexion de réseaux locaux

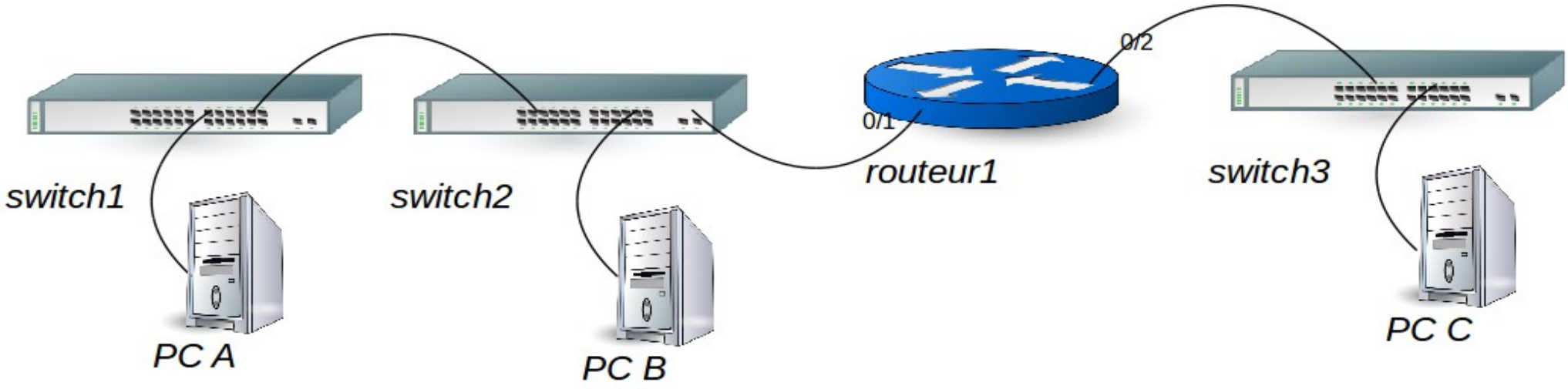
– Échange de tables de routage pour déterminer le chemin optimum.

– Firewall...



acheminement local / global
Ex : Ethernet

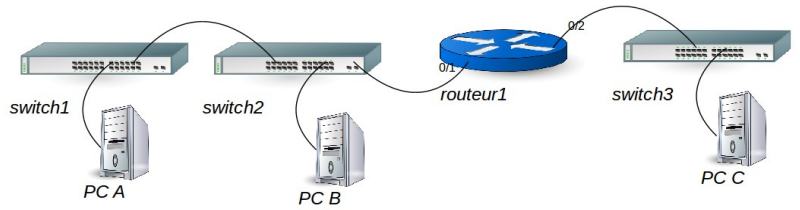
Voici le schéma physique d'un réseau



1 -Dessinez le schéma logique

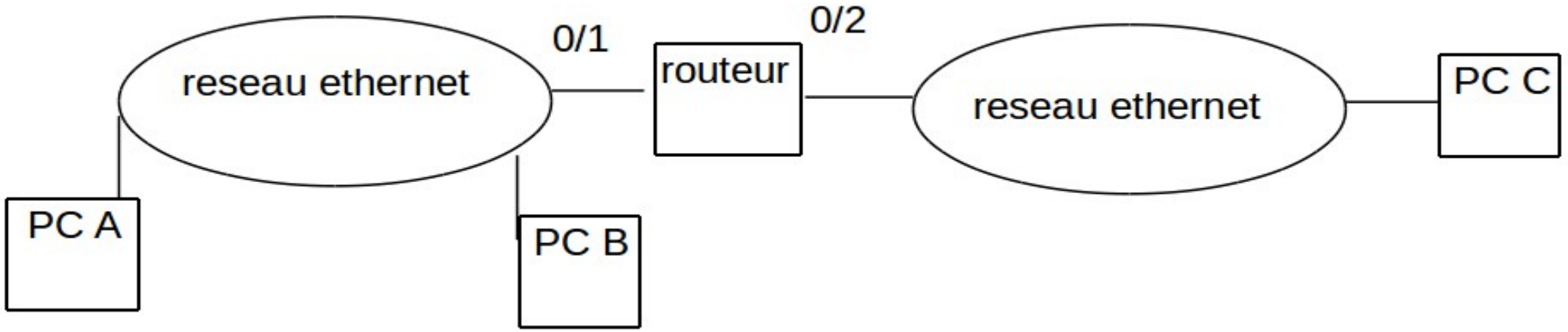
acheminement local / global
Ex : Ethernet

Voici le schéma physique d'un réseau



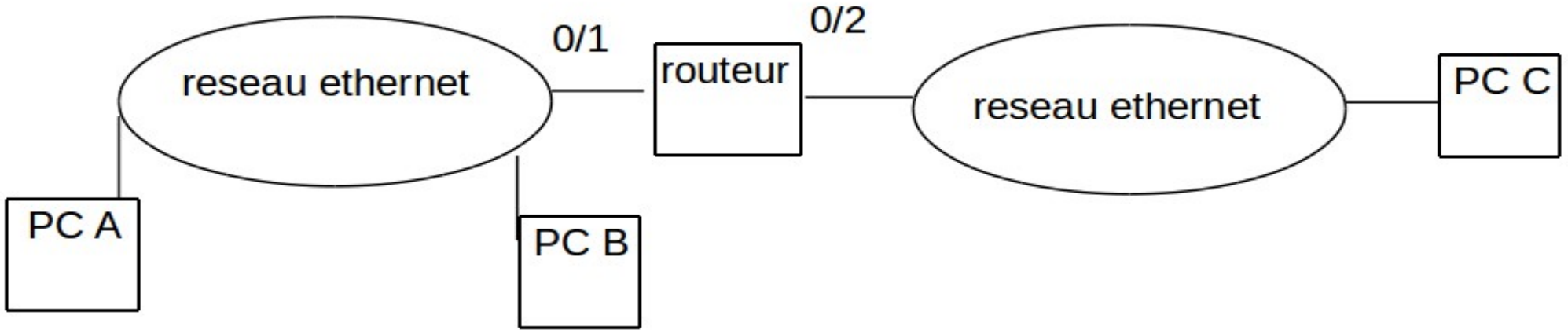
1 -Dessinez le schéma logique

Schéma Logique



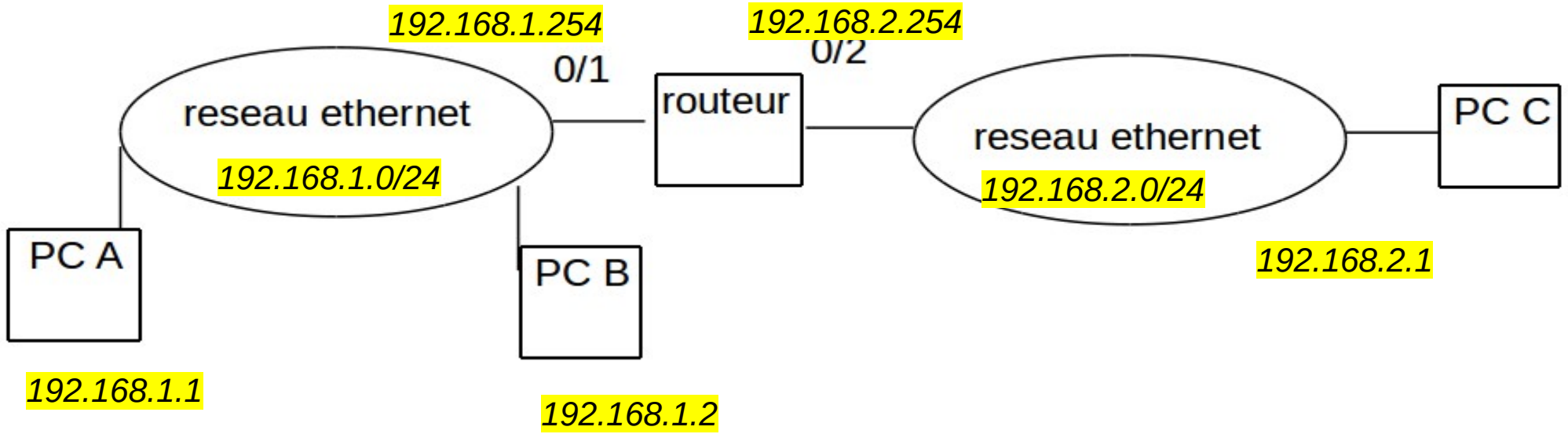
acheminement local / global
Ex : Ethernet

2 - Donnez des adresses aux machines parmi 192.168.0.0/16



acheminement local / global
Ex : Ethernet

2 - Donnez des adresses aux machines parmi 192.168.0.0/16



acheminement local / global
Ex : Ethernet

3 – Complétez le schéma suivant

	Adresse MAC source	Adresse MAC destination	Adresse IP source	Adresse IP destination
Trame de A vers B vue de B				
Trame de A vers C vue de A				
Trame de A vers C vue de C				

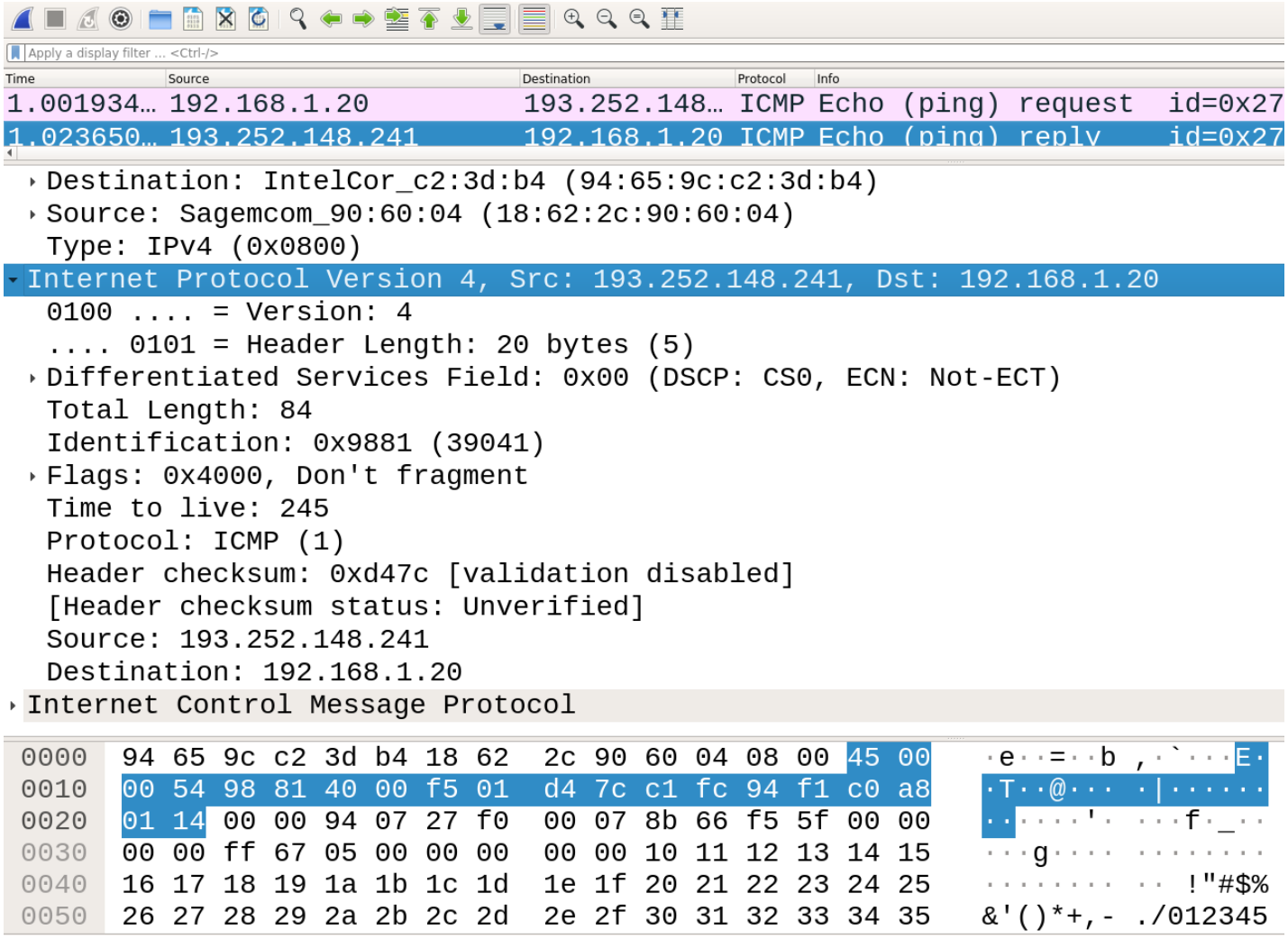
acheminement local / global
Ex : Ethernet

3 – Complétez le schéma suivant

	Adresse MAC source	Adresse MAC destination	Adresse IP source	Adresse IP destination
Trame de A vers B vue de B	***@MAC de A***	***@MAC de B***	***@IP de A***	***@IP de B***
Trame de A vers C vue de A	***@MAC de A***	***@MAC de 0/1***	***@IP de A***	***@IP de C***
Trame de A vers C vue de C	***@MAC de 0/2***	***@MAC de C***	***@IP de A***	***@IP de C***

acheminement local / global
Ex : Ethernet

exemple ping entre domicile et www.orange.fr capturé par wireshark au domicile.



Attribution des adresses

Une machine doit connaître :

- son adresse IP
- netmask
- passerelle
- @IP du DNS
- Éventuellement le domaine d'interrogation du DNS par défaut. Par exemple si le domaine est « ensicaen.fr », pour un utilisateur qui cherche à joindre « cybele », la requete au DNS sera « quelle est l'adresse IP de cybele.*ensicaen.fr* »

remarque l'adresse MAC est une caractéristique intrinsèque de la carte réseau (wifi/ethernet/bluetooth…).

Attribution des adresses

- **De manière dynamique**
  - serveur **DHCP** : Dynamic Host Configuration Protocole

    Le protocole le plus utilisé ; s'appuie sur UDP (port 53)

    4 phases :

    1) recherche de serveur par le client
    2) offres de la part des serveurs vers le client
    3) requete du client vers un serveur
    4) Accusé de réception de la part du serveur

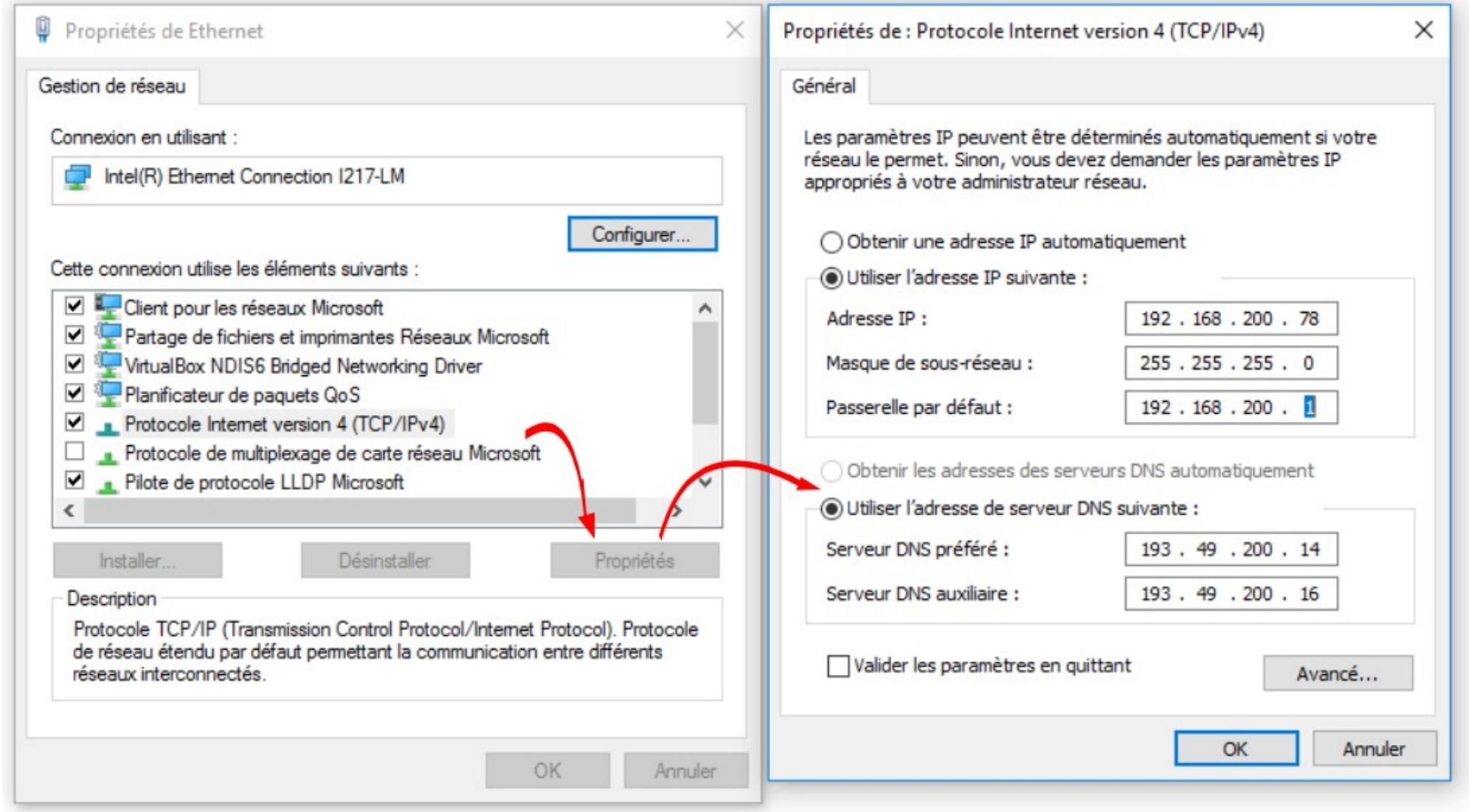
    Les phases 1 et 2 ne sont jouées par le client que si ce dernier vient d'être allumé.

  - Zeroconf / APIPA (Automatic Private Internet Protocol Addressing)

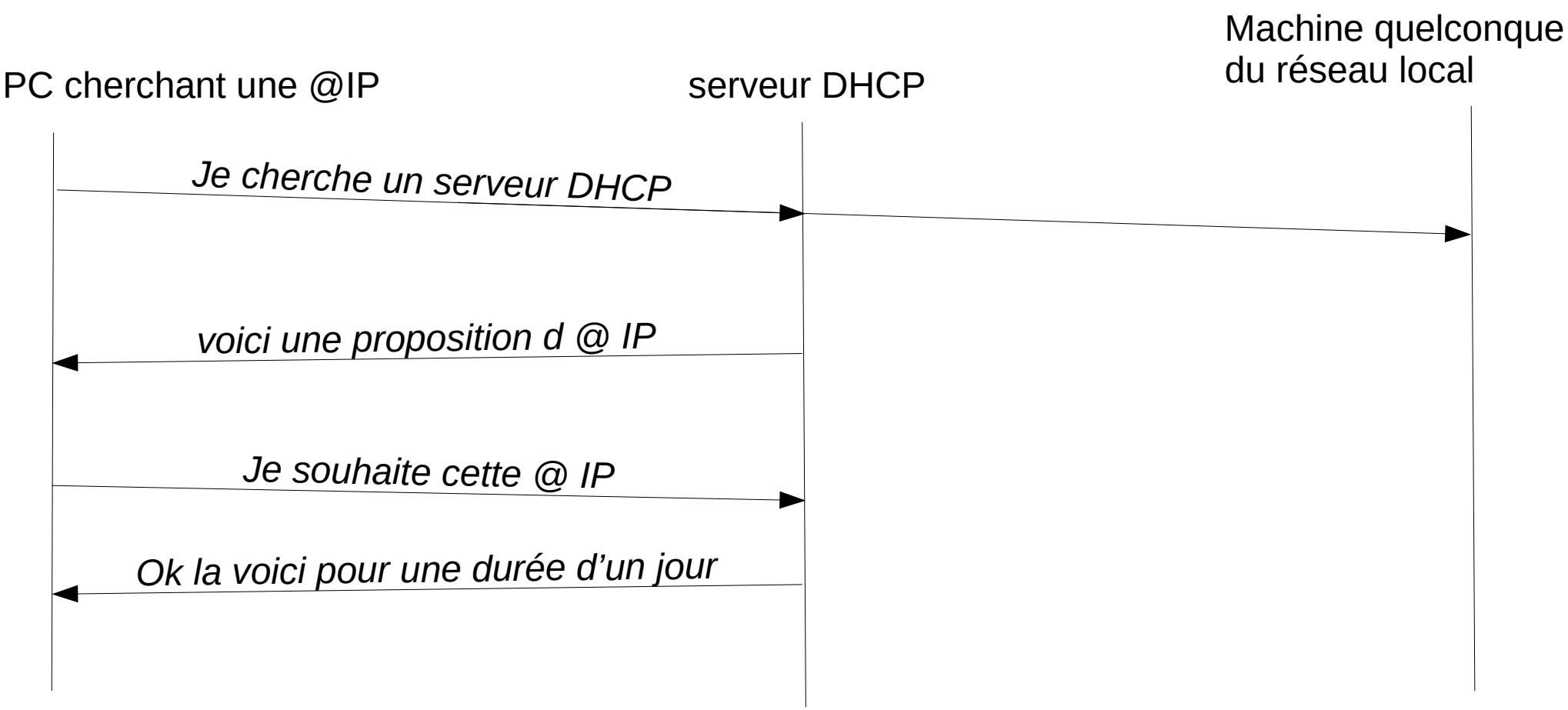
    choisi, en l'absence de DHCP, dans 169.254.0.0 à 169.254.255.255

- Attribué de manière **statique**, par l'administrateur de la machine.

Attribution statique avec windows 10



DHCP


PC cherchant une @IP
serveur DHCP
Machine quelconque du réseau local
Je cherche un serveur DHCP
voici une proposition d @ IP
Je souhaite cette @ IP
Ok la voici pour une durée d'un jour

Résolution de nom

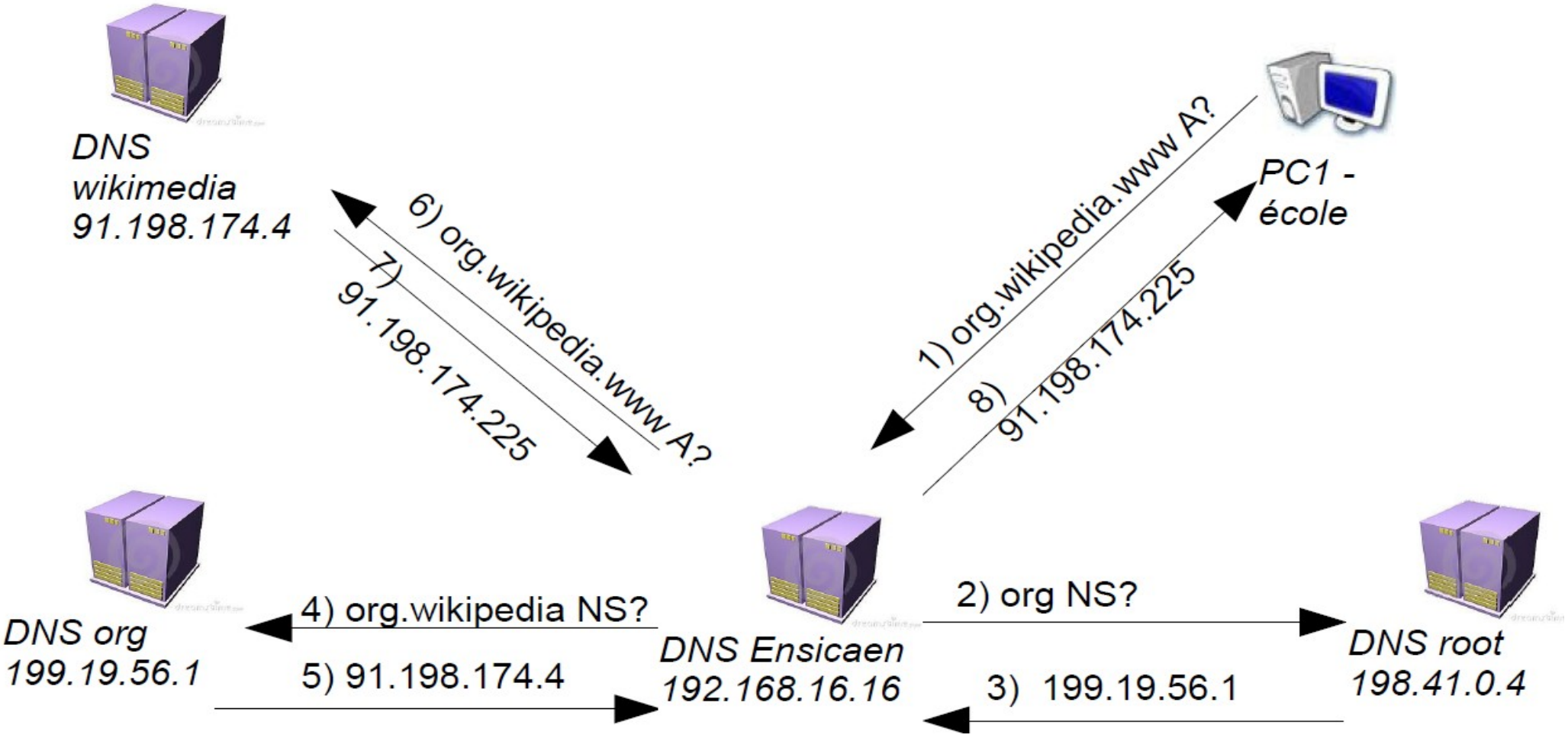
Plusieurs techniques existent pour connaître l'@ IP du correspondant :

- La machine possède un fichier /etc/hosts (Unix) ou lmhosts (Windows) dans lequel on a une correspondance entre des noms et des adresses IP ;
- La machine est connecté à un serveur NIS (Network Information Service de Unix) ou WINS (de Windows) et ce serveur met à disposition son propre fichier /etc/hosts ;
- Netbios Name Server, un service propore à Windows ;
- Domain Name Service ;
- Multicast DNS, un service de la série Zeroconf.

DNS : Domain Name System

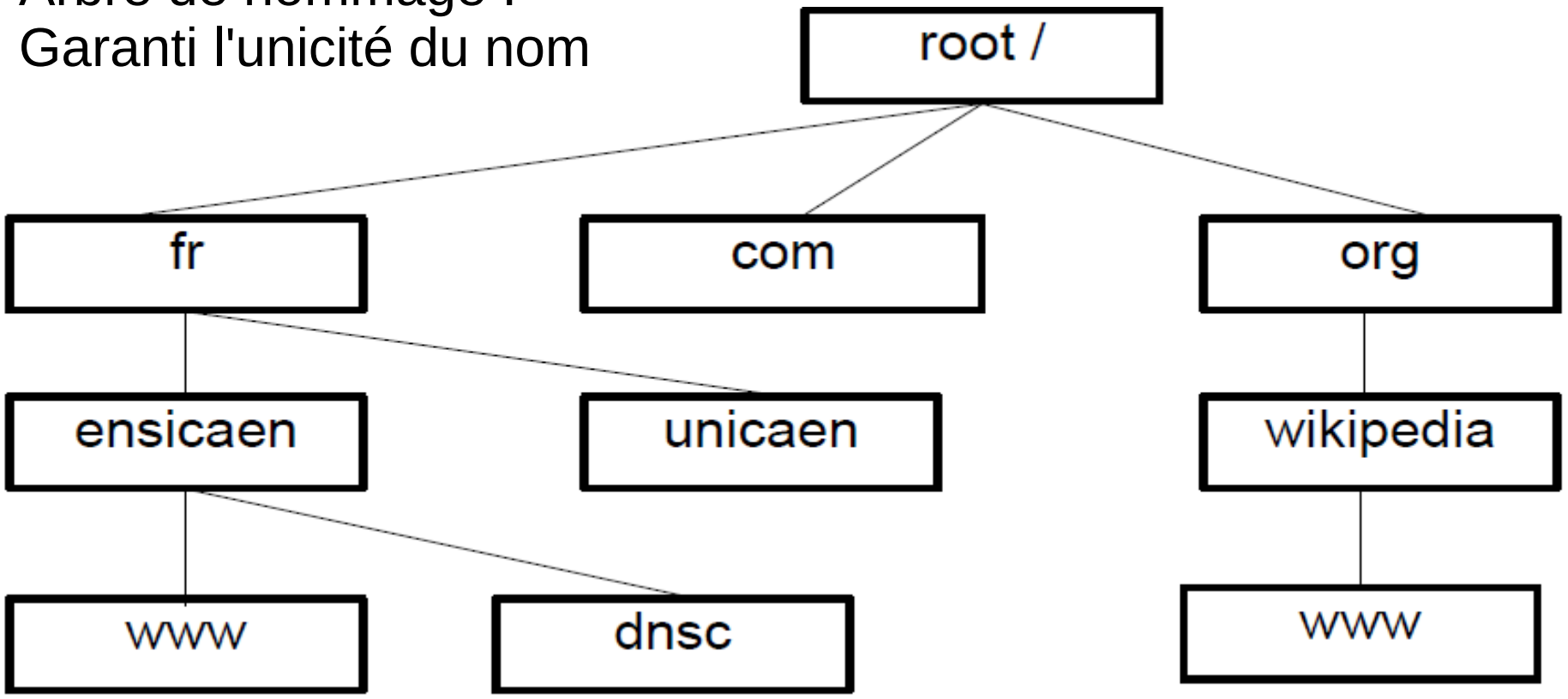
- Basé sur UDP
- 2 trames : 1 requête / 1 réponse
- Les requêtes DNS sont de plusieurs types parmi les plus fréquentes :
  - A : adresse Ipv4
  - AAAA : adresse Ipv6
  - NS : nom du DNS
  - MX : nom du mail exchanger (serveur SMTP)

DNS : exemple d'interrogation



DNS : hiérarchie de nommage :

Arbre de nommage :
Garanti l'unicité du nom



DNS : serveurs racines

- 13 serveurs racines : 10 hébergés aux US, 2 en Europe et 1 au Japon
- Dirigé par l'Internet Corporation for Assigned Names and Numbers en Californie
- L'ICANN a la possibilité de bloquer les sites par non diffusion des IP.
- L'ICANN gère les TLD : Top Level Domain : .com , .fr , .us , .org…
- L'AFNIC (Association française pour le nommage Internet en coopération) a reçu de la part de l'ICANN la délégation de gestion de .fr
- l'ICANN dépend encore du ministère du commerce américain.
- 185000 $ + 25000 $/an pour un nouveau suffixe (TLD).
- Il existe des serveurs de « racines ouvertes » encore peu déployés
- voir http://ipv4info.com/ pour se faire une idée de la répartition mondiale des adresses

Adress Resolution Protocol

- Permet de récupérer l'adresse MAC d'un correspondant
- Échange en 2 trames :
  - Une de broadcast fournissant l'adresse IP du correspondant.
  - Le correspondant répond en fournissant son adresse MAC

Trame : 28 octets

Hardware type		*Protocol type*
Hardware address length	*Protocol address length*	*Opération*
@ MAC source octets 1 à 4		
@ MAC source octets 5-6		*@ IP source octets 1-2*
@ IP source octets 3-4		*@ MAC destination octets 1-2*
@ MAC destination octets 3-6		
@ IP destination		

Réseaux de communication

Couches Transport

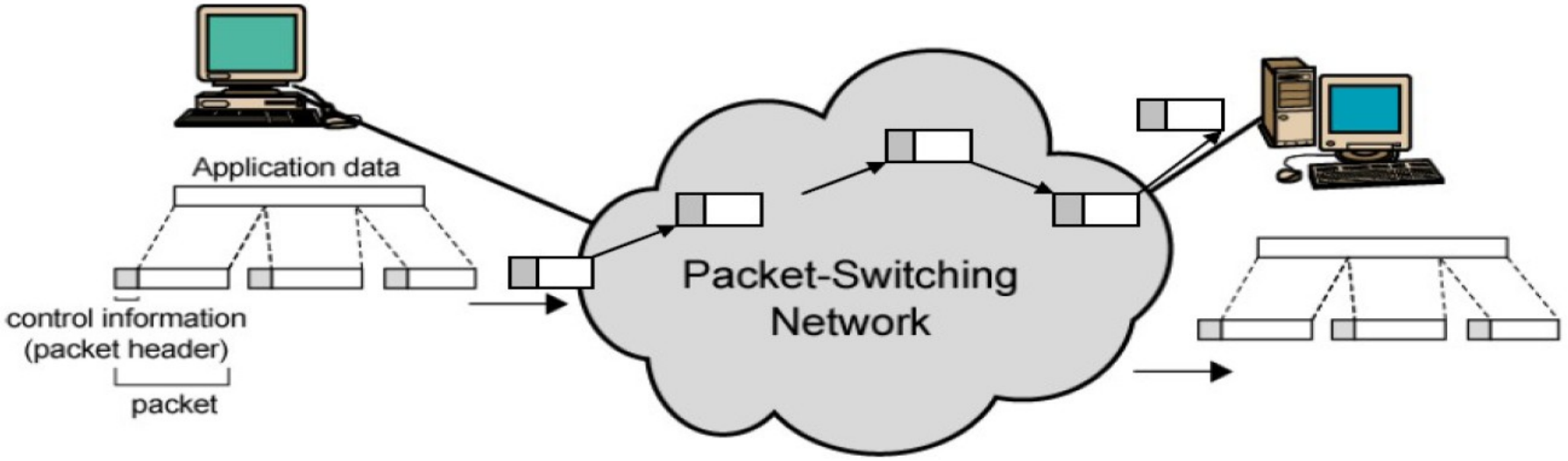
L'École des INGÉNIEURS Scientifiques

Normandie Université

Couche Transport

Transporter l'information de bout en bout en assurant :

- la remise de l'information à la bonne application : n° de port
- le contrôle des erreurs : détection / récupération



Couche Transport

- 2 protocoles
  - non Fiable : UDP
  - Fiable : TCP

UDP

- User Datagram Protocol
- Non fiable
- Entête :

port source	port dest.	longueur totale	checksum entete	Données transportées
2 o.	2 o.	2 o.	2 o.	

- Longueur maximale des données transportées :
  - MTU IP – long entete IP – entete UDP
  - soit 1500 – 20 – 8 = 1472 sur Ethernet.
  - Le reste peut être tronqué par UDP selon les implémentations.

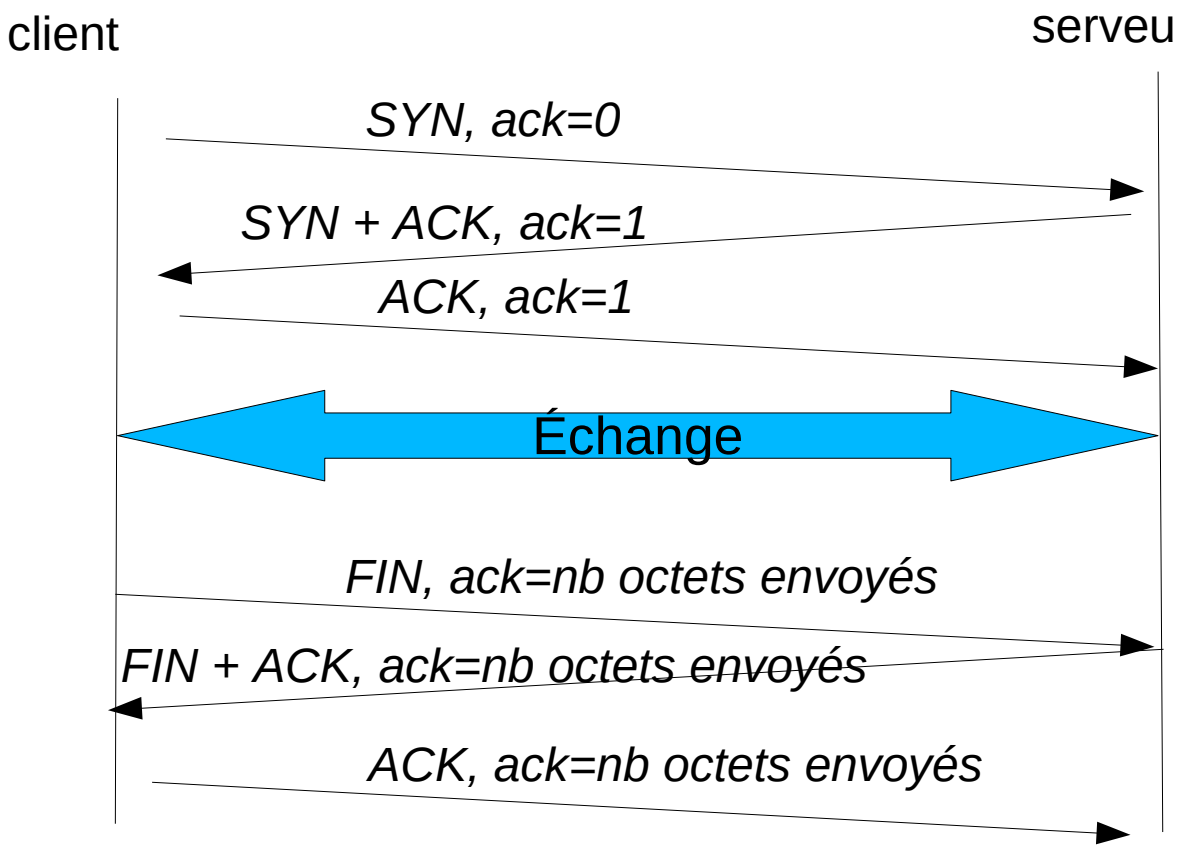
UDP

- Utilisé pour des **services simples**.
- Par exemple : DNS (port 53) , DHCP (port 67) , TFTP (port 69)
- Ou pour des services pour lesquels la reprise sur erreur arriverait trop tard comme en **téléphonie**.
- On distingue le **client** qui envoie la requête ;
- Et le **serveur** qui répond.
- Le **serveur** doit être lié à un **port connu à l'avance** par le client. Ces ports sont répertoriés pour les services classiques : 53 pour DNS, 67 pour DHCP...
- Le port du **client** est choisi par le système d'exploitation parmi les **ports libres**.

TCP

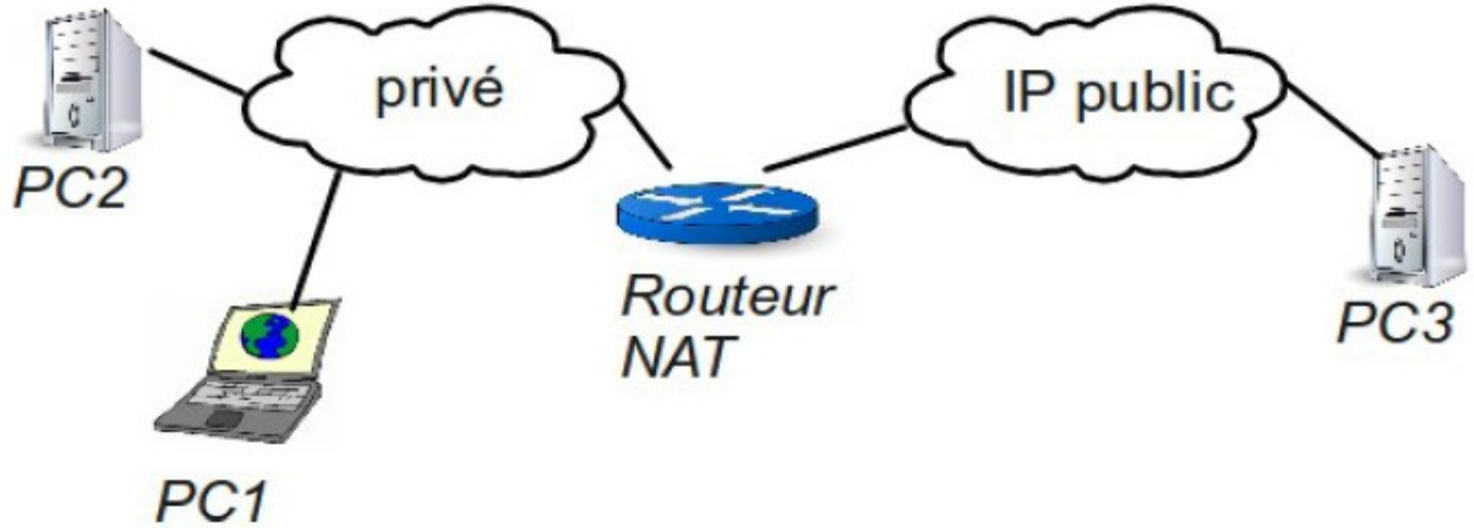
- Transmission Control Protocol
- **Transport Fiable** :
  - les paquets sont numérotés ;
  - les paquets **non acquittés en temps utile sont envoyés de nouveau**
  - l'application n'a pas à ce soucier de la gestion des erreurs. TCP s'en charge.
- Permet le **contrôle de flux**.
- Comme en UDP on distingue le **client** et le **serveur**
- Le **serveur** doit être lié à un **port connu à l'avance** par le client. Ces ports sont répertoriés pour les services classiques : 21 pour FTP, 22 pour SSH, 23 pour telnet, 25 pour SMTP, 80 pour HTTP...
- Le **port du client** est choisi par le système d'exploitation parmi les **ports libres**.

TCP – diagramme des séquences



NAT / PAT

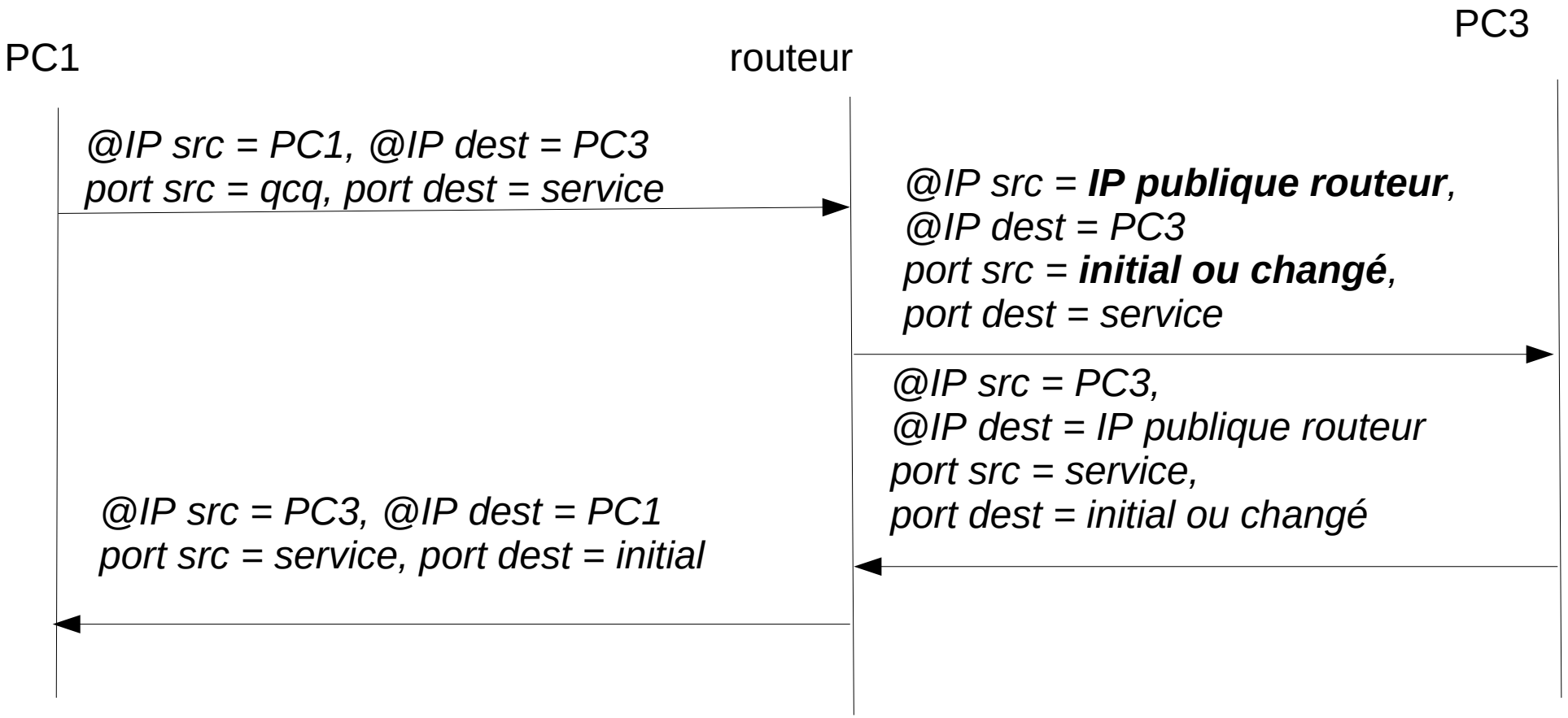
- Network Address Translation / Port Address Translation
- Pour faire face à la pénurie d'adresses IP et partager une connexion internet (c'est ce que faitune box ASDL)
- Mais aussi pour contrôler ce qui sort sur l'internet public (Toutes les connexions data sur réseau GSM et dérivés utilisent ce principe).



NAT / PAT

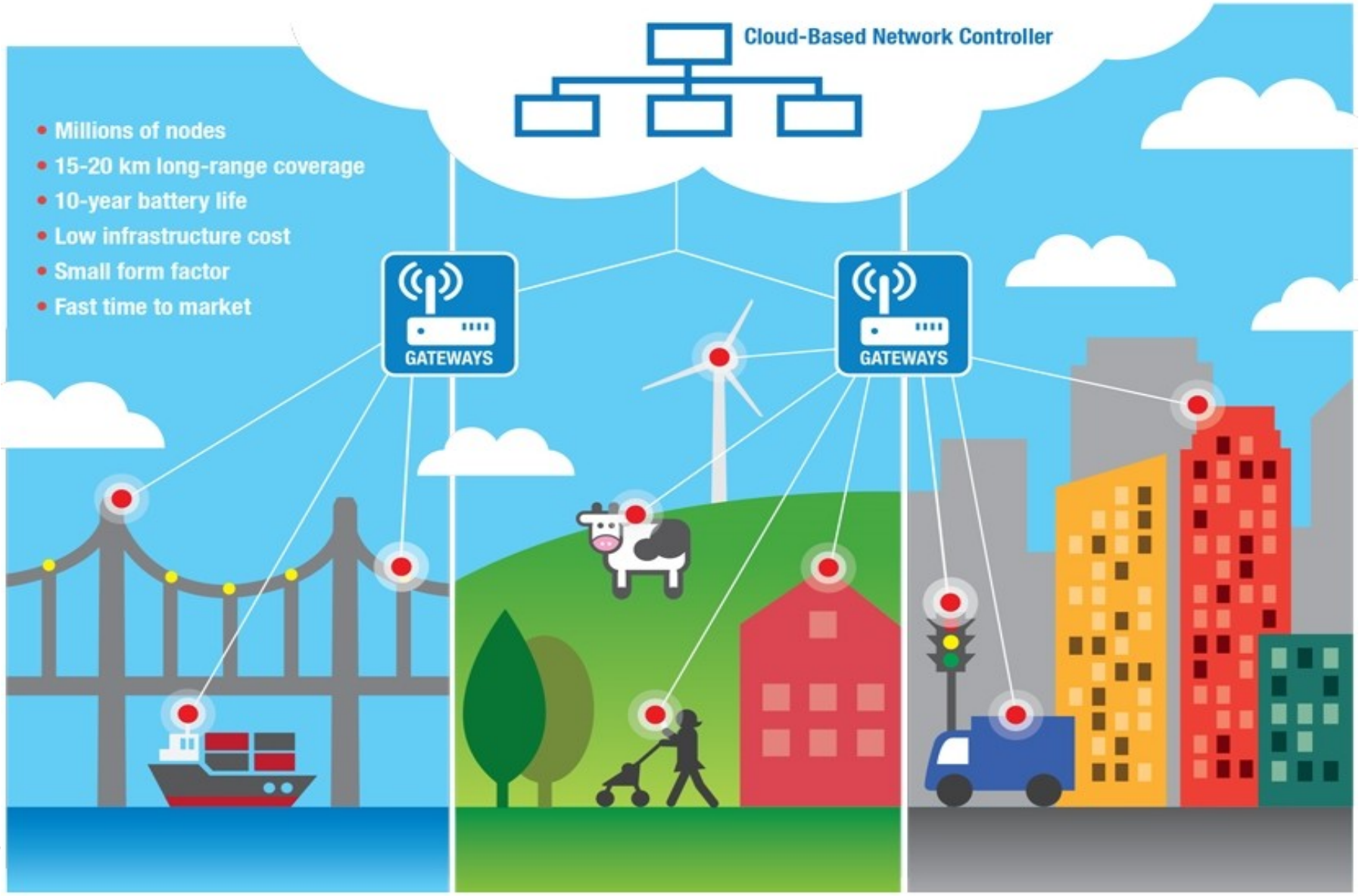
- Le routeur possède **2 adresses IP** car il a 2 interfaces.
- Le routeur possède une **@IP publique** et une **@IP privée**.
- Le routeur **mémorise** le numéro de **port source** utilisé par PC1.
- Si le routeur (ou PC2) utilise **déjà ce port source** à destination de PC3 et sur le même port de destination, le routeur **remplace également le port source**.
- PC3 répond donc au routeur qui renvoie à PC1 en changeant l'adresse destination et éventuellement le port destination.
- Comment PC3 peut-il ouvrir une connexion sur PC1 ? Il faut que cela soit prévu au niveau du routeur. (par exemple redirection de port d'une box ADSL).
- Pour ICMP, pas de notion de port → le routeur utilise le numéro de séquence ICMP

NAT / PAT


PC1
routeur
PC3
@IP src = PC1, @IP dest = PC3
port src = qcq, port dest = service
@IP src = IP publique routeur,
@IP dest = PC3
port src = initial ou changé,
port dest = service
@IP src = PC3,
@IP dest = IP publique routeur
port src = service,
port dest = initial ou changé
@IP src = PC3, @IP dest = PC1
port src = service, port dest = initial

L'IOT c'est quoi

Internet Of Things

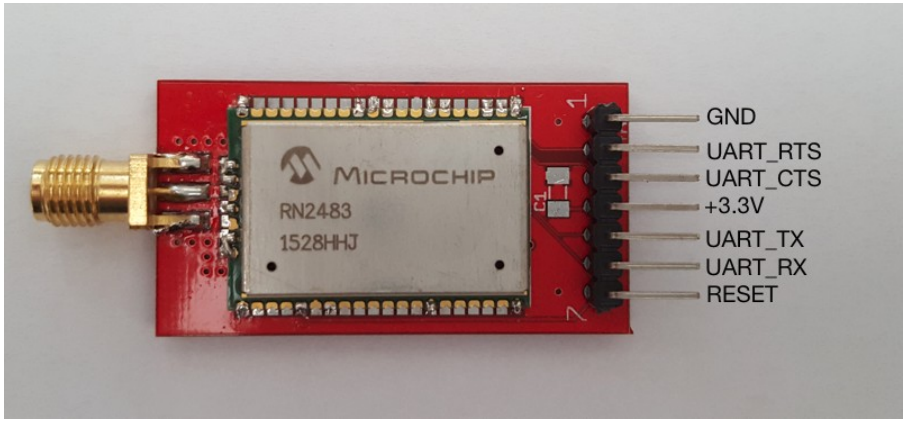


from www.real-iot.com

LoraWAN : Couche physique

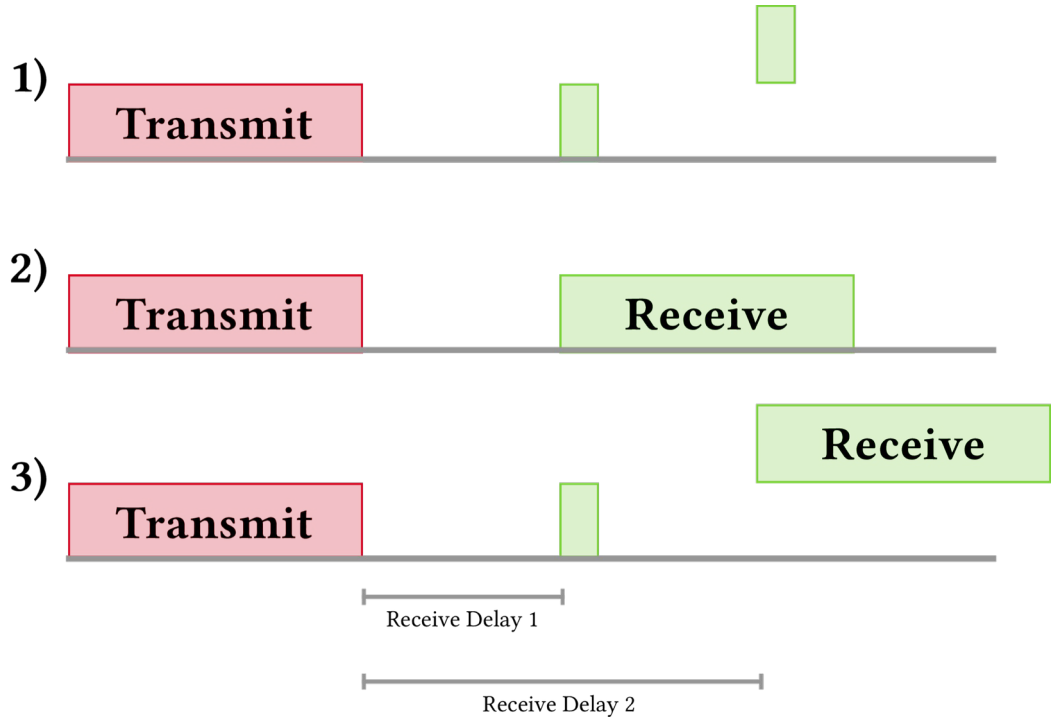
- Exemple de matériel
- Gateway :

- modem lora : Microchip RN2483
  – interface série ;
  – qq 10 €
  – mode sommeil



LoraWAN : Couche physique

- LoraWan est la couche MAC du modèle pour l'IOT du Lora Alliance
- spécifie la communication nœuds ↔ gateways
- Communication bidirectionnelle mais toujours à l'initiative du noeud.
- 3 classes de communication, mais la plus utilisée est la classe A : la gateway ne peut envoyer des données que soit 1s ou soit 2s après le uplink.



LoraWAN : Couche physique

- LoraWan utilise des bandes de fréquences libres
- En Europe : 863-870 MHz

  Les nœuds ne doivent pas émettre plus de 1 % du temps sur le canal
- Aux US : 902 Mhz-928MHz

- Modulation à étalement de spectre :

  → bonne immunité au bruit

  → bonne résistance au multipath fading

  bonne efficacité en bits/Watt (un peu meilleur que Zigbee)
- Canaux de largeur 125 kHz. Possibilité d'utiliser 1, 2 ou 4 canaux pour augmenter l'étalement ainsi que d'augmenter la longueur de la séquence codante (chirp)

  → meilleur immunité au bruit (donc distance plus grande)

  → mais à cause de la règle des 1 %, diminution du débit.

  → débits de 250 bps à 27 kbps (en Europe)

LoraWAN : Couche physique

- Géolocalisation possible par mesure de temps de vol en associant plusieurs passe-relle.
- chiffrement symétrique à clé secrète de type AES-128
- puissance maximum d'émission : 25 mW pendant 1 % du temps
- portée maximum en champ libre : 15 km
- Accès au média de type ALOHA : pas de vérification que le canal est libre. Les codes sont ortogonaux et permettent des émissions simultanées par plusieurs nœuds.
  → 2 modes de fonctionnement : avec ou sans acquittement de ma parte de la gate-way
- Adressage : par deux identifiants :
  DevEUI, identifiant du nœud (64 bits)
  AppEUI, identifiant de l'application côté broker (64 bits)

- Le nœud doit d'abord rejoindre le réseau en utilisant au choix 3 canaux réservés à cet usage

LoraWAN : Couche physique

- En fonction du spreading factor et de la règle des 1 %, le nombre d'octets de la charge utili varie de 51 à 222 octets.

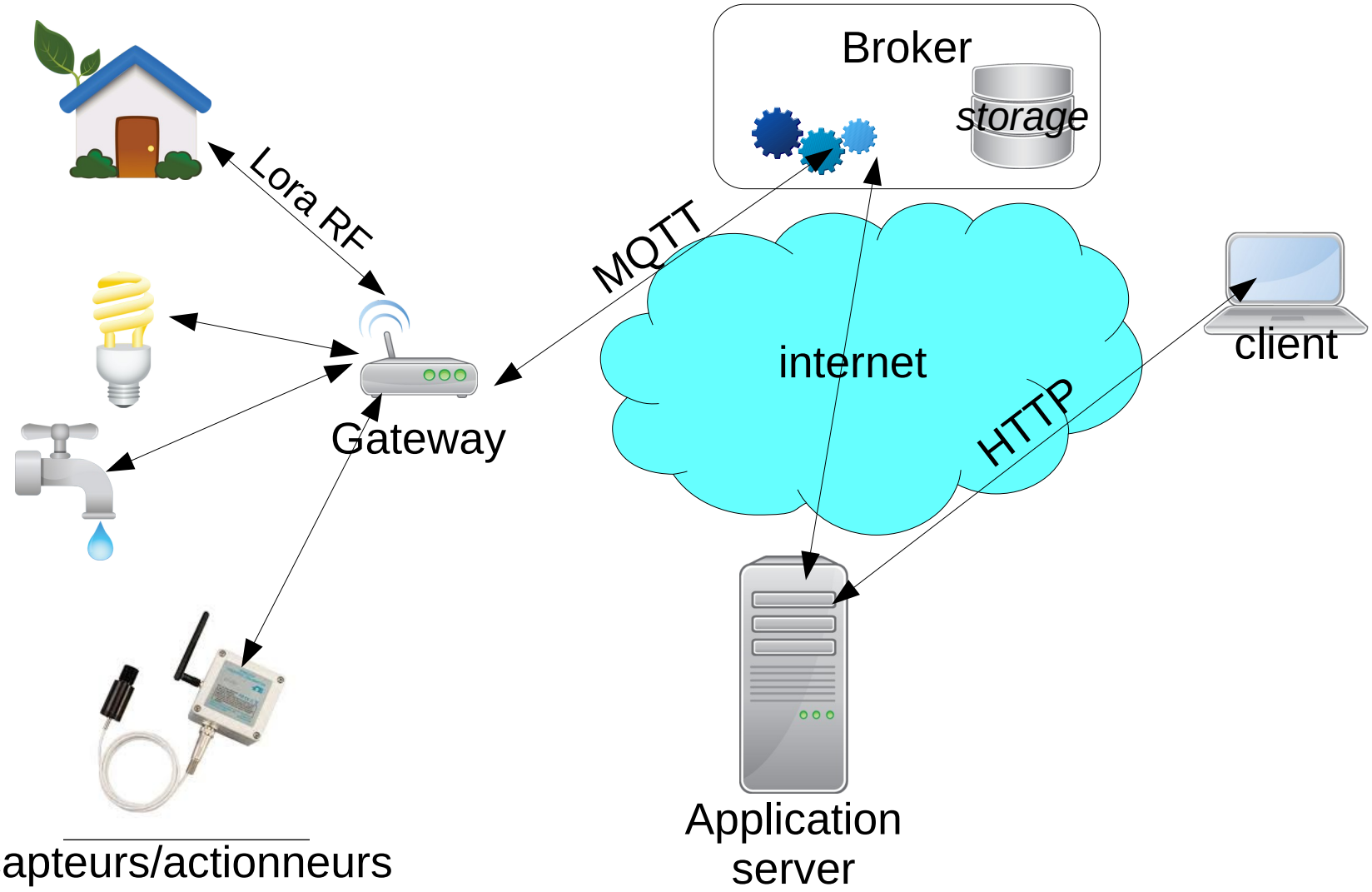
- Il existe différents broker :
  privé : Orange, Bouygues…
  communautaire : The Things Network

- https://www.thethingsnetwork.org/

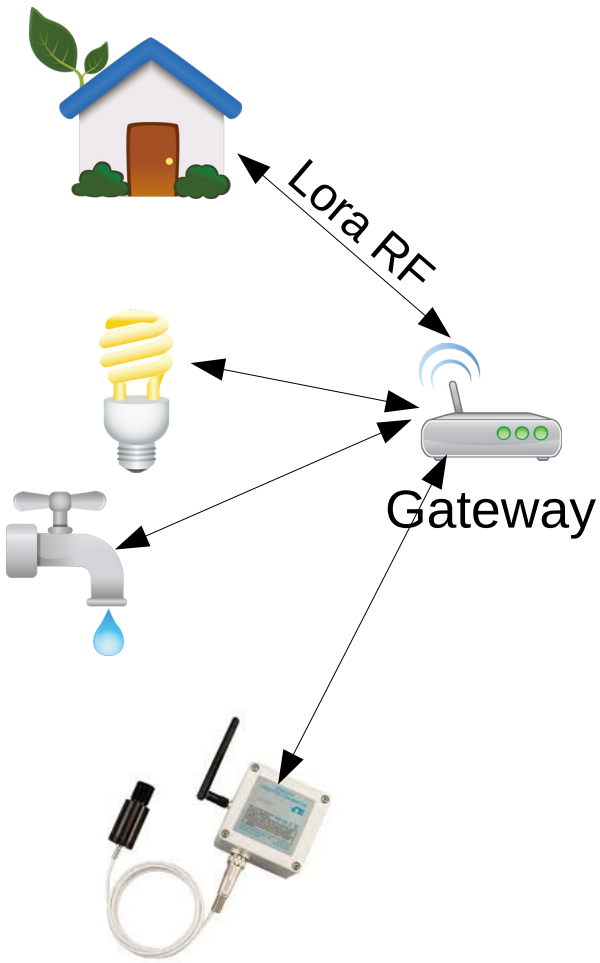
MQTT

- Créé pour l'IOT
- Moins d'overhead que pour HTTP
- Donc plus rapide
- Consomme aussi moins d'énergie
- Mais :
  - sur un port particulier, donc attention au firewall
  - sécurisation via MQTTs
- Donc pas sûr qu'il résiste à la HTTPisation

L'IOT c'est quoi



L'IOT c'est quoi



capteurs ↔ Gateway

– faible coût

– faible consommation

– longue portée

– chiffrement

exemples :

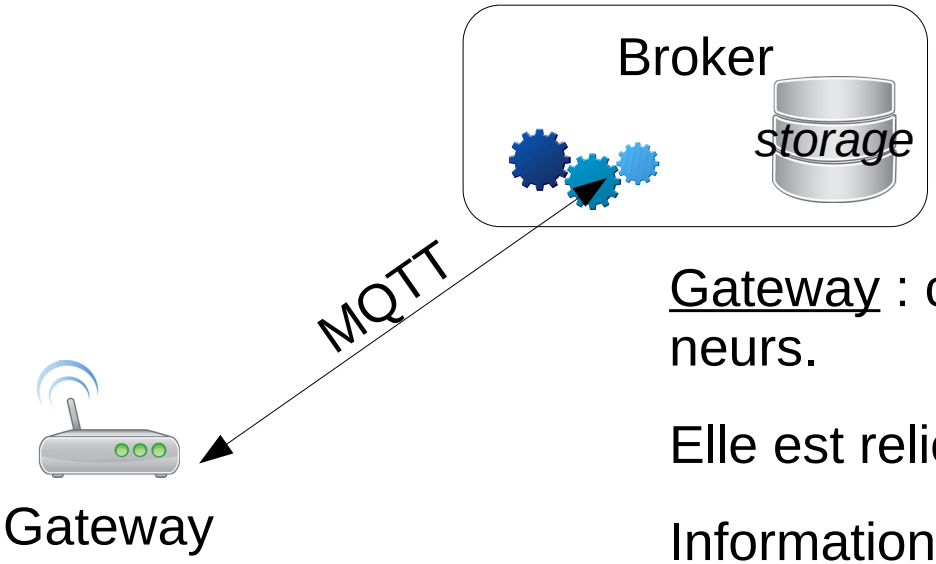
Lora

Sigfox

Zigbee *(faible portée)*

Bluetooth Low Energy *(faible portée)*

L'IOT c'est quoi



Gateway ↔ Broker

Gateway : collecte/dialogue avec les capteurs action-neurs.

Elle est reliée par internet au Broker

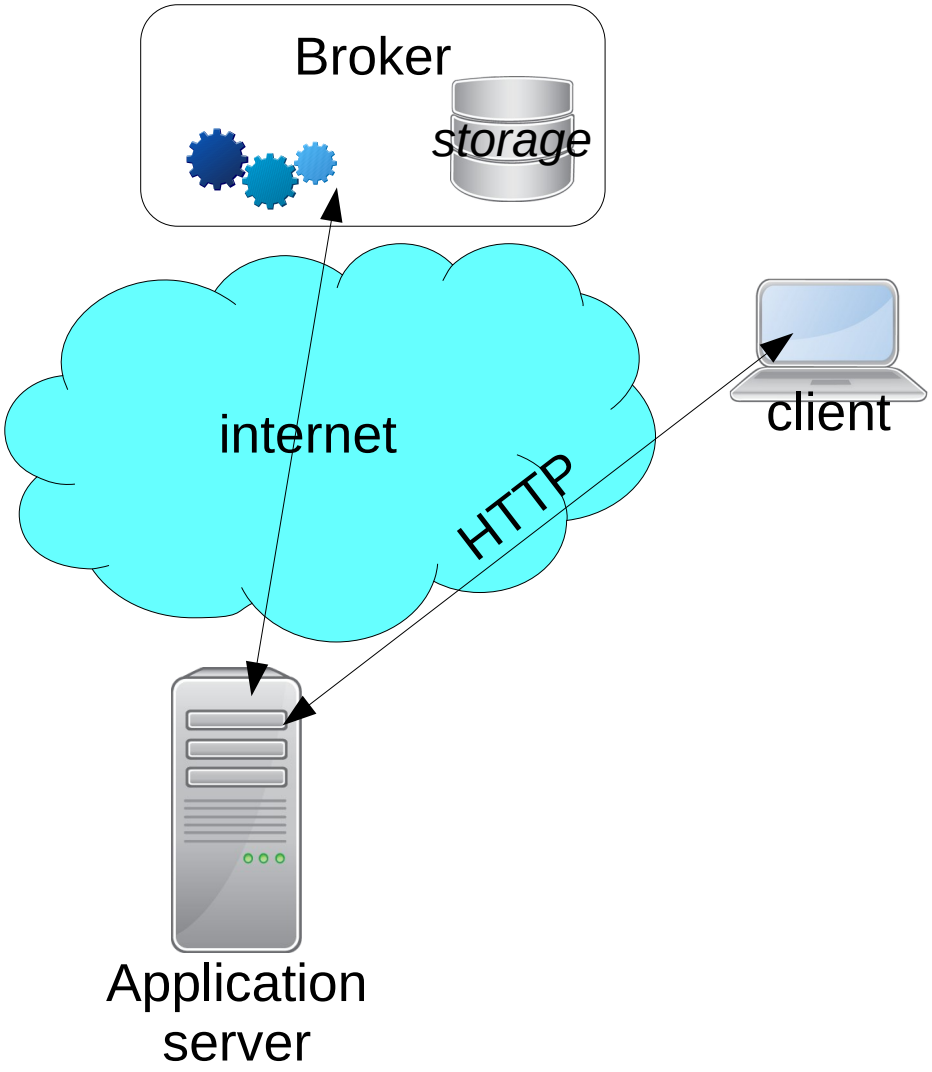
Informations transportées par un protocole léger : MQTT

Message Queue Telemetry Transfer protocol

Broker : transforme l'information binaire en information plus exploitable et la stocke. Ex :

« 0x0B » → { "water level" : 12 }

L'IOT c'est quoi



Gateway ↔ Broker

Application Server : Présentation de l'information. Analyse des données. remontées d'alarmes….

Client : PC / smartphones. Applications de monitoring.

Protocole HTTP
Généralités

- présentation inspirée de Stéphane Bortzmeyer : **www.bortzmeyer.org**
- Hypertext Transfer Protocol : protocole de couche 7, **applicatif**
- Protocole de transfert de fichiers. C'est un des protocoles du Wold Wide Web.
- Certes, mais qu'est-ce que la **Toile (Web)** ? : *« Information space where documents [...] are identified by Uniform Resource Locators (URLs, such as http://machine.domaine.org/ ressource.txt» (wikipedia)*
- Ne transporte pas que des documents hypertextes (HTML, mais aussi des images, ou **tout autre fichier**)
- Utilisé aussi pour **faire autre chose que du web** : mise à jour logiciel, internet des objets...

Protocole HTTP
principe

- **Client/serveur**
- repose sur **TCP**
  - **ouverture** de connexion TCP en 3 trames
  - **requête** HTTP
  - **réponse** du serveur
  - **fermeture** de la connexion TCP (ou pas…)
- En mode texte, dans la version 1
- port 80 pour HTTP
- port 443 pour HTTPS, c’est à dire HTTP over TLS (à priori…)

Protocole HTTP
requête

- Agit sur une ressource par une **méthode** :
  - **GET** : télécharger
  - PUT : téléverser
  - **POST** : modifier
  - delete : effacer
- Le format de la requête (au format texte) :

```
Méthode chemin/ressource HTTP/version
Host :
autres champs optionnels (1 par ligne)...

ligne vide ( \r\n )
corps de la requête (optionnel)
```

Protocole HTTP
requête

- La ressource est désignée par un emplacement : site **répertoire/fichier**
  - une ressource n’est pas forcément un “vrai” fichier. Il peut être créé de manière dynamique...
- Le site est spécifié par le champ **HOST**
  - En effet, une machine peut héberger plusieurs site
  - le répertoire est un répertoire relatif aux répertoires du site
- Les autres champs sont **facultatifs**

Protocole HTTP
requête

Exemple : demande dela page d'accueil d'ADE

```
GET /direct/myplanning.jsp?ticket=ST-1173f8ae9&lang=en HTTP/1.1
Host: ade.ensicaen.fr:8080
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/73.0.3683.103 Safari/537.36
Accept: text/html,application/xhtml+xml,image/webp,image/apng
Accept-Encoding: gzip, deflate
Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: _ga=GA1.2.377812723.1537194135
```

Protocole HTTP
requête

- codage des **caractères particuliers** des ressources
  - pas d'espaces dans les URL : remplacé par +
  - les codes ASCII différents de **majuscules minuscules - _ .** doivent être remplacés par le code utf-8 précédé pour chaque valeur d'octet du signe %

    exemple « `déjà l'Automne` » → `d%C3%A9j%C3%A0+l%27Automne`

  - **?** utilisé pour séparer la ressources d'une liste de **champ=valeur** *(cf. exemple précédent)*
  - **&** utilisé pour séparer les champs *(cf. exemple précédent)*

- **;q=valeur** : facteur de préférence relative. Par exemple :

  ```
  Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
  ```

  signifie que le français est préféré, suivi de l'anglais US, puis de l'anglais générique.

Protocole HTTP
Cookies

- cookies :
  - utilisés par les serveurs pour enregistrer des **informations sur le client**.
  - envoyé par le client **automatiquement** au serveur lorsque celui-ci possède un cookie relatif au site
- Le serveur envoie le champs « Set-Cookie » dans sa réponse :
  Set-Cookie:name=value
- Avant d'envoyer une requête, le client vérifie dans sa base de cookies s'il possède des cookies pour ce site et les envoie dans l'entête de la requête :
  Cookie:name=value
- Il est préférable que le serveur envoie des valeurs qui n'ont de sens que pour lui…
- Servent à maintenir les sessions… même celles sécurisées (google) !
  - les cookies sont stockés en clair dans le système de fichier

Protocole HTTP
réponse

- Message de réponse au format texte :

```
code_de_retour message explicite
Content-Type: type_mime
autres champs (1 par ligne)...
ligne vide ( \r\n )
corps de la réponses (souvent)
```

Protocole HTTP
réponse

- **code de retour sur 3 chiffres :**
  - 2xx : tout va bien (200 OK)
  - 3xx : succès partiel → souvent redirection
  - 4xx : problème (403 interdit, 404 not found, 451…)
  - 5xx : erreur sur le serveur (500 : crash code sur le serveur)
  -

- **Content-Type :**

  suivi du type MIME et du sous-type

  Exemple : text/html, text/plain, image/jpg, image/png…

  MIME : *Multipurpose Internet Mail Extensions*, inventé initialement pour le courrier

Protocole HTTP
réponse

Exemple :

```
HTTP/1.1 200 OK
Date: Mon, 29 Apr 2019 09:56:56 GMT
Server: Apache/2.4.6 (CentOS)
Last-Modified: Wed, 01 Jun 2016 10:50:58 GMT
Content-Length: 107862
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>...
```

Protocole HTTP
codage binaire

- Lorsque HTTP transporte du HTML qui lui même transporte une ressource binaire (image, video, sons...), il pourrait arriver que les octets représentent une suite de caractères ayant un sens. Par exemple </HTML>
- Pour éviter cela, on utilise un encodage dit **base64**
  - les octets sont regroupés par 3 pour former un mot de 24 bit
  - le mot de 24 bits est divisé en 4 mots de 6 bits
  - Les mots de 6 bits sont codés en utilisant une table de codage n'utilisant que les caractères A-Z, a-z, 0-9, + et /

    exemple : 00000000 11111111 00000000

    → 000000 001111 111100 000000 → AP8A (cf. table base64)

- utilisé aussi pour transporter des images dans le courrier électronique

Protocole HTTP
codage base64

Valeur	Codage	Valeur	Codage	Valeur	Codage	Valeur	Codage
0	000000 A	17	010001 R	34	100010 i	51	110011 z
1	000001 B	18	010010 S	35	100011 j	52	110100 0
2	000010 C	19	010011 T	36	100100 k	53	110101 1
3	000011 D	20	010100 U	37	100101 l	54	110110 2
4	000100 E	21	010101 V	38	100110 m	55	110111 3
5	000101 F	22	010110 W	39	100111 n	56	111000 4
6	000110 G	23	010111 X	40	101000 o	57	111001 5
7	000111 H	24	011000 Y	41	101001 p	58	111010 6
8	001000 I	25	011001 Z	42	101010 q	59	111011 7
9	001001 J	26	011010 a	43	101011 r	60	111100 8
10	001010 K	27	011011 b	44	101100 s	61	111101 9
11	001011 L	28	011100 c	45	101101 t	62	111110 +
12	001100 M	29	011101 d	46	101110 u	63	111111 /
13	001101 N	30	011110 e	47	101111 v		
14	001110 O	31	011111 f	48	110000 w		
15	001111 P	32	100000 g	49	110001 x		
16	010000 Q	33	100001 h	50	110010 y		

Protocole HTTP
utilisation

- Un navigateur web Edge, Safari, Firefox, Chrome…
  - récupère une ressource,
  - l'analyse et interprète la ressource demandée (html, javascript…)
  - demande d'autres ressources si nécessaire (css, favicon, images, javascript…)
  - et enfin l'affiche.

- Programmation d'un client HTTP (qui ne ferait pas d'affichage…ouf!) :

  Bibliothèque C : CURL

  https://curl.haxx.se/libcurl/c/example.html

  Beaucoup d'exemples notamment HTTPS

  curl est aussi une commande Linux

Protocole HTTP
utilisation

- Un navigateur web Edge, Safari, Firefox, Chrome…
  - récupère une ressource,
  - l'analyse et interprète la ressource demandée (html, javascript…)
  - demande d'autres ressources si nécessaire (css, favicon, images, javascript…)
  - et enfin l'affiche.

- curl est une commande Linux et une bibliothèque de programmation

  Page suivante, une requête GET sur un site en HTTPS

Protocole HTTP
CURL

```
curl -v https://www.google.com/
* Trying 193.49.200.22...
* Connected to proxy.ensicaen.fr (193.49.200.22) port 3128 (#0)
* Establish HTTP proxy tunnel to www.google.com:443
> CONNECT www.google.com:443 HTTP/1.1
> Host: www.google.com:443
> User-Agent: curl/7.47.0
> Proxy-Connection: Keep-Alive
>
< HTTP/1.1 200 Connection established
<
* Proxy replied OK to CONNECT request
* found 148 certificates in /etc/ssl/certs/ca-certificates.crt
* found 597 certificates in /etc/ssl/certs
* SSL connection using TLS1.2 / ECDHE_ECDSA_AES_128_GCM_SHA256
*        server certificate verification OK
*        common name: www.google.com (matched)
*        server certificate expiration date OK
*        certificate public key: EC
*        subject: C=US,ST=California,L=Mountain View,O=Google LLC,CN=www.google.com
*        start date: Tue, 26 Mar 2019 13:38:23 GMT
*        expire date: Tue, 18 Jun 2019 13:24:00 GMT
*        issuer: C=US,O=Google Trust Services,CN=Google Internet Authority G3
* ALPN, server accepted to use http/1.1
> GET / HTTP/1.1
> Host: www.google.com
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
```

Protocole HTTP
utilisation

- Programmation d'un client HTTP (qui ne ferait pas d'affichage...ouf!) :

  Bibliothèque C : CURL

  https://curl.haxx.se/libcurl/c/example.html

  Beaucoup d'exemples notamment HTTPS

- Par défaut, Curl utilise les certificats du système hôte.

  Sur Linux ils sont stockés dans : /etc/ssl/certs/

Protocole HTTP
CURL

```c
/*  Simple HTTPS GET </DESC> */
#include <stdio.h>
#include <curl/curl.h>

int main(void) {
  CURL *curl;
  CURLcode res;
  curl_global_init(CURL_GLOBAL_DEFAULT);
  curl = curl_easy_init();
  if(curl) {
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
    // If you want to connect to a site who isn't using a certificate that is signed
    // by one of the certs in the CA bundle you have, you can skip this step
    curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, 0L);
    // If the site you're connecting to uses a different host name that what they have mentioned in
    // their server certificate's commonName (or subjectAltName) fields, libcurl will refuse to
    // connect. You can skip this check, but this will make the connection less secure.
    curl_easy_setopt(curl, CURLOPT_SSL_VERIFYHOST, 0L);
    res = curl_easy_perform(curl); /* Perform the request, res will get the return code */
    if(res != CURLE_OK)  /* Check for errors */
      fprintf(stderr, "curl_easy_perform() failed: %s\n",
              curl_easy_strerror(res));
    curl_easy_cleanup(curl);  /* always cleanup */
  }
  curl_global_cleanup();
  return 0;
}
```

Protocole HTTP
Autres utilisations

- git : logiciel de suivi de version pour le développements de projets
- apt : gestionnaire de paquets linux
- API REST (Representational state transfer)

  Architecture pour accéder à des services au dessus de HTTP
  - Exemple API CRUD : Create, Read, Update, Delete pour les bases de données

    tester par exemple *http://dummy.restapiexample.com/api/v1/employees*
  - réponse en Json *(cf. exemple suivant)*
- web service : ne veut pas dire grand-chose…
- API autres : services “HTTPisés” pour lutter contre l’ossification d’internet

Protocole HTTP
API REST exemple

Exemple d'interrogation pour connaître le cours en euro du bitcoin

```
curl -s https://api.coindesk.com/v1/bpi/currentprice.json


{"time":{"updated":"Apr 29, 2019 11:33:00 UTC","updatedISO":"2019-04-
   29T11:33:00+00:00","updateduk":"Apr 29, 2019 at 12:33 BST"},
"disclaimer":"This data was produced from the CoinDesk Bitcoin Price Index
   (USD). Non-USD currency data converted using hourly conversion rate from
   openexchangerates.org","chartName":"Bitcoin",
"bpi": {
   "EUR":{
     "code":"EUR",
     "symbol":"&euro;",
     "rate":"4,709.4788",
     "description":"Euro",
     "rate_float":4709.4788
     }
  }
}
```

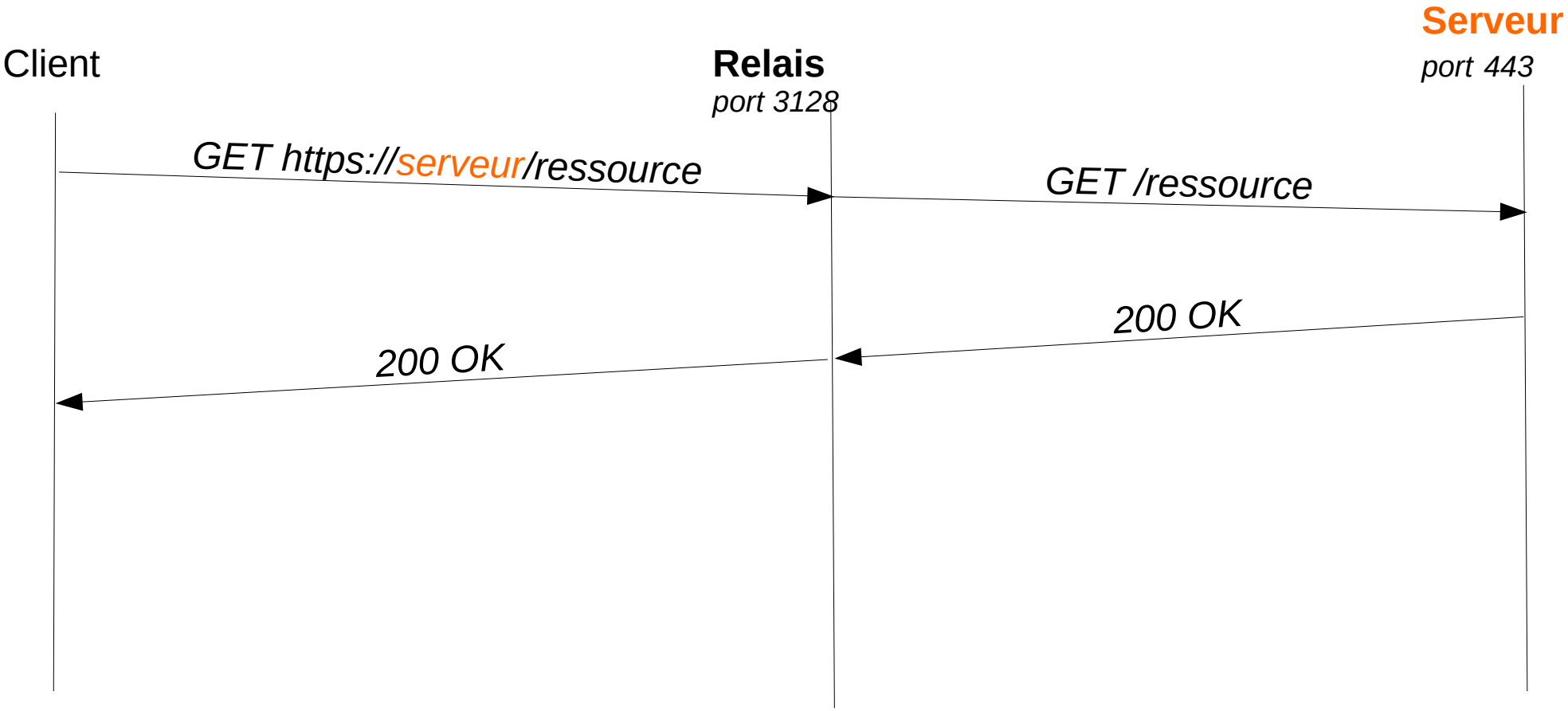
Protocole HTTP
Serveurs

Les serveurs web connus :

- Apache, l’historique
- IIS, Microsoft
- Nginx, le plus rapide...

- Mais aussi des classes de certains langages :
  - Java HttpServer class
  - Python http.server.HTTPServer

Relais HTTP


Client
Relais
port 3128
Serveur
port 443
GET https://serveur/ressource
GET /ressource
200 OK
200 OK

Protocole HTTP
Proxy

Utilités des proxies

- Relais pour éviter des parefeu.
  - Le relais est autorisé à passer le parefeu
- Caches.
  - Si plusieurs utilisateurs font la même requête…
  - peu d’intérêt si trafic essentiellement en HTTPS

Protocole HTTP
« HTTPisation » des protocoles

- De plus en plus de pare-feu
  - laissent passer seulement les ports 80 et 443
- Difficile de développer de nouveaux services : ossification d'internet
- Souvent dans un seul sens
- Exemple DNS sur TLS (port 853) pour éviter que le trafic DNS passe en clair
  - la censure de certains pays bloque le port 853
  - solution DoH (DNS sur HTTPS)

- Vers un tout sur le port 443 !

Sécurité

La sécurité des données consiste à garantir un ou plusieurs des objectifs suivants :

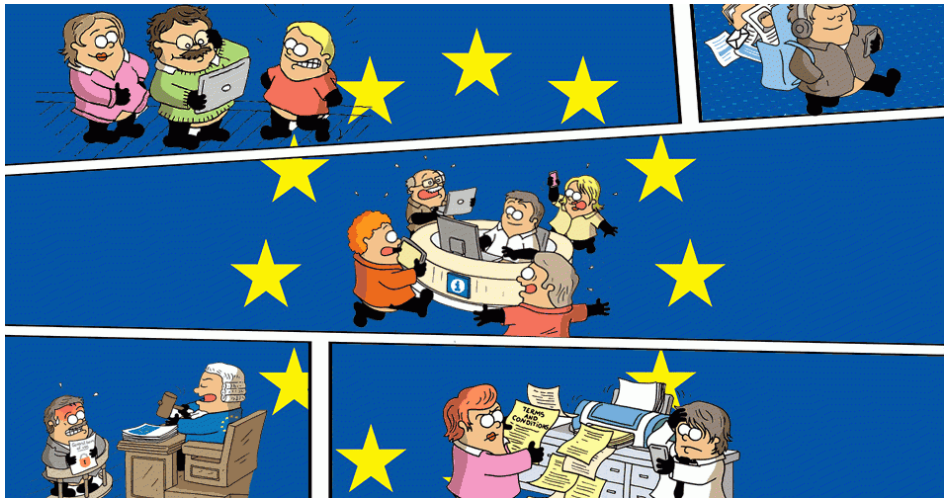
- L'**intégrité :** assurer que les données n'ont pas été altérées.
- La **confidentialité** : assurer que les données ne peuvent être lues que par des personnes autorisées.
- La **disponibilité** : assurer que les données sont accessibles 24h/24.
- La **non répudiation** : assurer que l'émetteur des données ne peut nier en être l'auteur.
- L'**authentification**, assurer que les parties ayant accès à l'information sont connues (au sens civil).

→ Le réseau n'est qu'un maillon de la chaîne !

Sécurité des données personnelles : un devoir des entreprises au niveau européen

Adoption par l'UE en avril 2016 de la RGPD : General Data Protection Regulation

- Principe de privacy by design :
  Les logiciels doivent intégrerla sécurité des données à la conception.
- le citoyen doit pouvoir récupérer les données qu'il a communiquées à une plate-forme.
- En cas de violation des droits,
  l'entreprise responsable encourt une sanction pouvant s'élever à 4% de son chiffre d'affaires mondial.



- Un citoyen peux demander à ce qu'un lien soit déréférencé d'un moteur de recherche ou qu'une information soit supprimée s'ils portent atteinte à sa vie privée.
- Les services en ligne doivent obtenir le consentement des parents des mineurs de moins de 16 ans avant leur inscription.
- En cas de problème, le citoyen s'adresse à l'autorité de protection des données de son pays, quelque soit le lieu d'implantation de l'entreprise qui traite mes données.

Sécurité

La sécurité du système d'information :

- c'est l'affaire de tous :
  - avoir des mots de passe solide
  - savoir reconnaître un mail de « phishing »
  - lutter contre les logiciels malveillants (venant d'une clé USB ou d'un site web…)
  - connaître les bonnes pratiques du paiement sécurisé.
  - avoir des systèmes d'exploitation et des logiciels à jour.

- Difficile de donner des chiffres, mais un coût réel pour l'entreprise. On parle d'une moyenne de 700 000 € par attaque et 9 semaines* pour s'en remettre :
  - perte d'image
  - frais juridiques
  - compensation clients….

** source NTT Com Security 2016*

Sécurité

C’est du **bon sens**, de la **rigueur** et des connaissances à jour.

En France, l’Agence nationale de la sécurité des systèmes d’information – **ANSSI** – accompagne les entreprises par des actions de conseil et de réglementation :

- formation des personnels
- mise en place de firewall
- méthodes de chiffrement
- alertes de sécurité (notamment systèmes…)
- mise en place de PSSI : politiques de sécurité des systèmes d’information

Chiffrement des données

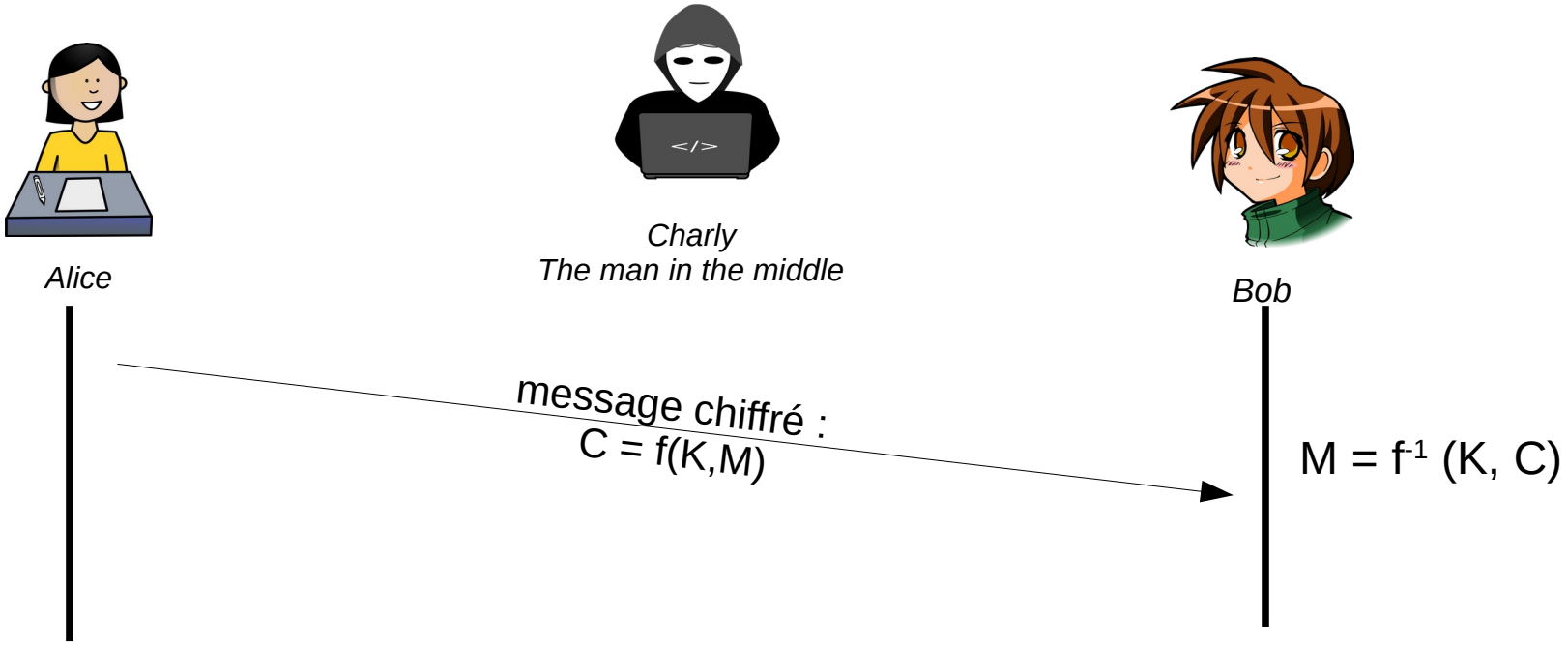
Pour sécuriser les données on a recourt au chiffrement (ou « cryptage »)
qui consiste à **transformer** les données.

L'opération de chiffrement peut présenter plusieurs caractéristiques :

- **inversible** ou non
- rapide ou nécessitant de longs temps de calculs
- **symétrique** (la même clé sert au codage et au décodage)
- ou **asymétrique** : existence de 2 clés.

Enigma

Chiffrement symétrique



- f : algorithme de chiffrement : en général connu
- M : message à chiffrer
- K : clé de chiffrement secrète connue uniquement de Alice et Bob
- C : message chiffré

Chiffrement symétrique

Exemple de code simple symétrique le codage par substitution :

Un symbole (un caractère par exemple) est substitué par un autre symbole selon une règle qui dépend d'une clé.

Exemple de règle simpliste : on remplace un caractère dont le code ASCII est N par le caractère de code ASCII $(N+C_i)\%256$ (appelé chiffrement de Vigenère)

C_i est un des code ASCII de la clé.

- Par exemple avec le message « Bonjour le monde » et la clé « 12345 » (on suppose que le code ASCII de 1 est 1, celui de 2 est 2... pour plus de simplicité.

```
  Bonjour le Monde
+ 1234512345123451
------------------
  Cqqntvt#pj!Orrif = Message chiffré.
```

- Le décodage est évident.

Chiffrement symétrique

Dans l'exemple précédent :

- Un symbole n'est pas toujours remplacé par le même symbole (il a fallu attendre 1853 pour casser le chiffre de Vigenère)
- L'algorithme est simple et donc ne demande que peu de ressource CPU

MAIS

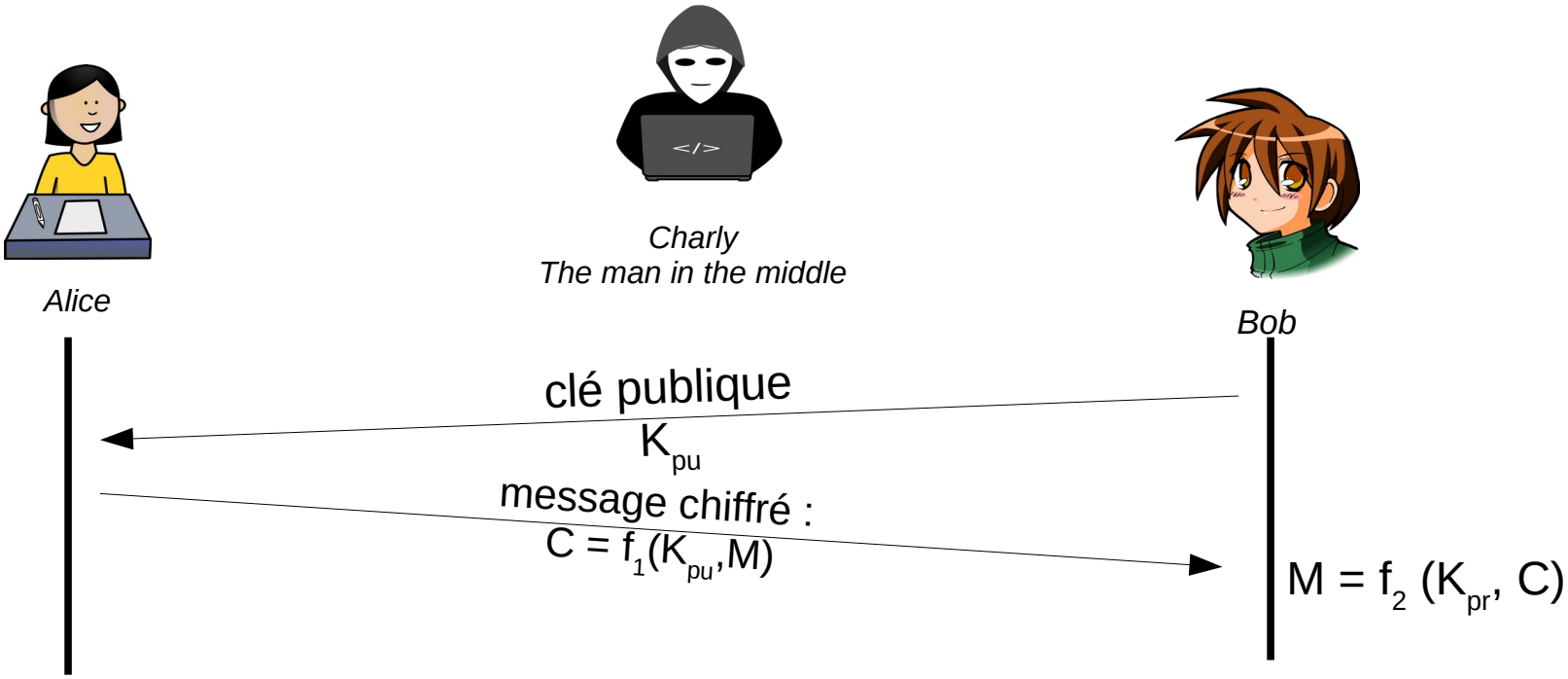
- Par attaque statistique, on peut retrouver la longueur de la clé : L' «espace» va produire des codes dont la fréquence sera élevée. Le « e » aussi…
- Puis connaissant la longueur, on peut faire une analyse statistique des fréquences d'apparition d'un code et supposer qu'il code telle lettre.

→ Des algorithmes symétriques **robustes** et **rapides** existes : **AES** avec des clés de 128 ou 256 bits (AES est approuvé par la NSA… euh, ça veut dire quoi?)

Remarque DES a été abandonné car sa clé de 56 bits était trop faible.

- Comment Alice et Bob se sont-ils échangés la clé ?
- Et si N machines doivent s'échanger des données, il faut N*(N-1)/2 clés…

Chiffrement asymétrique



- f_1: algorithme de chiffrement, f_2 algorithme de déchiffrement
- M : message à chiffrer
- K_{pu}: clé de chiffrement publique de Bob
- K_{pr}: clé de déchiffrement privée de Bob (inconnue des autres)
- C : message chiffré

Chiffrement asymétrique

Les algorithmes f_1 et f_2 sont en général connus.

- Connaitre K_{pu} ne révèle rien sur K_{pr}
- Algorithme le plus utilisé : RSA du nom des inventeurs Rivest Shamir Aldleman alors étudiants au MIT.
- Basé sur le fait qu'il est difficile de décomposer un grand nombre (2048 bits) en facteurs premiers surtout si ce grand nombre est le résultat de la multiplication de seulement 2 nombres premiers.
- Utilise le petit théorème de Fermat : *http://villemin.gerard.free.fr/Wwwgvmm/Decompos/DivisiFe.htm*

On admet que si p est premier $(n^p)\%p = n\%p$ On dit ausi que n^p est congru à n , modulo p.

Cela veut dire que $n^p = n + k.p$ ou encore $n.n^{p-1} = n + k.p$ ou encore $n.(n^{p-1} - 1) = k.p$

Donc si n et p sont premiers entre eux (n et p sont étrangers)

on déduit si dessus que $(n^{p-1} - 1)$ est divisible par p donc : $n^{p-1} - 1 = k_2.p$ noté

$$\mathbf{n^{p-1} \equiv 1 \bmod p}$$

Chiffrement asymétrique

Algorithme RSA :	Exemple
Choisir 2 nombres premiers	p = 11, q = 13
calculer le produit n	**n = p.q = 143**
calculer f = (p-1)(q-1)	f = 10*12 = 120
choisir un nombre e premier avec f	**e = 7** par exemple
calculer d tel que e.d=1 mod f	**d=103** car (7*103) mod 120 = 1
Diffuser n et e qui sont la clé publique	K_{pub}= (n,e) et K_{priv}= (n,d)
Découpons notre flux binaire à coder en nombres tous < n	Par exemple découpons notre flux binaire en mots de 7 bits. Supposons que le premier symbole du message à coder soit **x = 4**
Code $y = x^e \bmod n$	Code $y = 4^7 \bmod 143 = 16384 \bmod 143 =$ **82 = y**
A la réception calcul de $x = y^d \bmod n$	$x = 82^{103} \bmod 143 = (82 * (82*82)^{51}) \bmod 143$ $x = (82 * (6724 \bmod 143)^{51}) \bmod 143$ $x = (82 * 3^{51}) \bmod 143 = (82 * 27 * 3^{48}) \bmod 143$ $x = (82 * 27 * (3^{12} \bmod 143)^4) \bmod 143$ **= 4 = X**

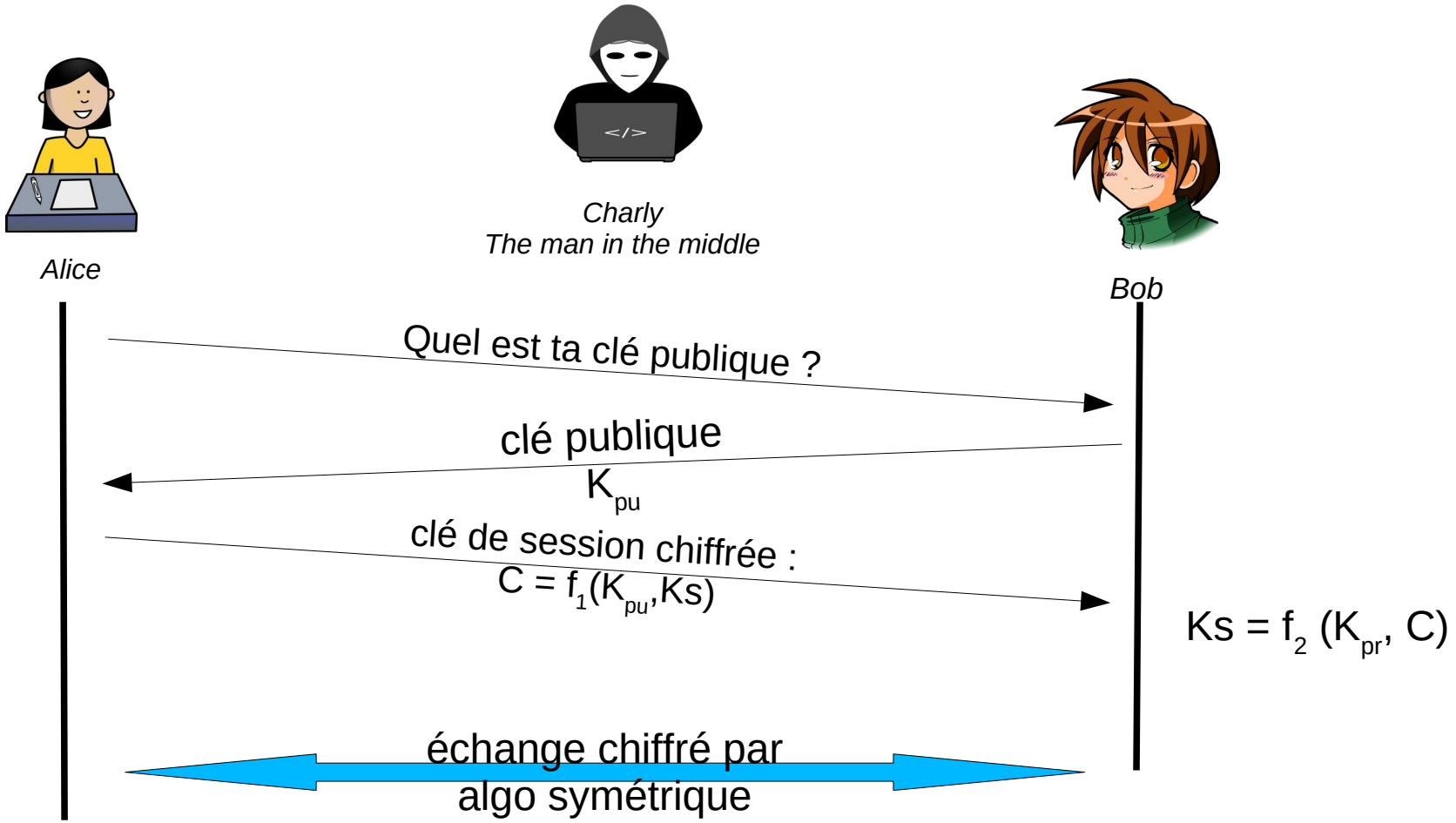
distributivité de l'opérateur modulo :
le modulo de la somme est le modulo de la somme des modulo
le modulo du produit est le modulo du produit des modulo

Chiffrement asymétrique

Démonstration

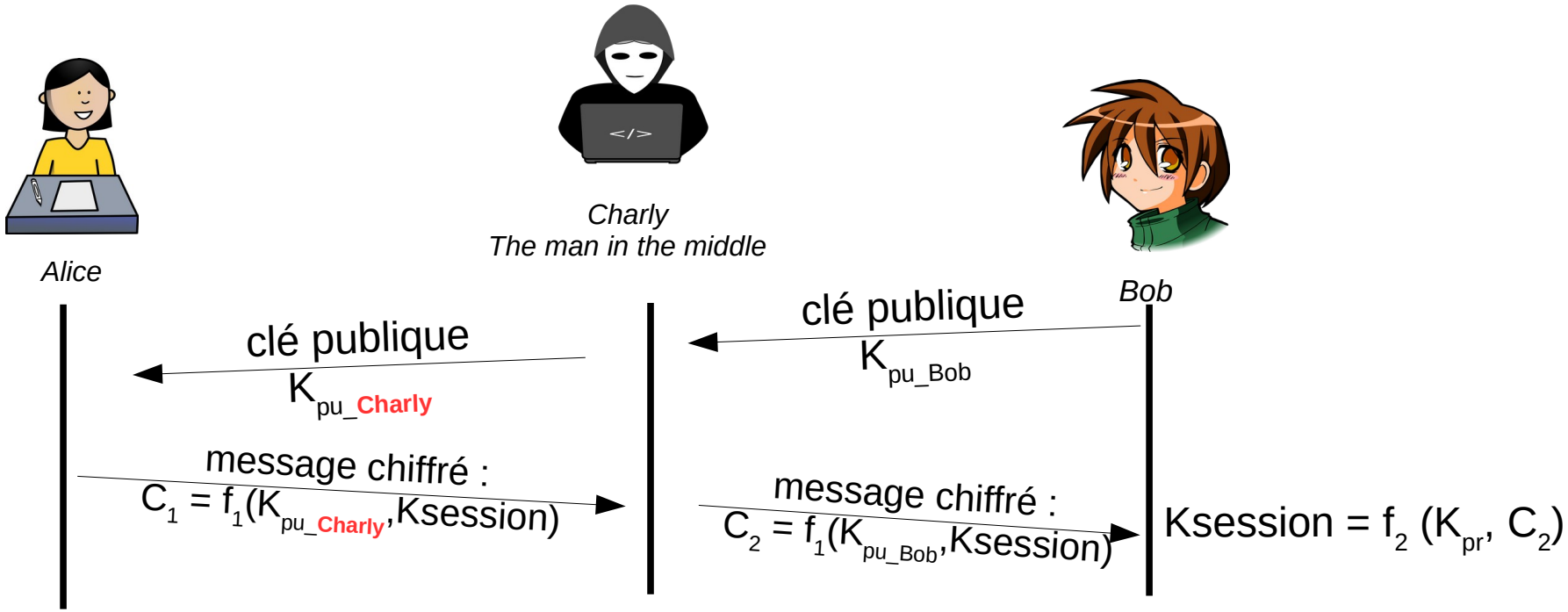
1) $f = (p-1)(q-1)$
2) $e.d = 1 \bmod f \rightarrow e.d = 1 + k.f \rightarrow e.d - 1 = k.f$
3) calculons $z = y^d \bmod n$ et prouvons qu'il est égal à x
4) $y = x^e \bmod n \rightarrow z = x^{e.d} \bmod n$
5) $z = x.x^{(e.d\ -1)} \bmod n$
6) de 2) on tire $z = x.x^{k.f} \bmod n = x.\ x^{kf} \bmod n = x\ .\ k_1{}^f \bmod n$
   $z = x.k_1{}^{(p-1)(q-1)} \bmod n$
7) petit théo de Fermat comme $k_1{}^{(p-1)(q-1)} \bmod n = 1 \bmod n \rightarrow z = x$

Chiffrement asymétrique + clé de session symétrique



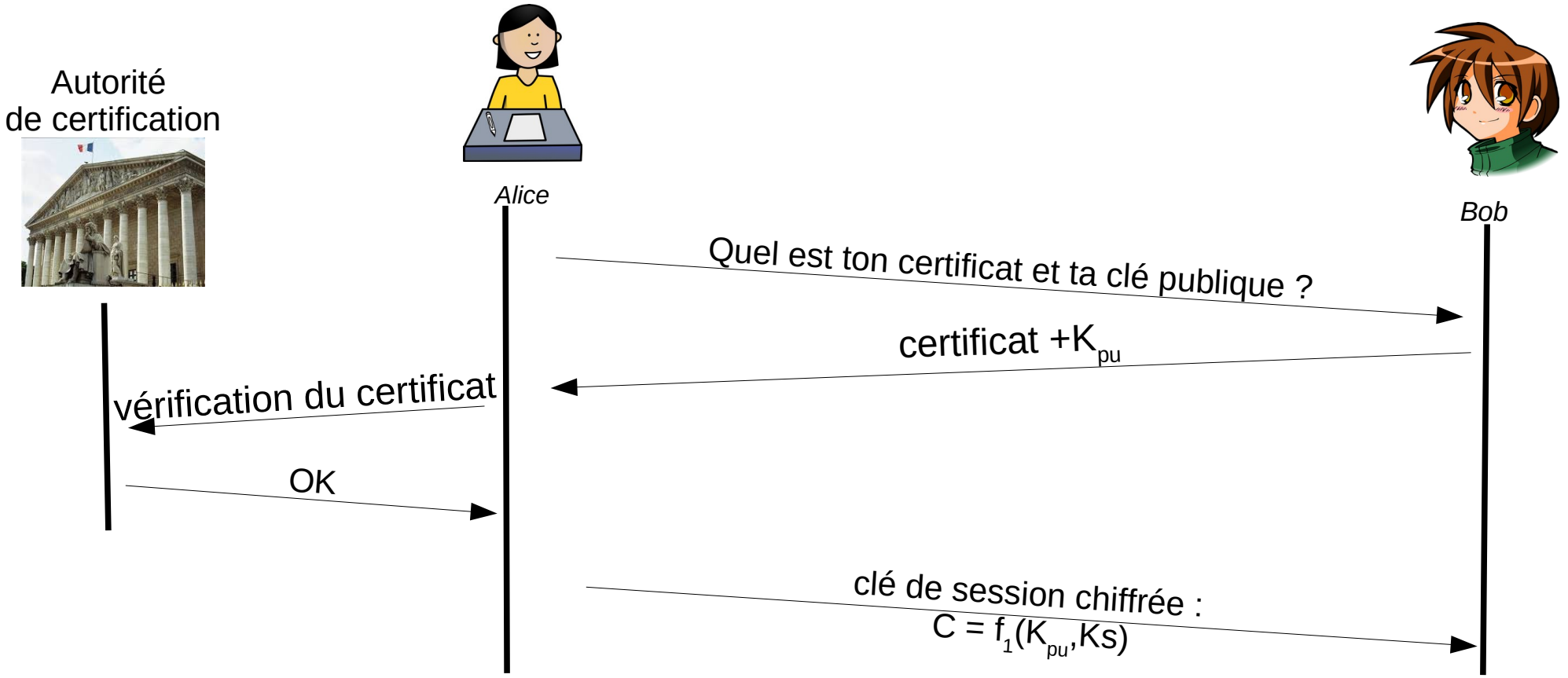
- Le chiffrement asymétrique est lent : mieux vaut utiliser un chiffrement symétrique dont la clé sera échangée par cheffrement asymétrique

Certificats



- Charly vient d'intercepter la clé de session entre Bob et Alice… il va pouvoir tranquillement écouter la communication !
- Comment Alice peut-elle être sure qu'elle vient de recevoir la clé de Bob ?
- En plus, Bob ne s'en rend pas compte….

Certificats



– Comment Alice sait-elle qu'elle dialogue avec l'autorité de certification ?

Autorité de certification connue : Thawte, Verisign...

Certificats

Structure des certificats X.509 de l'UIT :

- Version de X.509 à laquelle le certificat correspond ;
- Numéro de série du certificat ;
- Algorithme de chiffrement utilisé pour signer le certificat ;
- Nom (Distinguished Name) de l'autorité de certification ;
- Date de début de validité du certificat ;
- Date de fin de validité du certificat ;
- Objet de l'utilisation de la clé publique ;
- Clé publique du propriétaire du certificat ;
- Signature de l'émetteur du certificat.

Le navigateur (firefox…) connaît au préalable (nativement) la clé publique de l'autorité de certification

encore un problème !

La confidentialité persistante ou comment garantir que les échanges entre Alice et Bob seront toujours protégés même si

- si la clé privé de Bob (*cf. transparent pécédent*) est récupérée plus tard (perquisition, torture…)
- si les échanges entre Alice et bob ont été enregistrés (ce n'est pas de la Sci-Fi… *cf. Snowden*)

- Solution :

  → Il faut que même si l'on découvre la clé privée de Bob ou d'Alice on ne puisse déchiffrer les transactions passées. Ceci peut être assuré par un échange de clé de session. Il faudra pour déchiffrer, les clés privées des 2 protagonistes. On peut même se protéger de cela en tirant de nouveaux couples de clés asymétriques juste pour l'échange et signer ces nouvelles clés avec la clé officielle.

  → une autre solution consiste à utiliser l'échange de clés Diffie-Hellman (prix Turing 2015)

  Basé sur la difficulté algorithmique à calculer des logarithmes discrets.

  cf. http://images.math.cnrs.fr/Le-probleme-du-logarithme-discret-en-cryptographie.html

Échange de clés Diffie-Hellman

Exemple : Source wikipedia

- Alice et Bob ont choisi un nombre premier p et une base g. Dans notre exemple, p=23 et g=3
- Alice choisit un nombre secret a=6
- Elle envoie à Bob la valeur A = ga [mod p] = 36 [23] = 16
- Bob choisit à son tour un nombre secret b=15
- Bob envoie à Alice la valeur B = gb [mod p] = 315 [23] = 12
- Alice peut maintenant calculer **la clé secrète** : Ba [mod p] = 126 [23] = 9
- Bob fait de même et obtient la **même clé qu'Alice** : Ab [mod p] = 1615 [23] = 9

Il est difficile de retrouver a et b en connaissant A,B, p et g si p et g sont de grandes tailles.

Attention aux attaques type man in the middle. En effet, Si Charly remplace g^a par $g^{a'}$ Charly pourra se faire passer pour Alice… Et de même dans le l'autre sens.

Il suffit pour éviter cela de signer les échanges par RSA.

Signature numérique

- permet de garantir l'authenticité de l'expéditeur ;
- permet de vérifier l'intégrité du message reçu ;
- Basé sur les fonctions de hachage :
  - l’émetteur calcul un condensé du message (sorte de résumé)
  - l’émetteur chiffre avec sa clé privée le condensé
  - Le récepteur décode avec la clé publique le condensé.
  - Le récepteur calcul le condensé du message reçu et le compare au condensé chiffré reçu.
- Algo de hachage : SHA-256
- Intérêt du condensé : plus rapide que de chiffrer l’intégralité du message. En effet, le chiffrement utilisé est asymétrique.

Sécurité

Algorithmes reconnus sûrs

- SHA-256, SHA-512 ou SHA-3 comme fonction de hachage ;
- HMAC utilisant SHA-256, bcrypt, scrypt ou PBKDF2 **pour stocker les mots de passe** (dans une base de données, on ne stocke jamais les mots de passe ; seulement les signatures) ;
- AES ou AES-CBC pour le chiffrement symétrique ;
- RSA-OAEP comme défini dans PKCS#1 v2.1 pour le chiffrement asymétrique ;
- enfin, pour les signatures, RSA-SSA-PSS comme spécifié dans PKCS#1 v2.1.
- Utiliser les tailles de clés suffisantes pour AES il est recommandé d'utiliser des clés de 128 bits et, pour les algorithmes basés sur RSA, des modules et exposants secrets d'au moins 2048 bits ou 3072 bits, avec des exposants publics, pour le chiffrement, supérieurs à 65536.
- Enfin **ne plus utiliser** les chiffrements DES et 3DES ou les fonctions de hachage MD5 et SHA1, devenus obsolètes

Sécurité

Mise en œuvre :

- SSL v2 (1995, pas de v1) , v3

  puis remplacé par

- TLS 1.0 (1999), 1.1, 1.2, 1.3 (2018)

Développement :

- https://openweb.eu.org/articles/https-de-ssl-a-tls-1-3
- https://wiki.openssl.org/index.php/Simple_TLS_Server
- https://wiki.openssl.org/index.php/SSL/TLS_Client
- curl : https://curl.haxx.se/libcurl/c/https.html

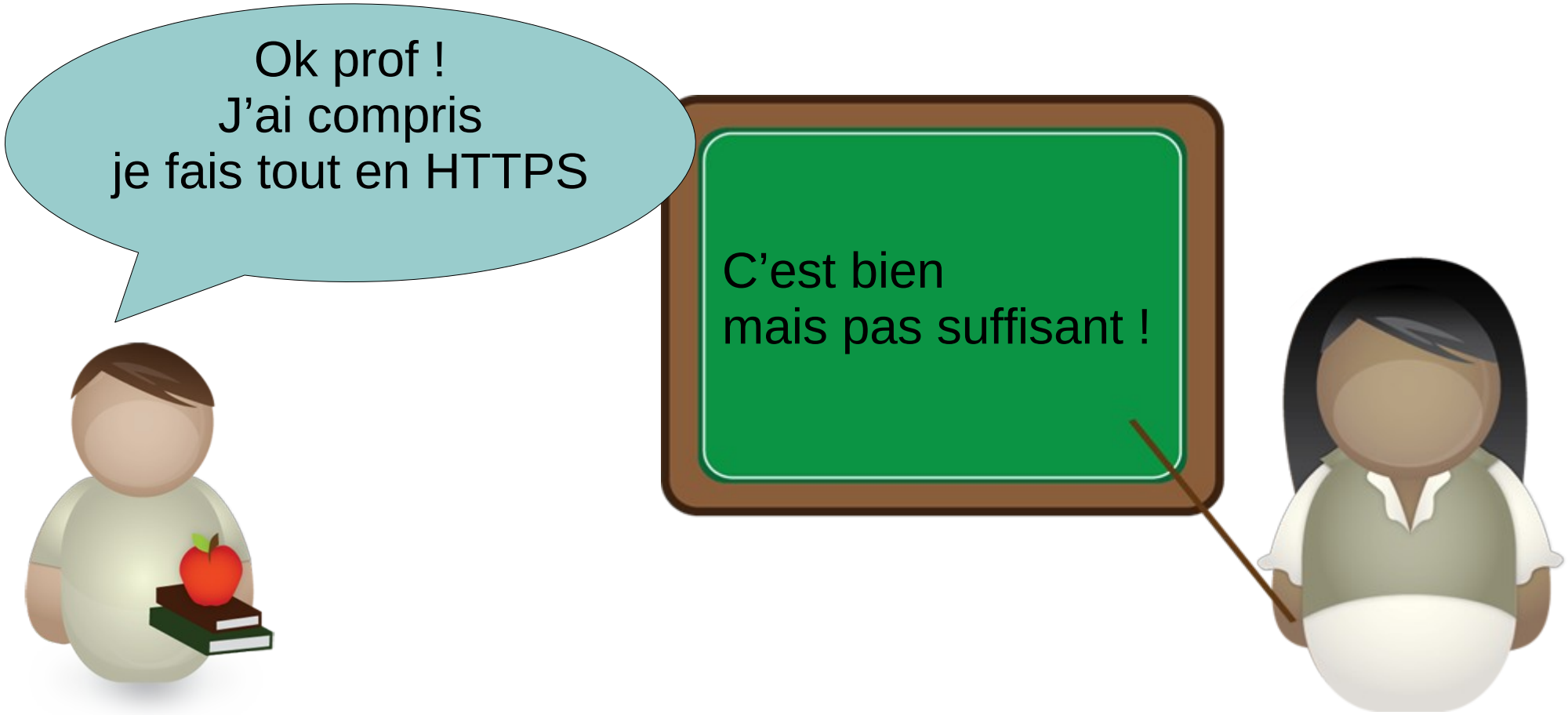
Sécurité

TLS 1.3 (août 2018 / RFC 8446)

- **sécurité persistante**
- Bien sûr si l'algorithme de chiffrement est cassé dans le futur, rien n'empêchera de décoder une transaction passée qui aura été mémorisée (c'est ce qu'espèrent les états…)
- La sécurité persistante était présente dans TLS 1.2, mais optionnelle et chiffrée avec des clés faibles…
- De nombreux lobbies ont essayé de combattre TLS 1.3

- Programmation sous Linux avec libssh. Excellent tuto ici : http://api.libssh.org/master/libssh_tutor_guided_tour.html

Sécurité


Ok prof !
J'ai compris
je fais tout en HTTPS
C'est bien
mais pas suffisant !

Sécurité

Pourquoi utiliser HTTPS n’est pas suffisant :

- Le site visité utilise-t-il HTTPS sur TLS 1.0 qui est connu comme vulnérable ?
- Même si le contenu est chiffré, l’ @IP est elle chiffrée ?
- Les requêtes DNS sont-elles chiffrées ?
- Le traffic est-il une signature du site visité ? autrement dit, puis-je en analysant la quantité et la fréquence des échanges deviner de quelle page il s’agit ?
  - En effet , TLS ne fait pas de remplissage par défaut…
  - le nombre de lien qu’une page possède est sa signature…

→ Le protocole QUIC (Quick UDP Internet Connections) essaye d’apporter des solutions… mais n’est encore utilisé que dans Chrome (bien que Google ne semble pas l’utiliser encore…)

Internet version 6

- les 2 **inconvénients de IPV4** :
  - **Manque de qualité de service en natif**

    Il existe bien DiffServ, RTP/RTCP/RSVP en ipv4, mais pas implémenté partout !
  - **pénurie des adresses IPv4**

    En IPv6 adresse sur 128 bits → $3{,}4.10^{38}$ adresses !

    Surface de la Terre 5.10^{20} mm²

    → IP v6 : RFC 2460

Internet version 6

IPV6 c’est :

- **En-tête simplifiée** (efficacité accrue pour le routage) : les éventuelles options viennent s’intercaler entre l’en-tête et les données de la couche supérieure (transport).
  - Les options peuvent être simplement ignorées par les routeurs.
  - Les options peuvent être plus élaborées qu’en IPv4.
- Qualité de Service (**QoS**) prévue.
- Authentification, intégrité et confidentialité **(IPSec**).
- **Plus de fragmentation** par les routers intermédiaires
- **Plus de calcul de checksum** de l’en-tête
- Mise à jour importante du protocole **ICMP** et mise à jour mineure des protocoles de niveau 4 **UDP** et **TCP**.

Internet version 6

La trame IPv6 (taille minimum de l'entête : 40 octets)

N° de version (4b)	Champ differentiate service (8 b)	Indentificateur de flot (20b)	
Longueur des données (16 b)		Entête prochain (8 b)	nombre de sauts (8b)
Adresse émetteur (128 b)			
Adresse destinataire (128 b)			
Entête suivant s'il existe			
Données			

- Par rapport à IPv4, les champs CRC, flags et Frag offset ont été supprimés.

Internet version 6

- Le type de l'entête suivant est précisé par le champs «entête prochain». Ce champ existe afin de de **chaîner des informations supplémentaires** (informations de routage, QoS...). Le numéro du protocole transporté en couche 4 (06 pour TCP par exemple) est transporté par le dernier entete.
- L'identificateur de flot sert à indiquer au routeur que le paquet fait partie d'un flux qui doit avoir un **traitement spécial** (QoS particulière).
- Nombre de sauts : remplace le **TTL** ipv4
- Le champ differentiate service indique, comme le **TOS** d'IPv4, une classe de service et une priorité.
- Ipv6 intègre de façon native des possibilités **d'authentification ainsi que du chiffrement.**

Internet version 6

- En Ipv6 on utilise **la plus petite taille** de la taille maximale des paquets (MTU) transportables par tous les réseaux traversés.
- En effet, si on considère que le paquet doit être intégralement reçu pour être renvoyé, un petit paquet mettra moins de temps à traverser un routeur qu'un gros et l**e temps d'acheminement d'un ensemble de petits paquets sera moins long que celui d'un gros** (c'est aussi pour cela que les cellules d'ATM sont si petites).
- Le MTU de départ est celui du **réseau d'origine**, par exemple 1500 octets si ethernet
- Si le paquet doit traverser un réseau de MTU plus petit, un message **ICMPv6 "too big packet"** est envoyé à l'émetteur.

Internet version 6

ICMPv6

- Détection d'erreurs ;
- tests de liaison (ping) ;
- Configuration automatique des équipements ;
- Gestion des groupes de multicast ;
- ICMPv6 reprend les fonctions du protocole ARP.

Internet version 6

Adresse ipv6

- notation en groupe de 4 chiffres hexadécimaux séparés par :
  - ex, 2001:0db8:0000:0000:0008:0800:200c:417a
- on simplifie la première plus longue suite de zéro par ::
  - ex, 2001:0db8::0008:0800:200c:417a
- on omet les zéros de début de chaque groupe
  - ex, 2001:db8::8:800:200c:417a

  cette forme est appelée la forme canonique si elle est notée en minuscule.
- Netmask : uniquement en notation CIDR.
  - ex, 2001:db8::1 / 64
- Dans un navigateur, utilisation des crochets pour lever l'ambiguïté sur le numéro de port : *http://[2001:1234:12::1]:8080*

Internet version 6

Adressage IPv6

64 bits : Network ID	64 bits : interface ID

- Network ID est découpé lui même en différentes parties, permettant de numéroter le FAI et le site
- Exemple :
  - RIR RIPE-NCC (europe) ⇒ 2001:0600:: / 23
  - Renater ⇒ 2001:0660::/32
  - Ensicaen ⇒ 2001:0660:7105::/48
  - Reste 16 bits à l'Ensicaen pour créer des sous réseaux

Internet version 6

Préfixes réservés

- : : 1/128 Adresse de loopback
- 2000 ::/3 Toutes les adresses Unicast de l'Internet mondial
- fe80::/10 Adresse de lien local
- fec0::/10 Adresse de site local (déprécié)
- ff00::/8 Addresses de multicast
- ::ffff:0:0/96 Adresses mappées sur IPv4
- 2002::/96 préfixe de d'adresses pour le routage 6to4

Internet version 6

Répartition des adresses unicast

- http://www.iana.org/assignments/ipv6-unicast-address-assignments

Voici les premières allocations pour les Regional Internet Registry (RIR)

- 2001:0000::/23 IANA (Internet mondial)
- 2001:0200::/23 APNIC (Asie)
- 2001:0400::/23 ARIN (Amérique du nord)
- 2001:0600::/23 RIPE NCC (Europe)
- 2001:1200::/23 LACNIC (Amérique du Sud)
- 2001:4200::/23 AFRINIC (Afrique)
- ...
- 2A00 :0000::/12 RIPE NCC (Europe)

Internet version 6

Exemple sur une interface Linux

- Une interface réseau IPv6 possède généralement plusieurs adresses

```
eth1 Link encap:Ethernet HWaddr 00:30:48:2E:3D:7D
  inet addr:193.49.200.59 Bcast:193.49.200.255 \
  Mask:255.255.255.0
  inet6 addr: 2001:660:7105::10/64 Scope:Global
  inet6 addr: fe80::230:48ff:fe2e:3d7d/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  ...
```

Internet version 6

adresses EUI-64

- Construction automatique d'une @ IPv6 quand un machine connaît le réseau sur lequel elle es située
- Construction des 64 derniers bits de l'adresse à l'aide de l'adresse MAC
(3 octets de poids forts MAC) | (0x020000)	FF FE	3 octets de poids faibles MAC
- Exemple adresse réseau : 2002:1::/64

  adresse MAC 00:11:22:33:44:55

  → @IP EUI-64 construite : 2002:1:: 0211 : 22FF : FE33 : 4455
- N'est plus utilisé par Linux, pour éviter le "traçage" des appareils par leurs adresses MAC. En effet dans le cas EUI-64 l'@ MAC devient visible des correspondants.

Internet version 6

Mappage adresse IPv4 → IPv6

- Utilisation du préfixe ::ffff:0:0/96 (IPv4-mapped prefix)
- Par exemple 192.168.60.1 est mappé en

  ::ffff:192.168.60.1

  soit ::ffff:c0a8:4401 en notation canonique

- Exemple d'utilisation : un serveur qui travaille en interne uniquement avec des adresses IPv6 mais qui écoute sur un réseau IPv4 et IPv6.
- Le mappage IPv4 ⇒ IPv6 est effectué en entrée par la pile TCP/IP du serveur (resp. IPv6 ⇒ IPv4 en sortie) si la requête vient d'un client IPv4.

Internet version 6

Autoconfiguration d'une interface

- Soit utilisation classique de DHCPv6 [RFC3315]
- Soit récupération du préfixe du site à partir d'un routeur du lien et construction de sa propre adresse (utilisation d'ICMPv6)
- lien local (FE80::/64) avec vérification de l'unicité

Remarque : l'autoconfiguration via ICMPv6 ne permet pas de récupérer l'@ du DNS. DHCPv6 est donc quasi obligatoire.

Internet version 6

Attribution automatique

1) Découvrir un **préfixe**, un **netmask** et une **passerelle par défaut** grâce à un routeur

   → Trame ICMPv6 type 133 (Sollicitation d'un routeur) @ff02::2

   ← Trame ICMPv6 type 134 (information du routeur)

2) construire automatiquement une adresse d'interface soit par EUI-64, soit aléatoirement.

3) Vérification d'unicité.

   → Trame ICMPv6 type 135 (Sollicitation d'un voisin) @ff02::1

   ← ICMPv6 type 136 (Annonce d'un voisin).

   Sans réponse, l'adresse est libre.

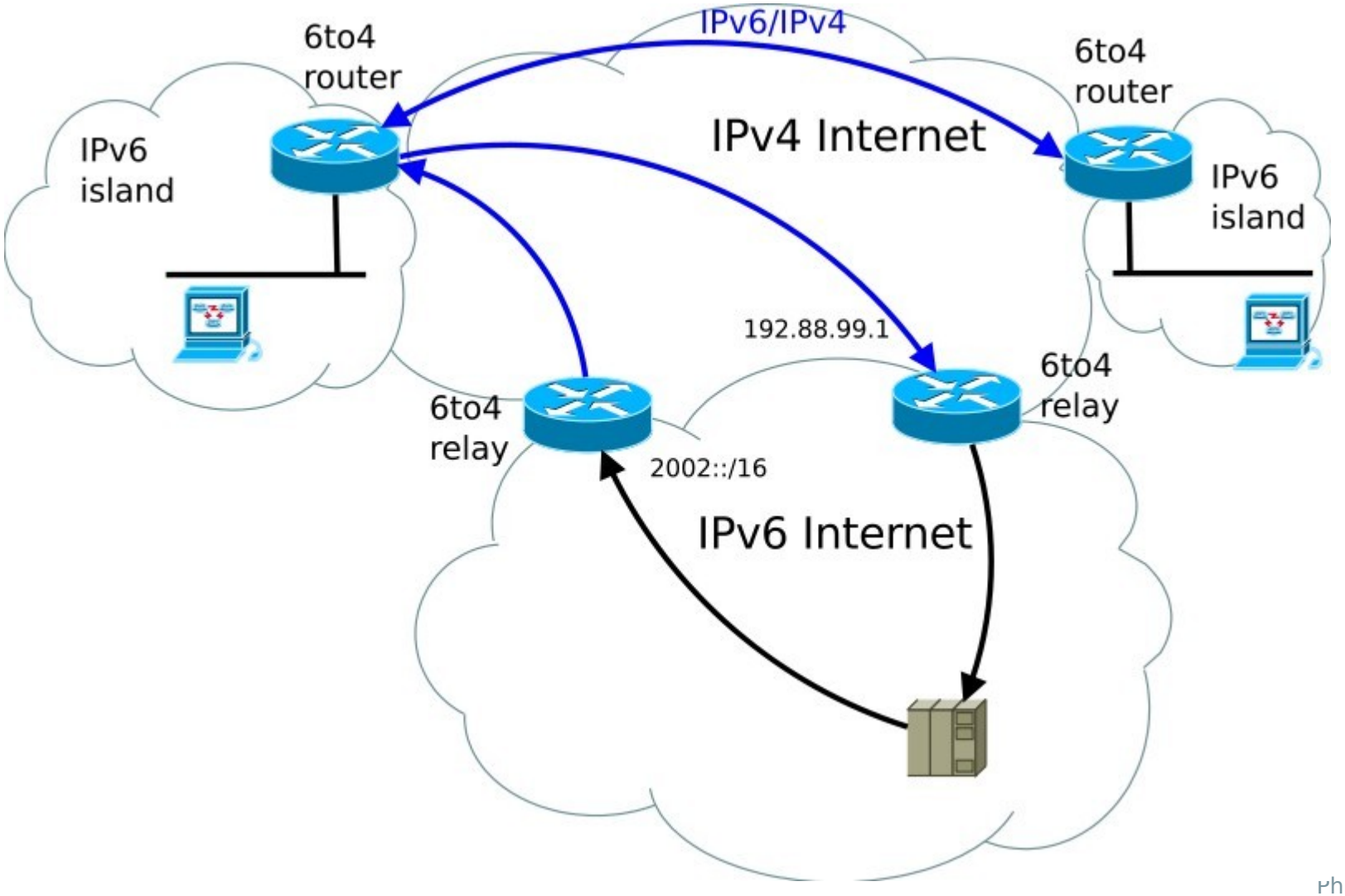
Internet version 6

Transition ipv4 vers ipv6

- Plusieurs mécanismes existent. Le plus fréquent est 6to4 et plus récemment 6to4rd
- ne permet pas de faire le lien entre des hôtes ipv6 pures et des hôtes ipv4 pures. Pour faire cela, utiliser NAT64 *(cf. TP)*
- Des mécanismes de tunnel peuvent être également employés
- 6to4 permet de relier des hôtes ipv6 isolés à des réseaux Ipv6 en utilisant au milieu le réseau ipv4.

Internet version 6

Transition ipv4 vers ipv6 (6to4)



Internet version 6

Transition ipv4 vers ipv6 (6to4)

- L'hôte isolé reçoit un préfixe ipv6 particulier puisqu'il n'est pas connecté au « monde ipv6 »
- Ce préfixe est envoyé par le routeur 6to4 et construit en accolant le préfixe 2002 :: à l'adresse ipv4 publique du routeur.
  - Par exemple si le routeur a comme adresse IP 193.49.200.204, l'hôte reçoit 2002 :C131:C8CC ::/48
  - puis l'hôte construit normalement son adresse ipv6 par autoconfiguration
- L'hôte envoie normalement ses paquets ipv6 au routeur 6to4.
- Celui-ci encapsule les paquets ipv6 dans des paquets ipv4 dont le champ protocole transporté sera égale à 41.

Internet version 6

Transition ipv4 vers ipv6 (6to4)

2 cas se présentent alors :

1) La destination est un hôte isolé. Dans ce cas le routeur 6to4 déduit l'@ ipv4 du routeur 6to4 du correspondant par le préfixe ipv6 de l'@ destination.

2) la destination est le réseau Ipv6 natif. Dans ce cas, le routeur 6to4 envoie son paquet à un relai v4/v6 dont il connaît l'@ ipv4.

Cependant, pour simplifier les configurations, si le réseau du FAI héberge un tel relais, le FAI a configuré son relai avec l'@ 192.88.99.1. qui est donc le routeur par défaut du routeur 6to4

Internet version 6

Transition ipv4 vers ipv6 (6to4)

Cependant, en voyageant dans le monde ipv6, le paquet sort du réseau du FAI et la réponse peut revenir par le réseau Ipv4 d'un autre FAI n'ayant pas ou mal implémenté 6to4.

- Il faut donc obliger le paquet à revenir dans le monde par le relais 6to4 d'origine.
- Pour cela il suffit au FAI non pas d'utiliser les adresses en 2002 :: mais d'utiliser son propre préfixe Ipv6 et d'y accoler l'adresse IP du routeur 6to4. Ainsi le paquet reviendra au relai ipv6 du FAI.

crédits

- https://www.ssi.gouv.fr
- https://www.cnil.fr/fr
- https://openclipart.org/
- Sébastien Fourey / Freddy Levée – Ensicaen