Networking 101

Réseaux TCP/IP

Hugo Blanc -

Sommaire

Contenu

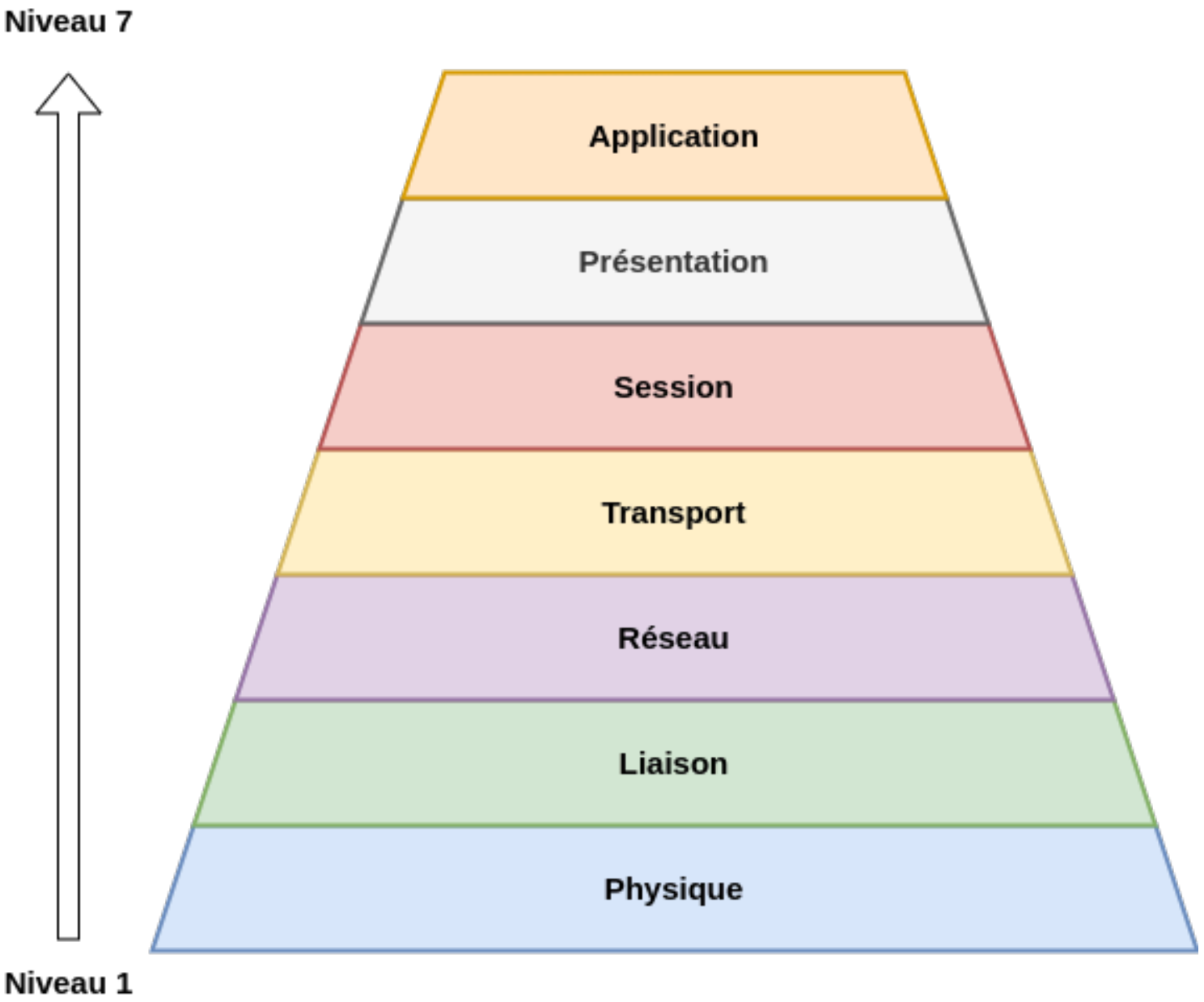
Fondamentaux de TCP/IP & Ethernet

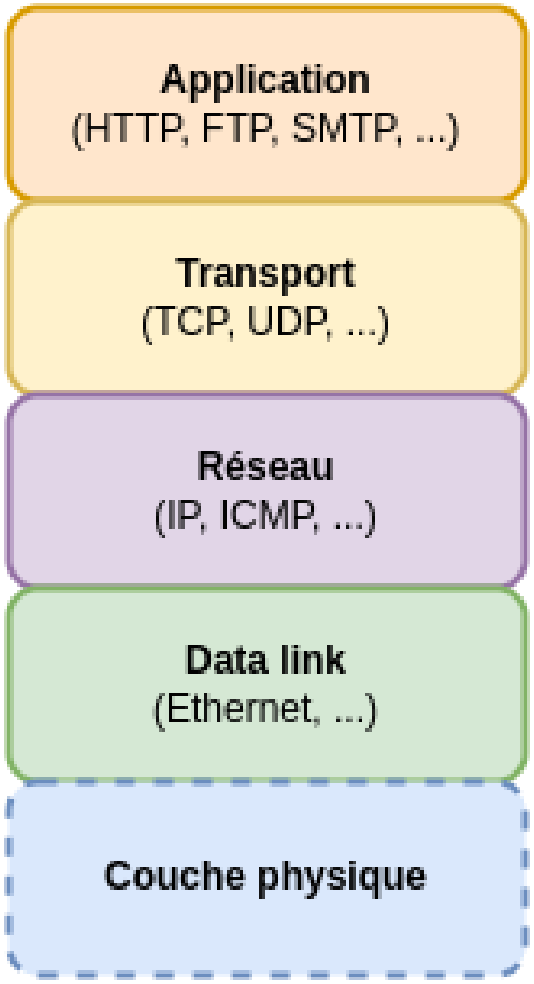
- Adressage
- Routage
- Ethernet: description, équipements, VLANs…
- Pile IP: ARP, IP, ICMP, UDP, TCP…

Réseau TCP/IP: bases

Modèle en 7 couches qui décrit comment les systèmes communiquent sur un réseau, chaque couche ayant un rôle spécifique, de la transmission physique des données à leur application finale.

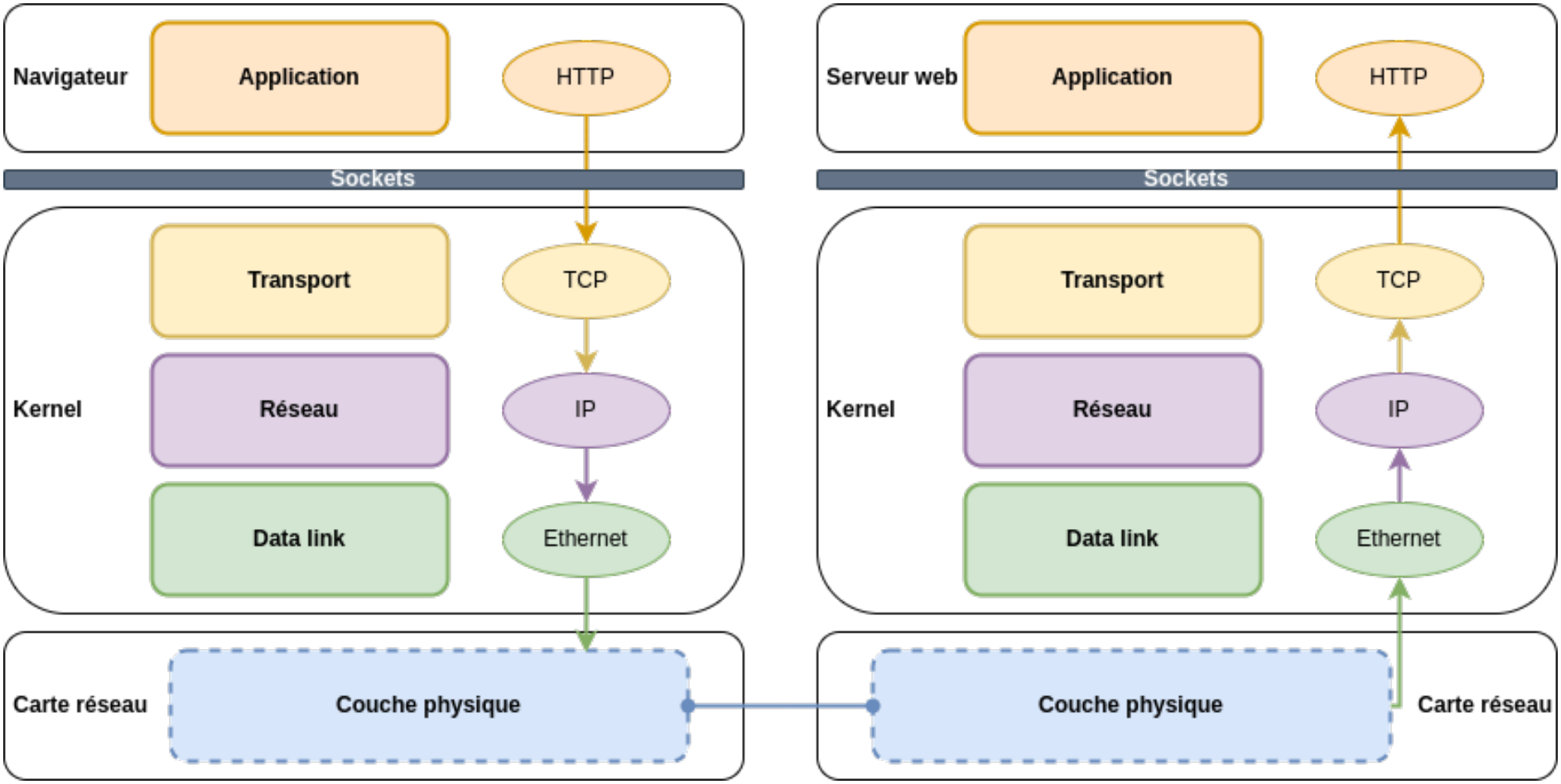
Réseau TCP/IP, modèle OSI



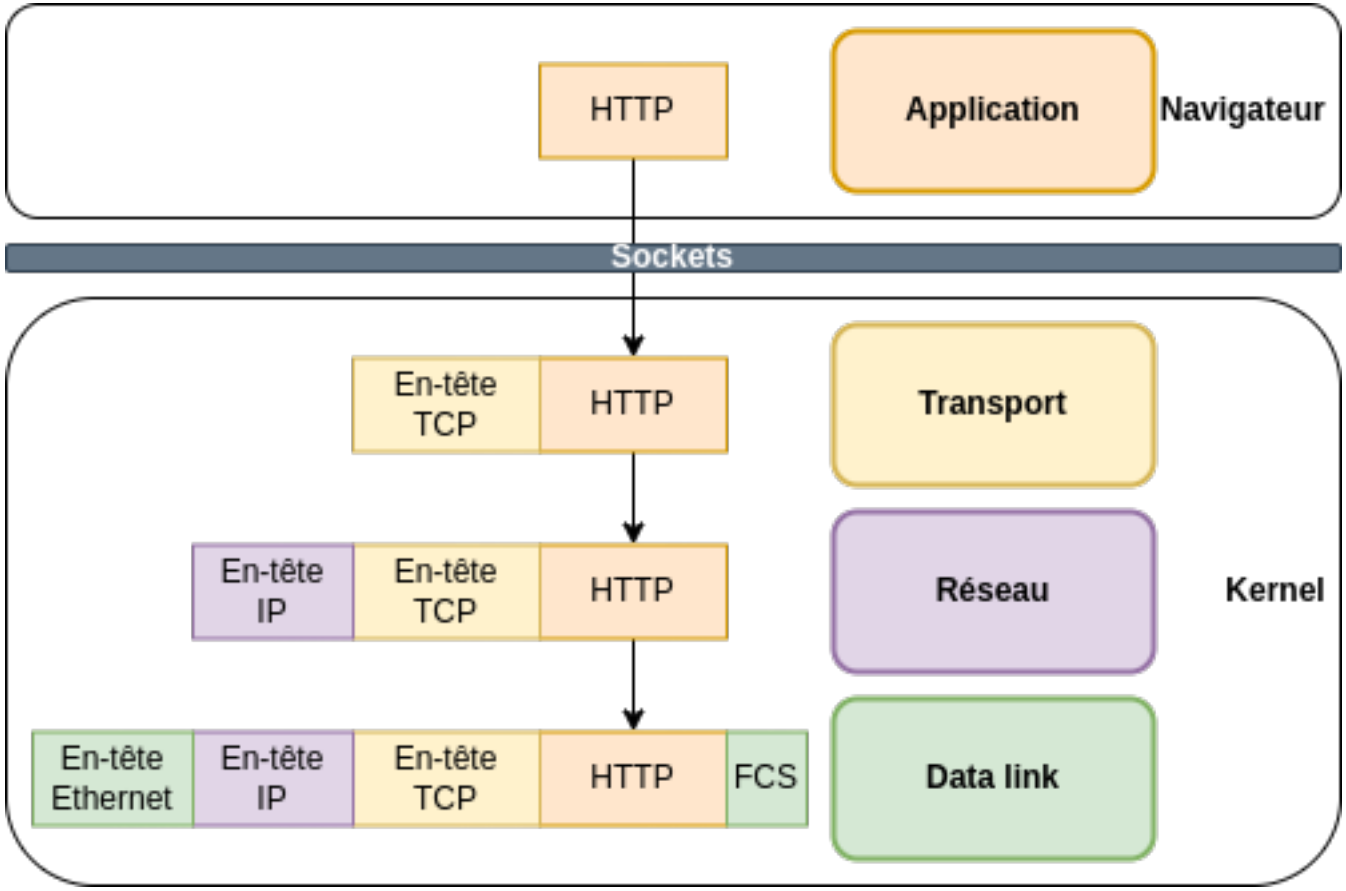


- Les réseaux sont généralement organisés en “piles protocolaires”
- chaque couche de la pile offre un niveau d’abstraction supplémentaire à la couche supérieure
- chaque couche offre un service supplémentaire par rapport à la couche inférieure
- les protocoles de la pile TCP/IP sont documentés dans les RFC (*Request For Comments*)

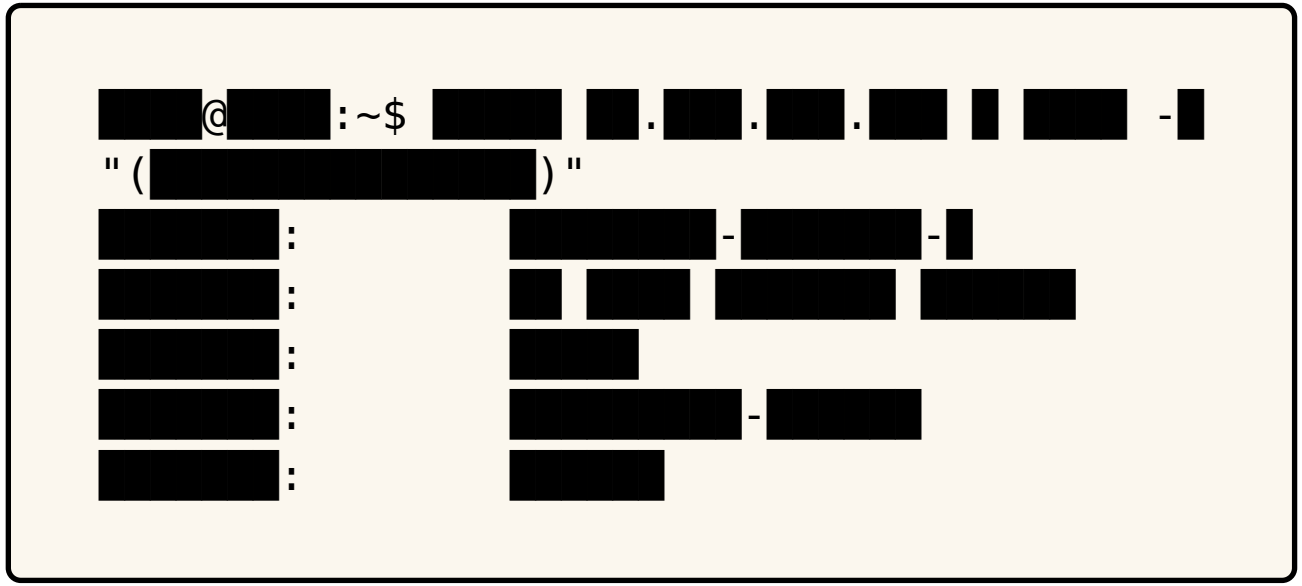
Communication navigateur ⟺ serveur web



Encapsulation (et désencapsulation) des trames



- 4 octets (32 bits)
- 0.0.0.0 $\Longleftrightarrow$ 255.255.255.255
- 2^{32} adresses possibles (~ 4.3 milliards)
- Adresses publiques attribuées par l'ICANN via différents RIRs: (RIPE pour l'Europe, ARIN pour l'Amérique du nord, ...)



Plages d'adresses privées d'usage libre (RFC1918):

- 10.0.0.0/8: 10.0.0.0 ⟺ 10.255.255.255
- 172.16.0.0/12: 172.16.0.0 ⟺ 172.31.255.255
- 192.168.0.0/16: 192.168.0.0 ⟺ 192.168.255.255

- Voir les adresses

```
ip addr show [dev device]
```

- Affecter une adresse avec ip

```
ip addr add <adresse/msk> dev <device>
```

(en cas d'appels multiples, `ip` ajoute des alias à l'interface)

- Supprimer une adresse

```
ip addr del <adresse/msk> dev <device>
```

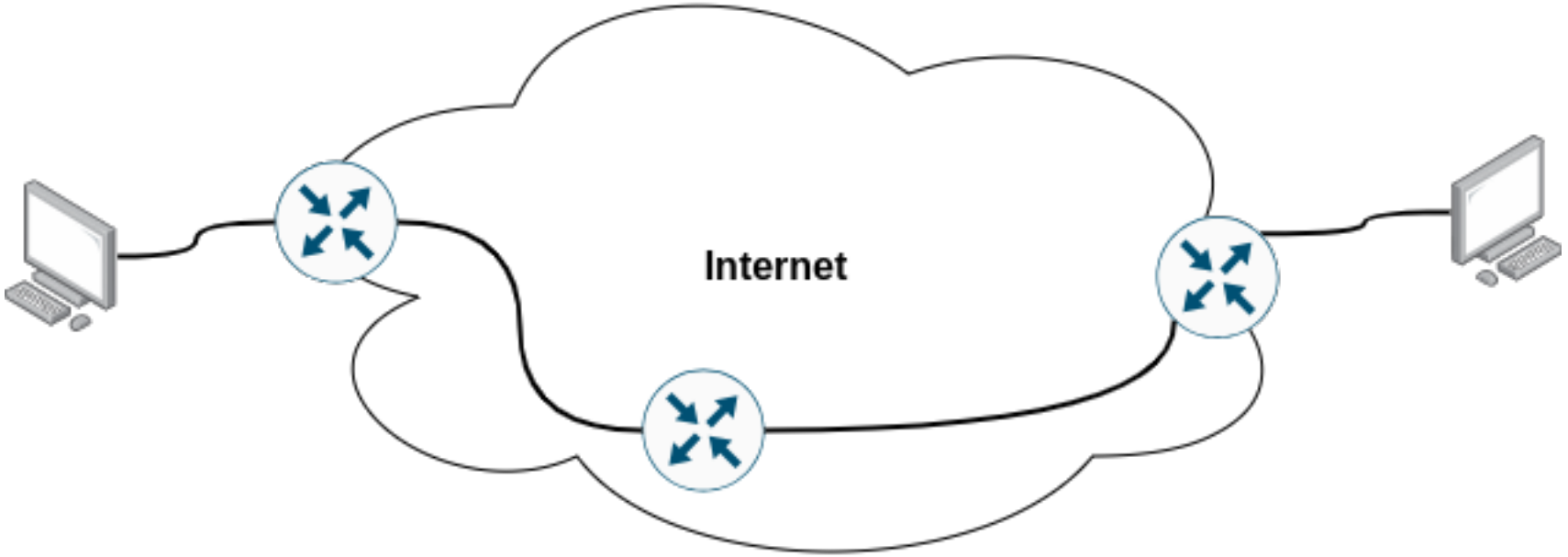
- Supprimer toutes les adresses

```
ip addr flush dev <device>
```

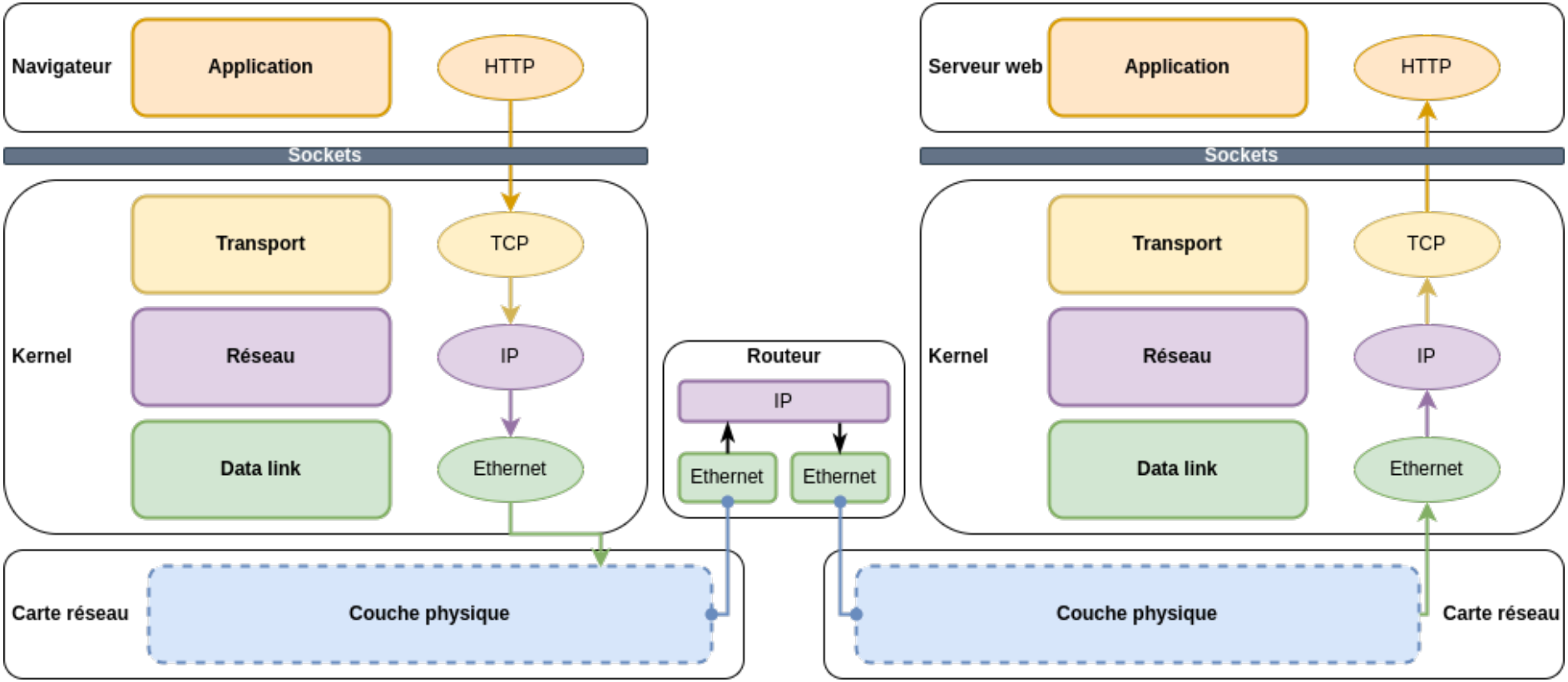
- Mettre en route/couper une l'interface

```
ip link set <device> <up|down>
```

Réseau TCP/IP: routage des datagrammes IP

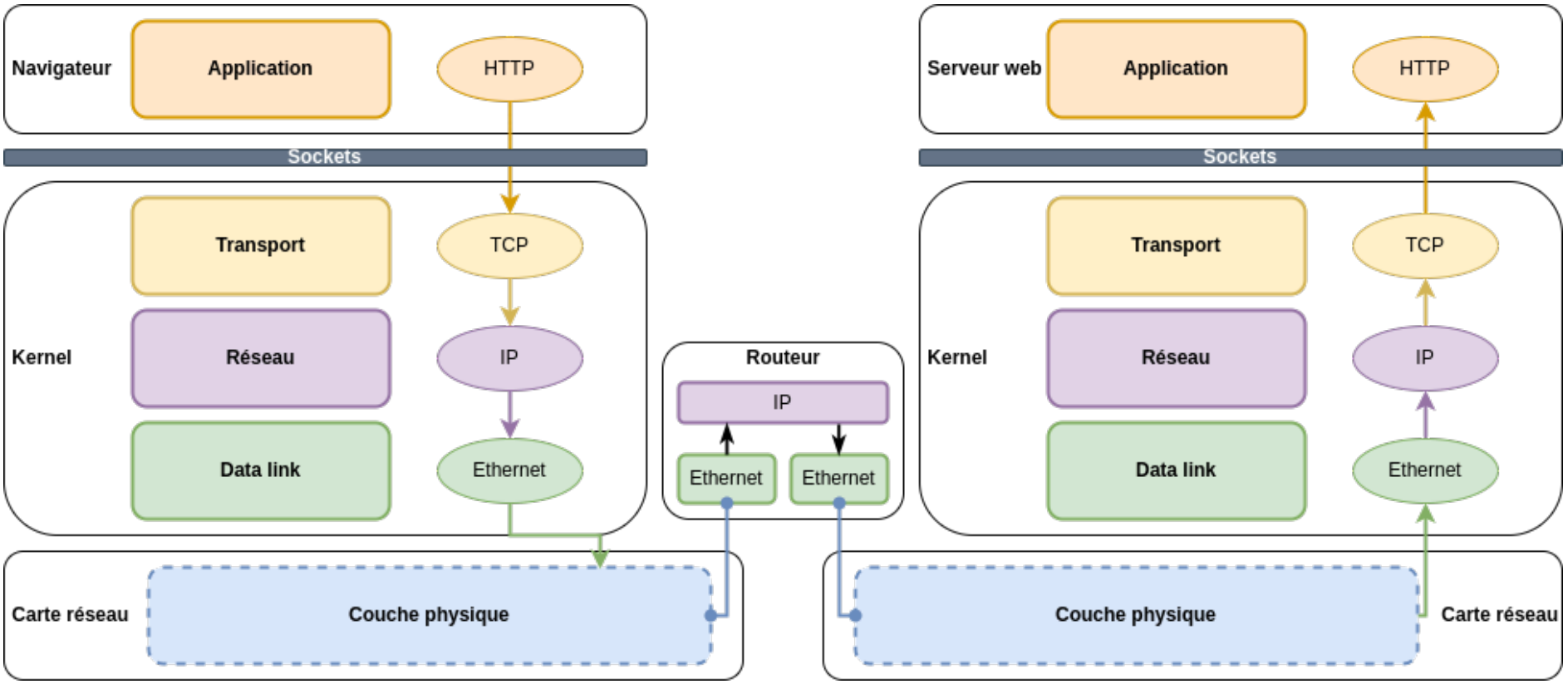


- Pour communiquer, les hôtes transmettent leurs paquets IP à des “routeurs”
- Les “routeurs” sont aussi des hôtes au sens TCP/IP, avec la particularité:
  - d’avoir plusieurs interfaces ;
  - de faire passer des paquets d’une interface à l’autre.



Un datagramme voyage dans plusieurs dimensions:

- entre les équipements ;
- dans la pile IP de ces équipements.



- Décapsulation/encapsulation à chaque noeud selon le “niveau” du “switch”
- Le routage IP s’effectue par prise de décision en fonction de l’IP de destination

- Tous les équipements possédant une pile IP doivent prendre des décisions de routage
- Ces décisions de routage se font normalement sur l'adresse de destination des:
  - paquets générés localement (hôtes et routeurs)
  - paquets transmis (routeurs)
- Les règles de décision sont contenues dans **la table de routage** de l'équipement
- La sélection de la route à prendre suit le principe de la plus grande correspondance pour l'adresse réseau ("longest prefix match")

- Lorsqu'un hôte **A** émet un paquet à destination d'un hôte **B**, il l'encapsule dans une trame Ethernet
- Cette trame sera envoyée:
  - directement à l'hôte **B** s'il est sur le **réseau local**
  - à **un routeur** si **B** n'est **pas sur le réseau local**, le routeur faisant son affaire de transmettre le contenu de la trame au bon endroit

- **A** peut savoir si **B** est sur le même réseau local en consultant sa table de routage:
  - grâce à son masque, **A** connaît l'étendue de son réseau local
  - il peut donc vérifier si **B** appartient à ce réseau

Historiquement, les hôtes avaient des tables de routage assez simples:

```
user@host:~$ ip ro
192.168.0.0/24 dev eth1 proto kernel scope link src
192.168.0.228
default via 192.168.0.254 dev eth1
```

Mais avec l'essor de la conteneurisation pour le développement, les tables de routages deviennent plus complexes...

Ethernet

- Encapsulation de plus bas niveau généralement utilisée sur les LANs
- Modulée directement sur le support physique (*Manchester encoding*)
- Les stations partagent le média physique
- Chaque station à une adresse **unique au monde** (adresse MAC) codée sur 6 octets

- Dans un réseau Ethernet, les stations sont identifiées par leur **adresse MAC** (*Media Access Control*)
- Ces adresses sont codées sur 48 bits (6 octets)
  - 24 premiers bits désignent le fabricant (*Organisational Unique Identifier*)
  - 24 derniers bits à sa discrétion (mais uniques)
- Ces adresses sont représentées sous la forme:

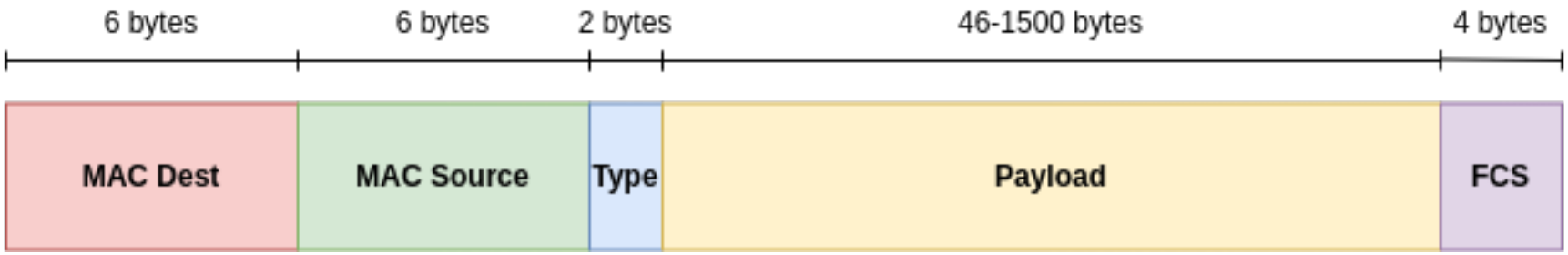
```
aa:bb:cc:dd:ee:ff
```

Attention

Certains équipements/OS permettent de modifier cette adresse MAC !

C'est par exemple le cas des smartphones qui pour la plupart randomisent leur adresse MAC régulièrement.

Trame Ethernet II (RFC 894)



Type d'Ethernet:

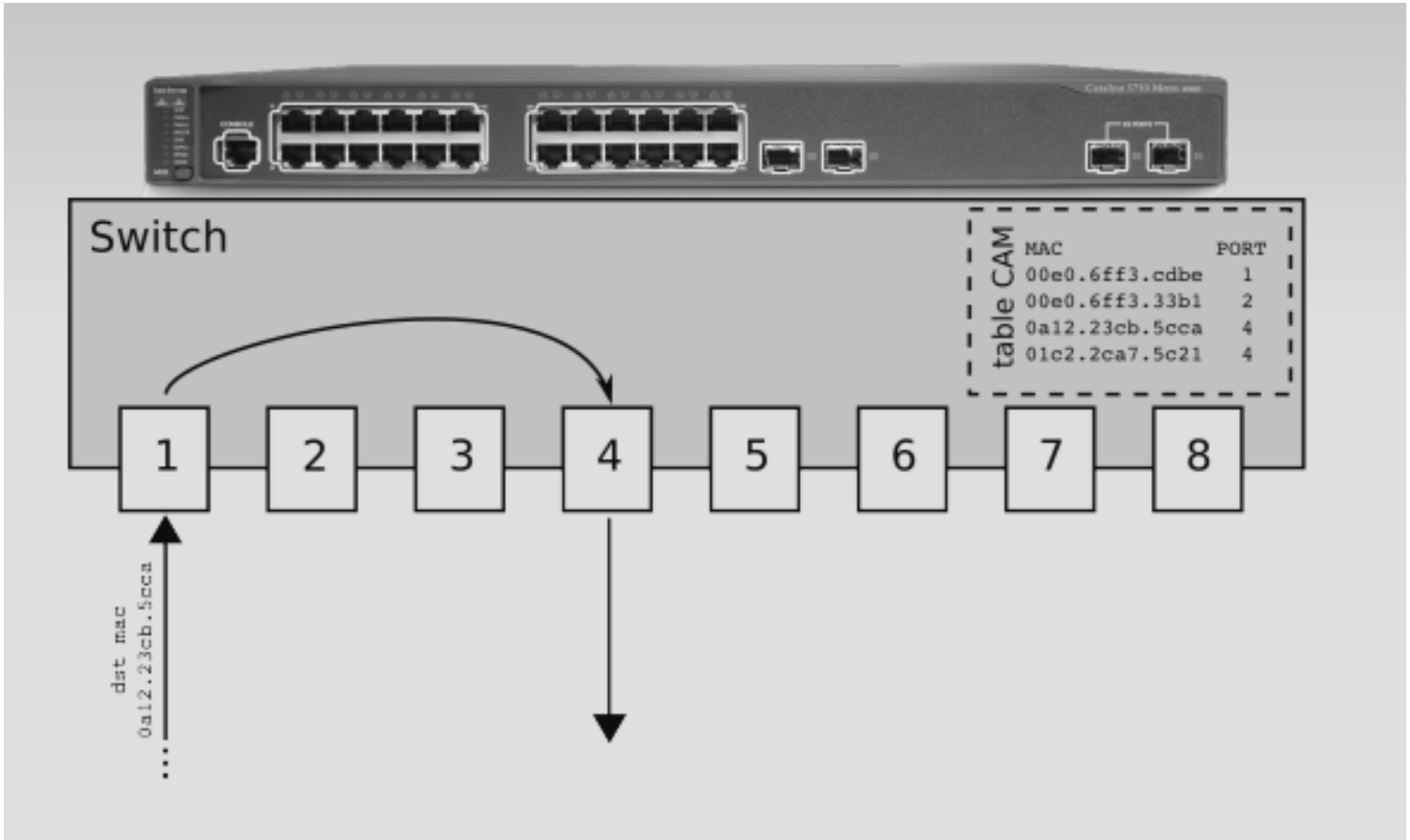
- `0x0800`: IP
- `0x0806`: ARP
- ...

Ethernet, équipements associés (niveau 1)

- **Répéteurs**
  - équipements à deux ports
  - “répète” vers l’autre port le signal électrique reçu sur un port
- **Hubs**
  - un hub est un répéteur possédant plus de deux ports
  - il réémet le signal électrique reçu d’un port vers tous les autres ports
- **Transceivers**
  - répéteurs permettant de passer d’un médium physique à un autre (fibre-cuivre, RJ45, ...)

- **Bridge/Switch**
  - consulte les adresses MAC dans les trames reçues
  - stocke des associations @ MAC/port
  - n'émet les trames vers un port que si la station avec l'adresse MAC de destination s'y trouve
  - émet les broadcasts sur tous les port

Ethernet, équipements associés (niveau 2)



Ethernet, Unicast, Broadcast, Multicast

- trame Ethernet unicast
  - un destinataire unique (représenté par son @ MAC)
- trame Ethernet broadcast
  - MAC dest = `ffff.ffff.ffff`
  - transmise à toutes les stations directement accessibles en Ethernet
- trame Ethernet multicast
  - (en pratique) la MAC débute par `0100.5e`
  - `0100.5eyy.zzzz`
  - transmise à toutes les stations directement accessibles en Ethernet

Ethernet met en oeuvre un algorithme de détection de collision, **CSMA/CD**:

Carrier Sense Multi-Access w/ Collision Detection

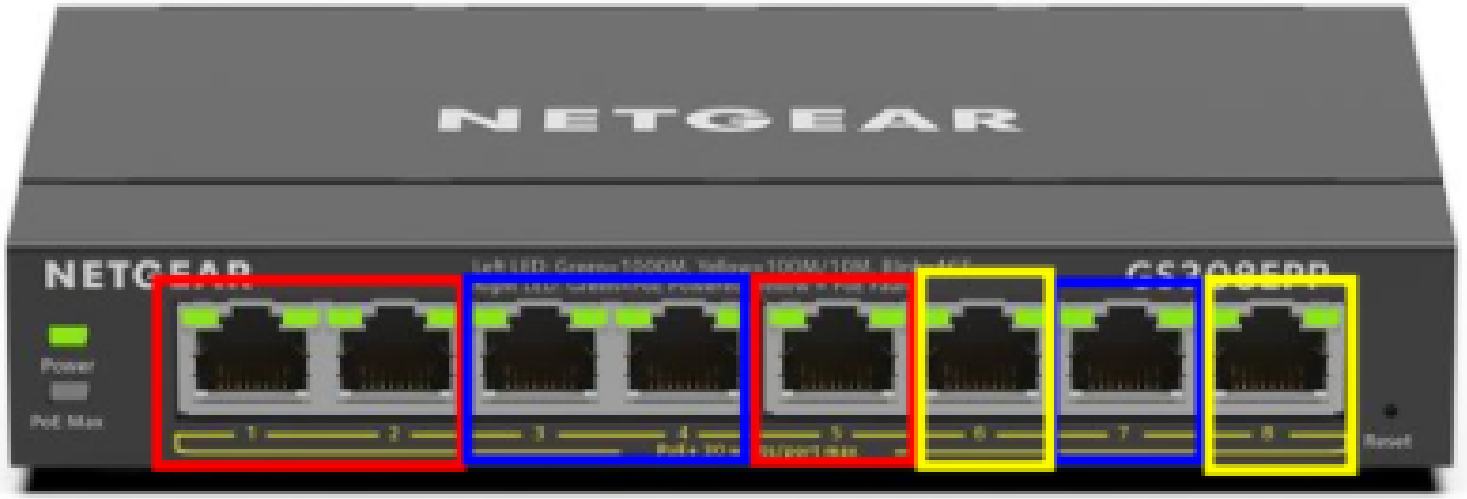
Les collisions sont normales tant qu'elles restent dans des proportions raisonnables (< ~10%)

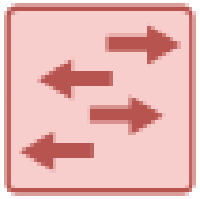
Les VLANs (LANs virtuels) permettent de découper des domaines de broadcast au sein d'un même switch

- chaque port du switch est affecté à un VLAN
- tous les ports d'un même VLAN partagent un domaine de broadcast
- des ports situés dans dans VLANs différents ne peuvent communiquer
- les VLANs peuvent communiquer s'ils sont reliés par des routeurs

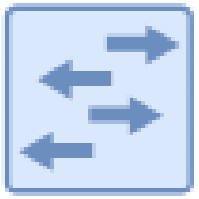
→ séparation de réseaux Ethernet

Ethernet, équipements associés (niveau 2)

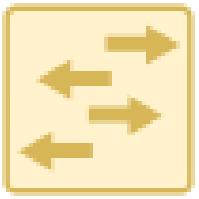




Data prod



Data staging



Dev

Ethernet, VLAN tagging & trunking (802.1q)

La norme 802.1q permet à un VLAN de s'étendre sur plusieurs switches

- les trames ethernet tagguées `dot1q` possèdent un VLAN ID (*VLAN tagging*)
- cet ID est manipulé par les switches, pas par des stations
- plusieurs trames appartenant à des VLANs différent peuvent ainsi circuler sur une même interface (*VLAN trunking*) appelée alors *trunk*

Linux permet de tagger ses paquets avec `vconfig`.

```
user@host:~$ vconfig add eth0 12
Added VLAN with VID == 12 to IF :eth0:
user@host:~$ ip add
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
qlen 1000
link/ether 00:15:c5:ab:dc:3b brd ff:ff:ff:ff:ff:ff
8: eth0.12@eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
link/ether 00:15:c5:ab:dc:3b brd ff:ff:ff:ff:ff:ff
```

Ethernet, VLAN tagging & trunking (802.1q)

Pourquoi faire ?

- un routeur avec une seule interface réseau
- participer à plusieurs réseaux Ethernet sans avoir besoin d'un routeur
- participer à plusieurs réseaux Ethernet quand on ne peut pas les faire communiquer au niveau 3

ARP

ARP, fonction

- Le protocole ARP permet à un hôte sur un réseau Ethernet+IP de **découvrir l'adresse MAC** d'une station donnée
- Une fois l'adresse MAC du destinataire déterminée, les trames Ethernet peuvent lui être envoyées
- Chaque hôte maintient une table ARP de correspondance @ IP/@ MAC des stations avec lesquelles il a récemment communiqué
- Le protocole ARP fonctionne sur un domaine de broadcast (réseau Ethernet) donné: il n'est pas routé !

ARP, mécanisme

- L'hôte 192.168.0.1 cherche à envoyer des paquets IP à 192.168.0.3, situé dans le même réseau local
  - il va donc vouloir lui envoyer une trame Ethernet "contenant" ces paquets IP "ping"
  - pour cela, il doit connaître l'adresse MAC l'hôte ayant l'adresse 192.168.0.3
- il envoie alors une requête **ARP en broadcast** sur le réseau, en posant la question: "qui possède l'adresse IP 192.168.0.3 ?"
- le locataire de 192.168.0.3 lui répondra (en **unicast**): "c'est moi, j'ai l'adresse MAC aa:bb:cc:dd:ee:ff"
- l'hôte recevant la réponse mettra alors à jour son cache ARP et pourra commencer à envoyer des paquets IP

ARP, mécanisme


```
user@host:~$ ip neigh
user@host:~$ ping c1 192.168.0.3
PING 192.168.0.227 (192.168.0.227) 56(84) bytes of data.
64 bytes from 192.168.0.227: icmp_seq=1 ttl=128 time=5.34 ms
...
1 packets transmitted, 1 received, 0% packet loss, time 0ms
user@host:~$ ip neigh
192.168.0.3 dev eth1 lladdr 00:13:8f:f7:2a:df REACHABLE
user@host:~$ ping 192.168.0.4
PING 192.168.0.4 56(84) bytes of data.
From 192.168.0.2 icmp_seq=1 Destination Host Unreachable
 192.168.0.4 ping statistics
1 packets transmitted, 0 received, 100% packet loss
user@host:~$ ip neigh
192.168.0.3 dev eth1 lladdr 00:13:8f:f7:2a:df REACHABLE
192.168.0.4 dev eth1 FAILED
```

ARP, gratuitous & unsollicited ARP

- *Gratuitous ARP* (ARP “gratuit”): envoi de paquets ARP afin de déterminer si son adresse IP est déjà utilisée sur le réseau
- *Unsollicited ARP* (ARP non sollicité): envoi de paquets ARP afin d’informer le reste du réseau local de son adresse MAC

- visualisation du cache ARP

```
ip neigh show arp
```

- ajout dans le cache ARP

```
ip neigh add lladdr <adresse_mac> to <addresse_ip> dev <if> arp [i <if>]
  s <addresse_ip> <adresse_mac>
```

- suppression d'une entrée dans cache ARP

```
ip neigh del to <addresse_ip> dev <if> arp [i <if>] d <addresse_ip>
```

- requètes ARP

```
arping I <if> <adresse_ip>
```

- gratuitous ARP

```
arping D I <if> <adresse_ip>
```

- ARP non sollicité

```
arping U I <if> <adresse_ip>
```

IP

IP, historique

- *Internet Protocol*, crée en 1973/1974 par Bob Kahn, Vint Cerf & Jon Postel (sur la base d'idées nées chez Xerox...)
- premiers tests en 1975
- déploiement sur ARPANET en 1983

IP, généralités

IP est le protocole fondamental de la pile TCP/IP:

- il encapsule tous les autres protocoles de niveau 3+
- c'est le protocole le plus bas requis dans la pile

IP fournit un service **non-fiable** de transmission de données:

- perte possible
- corruption possible
- ordres de livraison non garantis
- délais de livraison non garantis
- duplication possible
- fragmentation possible...

→ au besoin, c'est aux protocoles de niveaux supérieurs de s'en prémunir

MTU (*Max Transfert Unit*): c'est la taille maximum en octets d'un paquet sur un médium réseau donné

Les liens traversés par les paquets IP ont parfois des MTU différents:

- Ethernet: 1500
- FDDI: 4352
- PPPoE: 1492
- X25: 576
- PPP: 500-2000
- AAL5: 65536

Lorsqu'un paquet est plus gros que le MTU de l'interface via laquelle il doit être envoyé, il est **fragmenté**.

ICMP

ICMP, description

ICMP (*Internet Control Message Protocol*) est un protocole transportant les messages administratifs pour la pile TCP/IP.

ICMP, description

Chaque paquet ICMP possède un type de message, et éventuellement un code précisant ce message.

Ces codes permettent de prévenir des hôtes de conditions exceptionnelles:

- réseau/hôte/protocole injoignables (type 3)
- temps dépassé (type 11)
- redirection (type 5)

ICMP, description

- Aucun paquet ICMP n'est généré en cas d'erreur d'un paquet ICMP !

  → risque de boucles

- Certains codes ICMP permettent aussi de vérifier la connectivité d'un hôte
  - type 8 [*echo request*]: émis par la commande `ping`
  - type 0 [*echo reply*]: réponse renvoyée par l'hôte

UDP

UDP, généralités

UDP (*User Datagram Protocol*) est un protocole *connexionless*: pas de connexion établie entre l'émetteur et le récepteur

→ UDP n'apporte donc rien en termes de garanties par rapport à IP (paquets perdus, en désordre, dupliqués, ...), c'est au protocole applicatif de gérer ces problèmes si besoin (e.g. tftp)

UDP, généralités

Intérêt ?

- peu d'overhead (en-tête petit: 8 bytes)
- complètement *stateless* (pas d'état à conserver entre les participants
- simple à diagnostiquer (chaque paquet s'explique de lui-même)

UDP, utilisation

- UDP est en général utilisé pour les communications courtes
  - DNS, SNMP, DHCP: dans 90% des cas, une conversation se résume à moins deux échanges
- UDP est en général utilisé pour les communications ne nécessitant pas de retransmission
  - streaming audio/vidéo: pas la peine de retransmettre une image perdue
- UDP est utilisé dans les communications ne permettant pas d'établir une connexion
  - multicast: un émetteur vers n récepteurs
  - broadcast: un émetteur vers tous les récepteurs

TCP

TCP est **orienté connexion**:

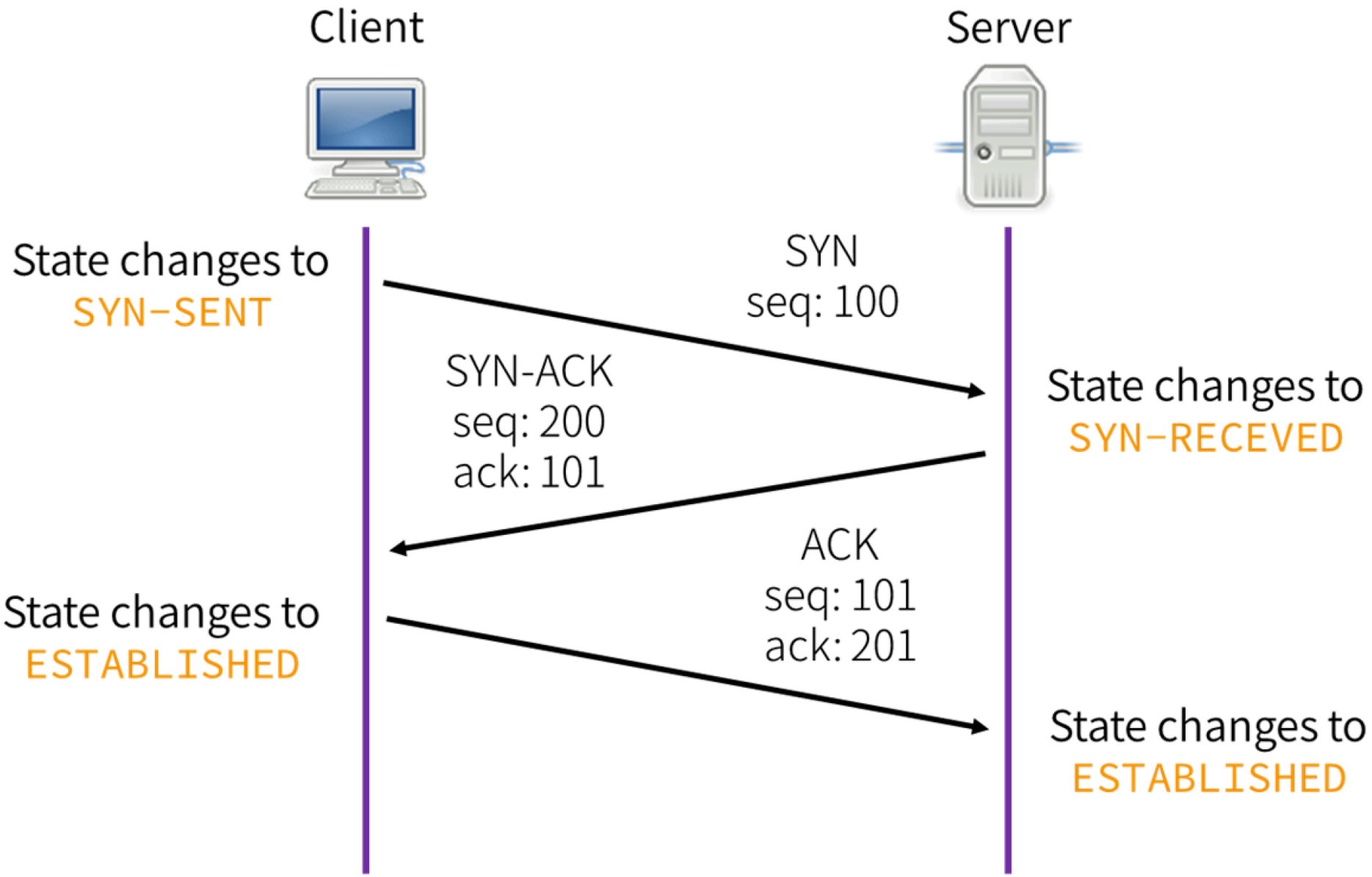
- les (2) participants vont établir une connexion négociée avant d'échanger
- chaque participant va maintenir un certain nombre de structures permettant de suivre la conversation
- les participants doivent acquitter les données reçues dans un certain délai
- TCP reçoit les données applicatives dans des buffers et décide des émissions sur le réseau
- TCP est full-duplex

TCP, généralités

TCP gère:

- la retransmission (avec *exponential backoff*)
- l'ordonancement
- les doublons

TCP, handshake



TCP, numéros de séquence

- Chaque hôte a son numéro de séquence pour la conversation TCP en cours.
- Ce numéro augmente de 1 à chaque octet envoyé (SYN et FIN comptent pour 1 aussi) à chaque paquet envoyé, l'émetteur doit indiquer quel numéro de séquence il attend
- Il acquitte ainsi tous les octets jusqu'à ACK+1.

TCP, numéros de séquence

Les numéros de séquence initiaux (ISN) sont choisis grâce à des fonctions cryptographiques

- plus seulement en fonction de l'heure
- évite les problèmes d'estimation de numéros de séquence...