IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RÉSEAUX IP ET BASES DU ROUTAGE

RES201

Printemps 2018

Christophe COUTURIER (Dpt. SRCD, Rennes): christophe.couturier@imt-atlantique.fr

SOMMAIRE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

INTRODUCTION

- **Organisation**
- **Contenu du cours**

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

OBJECTIFS

Cours: Bases des réseaux => RES 201

- **Modèles de référence (OSI, TCP/IP)**
- **Importance de la normalisation**
- **Adressage IP**
- **Réseaux locaux**

Apprendre par la pratique => RES 209

- **Conception d'un protocole**
- **Implémentation en Python (TP / projet)**
- **Mode projet**

PLAN DU COURS

- Rappels: modèles en couche
- Adressage
- Adressage IPv4
- Protocoles de transport (UDP/TCP)

- Introduction au routage
- Protocole IP (v4)
- Protocole ICMP
- Notion de tunnel
- Capture et analyse

RAPPELS

- Modèles en couches
- Notion de protocole

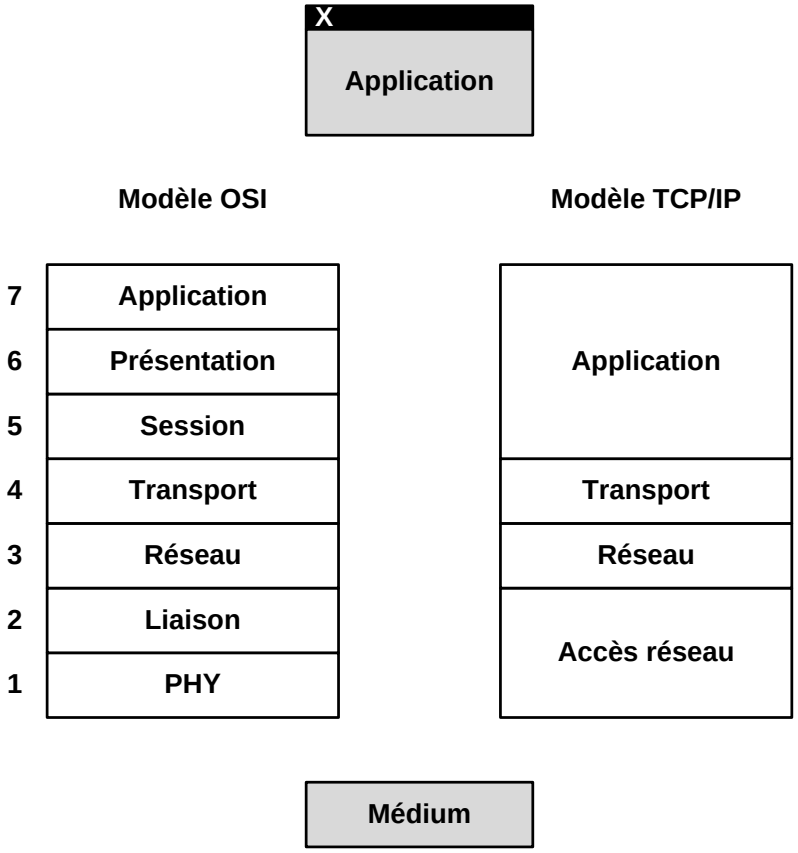
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

De la machine vers l'utilisateur

Intérêt:

- **Universalité**
  - Tout le mode se comprend; facilite l'interconnexion
- **Interopérabilité**
  - Un même protocole peut tourner sur des machines différentes (Cisco/Alcatel ou PC/Linux/MAC)
- **Adaptabilité**
  - Je peux utiliser IP sur du LTE, mais aussi sur Ethernet ou sur Wifi
  - Mon développement devient donc générique
- **Décomposition des problèmes et du travail**
  - Équipes de développement différentes (parallélisation)
  - Décomposition HW/SW
- **Évolutivité**
  - Un protocole peut évoluer indépendamment des autres car les interfaces entre les couches ne changent pas

EXEMPLES DES MODÈLES OSI ET TCP/IP



Modèle OSI

- Open System Interconnexion
- Modèle de référence

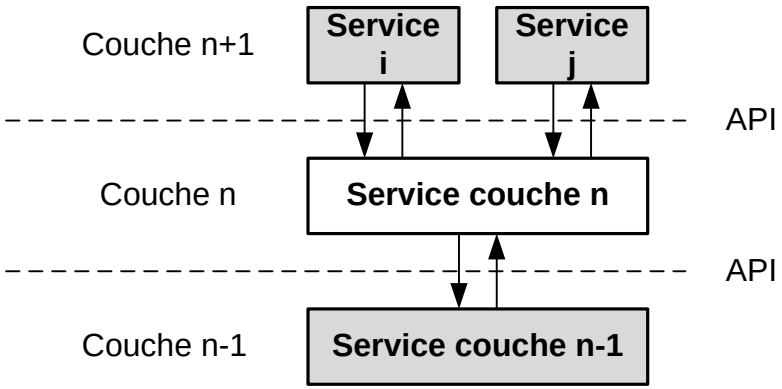
Modèle TCP/IP

- Modèle de fait d'Internet
- Fusion des couches 5 à 7
- Fusion de 1 et 2 (ex. Ethernet)

Important

- Les applicatifs et les media sont EXTERIEURS aux modèles
- Certains protocoles sont difficiles à situer (par ex ARP)

MODÈLE EN COUCHES: DÉFINITIONS



Chaque couche ajoute un ou plusieurs services à la précédente

- **Interactions via des primitives**
  API: Application Programming Interface
- **Services identifiés par leur SAP**
  SAP: Service Access Point

SDU (Service Data Unit)

- **Données utiles qui entrent/ sortent par le haut de la couche**

PDU (Protocol Data Unit)

- **Données enrichies des informations protocolaires**
- **Transitent par le bas de la couche**

EXEMPLES DE PROTOCOLES

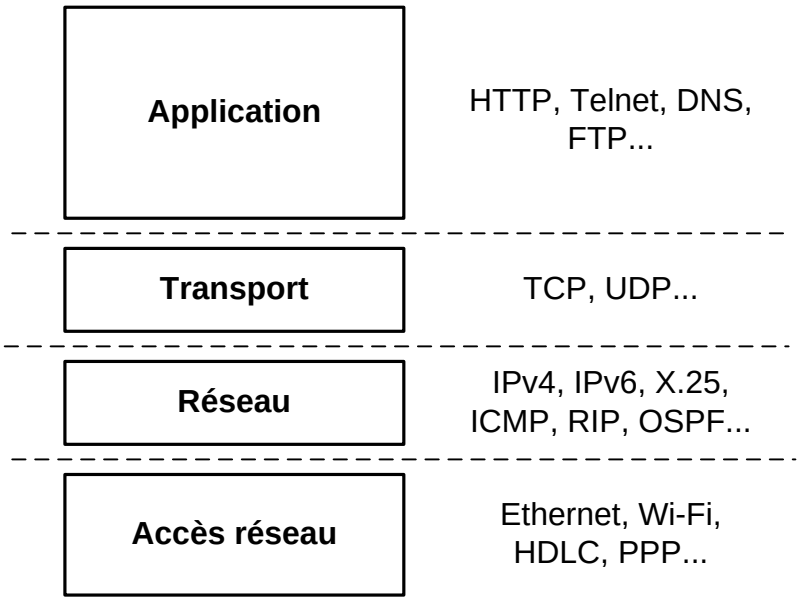
Certains sont difficiles à placer

- **Ex: ARP qui assure la correspondance @MAC/@IP**

Approche cross-layer

- **Entorse au modèle en couches**
- **Collaboration entre les différentes couches**
- **Par ex: router en fonction de la qualité du signal radio**

Exemples de protocoles répandus:



IMPLÉMENTATION SUR LES MACHINES

Espace utilisateur

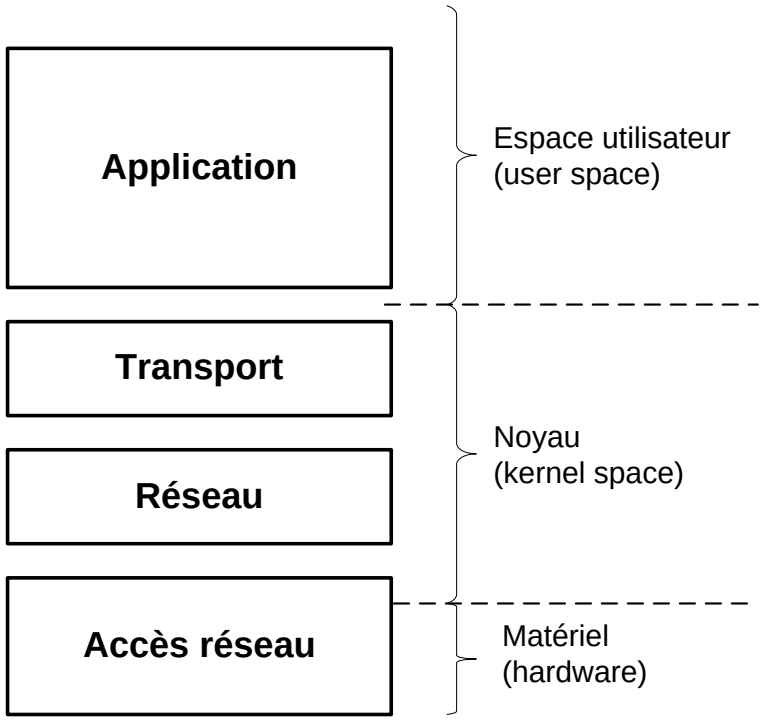
- **Applications et Protocoles applicatifs**
- **Modification facile (recompilation programme)**

Noyau

- **Protocoles de transport et réseau**
- **Couches haute de l'Accès (« drivers »)**
- **Modification => recompilation du noyau**

Matériel

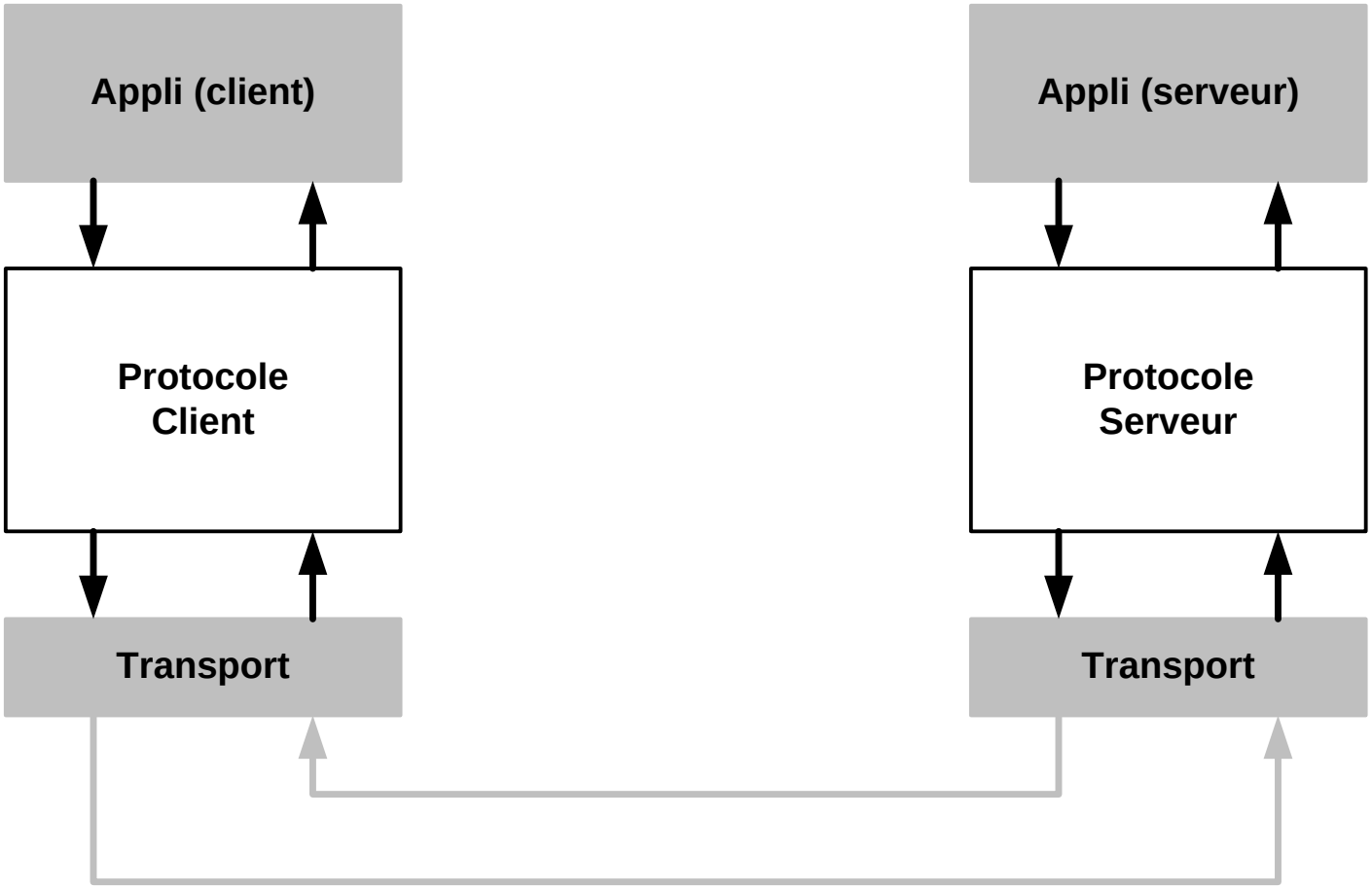
- **Couches basses de l'accès**
- **Couche Physique**
- **Modification => constructeur (microcode ou HW)**



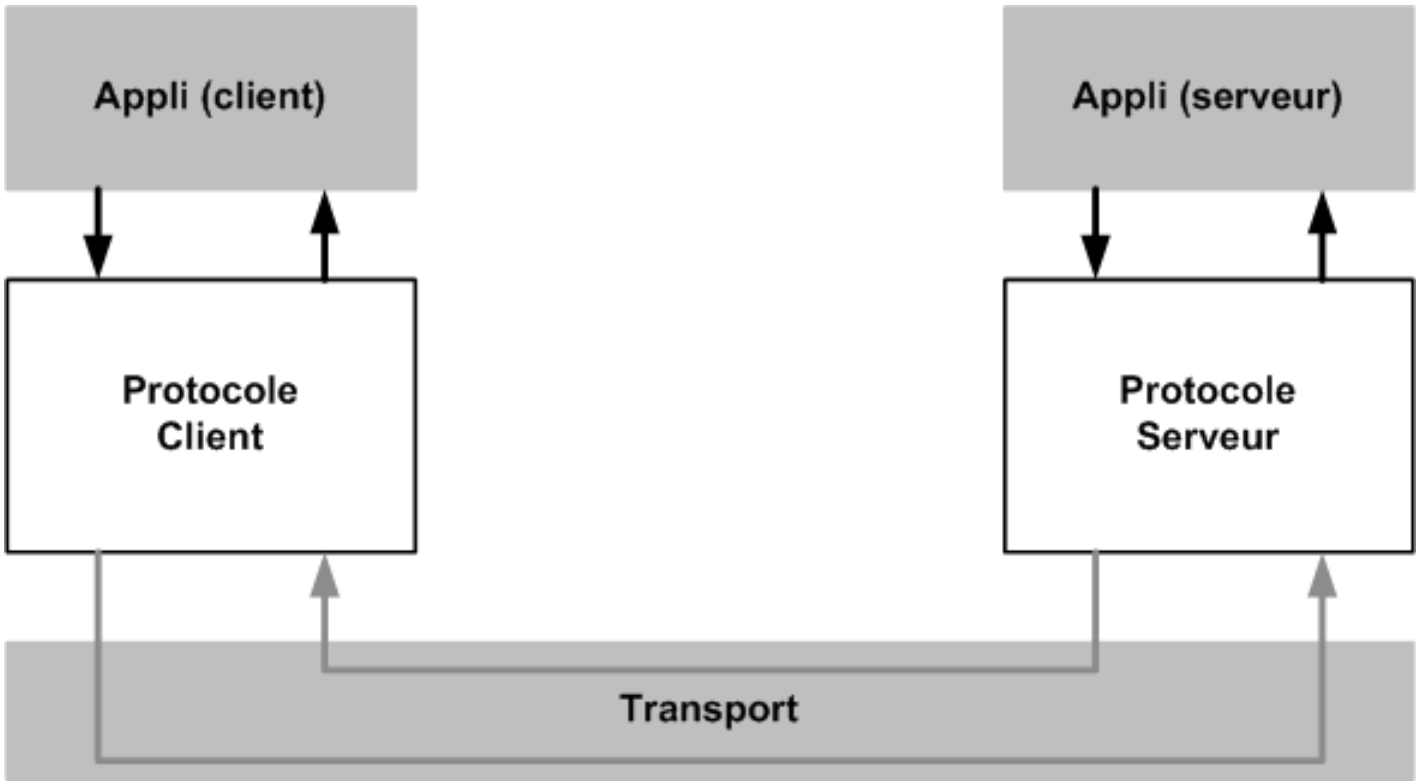
Ce modèle est « général », variantes possibles

- **Par ex. accélération matérielle (short path)**

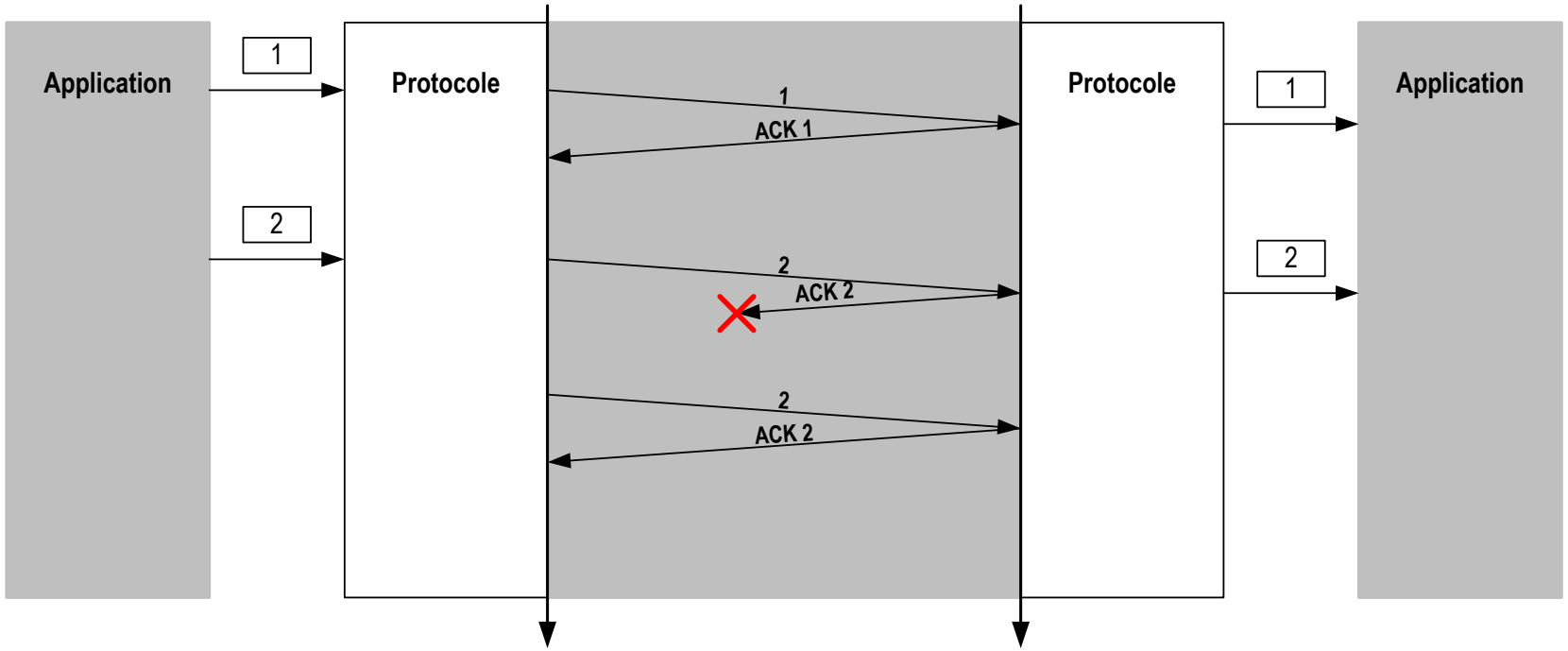
ENVIRONNEMENT DU PROTOCOLE (1/2)



ENVIRONNEMENT DU PROTOCOLE (2/2)



Le fonctionnement de la couche inférieure est transparent pour celle d'au dessus

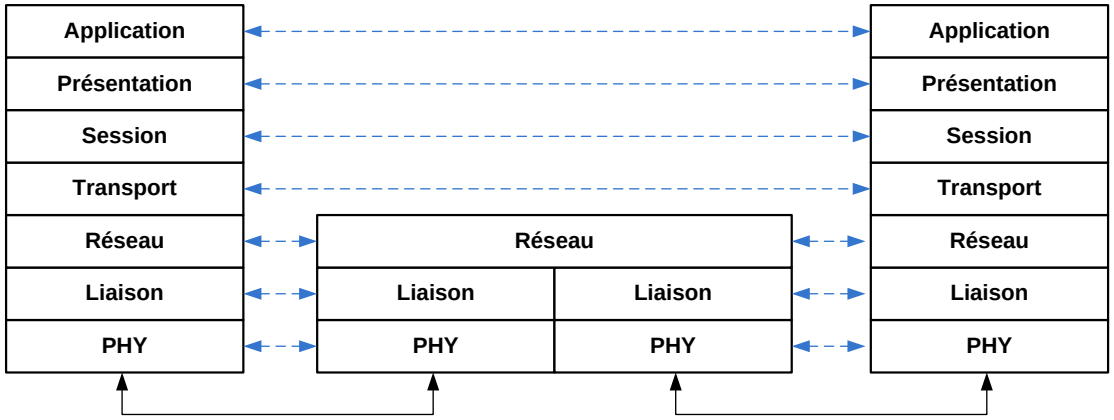


Ce qui est ajouté par une couche à l'émission est retiré par la même couche en réception

MODÈLE EN COUCHES: INTERCONNEXION

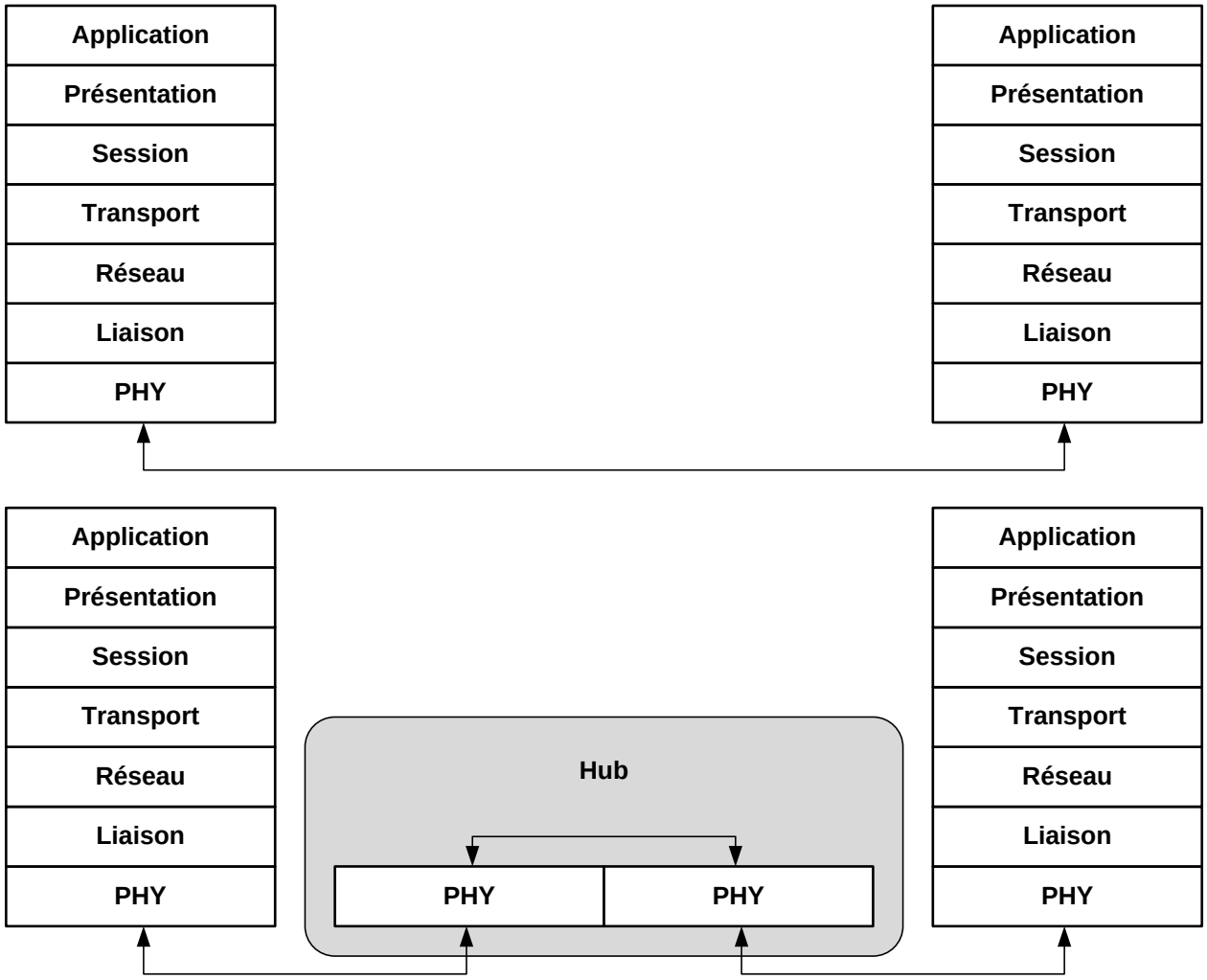
Point à point aux niveaux 1, 2 ou 3

Communication de bout en bout à partir du niveau 4

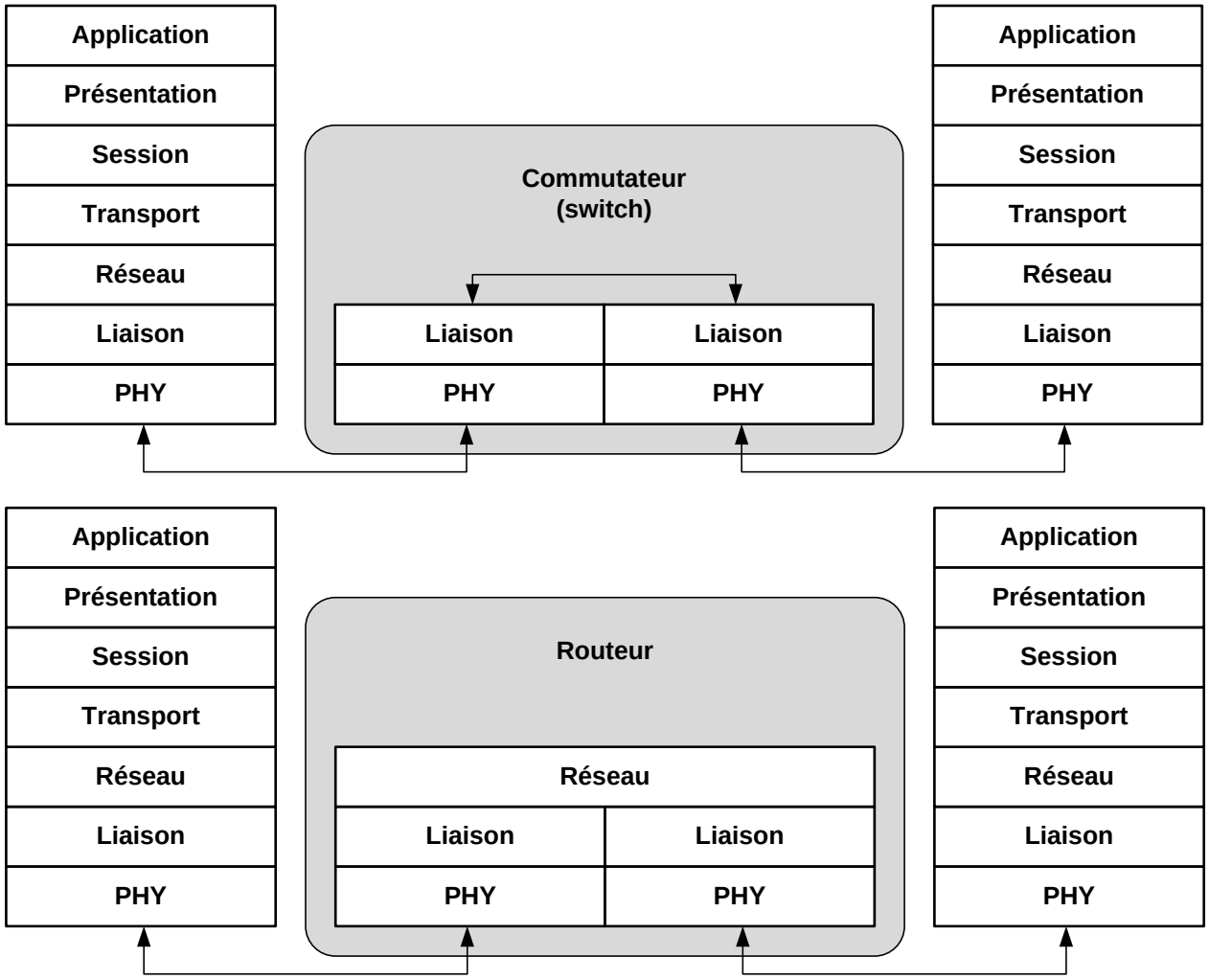


Niveau	Unité de données	Ex. d'équipements
4. Transport	Segment / datagramme	
3. Réseau	Paquet	Routeur
2. Liaison	Trame	Commutateur (switch) Modem, carte réseau
1. Physique	Signal	Hub, connecteur

MODÈLE EN COUCHES: INTERCONNEXION



MODÈLE EN COUCHES: INTERCONNEXION



PASSAGE À L'ÉCHELLE

« Scalabilité » : propriété fondamentale des protocoles/architectures

- **Généralisation à l'échelle de l'internet**
- **Dimensionnement** (par ex. taille des champs d'adresse)
- **Mode de fonctionnement** (centralisé, réparti, distribué…)
- **Puissance de calcul des équipements**
- **Capacité mémoire des équipements**
- **Tolérance aux pannes**
- **…**

Mais pas indispensable

- **Ex.: PPP n'assure que des liaisons point à point**
- **=> savoir quel est le besoin**

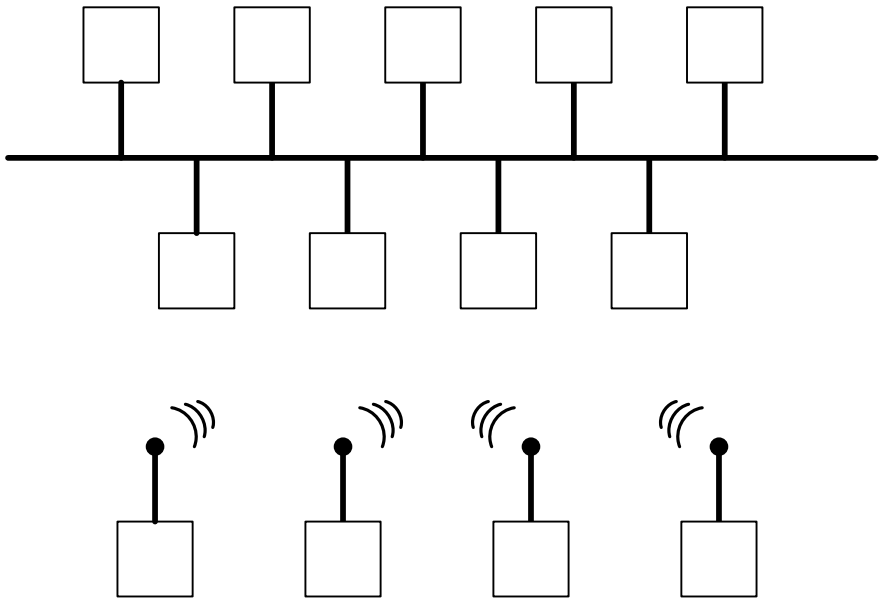
ADRESSAGE

- Topologies de réseau
- Notion d'adresse
- Modes d'adressage
- Types d'adresses et correspondance

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Bus

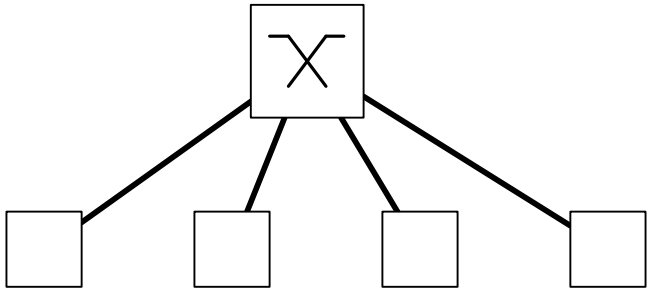
- **Tout le monde s'entend**



- Ex.: CAN, RS-485, PCI, Ethernet 10Base5, Wifi…

Étoile

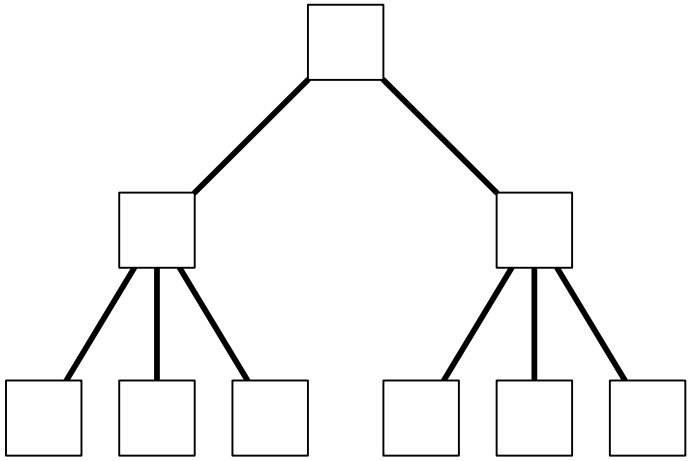
- **Liaisons point à point**
  entre nœuds terminaux (hosts) **et commutateur** (switch)



- Ex.: Ethernet (>10BaseT), RTC

Arbre

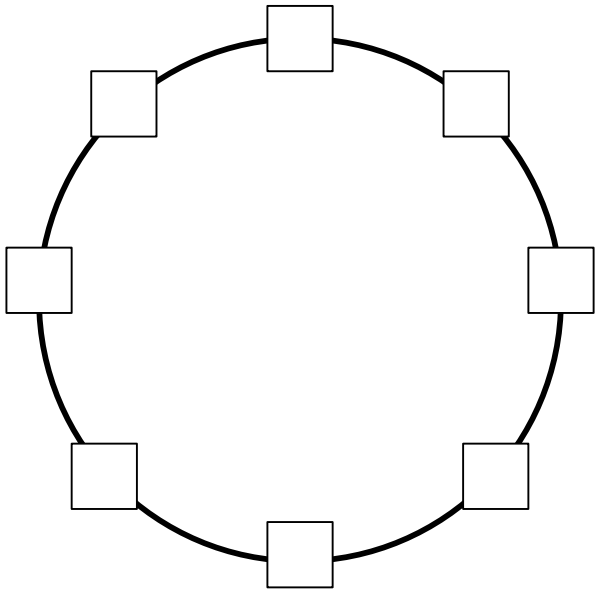
- **Relayage de proche en proche (par terminaux ou relais)**



- Empilement d'étoiles (type Ethernet), Réseaux multi-sauts…

Anneau

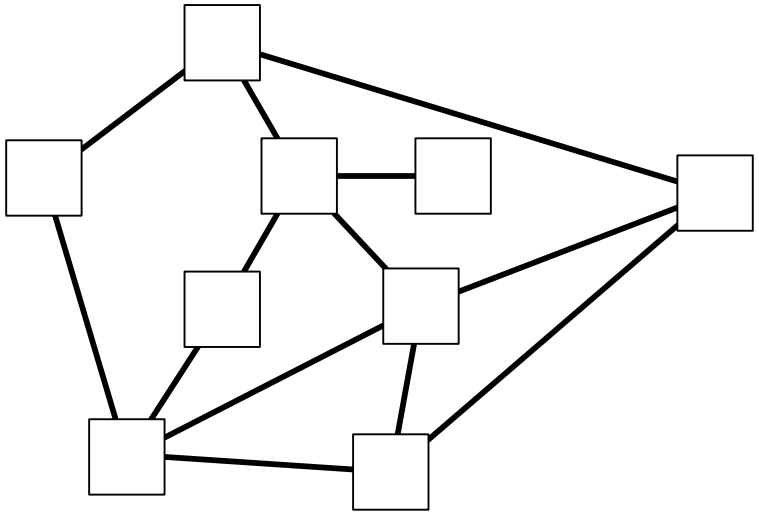
- **Relayage de proche en proche (par les terminaux)**
- **Résilience (résistance à la perte d'un lien ou d'un relai)**



- Ex.: Token Ring

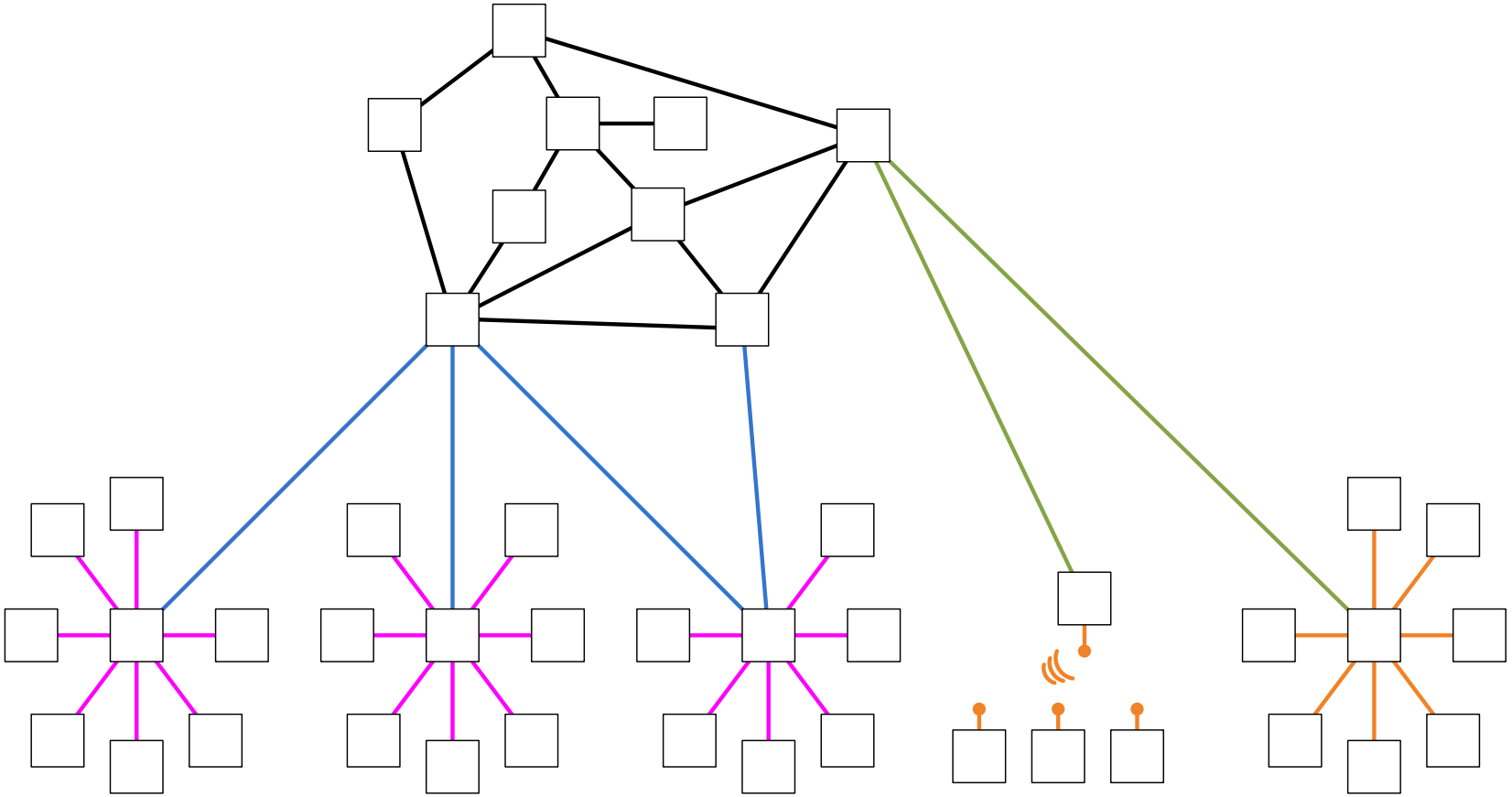
Maillage

- **Ajout de liens explicites (câblage) ou implicites (radio) pour redondance**
- **Plusieurs routes possibles => Routage**

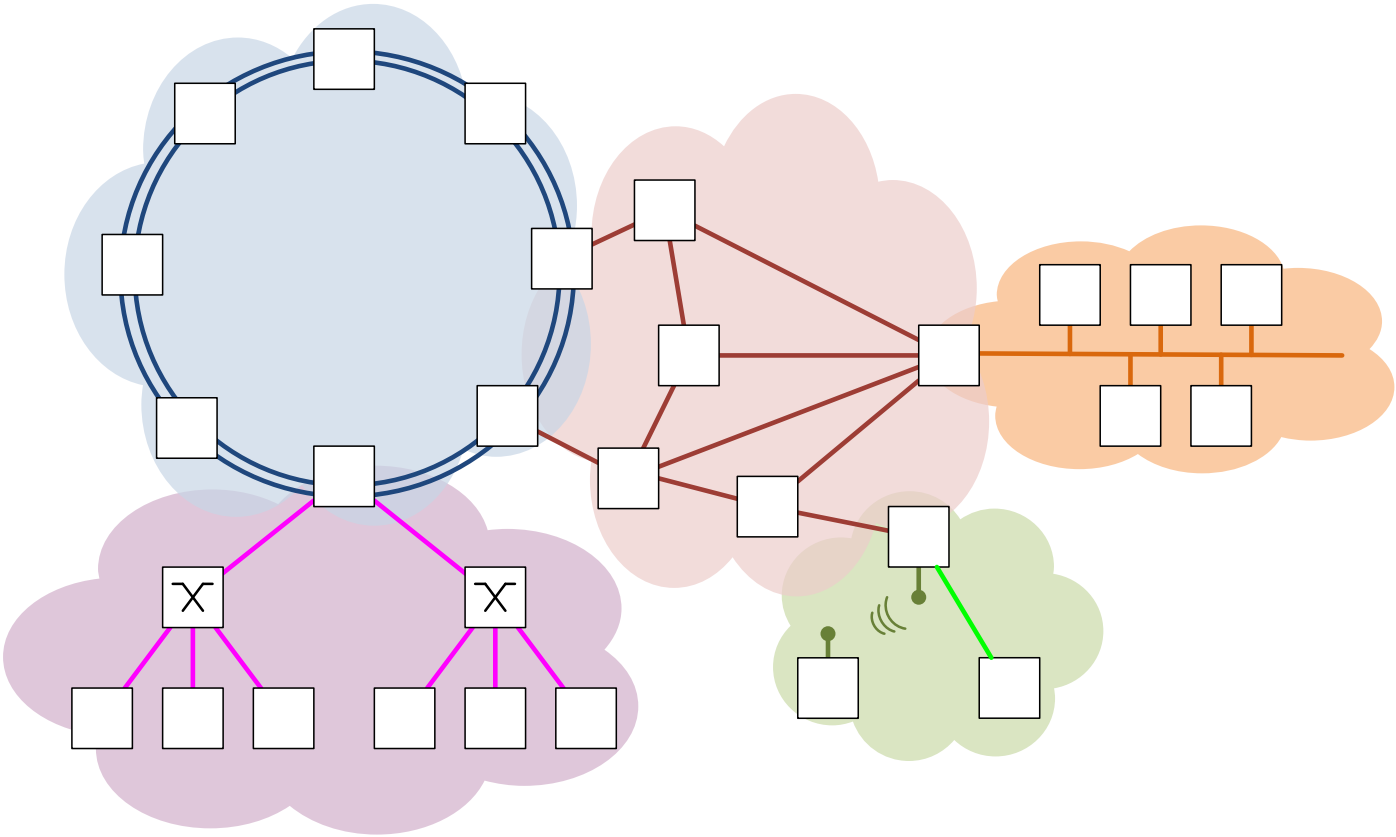


TOPOLOGIES RÉSEAU

Et toutes les combinaisons possibles!



Et toutes les combinaisons possibles!



IDENTIFICATION VS ADRESSAGE

Identifiant

- **Désigne les équipements de façon unique**

Adresse

- **Permet de retrouver le chemin** (la route)

D'une manière générale

- **Besoin d'unicité**
- **Comment allouer les adresses/identifiants?**

ADRESSAGE

Comment allouer les adresses?

Adressage « à plat »

- **Une seule entité alloue les adresses**

Allocation hiérarchique

- **Une entité centrale**
- **Délègue la gestion de sous groupes d'adresses**
- **Les sous-groupes peuvent eux même être re-divisés**
- **Les adresses d'un même sous groupe sont « voisines »**
- **Différentes organisations possibles:**
  - Institutionnelles, géographiques….

ALLOCATION DES ADRESSES IP

- **IANA: Internet Assigned Numbers Authority**
  - https://www.iana.org/numbers
- **Allocation hiérarchique par délégation**
  - Organisation géographique
  - Délégation de **blocs d'adresses** aux RIR: Regional Internet Registeries
  - Les RIR allouent eux même des sous blocs qui sont eux même re-divisés
- **Certains blocs ont un usage réservé**

Registry	Area Covered
AFRINIC	Africa Region
APNIC	Asia/Pacific Region
ARIN	Canada, USA, and some Caribbean Islands
LACNIC	Latin America and some Caribbean Islands
RIPE NCC	Europe, the Middle East, and Central Asia

Source: IANA

MODES D'ADRESSAGE

Unicast (uni-diffusion)

- **Une interface = une adresse**

Broadcast (diffusion)

- **Tout le réseau**

Multicast (multi-diffusion)

- **Un groupe d'équipements**

Anycast

- **Un parmi les membres d'un groupe**

TYPES D'ADRESSES

A chaque couche ses adresses

- **Applicatif: noms de protocoles / machines / domaine**
  ex.: sproj.rsm.entb.fr

- **Transport: n° de port (couplé à @Réseau)**
  ex: 10.10.1.2:80

- **Réseau: adresse réseau (@ IP)**
  ex: 192.168.10.124 / 2001:0db8:ac10:fe01::

- **Liaison: adresse MAC (Media Access Control)**
  ex.:16:6e:08:e1:a6:f2

RÉSOLUTION D'ADRESSES

- **Comment « traduire » les adresses d'un niveau à l'autre?**
  - Protocoles de **résolution d'adresse**
- **Noms de domaines -> @IP: DNS** (inversement: Reverse DNS)
- **Protocole -> n° de port: assignation centralisée (IANA)**
  - /etc/services
- **@IP -> @MAC: ARP** (Inversement: RARP)
  - Vu plus tard en PC

ADRESSAGE IPv4

- **Allocation des adresses**
- **Notion de préfixe**
- **Sous réseaux**
- **Adresses publiques / privées**

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

ALLOCATION DES ADRESSES IP

- **IANA: Internet Assigned Numbers Authority**
  - https://www.iana.org/numbers
- **Allocation hiérarchique par délégation**
  - Organisation géographique
  - Délégation de **blocs d'adresses** aux RIR: Regional Internet Registeries
  - Les RIR allouent eux même des sous blocs qui sont eux même re-divisés
- **Certains blocs ont un usage réservé**

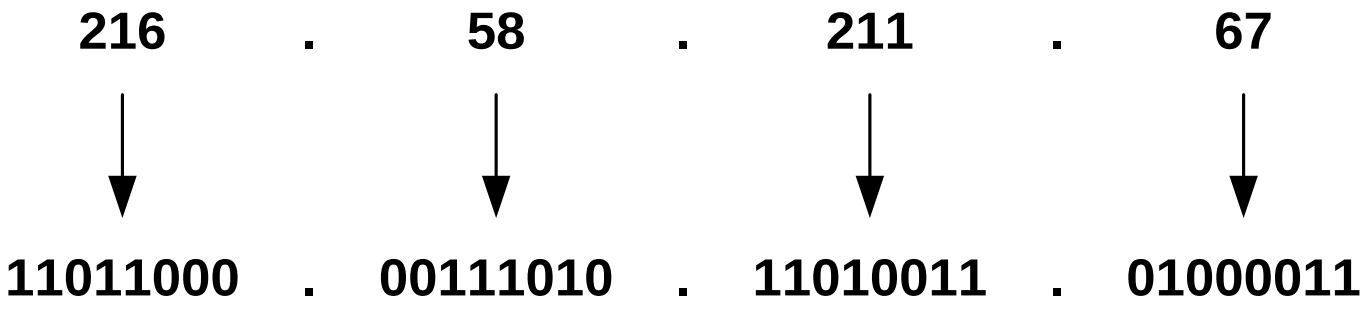
Registry	Area Covered
AFRINIC	Africa Region
APNIC	Asia/Pacific Region
ARIN	Canada, USA, and some Caribbean Islands
LACNIC	Latin America and some Caribbean Islands
RIPE NCC	Europe, the Middle East, and Central Asia

Source: IANA

ADRESSE IP

- **Taille: 32 bits en IPv4**
  - 128 bits (16 octets) en IPv6, hors cadre de ce cours
  - 2^{32} = 4 294 967 296 combinaisons

- **Représentation décimale**
  - 4 octets (0 à 255) séparés par des '.'
  - ex.: 216.58.211.67



COMMENT FAIRE DES « BLOCS D'ADRESSES » ?

- **CIDR: Classless Inter-Domain Routing**
- **« Bloc »: groupe d'adresses ayant un même préfixe**
  - Préfixe: séquence de bits de poids fort
  - **Longueur variable** (0 à 32 bits)
  - 2^{32} = 4 294 967 296 combinaisons
- **Exemple: préfixe de longueur 16**

211.58.211.67	→	**11011000**	**.**	**00111010**	**.**	**11010011**	**.**	**01000011**
211.58.2.36	→	**11011000**	**.**	**00111010**	**.**	**00000010**	**.**	**00100100**
211.59.105.24	→	**11011000**	**.**	**00111011**	**.**	**01101001**	**.**	**0001100**

Préfixe – 16 bits

- **Écriture**
  - / 16
  - Ou 255.255.0.0

EXERCICES

- **Soit A= 65.12.1.1 et B= 65.14.2.3**
  - Le prestataire Internet de A dispose d'un préfixe de taille 16 A et B appartiennent-ils au même bloc?
  - Même question pour préfixe de taille 12

- **Quel est le plus grand préfixe regroupant les adresses 132.16.128.24 et 132.16.130.13 ?**

NOTION DE SOUS RÉSEAU

- **Sous réseau = ensemble des @ disposant d'un même préfixe**
- **Masque de sous réseau**
  - Adresse constituée de
    ensemble des bits de poids fort du préfixe =1
    ensemble des bits de poids faible = 0
  - Ex: préfixe /13:
    `1111 1111.1111 1000.0000 0000.0000 0000` **soit 255.248.0.0**

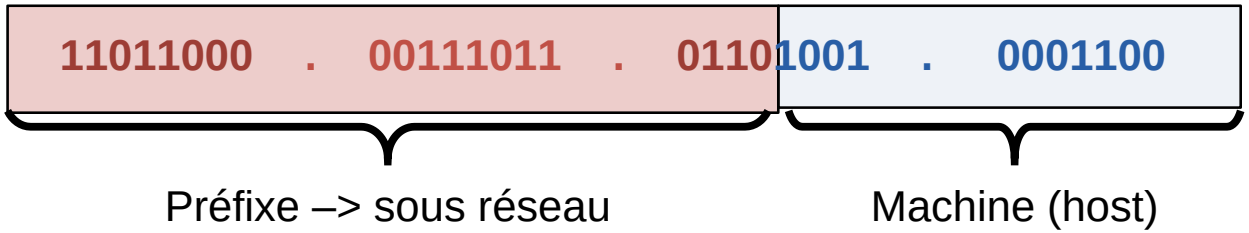
- **L'adresse d'un sous réseau d'une @IP est donnée par**
  - IP & Masque (ET bit à bit)
  - Ex. adresse sur SR de 16.17.18.19 / 20:

```
     0001 0000.0001 0001.0001 0010.0001 0011
&    1111 1111.1111 1111.1111 0000.0000 0000
-----------------------------------------------
     0001 0000.0001 0001.0001 0000.0000 0000
```

soit 16.17.16.0

DÉCOMPOSITION D'UNE ADRESSE IP

- **Le couple @IP/préfixe permet d'identifier**
  - Le sous réseau
  - L'adresse de la machine dans le sous réseau

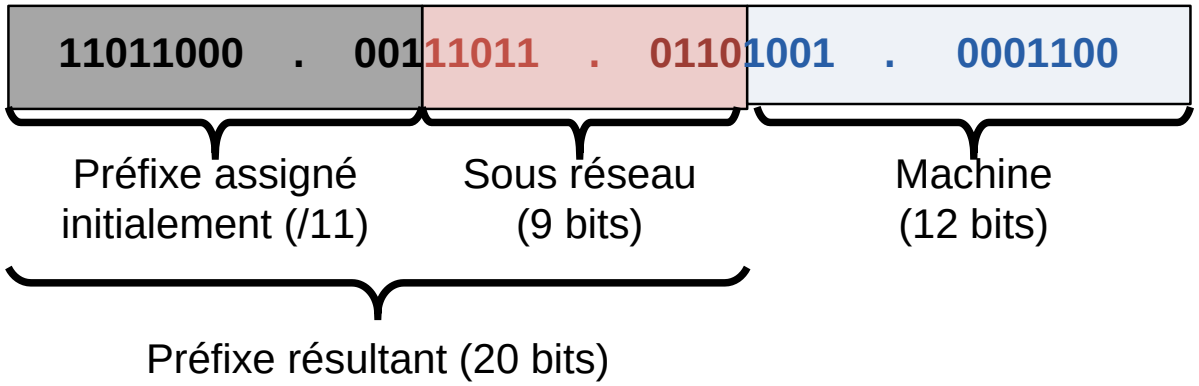


- **Adresses de host réservées:**
  - Tous les bits à 0 : adresse de réseau (raison historique)
  - Tous les bits à 1: adresse de broadcast (tous les host du SR)

- **Avec un préfixe de taille n**
  - On peut former 2^n sous réseaux
  - Chaque SR peut accueillir $2^{(32-n)}$-2 machines

RE-DÉCOMPOSITION D'UN SR

- **On peut re-découper une plage d'adresses**
  - En augmentant la taille du préfixe



- **Utile pour**
  - RIR: distribution aux opérateurs
  - Opérateurs: distribution aux clients
  - Administrateurs système: structuration logique de leur réseau

EXERCICE

- **Un opérateur se voit attribuer la plage 24.224.0.0/11**
  - Combien de préfixes /16 peut-il vendre à ses clients?
  - Combien de /24
  - Les adresses suivantes appartiennent-elles à cet opérateur:
    24.224.25.12 25.10.13.14 25.225.15.78 25.226.190.125

- **Un client désire décomposer son réseau en 7 SR de 100 machines chacun**
  - De quelle taille de préfixe a-t-il besoin?
  - Donnez un exemple de préfixe que l'opérateur pourrait lui assigner

ADRESSE PUBLIQUES / ADRESSES PRIVÉS

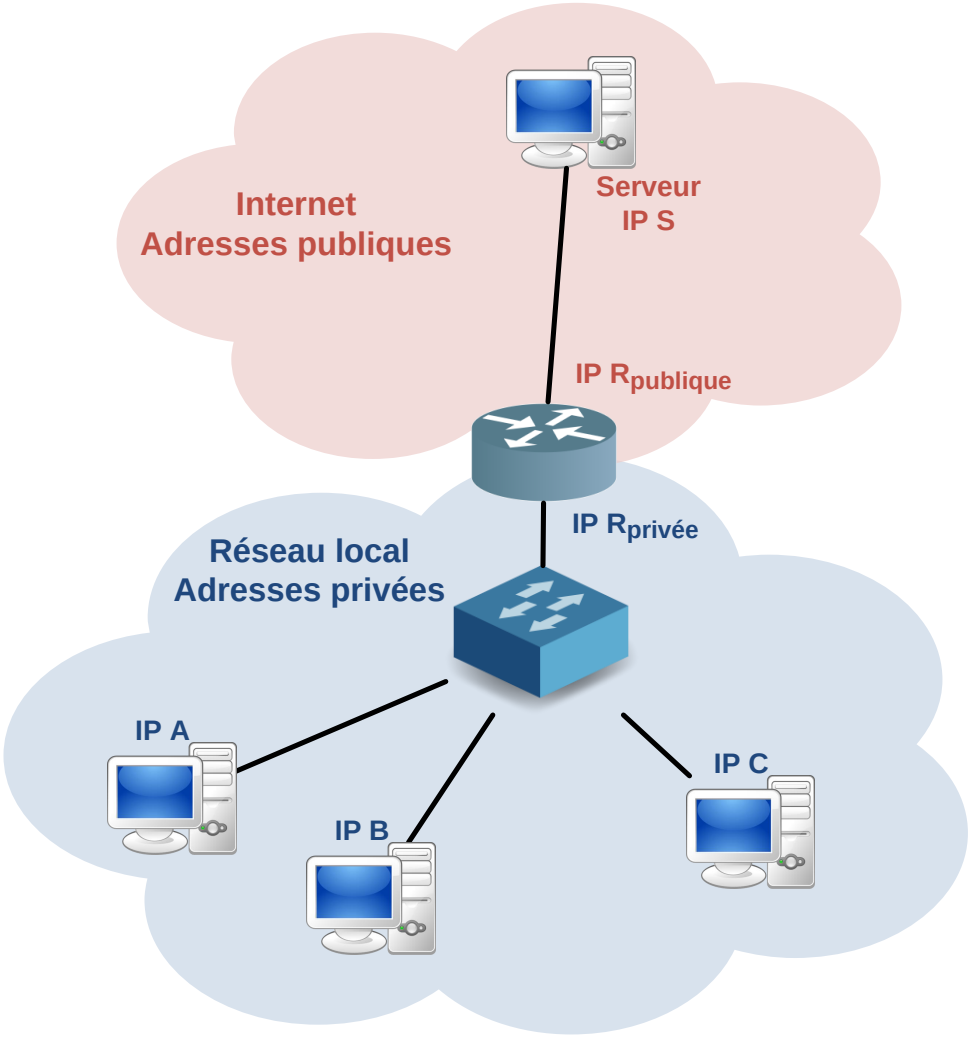
- **Certaines adresses sont réservées à un usage privé**
  - Elles sont dites « **privées** » ou aussi « **non routables** »
  - Elle ne sont pas assignées
  - Chacun peut les utiliser sur son réseau
  - Elles ne seront pas accessibles depuis l'internet
  - **Utile pour construire un réseau local**

- **Plages réservées**
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

  - **NAT: Network Address Translation**
  - Remplace ces adresses par des adresses publiques pour sortir sur internet
  - -> Petite Classe

NAT: NETWORK ADDRESS TRANSLATION

- **Problème**
  - Raréfaction des IPv4 publiques
  - Multiplication des machines
- **Solution proposée par NAT**
  - Un routeur « prête » son IP publique aux machines de son réseau local
- **Vu des machines**
  - C'est transparent
- **Vu du serveur**
  - Plusieurs connections avec une même machine (IP publique du routeur) sur des ports différents
- **Le routeur assure la « translation »**
  - Table de correspondance
  - Fonctionnement « **statefull** »
  - Différents algorithmes possibles



PROTOCOLES DE TRANSPORT

- **UDP**
- **TCP**
- **Vue utilisateur:**
  mode datagramme VS mode flux

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

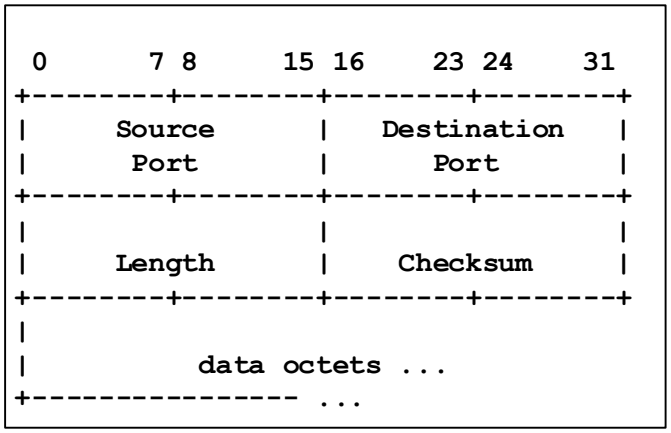
PROTOCOLES DE TRANSPORT: INTRODUCTION

- **Il existe plusieurs protocoles de transport**
  - On présentera **TCP** et **UDP**
  - Mais il y en a d'autres:
    SCTP (multi flux), RTP (temps réel), RSVP (Réservation)…
  - Même niveau dans la pile, mais offrent des services différents

- **Implémentation**
  - Transport implémenté dans le noyau du système d'exploitation (OS:Operating System)
  - => Implémentation logicielle mais modification difficile (recompilation noyau)
  - Interface de programmation standard (API): les **sockets**
    (en projet: Twisted est une surcouche au dessus des sockets)

UDP: USER DATAGRAM PROTOCOL

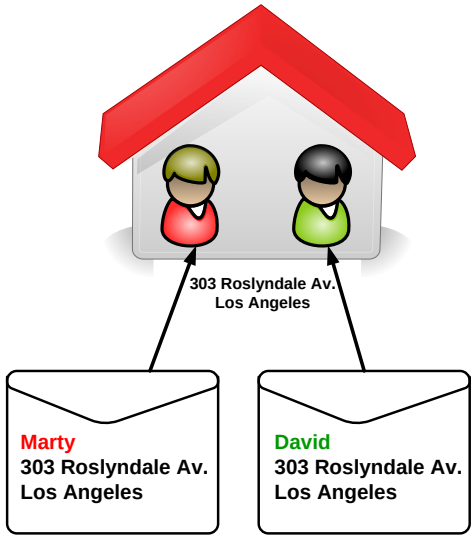
- **UDP: User Datagram Protocol (RFC 768)**
  - Un protocole très simple
- **Mode datagramme**
  - Un envoi pour chaque message
  - Si message trop gros => le décomposer
    Si ma lettre dépasse 50g, je la décompose en 2
- **Notion de port source et port destination**
  - Destinataire à l'adresse donnée
  - Couplage fort à IP pour l'adressage

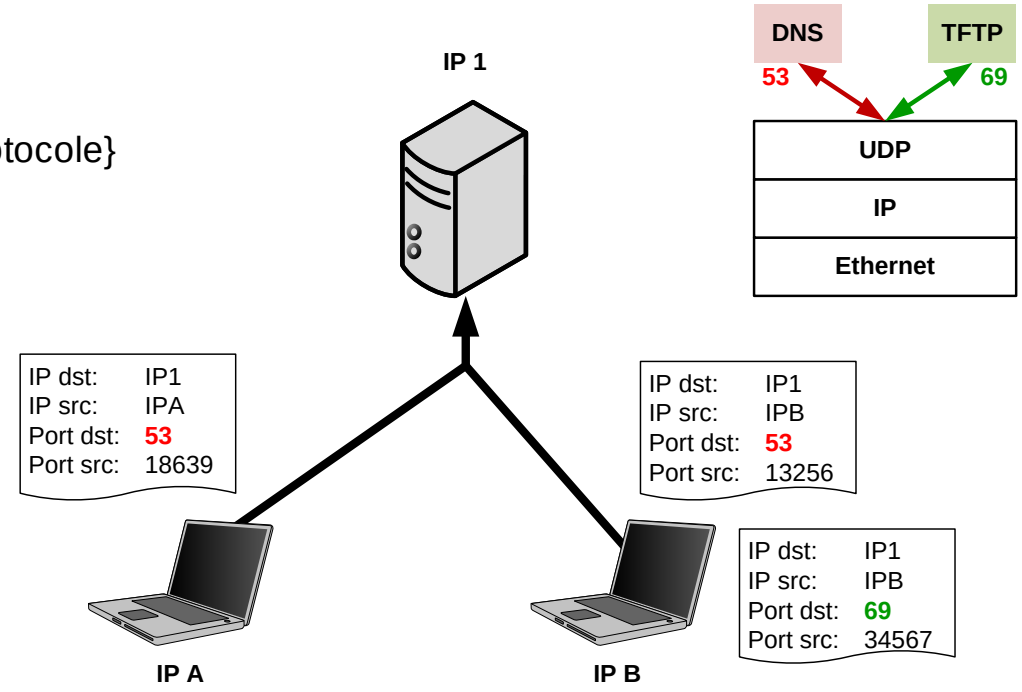


- **Propriétés**
  - **PAS** besoin d'ouvrir une connexion => rapidité, simplicité
  - **PAS** de garantie d'acheminement (les messages peuvent de perdre)
  - **PAS** de garantie de séquence (les messages peuvent arriver dans le désordre)
- **Applications**
  - Temps réel tolérant aux pertes: VoIP, diffusion vidéo temps réel….
  - Protocoles simples: DHCP, TFTP, SNMP…
  - Diffusion à plusieurs destinataires (broadcast, multicast…)

NOTION DE PORT

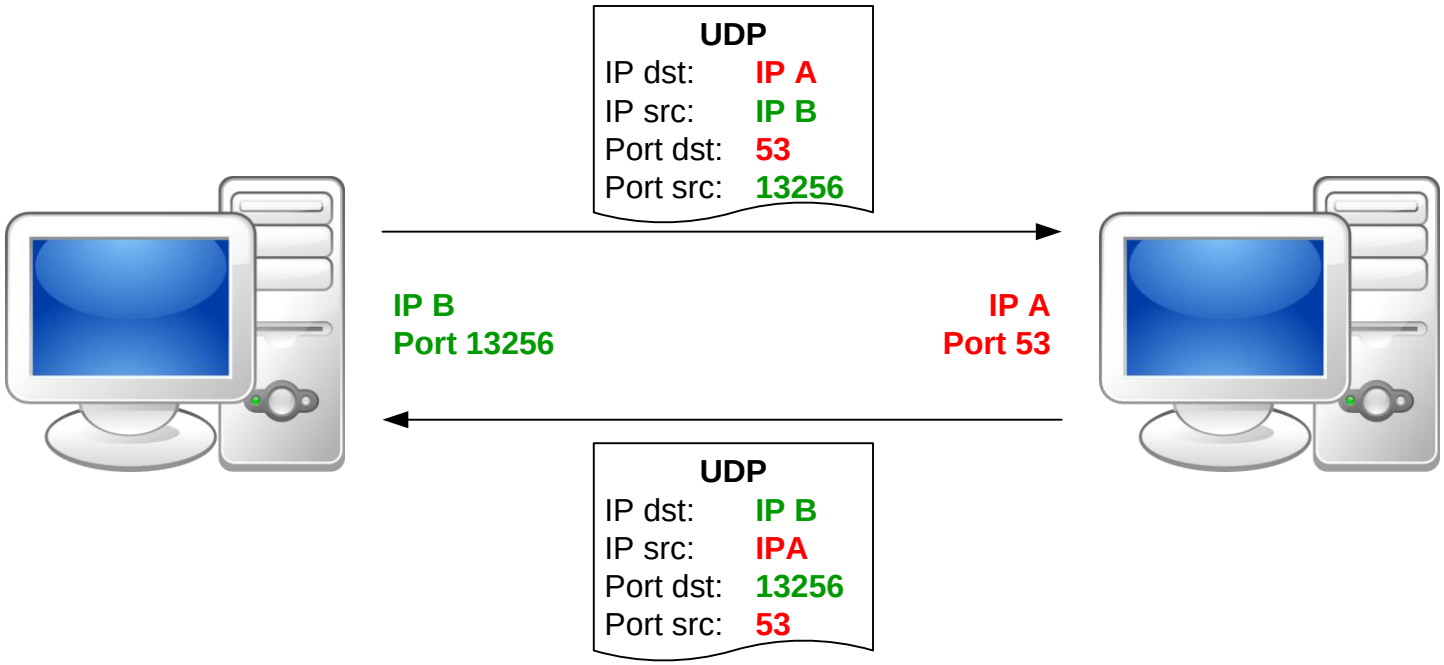
- **Valable pour UDP comme TCP**
- **N° de port = n° d'émetteur / destinataire sur la machine**
  - Plusieurs programmes peuvent utiliser UDP sur la même machine
  - Adresse service: couple (@IP, n°port) => couplage fort à IP
- **Un port destination ET un port source**
- **Identification d'une connexion**
  - {@IPsrc, @IPdst, PortSrc, PortDst, Protocole}



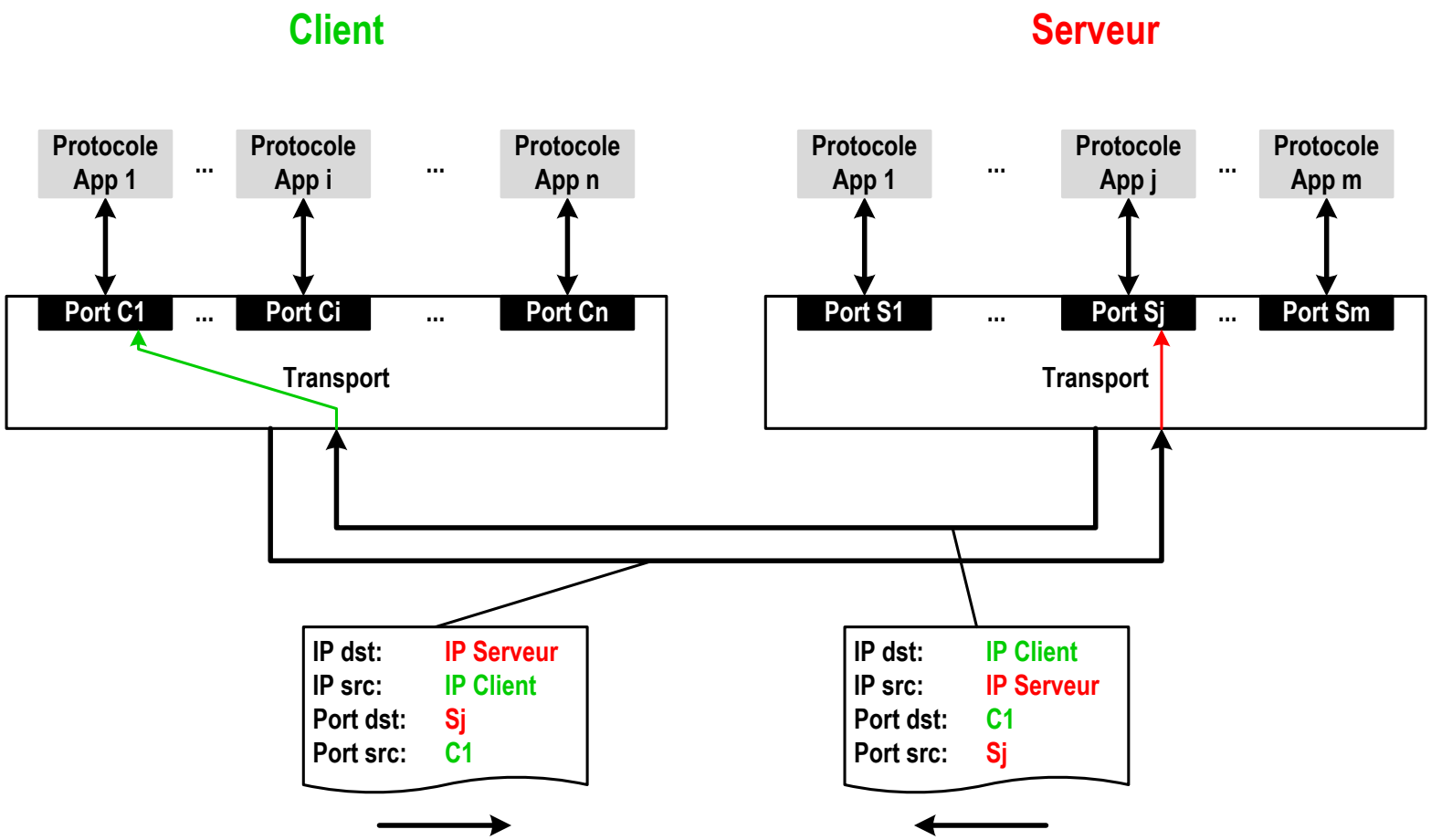


IDENTIFICATION D'UNE CONNEXION

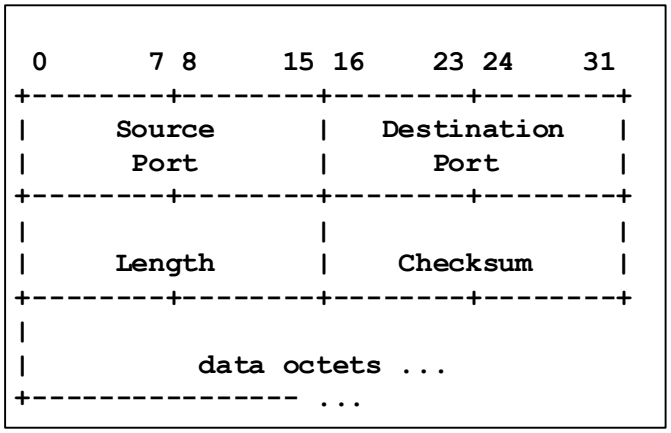
- **Identification par quintuplet**
  - @IP source, @IP destination
  - N° port source, N° port destination
  - Type de protocole de transport (UDP, TCP…)



INTERFACE SOCKET

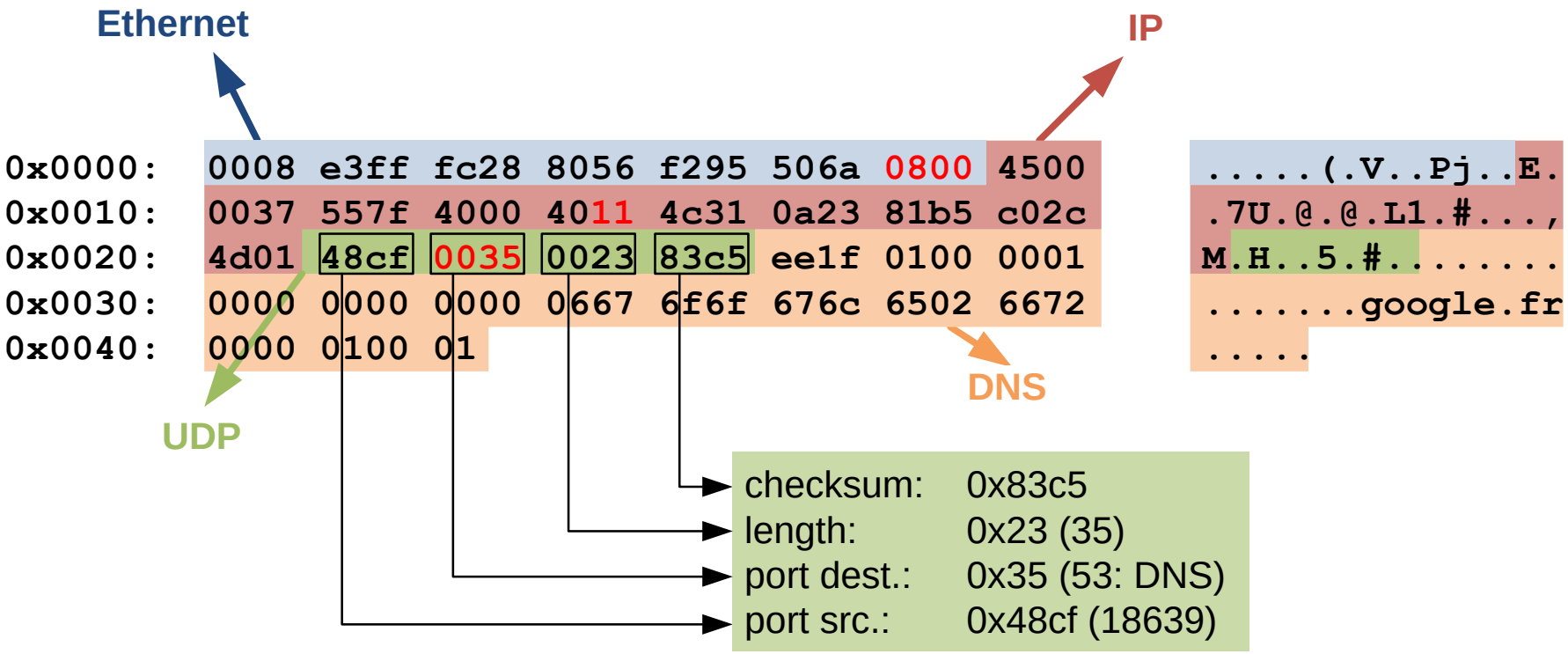


UDP: DÉTAIL DES CHAMPS



- **Port source: 16 bits**
  - Alloué par l'OS de machine source
  - Si client: tiré au hasard parmi les n° dispos
  - Si serveur: N° du service
- **Port destination: 16 bits**
  - Convenu à l'avance
    (ou lors d'un premier échange pour la voie retour)
- **Assignation des n° de ports par l'IANA (RFC 6335)**
  - 0-1023 / 1024-49151: System Ports / User Ports
  - 49152-65535: Dynamic and/or Private Ports
- **Length: 16 bits**
  - Longueur en octets du datagramme complet (entête + données)
- **Checksum: 16 bits**
  - Somme de contrôle portant sur le datagramme et un condensé de l'entête IP
  - => Dépendance à IP

UDP: DÉSASSEMBLAGE



Principales propriétés:

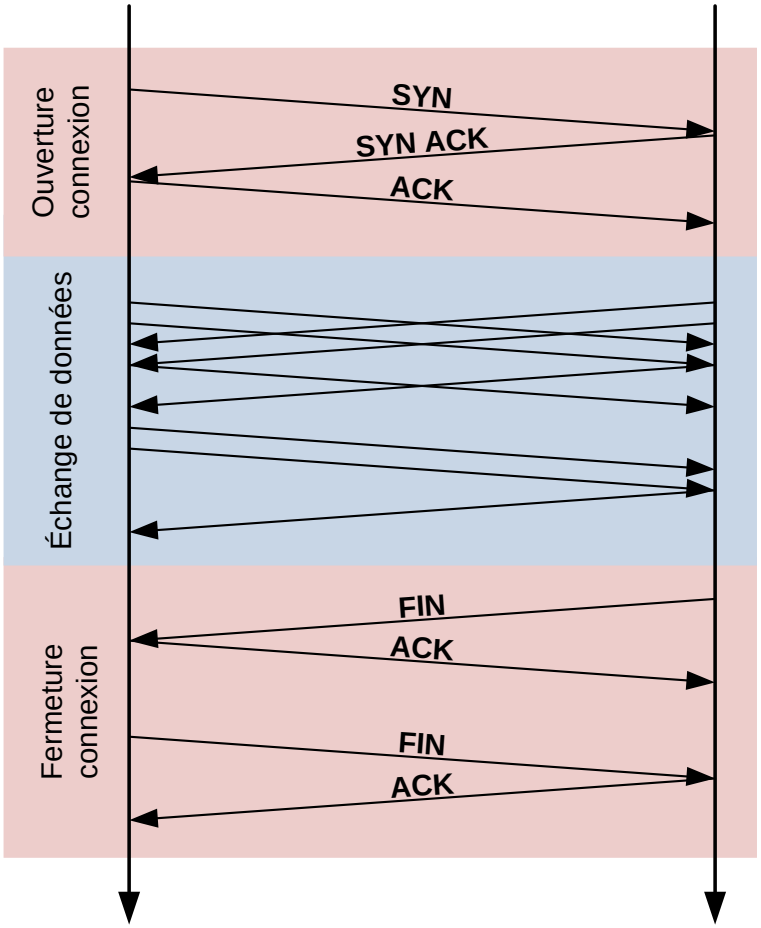
- **Mode connecté**
  - On envoie/reçoit des flux de données
  - Transparence de la taille des paquets
- **Fiabilité**
  - Retransmission en cas d'erreur
- **Garantie de séquencement**
  - Les données arrivent dans l'ordre où elles on été émises
- **Contrôle de flux**
  - Evite de saturer le récepteur
- **Contrôle de congestion**
  - Recherche de l'utilisation « optimale » des ressources disponibles
  - Partage équitable des ressources entre différents flux

TCP = Tuyau point à point

Le système d'exploitation implémente ces propriétés de façon transparente pour les applications

TCP: MODE CONNECTÉ

- **Connexion = Tuyau point à point**
- **Etablissement de la connexion**
  - Echange de signalisation en 3 étapes (Three-way handshake)
  - SYN / SYN-ACK / ACK
  - connexion / échange de paramètres
  - **Géré par le système d'exploitation**
- **Echange de données**
  - Mode flux => pas de notion de paquet
  - On pousse les données dans le tuyau
  - Données regroupées en « **segments** »
  - Besoin de repères en réception (framing)
- **Fermeture de la connexion**
  - Gracefull stop
  - 2x 2 étapes: FIN / ACK / FIN ACK
  - **Géré par le système d'exploitation**

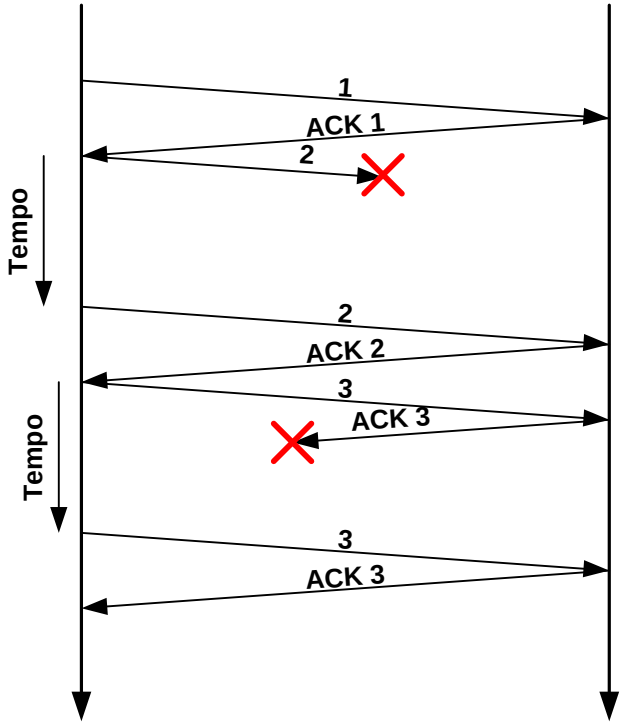


TCP: FIABILISATION

- **Détection d'erreurs**
  - Somme de contrôle
- **Acquittement des segments reçus**
- **Retransmission si erreur ou perte**
- **Go back N**
  - Plusieurs segments émis simultanément
  - Récepteur indique sa position dans la séquence reçue

FIABILISATION: SEND & WAIT

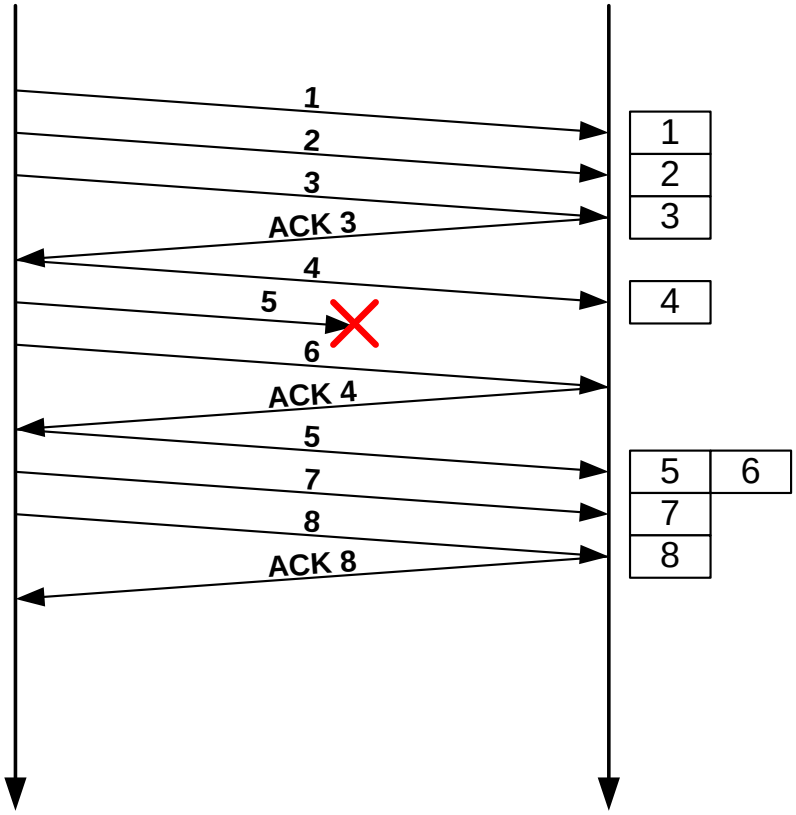
- **Emission d'un paquet**
- **Attente de l'acquittement (ACK)**
- **Si OK**
  - Emission du paquet suivant
- **Si pas d'ACK après un délais donné**
  - Réémission de la donnée
- **Besoin de dimensionner la temporisation de réémission**
  - En TCP: Algo dynamique
  - Basé sur mesure du RTT (Round Trip Time)
- **TCP utilise un AUTRE mode (Go Back N)**



Exercice: Faut-il numéroter les messages? Et les ACK?

FIABILISATION: GO BACK N

- **Emission consécutive de plusieurs segments**
  - Meilleure efficacité
  - Nombre max = « taille de fenêtre »
- **Récepteur indique le plus grand n° de segment consécutif reçu**
  - Possibilité de « sauter » des ACK
  - Rq.: TCP indique un N° de séquence et non un n° de segment
- **Réémission à partir du dernier ACK**
- **Le récepteur « bloque » les segments hors séquence**
- **Besoin de dimensionner la fenêtre**
  - -> Contrôle de congestion



CONTRÔLE DE CONGESTION

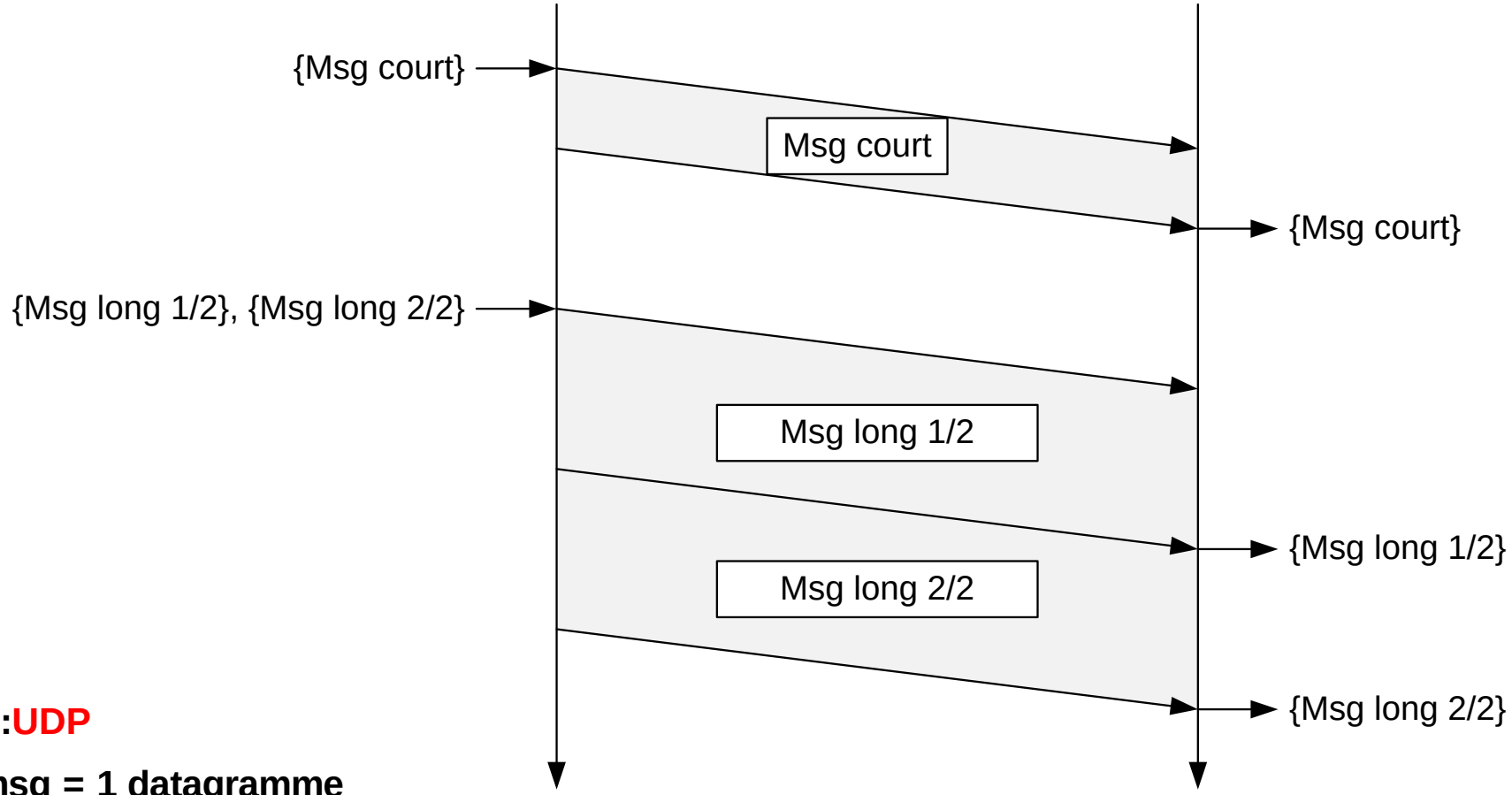
- **Congestion = saturation du réseau**
- **Différentes causes**
  - Capacité faible / débit nécessaire
  - Panne / perturbations
  - Changement de route…
- **Contrôle de congestion**
  - Adapter le trafic à la capacité du réseau
  - Fluctue en temps réel
- **Fonctionnement en TCP**
  - Hypothèse: En filaire perte de paquet ~= congestion
  - Si pas de perte => augmentation de la taille de fenêtre
  - Si perte => diminution de la fenêtre
  - Algo complexe qui dépasse le cadre du cours

CONTRÔLE DE FLUX

- **Éviter de saturer le récepteur**
  - Capacité de traitement limitée sur le récepteur
  - Capacité mémoire
  - Par ex: petite machine, ou soumise à forte charge

- **Le récepteur indique la taille de données qu'il peut recevoir**
  - L'émetteur ajuste son débit en conséquence

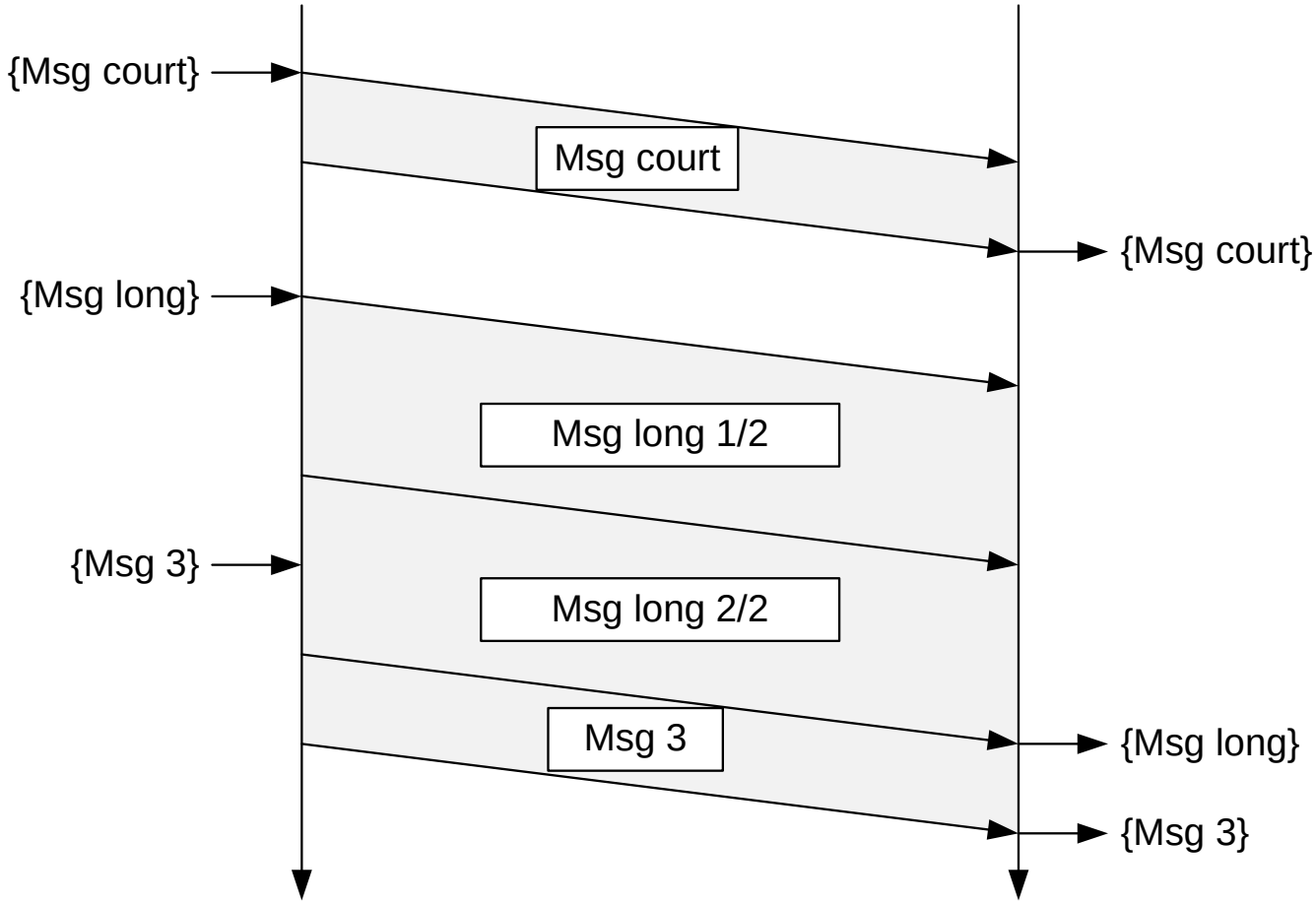
MODE DATAGRAMME



- **Ex.:UDP**
- **1 msg = 1 datagramme**

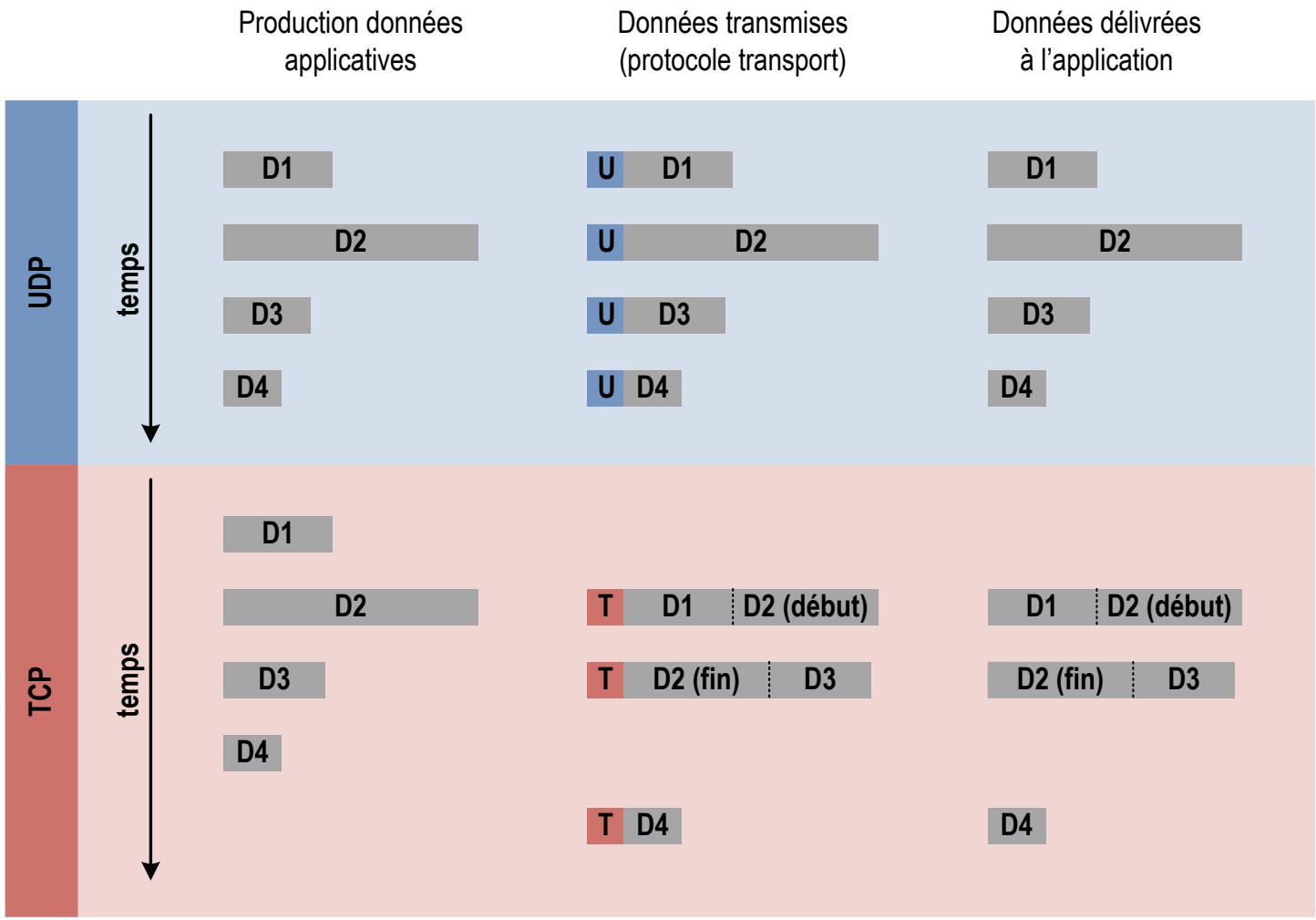
  => segmentation par l'application

MODE FLUX



- **Ex.: TCP**
- **Segmentation par la couche transport**
  => besoin de délimiter les messages

COMPARAISON DATAGRAMME / FLUX



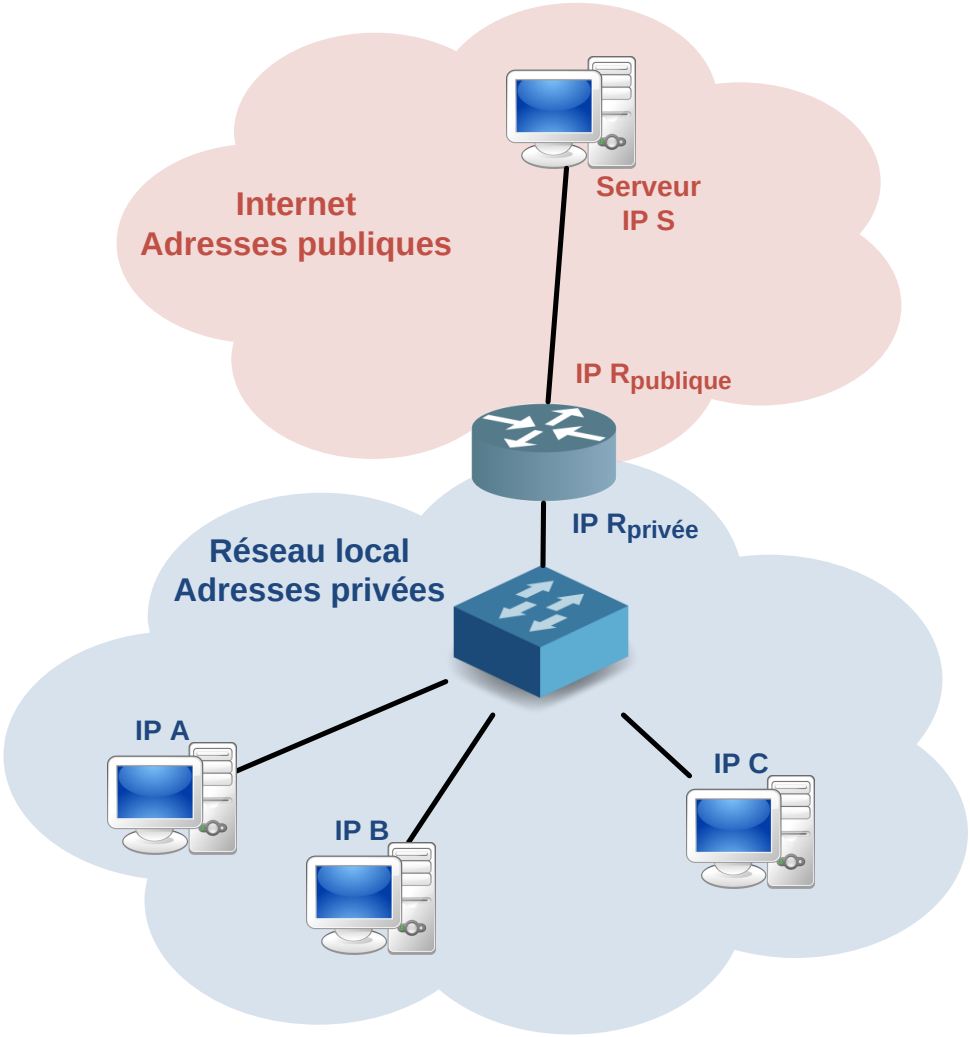
NAT

- **Network Address Translation**

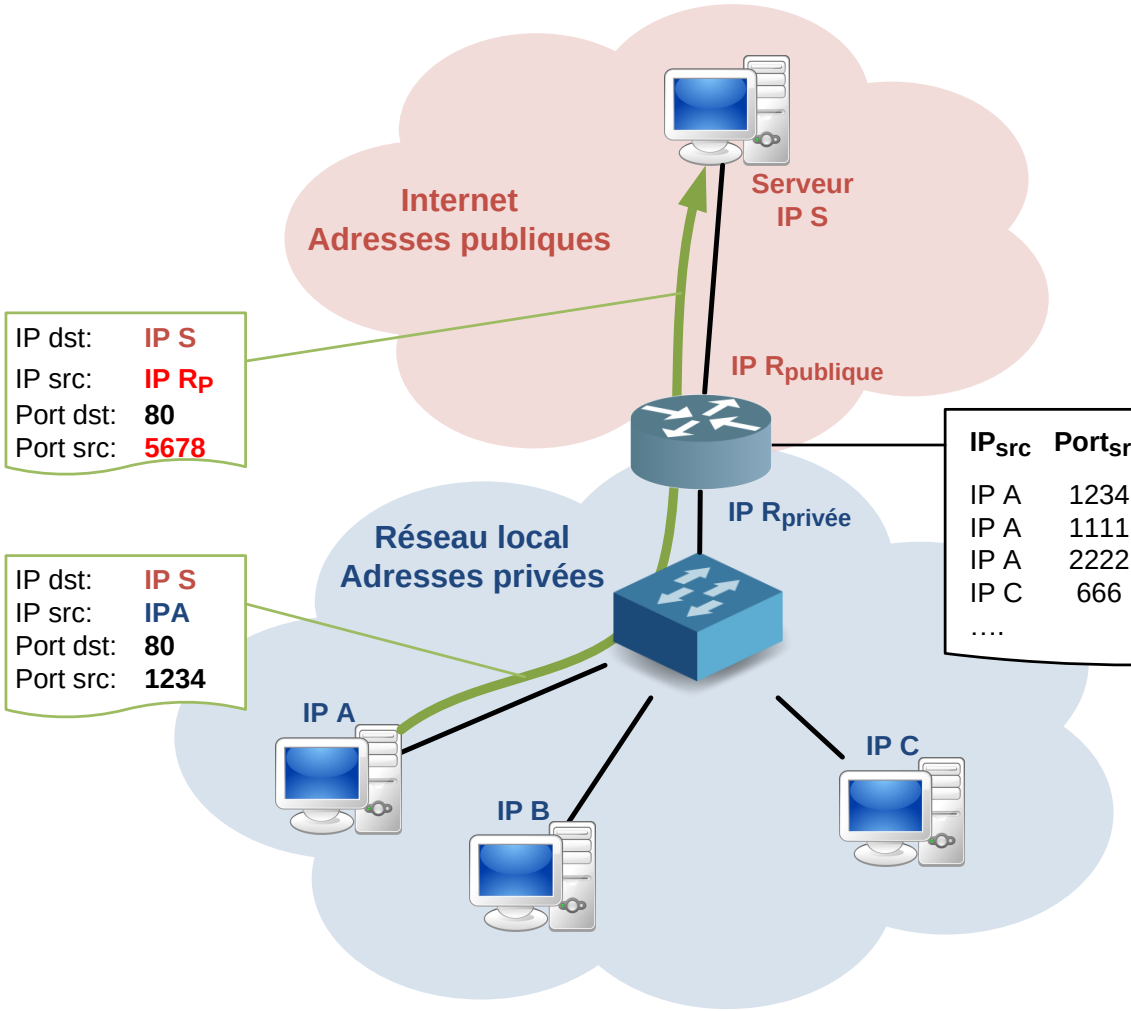
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

NAT: NETWORK ADDRESS TRANSLATION (1/2)

- **Problème**
  - Raréfaction des IPv4 publiques
  - Multiplication des machines
- **Solution proposée par NAT**
  - Un routeur « prête » son IP publique aux machines de son réseau local
- **Vu des machines**
  - C'est transparent
- **Vu du serveur**
  - Plusieurs connections avec une même machine (IP publique du routeur) sur des ports différents
- **Le routeur assure la « translation »**
  - Table de correspondance
  - Fonctionnement « **statefull** »
  - Différents algorithmes possibles



NAT: NETWORK ADDRESS TRANSLATION (2/2)



IP_{src}	$Port_{src}$	IP_{dst}	$Port_{dst}$	Proto	$Port_{modif}$
IP A	1234	IP S	80	TCP	5678
IP A	1111	IP S	80	TCP	5555
IP A	2222	IP S1	53	UDP	6423
IP C	666	IP S	80	TCP	123
….					

- **Table de correspondance**
  - Une entrée par connexion
    => statefull

INTRODUCTION AU ROUTAGE

- **Routage en réseau local**
- **Passerelle par défaut**
- **Principes de base du routage**

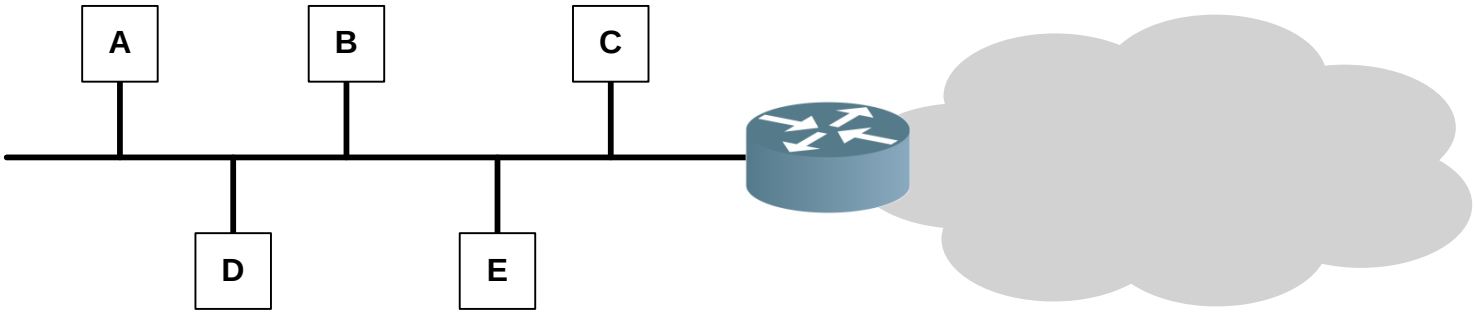
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

- **Chaque host connaît:**
  - Son adresse IP
  - Son masque de sous réseau (ie taille de préfixe) => son @ de SR
  - L'adresse du routeur par défaut

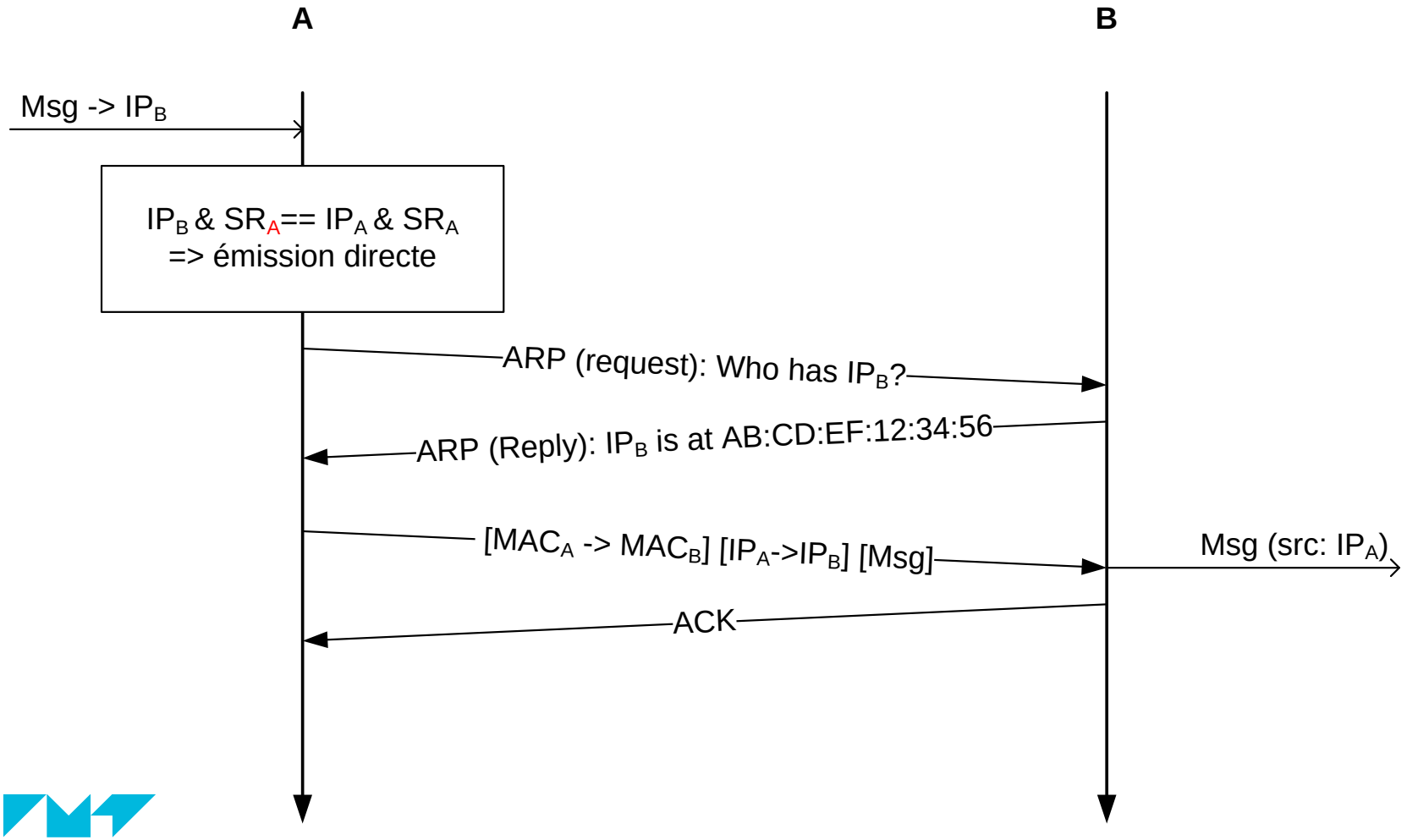
- **Algorithme:**
  - Pour tout paquet sortant:
    regarder destination
    calculer @SR de destination
  - Si SR_{dest}==SR_{local} :
    rechercher @MAC du destinataire (requête ARP)
    envoyer le paquet à l'@MAC du destinataire
  - Sinon:
    envoyer à l'@MAC sur routeur par défaut

ROUTAGE EN RÉSEAU LOCAL

CAS n°1: source et destination dans le même sous-réseau

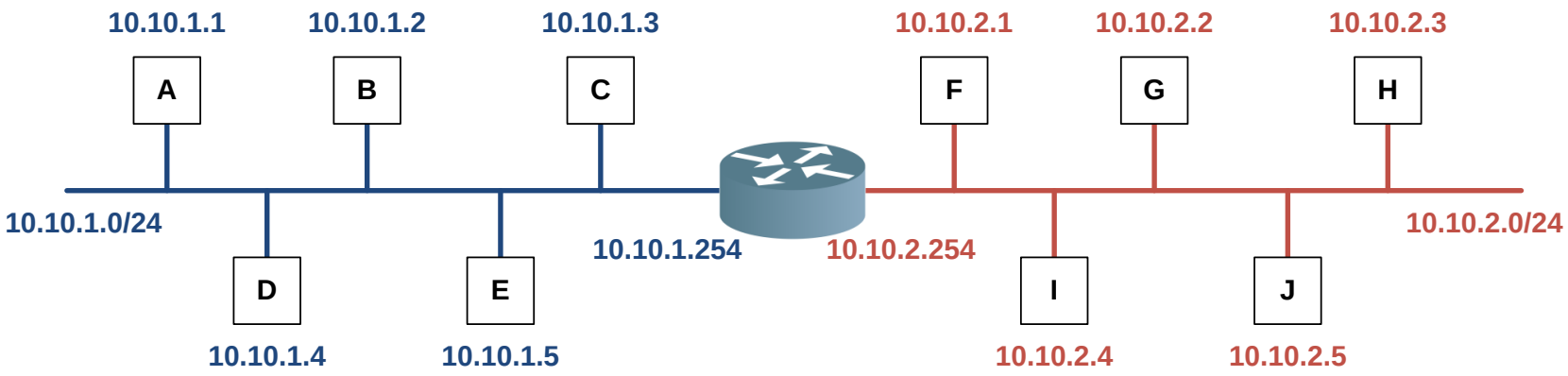


- **Résolution ARP puis émission au niveau 2**



ROUTAGE EN RÉSEAU LOCAL

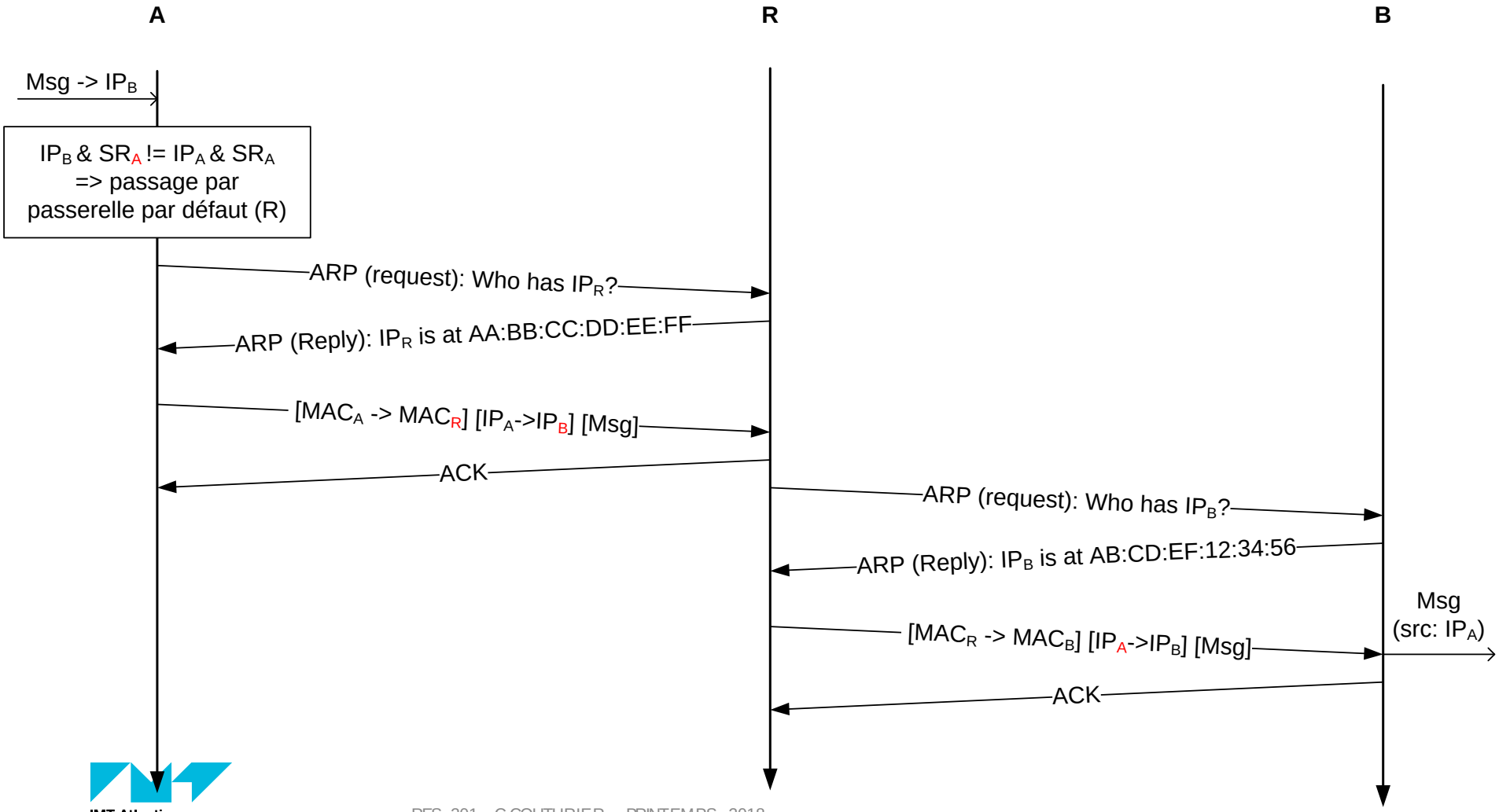
CAS n°2: source et destination dans des sous-réseaux différents



ROUTAGE EN RÉSEAU LOCAL

CAS n°2: source et destination dans des sous-réseaux différents

- **Émission au niveau 2 vers le Routeur par défaut**



PRINCIPE DE FONCTIONNEMENT D'UN ROUTEUR

Routeur = « Rond Point » de l'internet

- **Un routeur a plusieurs interfaces**
- **Table de commutation (forwarding table)**
  - Abusivement appelée « table de routage (routing table) »
  - Liste des destinations
  - Pour chaque destination:
    si la destination appartient au SR de l'interface
    => émission directe
    sinon
    => envoi au prochain routeur (next hop)
  - Les routes sont parcourues de la plus précise (plus grand préfixe) à la moins précise
  - Route par défaut: 0.0.0.0/0
- **Destination = plage d'adresse => adresse / préfixe (agrégation)**
- **La table de routage peut être construite**
  - De marinière statique (configuration manuelle)
  - Automatiquement (par des protocoles de routage, hors cadre du cours)

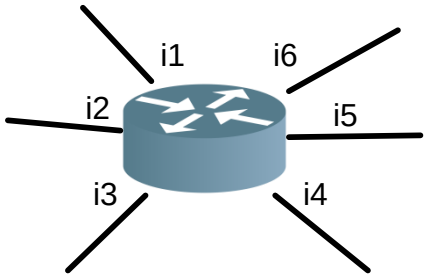
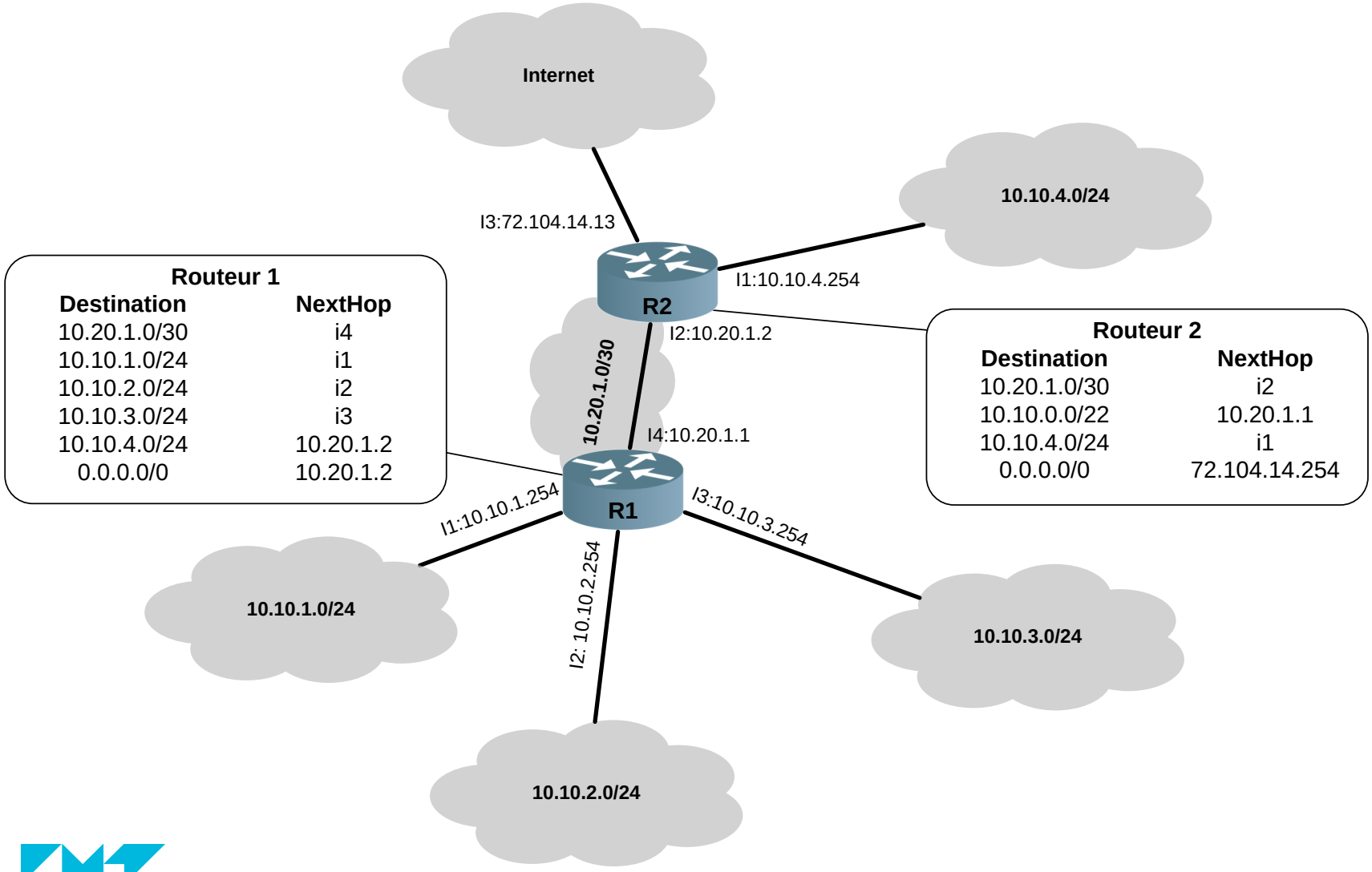


TABLE DE ROUTAGE



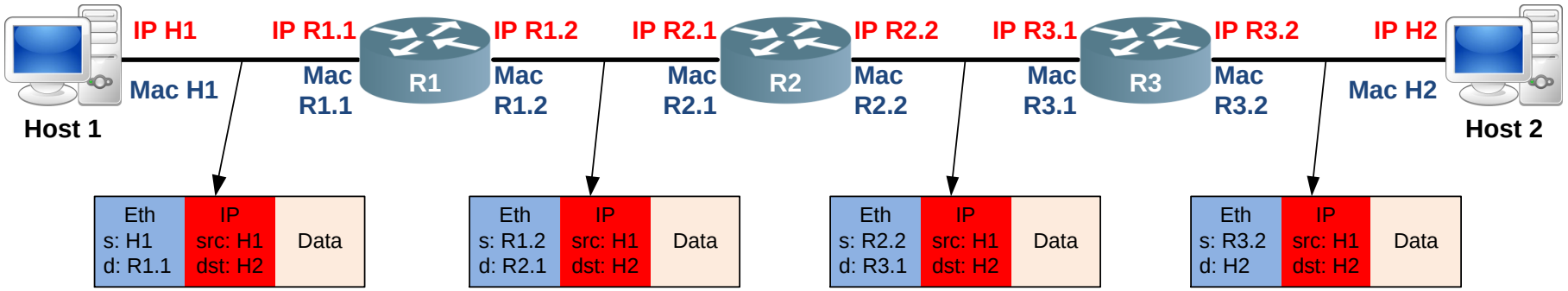
Routeur 1

Destination	NextHop
10.20.1.0/30	i4
10.10.1.0/24	i1
10.10.2.0/24	i2
10.10.3.0/24	i3
10.10.4.0/24	10.20.1.2
0.0.0.0/0	10.20.1.2

Routeur 2

Destination	NextHop
10.20.1.0/30	i2
10.10.0.0/22	10.20.1.1
10.10.4.0/24	i1
0.0.0.0/0	72.104.14.254

ADRESSES L2 / L3



LE PROTOCOLE IPv4

- Fonctionnalités / Propriétés
- Format de paquet IPv4
- Fragmentation
- TTL

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

IPV4: FONCTIONNALITÉS

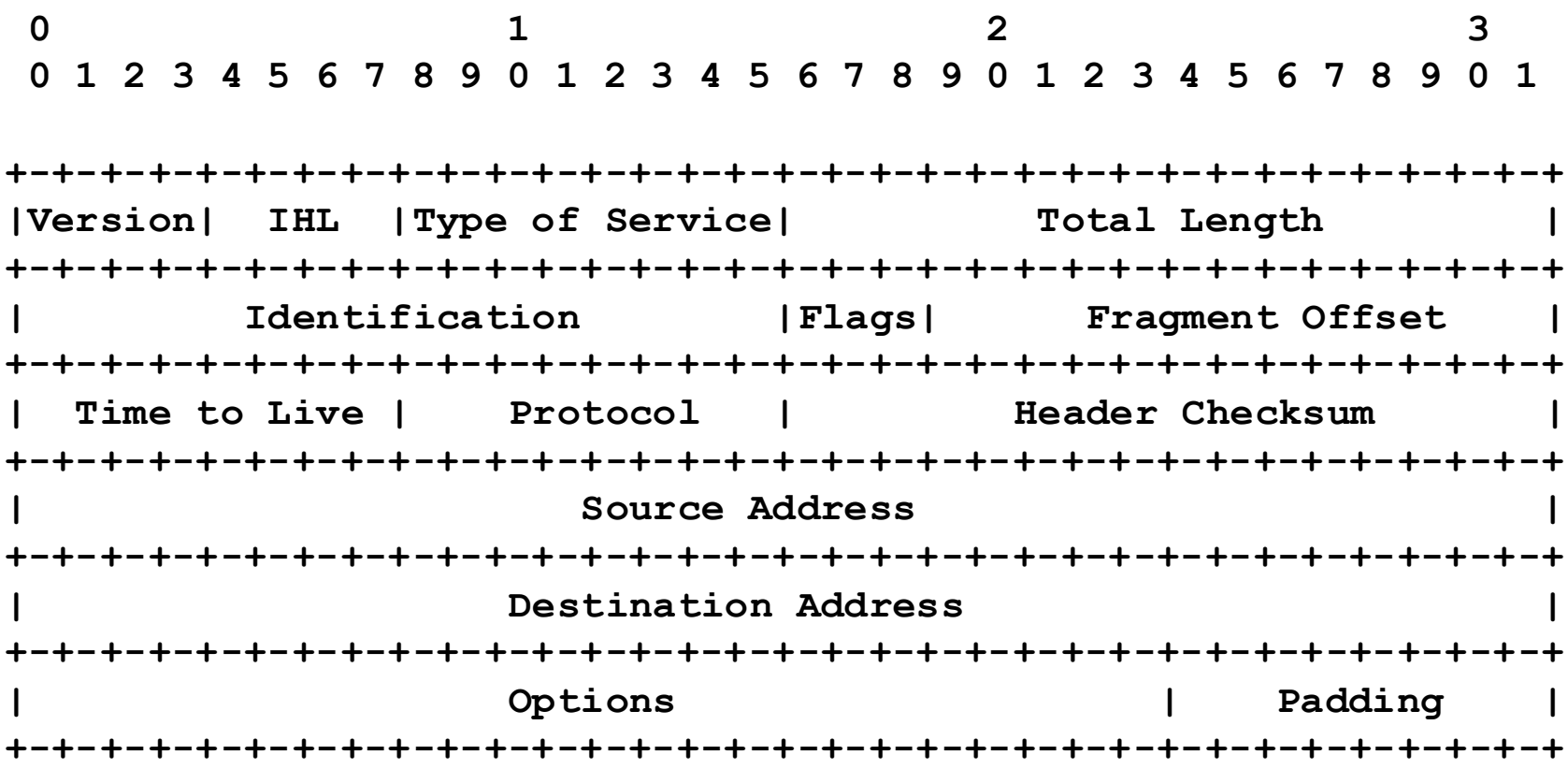
- **Adressage**
  - Identifiant unique à tous les nœuds
  - Adressage « universel »: indépendant du réseau et de la techno
- **Routage**
  - Acheminer les paquets en les « aiguillant » à chaque nœud
  - Assuré par les routeurs (« ronds points de l'Internet »)
- **Qualité de service (QoS)**
  - Identifier des « flux » présentant des besoins différents (latence, débit, coûts…) et les traiter en conséquence
- **Fragmentation / Assemblage**
  - Adapter la taille des paquets à la capacité de chaque lien (MTU: Maximum Transport Unit)
- **Autres fonctions**
  - Sécurité (IPsec) -> VPN
  - Mobilité (MobileIP) -> conserver son adresse en itinérance
  - Mieux intégrées en IPv6

IPv4: PROPRIÉTÉS

Mode datagramme

- **Non fiable**
  - Pertes (congestion, erreurs…)
  - Pas de répétition
- **Routage non déterministe**
  - 2 paquets vers un même destinataire peuvent suivre différents chemins
  - Le chemin aller peut être différent du retour
- **Pas de garantie d'ordre**
  - Retards variables (QoS peut améliorer un peu)
  - Routage non déterministe
- **Duplication possible**
  - Rare mais possible
  - Ex.: reprise de panne

IPv4: FORMAT DE PAQUET (1/3)



Taille entête (sans option) : 20 octets

IPv4: FORMAT DE PAQUET (2/3)

- **Version (4 bits)**
  - =4
- **IHL: Internet Header Length (4 bits)**
  - Longueur de l'entête en mots de 32 bits (4 oct)
- **ToS: Type of Service (8 bits)**
  - Spécification du niveau de QoS via des « codes de service » (DSCP)
- **Total Length (16 bits)**
  - Longueur totale du paquet (en octets), entête compris
- **Source / destination address (2x 32 bits)**
  - Adresses IP source et destination
- **Protocol (8 bits)**
  - Identifie le protocole (SAP) des données utiles
  - Ex: TCP=6 / UDP=17 / ICMP = 1

IPv4: FORMAT DE PAQUET (3/3)

- **Options / Padding**
  - Peu utilisé
  - Padding = bourrage pour maintenir l'alignement
- **Header Checksum (16 bits)**
  - Somme de contrôle de l'intégrité de l'entête (cpt à 1 de la somme des octets)
  - Si erroné, le paquet est jeté (drop)

- **TTL: Time To live (8 bits)**
  - Pour éviter les boucles de routage (**voir plus loin**)
- **Identification / Flags / Fragment Offset (total 32 bits)**
  - Pour gérer la fragmentation (**voir plus loin**)

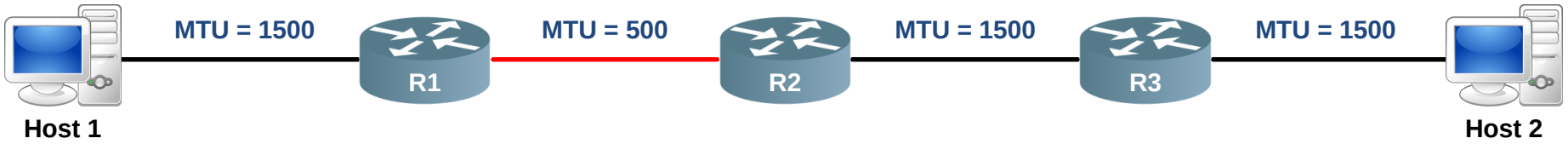
FRAGMENTATION 1/2

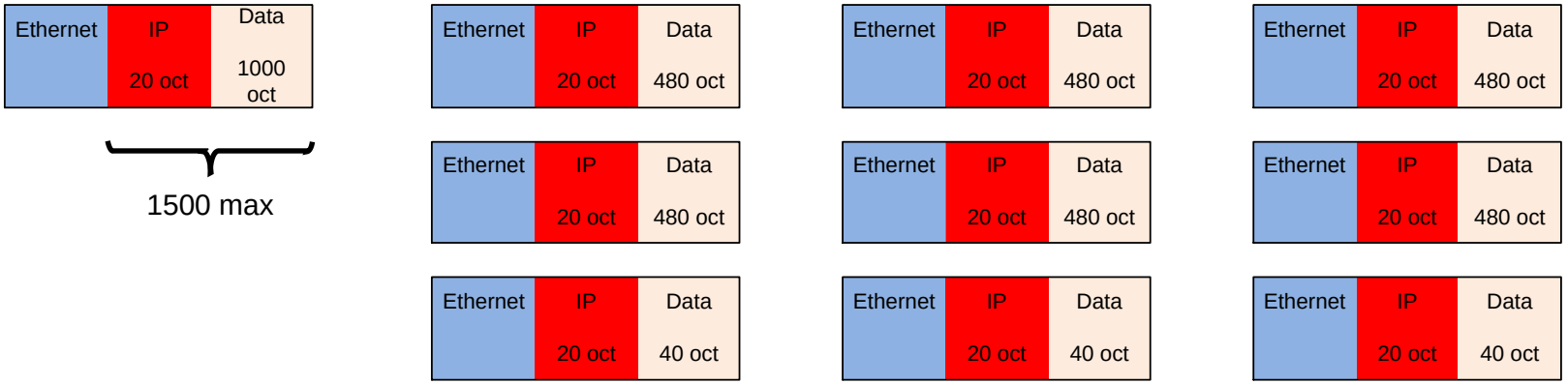
- **Besoin**
  - Toutes les couches de niveau 2 n'ont pas la même taille de trame maximale
  - MTU: Maximum Transport Unit
  - => si taille paquet IP > MTU
    la couche IP doit fragmenter le paquet sur plusieurs trames
  - Rq. MTU diminue avec l'encapsulation (tunnels)

Champs de l'entête:

- **Identification (16 bits)**
  - N° identique pour tous les fragments d'un même paquet
- **Fragment Offset (13 bits)**
  - Position du fragment dans le paquet
- **Flags (3 bits)**
  - Bit 0: réservé
  - Bit 1: DF: Don't Fragment: => Ne pas fragmenter le paquet
  - Bit 2: MF: More Fragment => 0: dernier fragment / 1: pas le dernier

FRAGMENTATION





- **Fragmentation**
  - Avant le lien limitant
- **Assemblage**
  - Par le **destinataire**

BOUCLES DE ROUTAGE: TTL

- **Elaboration des routes**
  - Statique ou dynamique
  - Dans tous les cas des erreurs peuvent se produire (continues ou transitoires)
  - Et former des **boucles**

- **Boucles**
  - Les paquets qui y rentrent n'en ressortent jamais!
  - => Avalanche de trafic

- **Champ TTL: Time to Live (8 bits)**
  - Décrémenté à chaque routeur
  - Si TTL = 0 => paquet supprimé (drop)
  - La source PEUT etre prévenue par un message ICMP Time Exceeded
  - Utilisé par **traceroute**

EXERCICE

- **Au cours du trajet d'un paquet, les champs de l'entête IP sont-ils modifiés?**

LE PROTOCOLE ICMP

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PROTOCOLE ICMP

- **ICMP: Internet Control Message Protocol**
  - Protocole de contrôle
  - Messages de test ou d'erreur
  - Pas de données « utiles »
  - Ne sert généralement pas à l'utilisateur final (sauf ping et traceroute)
- **Jeu de messages**
  - Cf. RFC 792 ou page wikipedia
  - Echo request / reply -> Ping
  - Time Exceeded -> TTL = 0
  - Destination Unreachable
  - ….
- **Activation optionnelle**
  - Dépend de la politique de sécurité

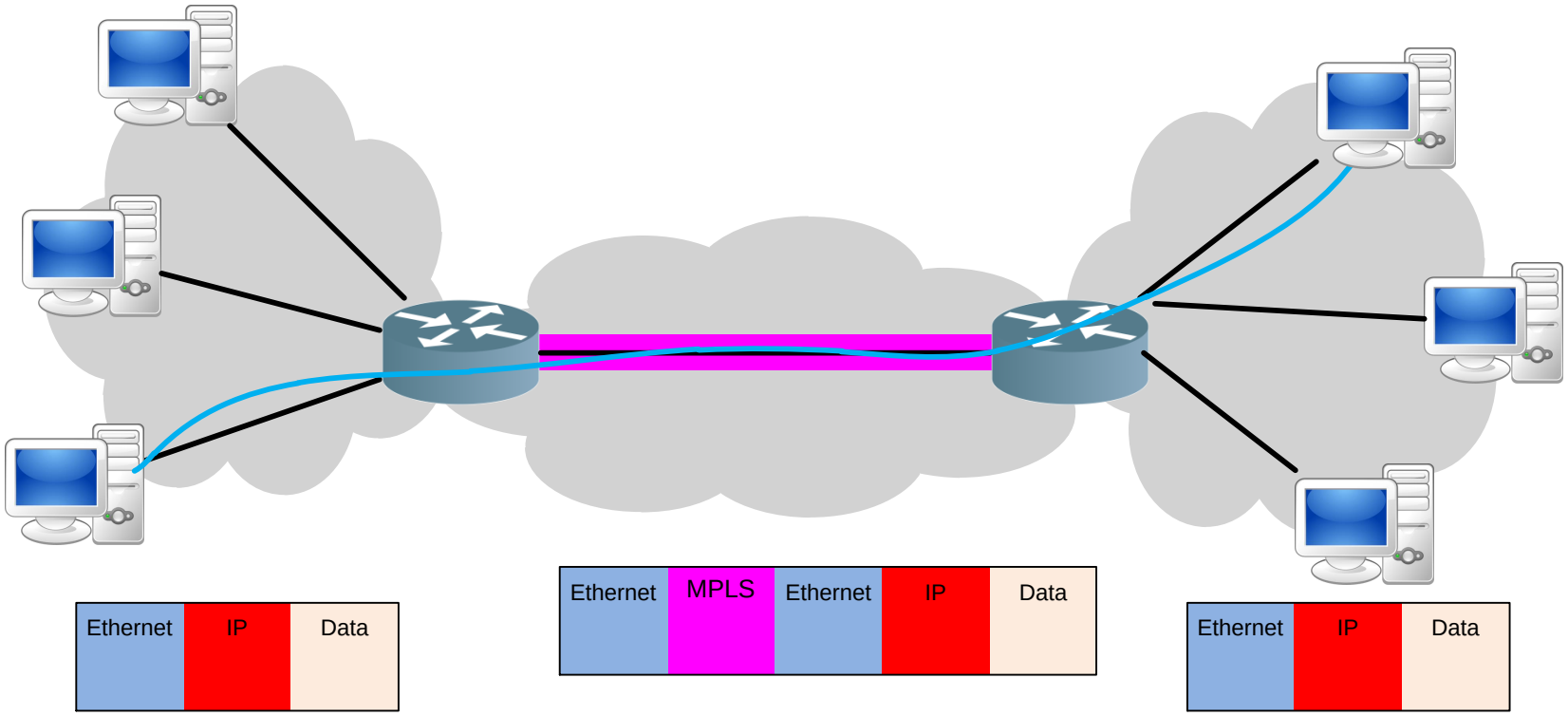
NOTION DE TUNNEL

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

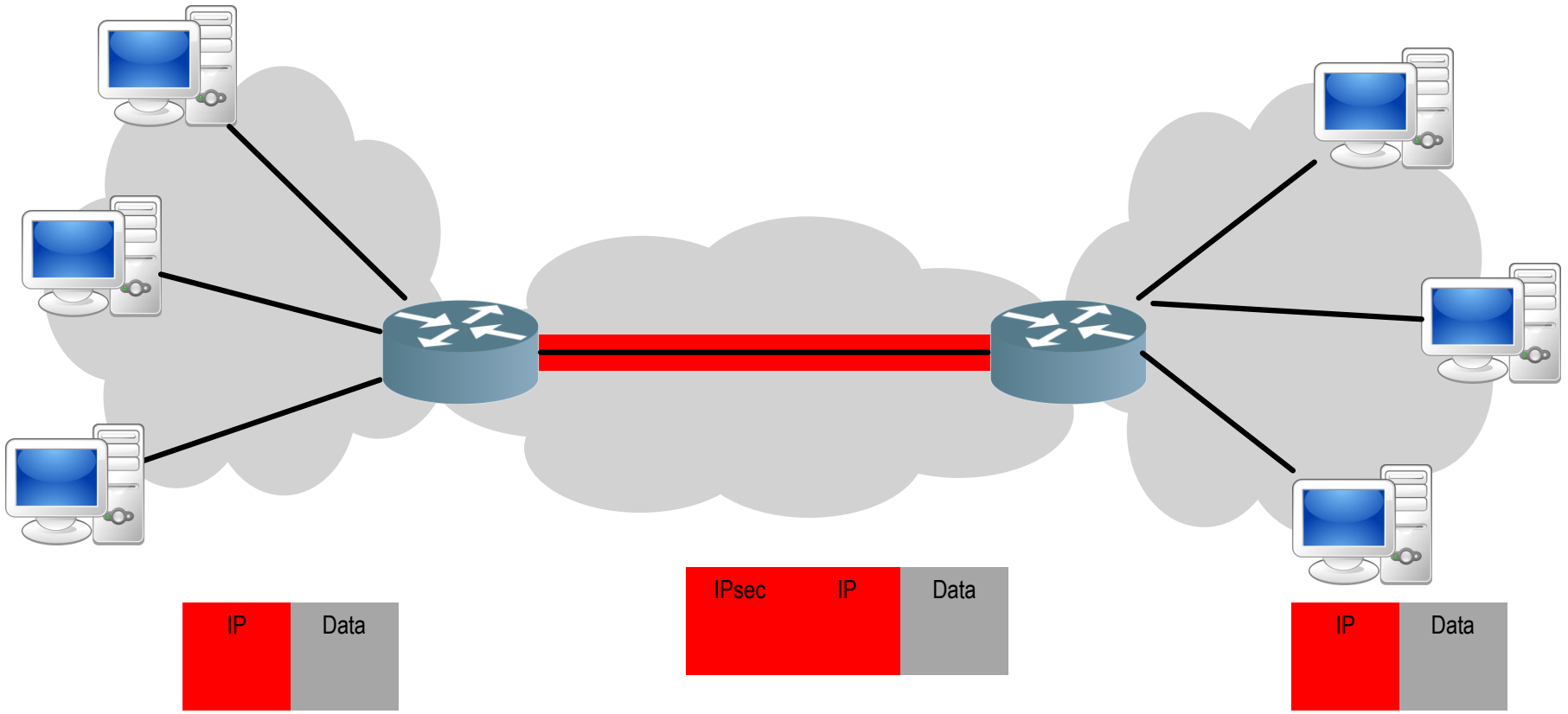
TUNNELS

- **Tunnel = Encapsulation**
  - Pour traverser des portions de réseau,
    on peut avoir besoin d' « adapter » un protocole
  - Pour cela: on encapsule le trafic dans le protocole cible
- **Exemples**
  - Tunnel IPsec: Chiffrement d'une connexion sur un tronçon donné
  - 6 in 4: transit de trafic IPv6 sur un réseau IPv4
  - 4 in 6: transit de trafic IPv4 sur un réseau IPv6
  - MPLS: transit de n'importe quel protocole sur un cœur de réseau
  - …
- **Impacts**
  - Augmentation de la charge protocolaire (overhead)
  - Diminution du MTU
  - Cout de traitement par les routeurs
  - Contournement de certains problèmes

TUNNELS: MPLS



TUNNELS: IPSEC



CAPTURE & ANALYSE

- **Tcpdump & Wireshark**

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TCPDUMP ET WIRESHARK

- **Logiciels libres**
  - Linux / Windows / MAC
  - Nécessite droits admin pour capture
- **Capture**
  - Trafic entrant/sortant au niveau 2
  - Enregistrement dans fichier PCAP
  - Exemples sur Moodle et internet
- **Analyse**
  - Désassemblage des trame
  - Analyse protocolaire
  - Reconstitution des connexions
  - …