

# Technologies de l'Internet partie I

**Eugen Dedu**

Maître de conférences  
Univ. de Franche-Comté, IUT Nord Franche-Comté  
Dépt. R&T, 1ère année  
Montbéliard, France  
janvier 2025

<https://dedu.fr>  
[eugen.dedu@univ-fcomte.fr](mailto:eugen.dedu@univ-fcomte.fr)

# Intégration dans la formation R&T

- Partie I :
  - formation initiale : 6h CM, 7h30 TD, 15h TP
  - formation par apprentissage : 6h CM, 6h TD, 10h30 TP
- Partie II : ...
- À la fin de la partie II, deux examens : théorique QCM et pratique
  
- Objectifs :
  - CM, TD : comprendre comment les paquets sont routés sur Internet pour les faire arriver à leur destination
  - TP : savoir configurer des machines et des routeurs pour assurer le routage des paquets

# Plan du cours

On voit 5 réseaux, dont 3 avec ordi et 2 entre routeurs.

Exemples pour les 3 réseaux avec ordinateurs : Besançon, Belfort, Montbéliard, ou RT/MMI/MP.

On suppose que tout le matériel (machines, câbles) est en place.

A (en haut) souhaite envoyer le message "Bonjour" à B (bas-droite).

1. On suppose que toutes les machines ont une @ip et que toutes les machines connaissent le réseau entièrement. Quelles sont les données que la carte réseau de A injecte dans le réseau pour que ce message arrive à B, et quelles données circulent sur les autres liens/réseaux ? => **encapsulation**, modèle multi-couche, en-têtes

1a. Que se passe-t-il si l'adresse de B n'existe pas ? => **ICMP**

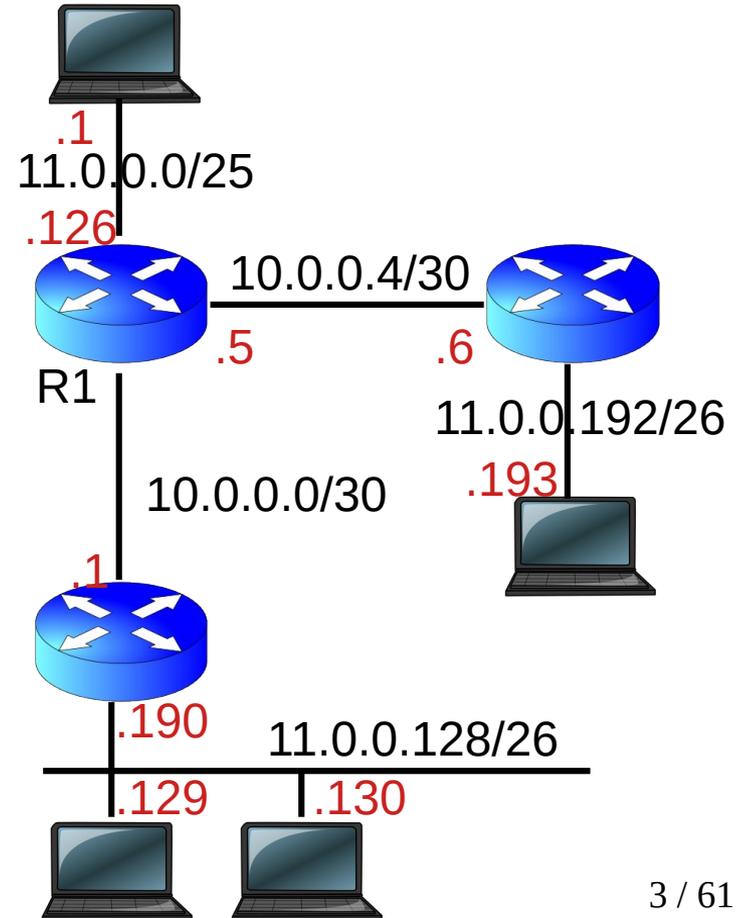
1b. Comment A spécifie/désigne B ? => **DNS**

2. Quelles adresses l'administrateur affecte aux machines/routeurs du réseau ? => **adressage IP (plan IP, @privées, masques)**

3. Comment l'admin affecte les adresses aux machines ? => **DHCP**

4. Comment les routeurs routent (aiguillent) les paquets pour arriver à B ? => **tables de routage**

5. Que faire si on ajoute d'autres machines, avec @privées ? => **NAT**



# Plan du cours (SAUTER)

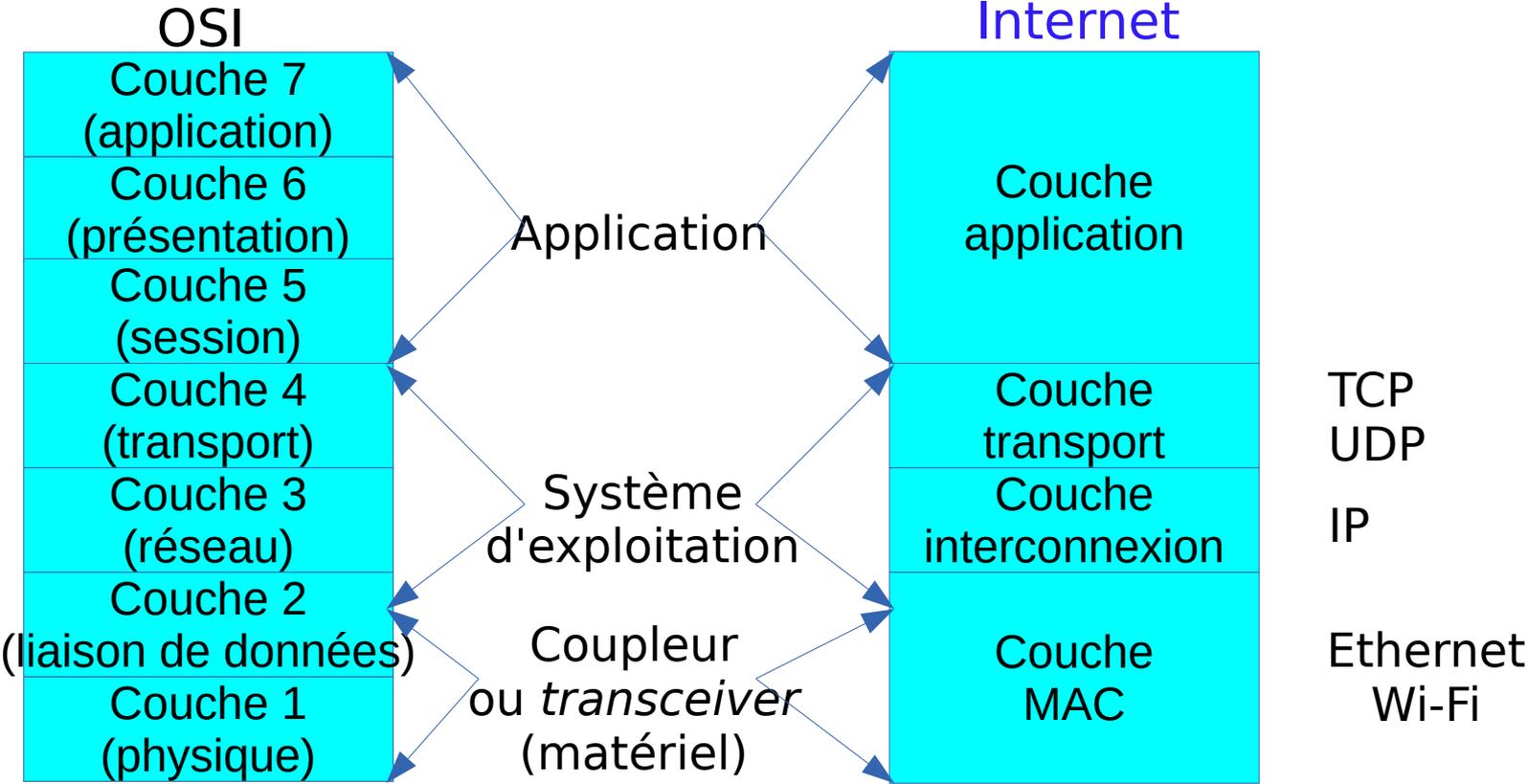
- Encapsulation des en-têtes (couche MAC, couche transport, en-tête IP, ICMP, DNS)
- Adressage IP (adresses privées, sous-adressage, plan d'adressage IP)
- Tables de routage
- Multicast, DHCP, NAT

## Partie II :

- routage dynamique interne (RIP et OSPF)
- filtrage ACL (listes d'accès)
- réseaux d'opérateurs (2ème année) : BGP et MPLS
- réseaux de campus : IPv6 ?

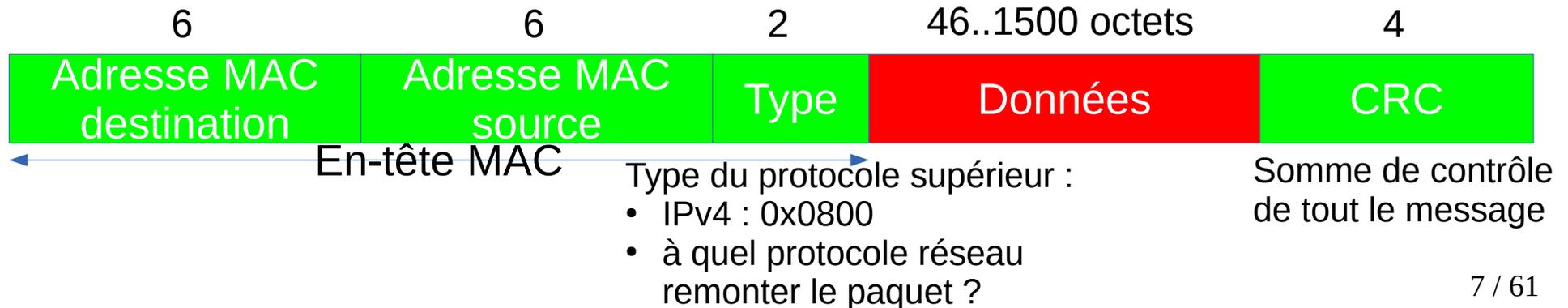
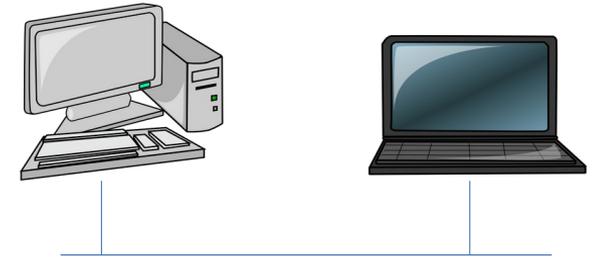
# Encapsulation des en-têtes

# Modèle OSI/Internet



# Début d'Internet : transmission couche MAC dans un câble (en Ethernet)

- Une machine veut envoyer un message à une autre machine
- Si les machines sont sur le même médium, la couche MAC de la source :
  - rajoute aux données un en-tête MAC avec l'adresse du destinataire
  - ensuite le met sur le médium
- Trame envoyée pour un câble (Ethernet type II) :



# Adresse MAC

- Chaque carte réseau a inscrit un numéro unique, appelé adresse MAC, adresse Ethernet ou adresse physique
  - MAC (*Medium Access Control*), contrôle d'accès au médium de communication
- Un ordinateur possède une adresse MAC pour chacune des cartes réseau
- Présentation de mes adresses MAC (ip addr)
- L'adresse avec tous les bits à 1 (48 bits à 1) est le broadcast

# Besoin de la couche 3 IP (réseau)

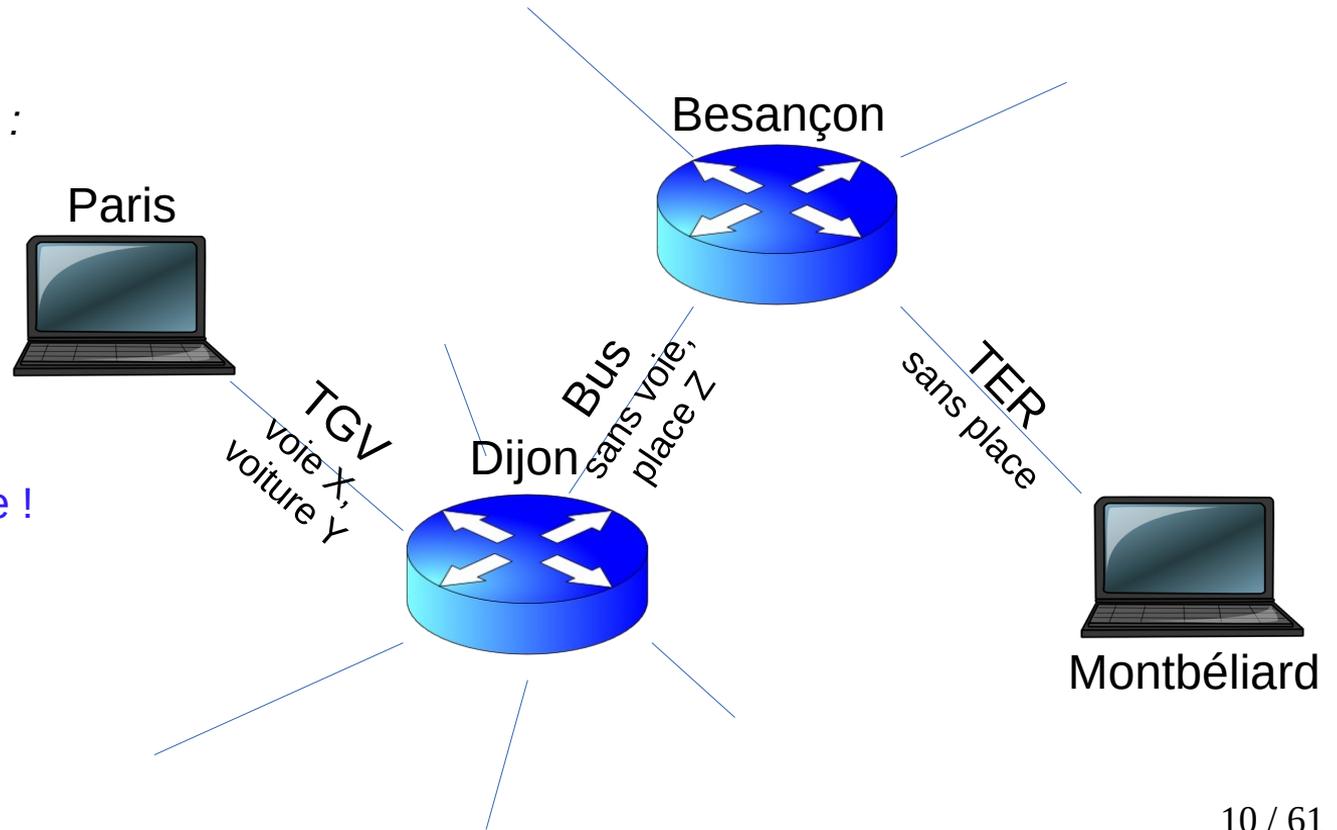
- Les adresses MAC permettent d'échanger des messages sur un même médium de communication
- Comment transférer des messages entre des machines lointaines ?
- IP (Internet Protocol) assure le routage, donc la connaissance des machines sur la route entre la source et la destination
- Chaque machine est identifiée par une adresse IP, voir plus loin
- Version couramment utilisée : IPv4
  - version qui lui succédera : IPv6, présentée plus tard dans une autre matière

# Vue d'ensemble d'une transmission, couches 2 et 3, exemple

*Analogie avec réception de courrier de Paris à Montbéliard :*

À l'attention de ...  
Dépt RT  
4 pl. Tharradin  
25200 Montbéliard

Analyser cette transmission :  
lecture et envoi à chaque étape !



# Travail de la couche MAC (carte réseau)

Pour les paquets qui viennent **d'en bas** :

- Une carte réseau lit toutes les trames du médium auquel elle est reliée
- Elle **traite** (envoi en haut, à IP) seulement les trames :
  - unicast avec soi-même comme destinataire
  - diffusion
  - multicast si elle s'est préalablement inscrite à l'adresse multicast
- Pour qu'elle traite toutes les trames, il faut la mettre en état promiscuous (beaucoup de logiciels le font automatiquement)
- Attention à la topologie : hub et radio vs switch

Pour les paquets qui viennent **d'en haut** :

- La carte réseau utilise ARP et l'adresse IP (spécifiée par la couche supérieure IP) pour trouver l'adresse MAC suivante
- Elle crée l'en-tête MAC et encapsule le paquet
- Elle le transmet à la couche 1 (physique) pour transmission
  - buffer FIFO, premier arrivé, premier sorti = PEPS
  - si la file de la carte réseau est pleine, elle rejette le paquet

aussi la même fig, mais abrégée

## SAUTER

Un routeur est un appareil soit dédié (par ex. Cisco), soit une machine quelconque (PC) avec les logiciels appropriés installés (par ex. Linux)

```
...  
read (); <-- lit le paquet du buffer  
...
```

chaque couche dans  
une boîte colorée

- vérifie CRC
- l'ajoute au buffer de l'app  
qui écoute sur portdest

vérifie CRC

suis-je le destinataire final ?

oui : réassemble fragments le cas échéant

- remonte paquet (proto sup) si entier

non : TTL-- et rejet si 0 etc.

- consulte TR

- envoie à l'interface reliée au routeur suivant

vérifie CRC

suis-je le prochain saut ?

oui : ôte en-tête et remonte paquet

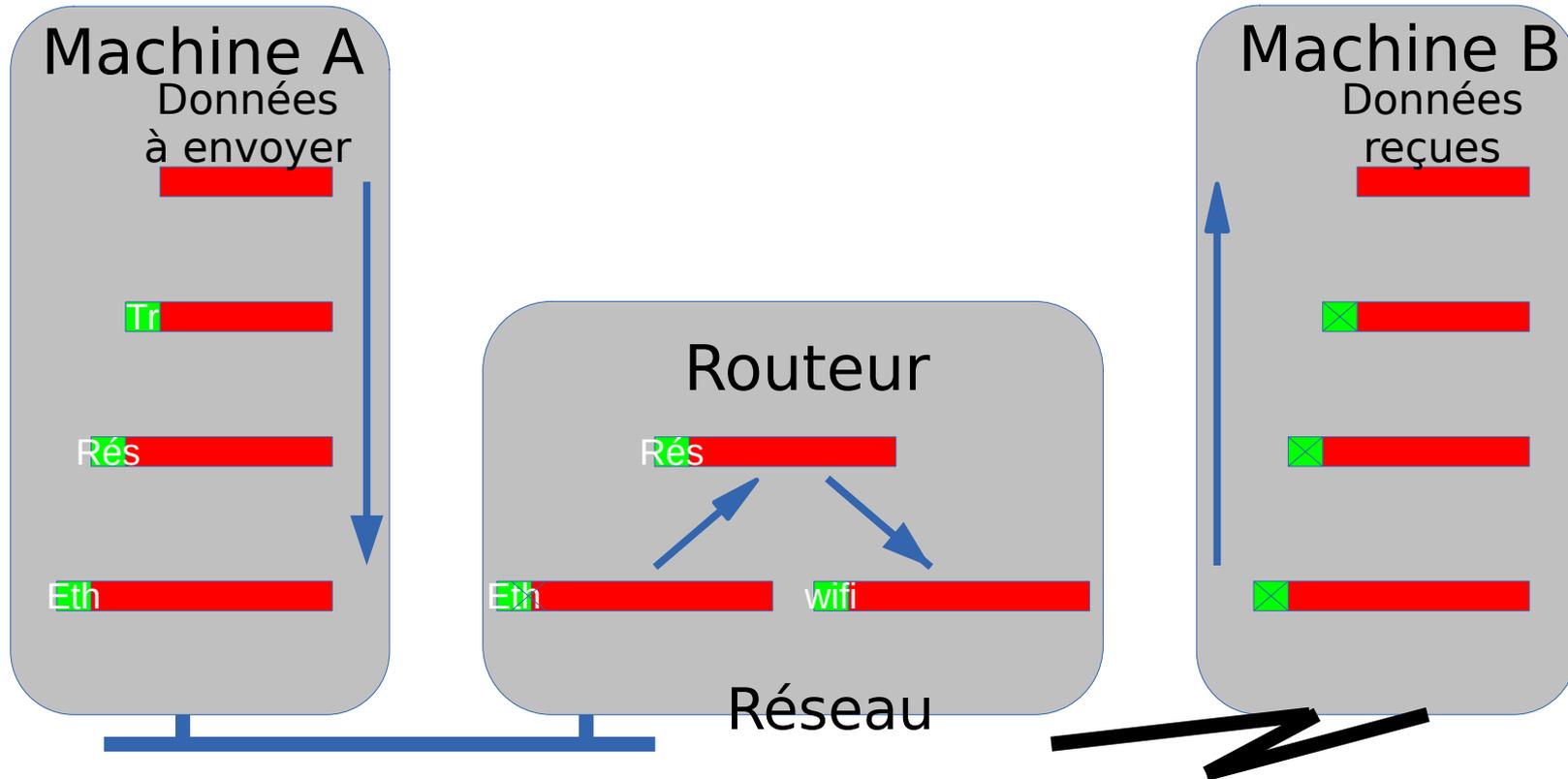
non : rejet paquet

utilise ARP pour récupérer l'@ MAC

fabrique l'en-tête MAC et l'ajoute au paquet

transmet le paquet sur le lien

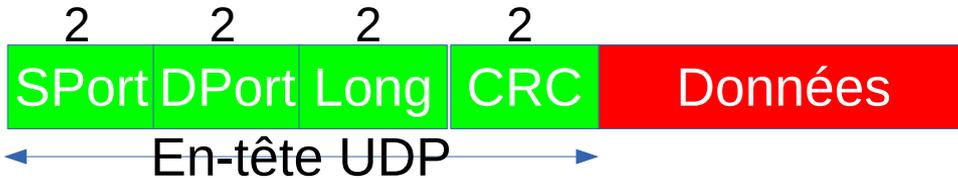
# Notion de pile TCP/IP



# Couche 4 (transport)

## UDP, User Datagram Protocol

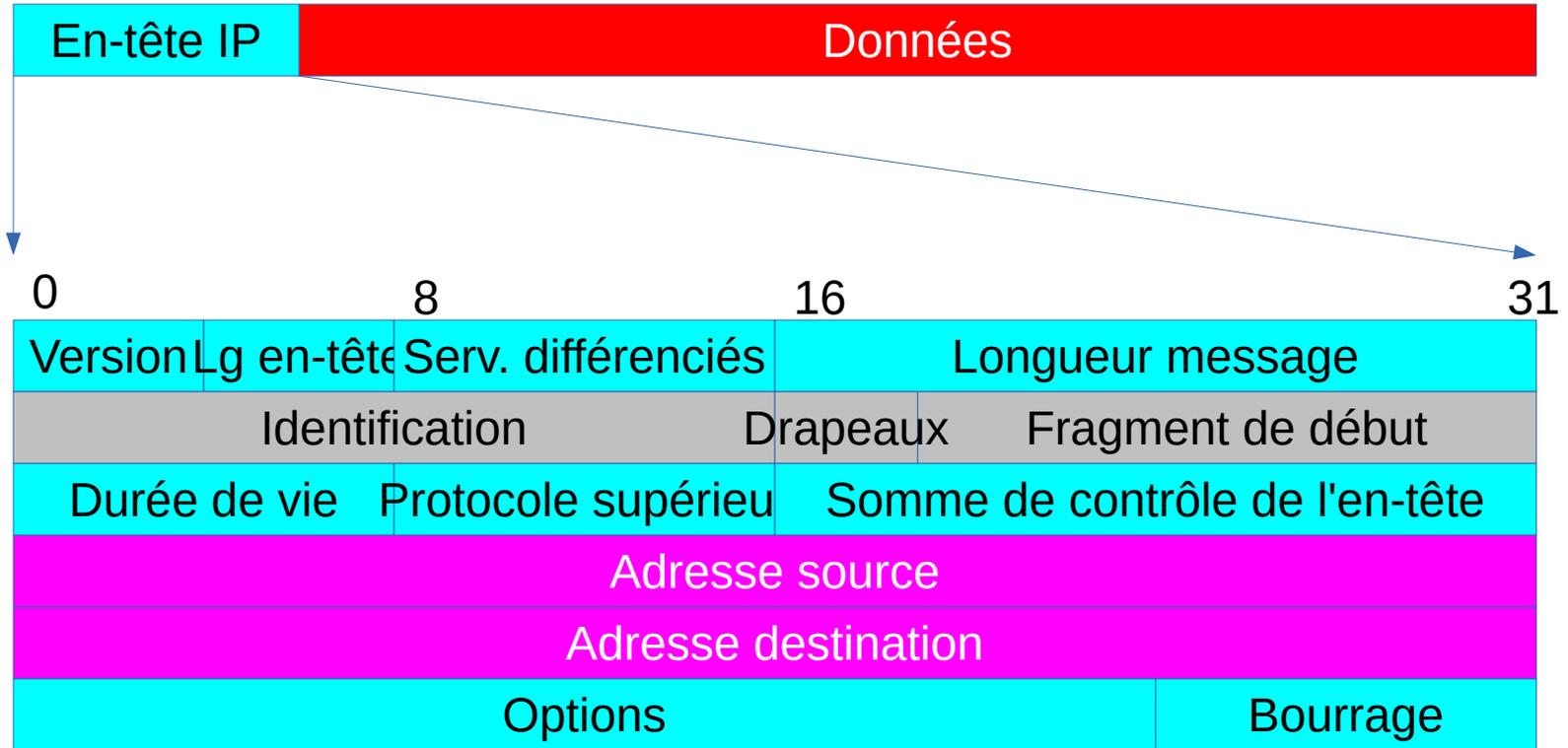
- Le port permet de distinguer plusieurs communications sur une même machine (DNS 53)
- Ne gère pas les pertes, les paquets dupliqués etc.
- Utilisation :
  - pour des services simples, comme DNS
  - pour des services où TCP est trop lent, e.g. streaming vidéo, jeux en ligne, VoIP, ...
- QUIC, au-dessus d'UDP, optimisé pour HTTP



## TCP, Transmission Control Protocol

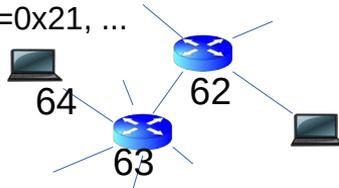
- Fournit les ports dans son en-tête (HTTP 80, SSH 22 etc.)
- Gère les paquets perdus (=> retransmission), dupliqués (=> effacement des dupliqués), désordonnés (=> réordonnancement)
- L'en-tête TCP contient un numéro de séquence pour chaque paquet
  - => la destination peut réordonner les paquets arrivés en désordre
  - => la destination peut ignorer les paquets dupliqués
- La destination, pour chaque paquet de données reçu, renvoie un accusé de réception
  - => la source sait quels paquets ont été perdus, et les retransmet
- Gère les ressources réseau par le contrôle de congestion : la source veut envoyer 1000 paquets, à quelle cadence les envoyer ?

# En-tête IP



# En-tête IP

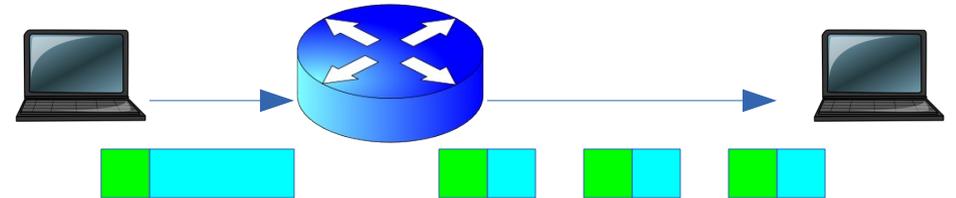
- Version : 4, pour IPv4, 6, pour IPv6
- Lg en-tête : longueur de l'en-tête en mots de 4 octets
  - le plus courant : 5 (donc en-tête de  $5 \times 4 = 20$  octets)
  - mais il peut y avoir des options + bourrage
- Longueur message : longueur totale du datagramme en octets
  - => taille des données = long message – lg en-tête
  - => taille max d'un datagramme : 64 ko
- TTL (Time To Live): décrémenté par chaque routeur
  - le paquet est détruit quand TTL arrive à 0
  - utile s'il y a des erreurs de routage (surtout des boucles)
- Protocole supérieur : quel protocole transport traitera les données
  - TCP=6, UDP=0x11, ICMP=1, DCCP=0x21, ...
- Services différenciés : on ne les présente pas
- Somme de contrôle : de l'en-tête seulement
- Adresses source et destination
  - jamais modifiées (sauf NAT)
- Options : optionnelles
  - enregistrement de route
  - routage défini par la source
  - un routeur, pour des raisons de sécurité, peut ignorer certaines options, voire rejeter le paquet
- Bourrage : pour faire aligner l'en-tête sur 32 bits (voir lg en-tête)
- Données : les données, de taille variable



# Fragmentation

- Rappel, couche 2 : toute technologie a un MTU (*Maximum Transmission Unit*) :
  - Ethernet : 1500 octets de données
  - Wi-Fi : 2313 octets de données
  - IoT (802.4 ?) : 120 octets
- Et si un routeur reçoit via Wi-Fi un paquet de 2000 octets et doit le transmettre via Ethernet (MTU=1500 octets) ?

- Les routeurs, si nécessaire, fragmentent les datagrammes



- Le destinataire réassemble les fragments
- La fragmentation est coûteuse => IPv6 ne la gère pas



# ICMP

- ICMP, Internet Control Message Protocol
  - IP n'est pas conçu pour être 100 % fiable
  - Lorsqu'une machine détecte une erreur IP, elle envoie un paquet ICMP à la source
  - Un paquet ICMP est comme les autres, il n'a pas de priorité particulière
  - Au-dessus d'IP
- Accessibilité et état d'une machine (**ping**)
    - si réponse, alors source, destination et tous les routeurs intermédiaires et le réseau sont fonctionnels
    - expliquer ping 127.0.0.1
  - **traceroute**, afficher les routeurs du chemin vers la destination
    - la source envoie des paquets avec des TTL successifs (à partir de 1)
    - pour chaque paquet, il reçoit un message ICMP (type 11) du routeur ayant détruit le paquet à cause du TTL à 0, et affiche son adresse (et autres informations)
  - Destination inaccessible (réseau ou ordinateur inaccessible, port fermé) :
    - ssh x.y.z.t, output : unreachable host : une adresse IP n'existe pas ou n'est pas joignable

En-tête IP

Message ICMP

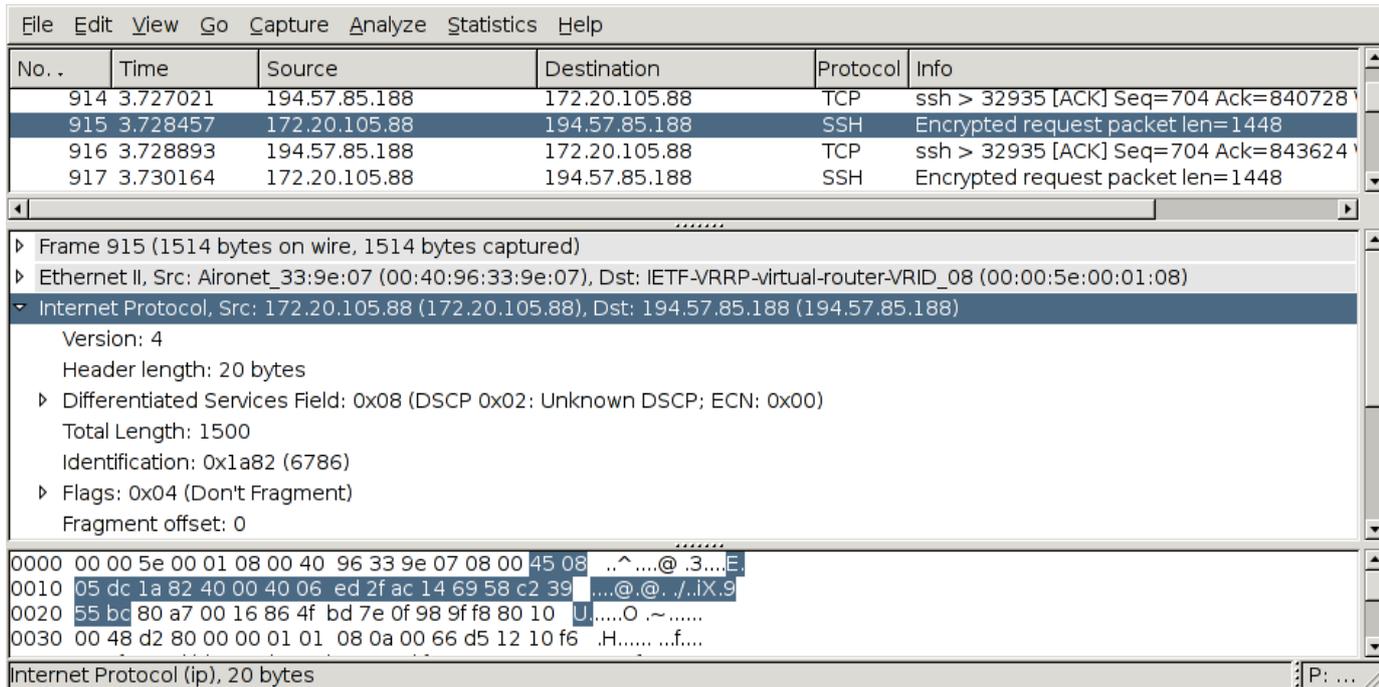
Protocol = 1 (ICMP)

# Taille des datagrammes et rendement de la transmission

- Exemple : transfert de 10 000 octets de données IP
- En-tête TCP + IP + MAC Ethernet = 20 + 20 + 18 = 58 octets
- Datagrammes de 1400 octets de données
  - 7 de 1400, 1 de 200
    - taille des en-têtes = 8 paquets \* 58 octets = 464 octets
    - rendement =  $10000/10464 = 0.95$
- Datagrammes de 200 octets de données
  - 50 de 200
    - taille des en-têtes = 50 paquets \* 58 octets = 2900 octets
    - rendement =  $10000/12900 = 0.77$

# Packet sniffers

- Logiciels qui captent le trafic d'une carte réseau : wireshark
  - utile pour *debugging*, *crack*, ...



# DNS

- Il est difficile de retenir des adresses IP
  - quelle est l'adresse IP des sites Web google, rt-serv, debian ?
- À part les administrateurs réseau, personne ne travaille avec des adresses IP
- DNS, *Domain Name System*, fournit une correspondance nom d'ordinateur => adr. IP
- Insensible à la casse (*case insensitive*) => utiliser les **minuscules**, plus simple à taper
- DNS est un protocole de couche application, car basé sur UDP (ou TCP plus récemment)
- host rt-serv.pu-pm.univ-fcomte.fr => 172.20.40.90
- host 172.20.40.90 => rt-serv.pu-pm.univ-fcomte.fr

# DNS, registrars

- Dernier mot du nom :
  - com, org, net, gov, eu, ... : international
  - fr, ro, es, uk, de, ... : par pays
  - ovh, leclerc, sncf, ... : certaines grosses entreprises
  - vous pouvez donner un nom en .ro à votre site alors qu'il est hébergé en Suisse (et que vous habitez en France) !
- Bureau d'enregistrement ("registrar") = organisme qui alloue des noms, en France c'est Afnic
  - mais les hébergeurs de sites Web peuvent intermédiaire la réservation d'un nom

# DNS, résolution du nom (en adresse IP)

- Un navigateur (ou une application quelconque) souhaite envoyer des données à une machine dont elle connaît le nom
- 1. Elle *résout le nom* en envoyant la requête de résolution au serveur DNS
  - toute machine connaît l'adresse IP d'un serveur DNS, habituellement récupérée lors du DHCP
    - par ex. 194.57.85.210 (tharradin) et 194.57.85.245 (lactel) sont les serveurs DNS de notre pôle
  - si le serveur peut faire tout seul la résolution (par ex. le nom est dans son domaine ou bien dans son cache), il renvoie l'adresse IP
  - sinon, il contacte un autre serveur DNS, de niveau plus haut, qui lui fournit (de manière récursive éventuellement) l'adresse IP
- 2. Elle envoie toutes les données à l'adresse IP retournée

Routeur  
(sauter cette section)

# Comparaison équipements réseau

- Répéteur, hub (1) : ne regarde rien
- Bridge, switch (2) : regarde l'adresse MAC destination
  - a la vision d'un réseau local seulement
- Routeur (3) : regarde l'adresse IP destination
  - achemine les paquets en dehors du réseau local
  - modifie l'en-tête niveau 2
- NAT (4) : regarde les adresses IP et les ports
  - modifie l'en-tête niveau 3
- Pare-feu (firewall), proxy web « filtrant » (5) : regarde l'en-tête application (n'autorise pas les commandes inconnues, les sites non autorisés)

# Équipements réseau

- Attention, certains équipements ont des fonctionnalités en dehors de leur rôle :
  - un point d'accès travaille au niveau 2, mais il peut intégrer une adresse IP (voire un serveur HTTP, niveau application) pour le configurer à distance
  - un hub travaille au niveau 1, mais il peut avoir une adresse IP (voire agents/serveurs SNMP, niveau application) pour le configurer/superviser à distance
- D'autres n'ont pas d'adresse IP, mais leur hôte en a
  - un modem (changement de technologie), tout comme une carte Ethernet, travaille au niveau 1/2, mais la machine qui l'utilise peut intégrer une pile TCP/IP s'il est utilisé pour connecter la machine à Internet

# Adressage IP

# Plages d'adresses IP

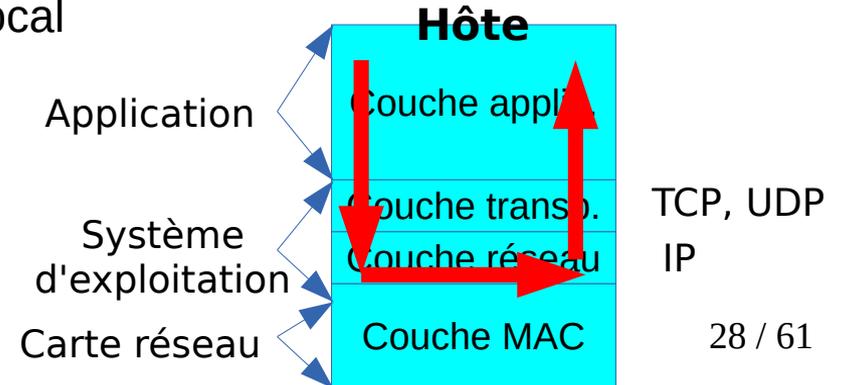
- Une adresse IPv4 a 32 bits, donc 4 octets
- A, B, C : unicast (classes générales)
- D : multicast, les 4 premiers bits sont 1110, voir plus tard
- E : réservée pour des usages futurs (jamais), les 4 premiers bits sont 1111



- Un organisme central, IANA, gère l'attribution de blocs d'adresses
  - le responsable d'une adresse IP est connu (<https://whois.iana.org>)

## Adresses spéciales

- 0.0.0.0 : si rien n'est connu
- 127.\*.\*.\* : boucle locale (*localhost*, *loopback*) => l'hôte lui-même
  - utile pour faire des tests entre deux applications sur une seule machine
  - une fois descendu à la couche IP, le paquet remonte sans passer par le réseau
- 255.255.255.255 : diffusion sur le réseau local



# Adresses privées

## Définition

- Une adresse est privée si elle appartient à :
  - **10.\*.\***, donc de 10.0.0.0 à 10.255.255.255 ( $2^{24}$  adresses)
  - **172.16–31.\*.\***, donc de 172.16.0.0 à 172.31.255.255 ( $16 * 2^{16}$  adresses)
  - **192.168.\*.\***, donc de 192.168.0.0 à 192.168.255.255 ( $256 * 256$  adresses)
- Adresse non privée = adresse publique

## Caractéristiques

- Les paquets avec adresse IP destination privée sont rejetés par les routeurs du cœur d'Internet, donc ces adresses **ne sont pas routées** sur Internet
- => Ces adresses sont utilisables uniquement dans le réseau local
- => Les mêmes adresses privées peuvent se trouver dans plusieurs réseaux
- => Les machines avec adresse privée sont inaccessibles depuis l'extérieur, ce qui est bénéfique d'pvt sécurité
- Toutefois, en utilisant le mécanisme NAT (présenté plus tard), les machines avec adresse IP privée :
  - peuvent communiquer sur Internet à condition que se soient elles qui **initient** la connexion
  - si configurées expressément, peuvent **recevoir** des connexions, et donc être serveur

# Appartenance d'une adresse à un réseau, parties réseau et hôte d'une adresse IP

- Toute adresse IP a deux parties : **partie réseau** et **partie hôte**
- Adresses A et B ont la même **partie réseau**  $\Leftrightarrow$  A et B sont dans le même réseau, autrement dit :
  - les machines avec la même **partie réseau** sont dans le même réseau
  - et vice-versa : toutes les machines d'un même réseau ont la même **partie réseau**
- Les adresses **1.2.3.4** et **1.2.3.5**, sont-elles dans le même réseau ou non ? Même question pour **1.2.3.4** et **1.2.2.5**, et pour **1.2.3.4** et **1.2.2.5**

# Les adresses d'un réseau, adresses réseau et diffusion

- Où commence et où finit le réseau contenant l'adresse 1.1.1.3 ? Même question pour 1.2.3.4
- => Un réseau ne peut pas commencer n'importe où et ne finit pas n'importe où !
- Quelle est la plus petite adresse du réseau contenant l'adresse 1.2.3.4 ? Même question pour 1.2.3.4.  
On l'appelle l'adresse (du) réseau
  - comment la calcule-t-on ? L'adresse réseau a toute la partie hôte à 0 !!
  - est-ce que les adresses 1.5.43.202 et 10.54.3.15 peuvent être des adresses réseau ? Pourquoi ?
  - les adresses réseau apparaissent dans les tables de routage (voir plus tard)
  - on ne peut pas l'affecter à une machine
- Quelle en est la plus grande adresse ? On l'appelle l'adresse de diffusion
  - comment la calcule-t-on ? L'adresse réseau a toute la partie hôte à 1 !!
  - quand un paquet est envoyé à cette adresse, toutes les machines du réseau la traitent
  - on ne peut pas l'affecter à une machine
- Traditionnellement, l'adresse juste avant l'adresse de diffusion est affectée au routeur du réseau

# Masque

- Le nombre de bits de la partie réseau s'appelle **masque**
  - 1.2.3.4 a 24 bits dans la partie réseau, donc elle s'écrit 1.2.3.4/24
  - 1.2.3.4 s'écrit 1.2.3.4/16
- Rappel : toutes les machines d'un même réseau (et le réseau lui-même) ont la même **partie réseau**, donc le même masque
- Pour spécifier une adresse de machine il faut toujours spécifier son adresse et le masque, par ex. 1.2.3.4/24 ou 1.2.3.4/16
- Pour spécifier un réseau il faut toujours spécifier l'**adresse réseau** et le masque, par ex. 1.2.3.0/24 ou 1.2.0.0/16
  - rappel : l'adresse réseau ne peut pas avoir n'importe quelle valeur (sa partie hôte doit être 0)
- Certaines machines anciennes demandent d'écrire le masque sous forme de 4 octets : que de 1 suivis que de 0, le masque étant le nombre de 1
  - 255.255.255.0 correspond à /24 (car il y a 24 bits à 1)
  - rappel : le masque n'est pas une adresse IP !!

# Adresses de réseau et de diffusion avec masque non multiple de 8

Une machine a l'adresse 192.65.129.74/26, donnez les adresses réseau, diffusion et routeur du réseau auquel elle appartient

Solution et méthodologie :

- 1) On écrit l'adresse en binaire
- 2) On identifie les parties réseau et hôte
- 3) L'adresse réseau est la première adresse (la plus petite) du réseau, l'écrire en décimal
- 4) L'adresse de diffusion (broadcast) de ce réseau est la dernière adresse (la plus grande) du réseau, l'écrire en décimal
- 5) L'avant-dernière adresse est affectée à l'interface du routeur, par convention, l'écrire en décimal

Peut-on avoir un réseau qui commence à 1.1.1.5/25 et finit à 1.1.1.19/25 ? Pourquoi ?

adr. IP : 11000000.01000001.10000001.01001010  
adr. rés : 11000000.01000001.10000001.01000000  
adr. diff : 11000000.01000001.10000001.01111111  
adr. rout : 11000000.01000001.10000001.01111110  
=> une adresse réseau a tous les bits de la partie hôte à 0

Écrivez l'adresse réseau et de diffusion du réseau contenant les adresses :

- 129.5.4.3/23
- 193.6.5.76/18

Regrouper en réseaux les adresses suivantes ...

# Nombre d'adresses

- Combien d'adresses a un réseau de masque /24 ? Même question pour /25, pour /26 et pour /18
  - => un /25 est une moitié de /24, un /26 est un quart de /24
  - => un /24 est formé de 4 réseaux /26, ou encore d'1 /25 et 2 /26
- Combien d'adresses a un réseau /30 ? Même question pour /31
  - en théorie, il y a 33 masques possibles, de /0 à /32, mais en pratique il n'y en a que 23 (voire moins) : /8 à /30
- Exemple : où commence et où finit le réseau contenant l'adresse 1.1.1.1/24, et combien d'adresses a-t-il ? Même question pour 1.1.1.1/25 et pour x.y.z.130/26

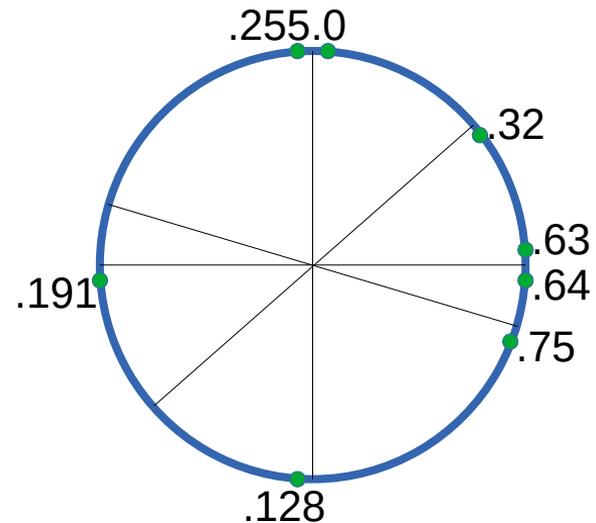
# Division d'un réseau en sous-réseaux et leur placement

## Division

- Une entreprise a acheté le réseau 9.1.59.0/24, donc les 256 adresses de .0 à .255
- Elle souhaite le diviser en deux sous-réseaux A et B, moitié chacun, qui ensemble représentent toute la plage achetée
- Quel est le masque des sous-réseaux ?

## Placement des sous-réseaux

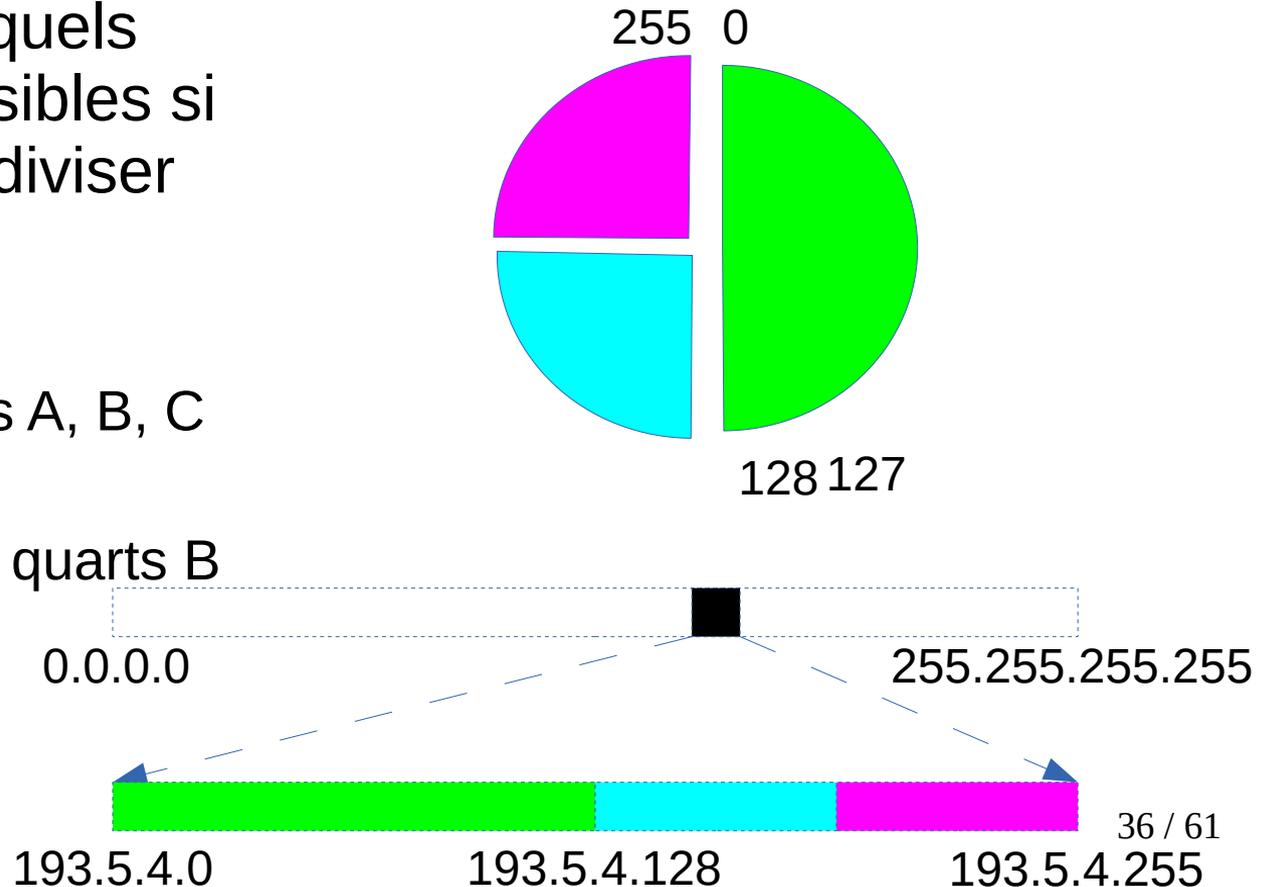
- Dans l'image de droite, spécifier l'adresse réseau de chaque moitié ; lesquels sont valides ? (rappel : un réseau ne peut pas commencer n'importe où !)
- Spécifier tous les placements possibles pour A et B



# Placement des sous-réseaux

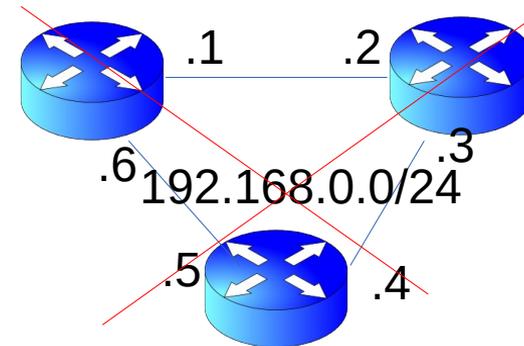
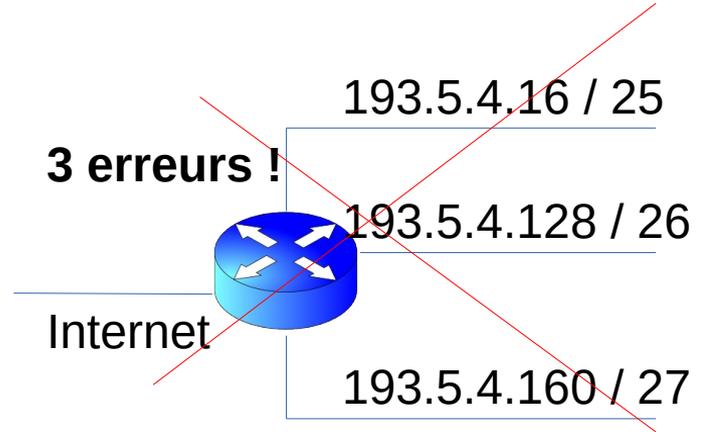
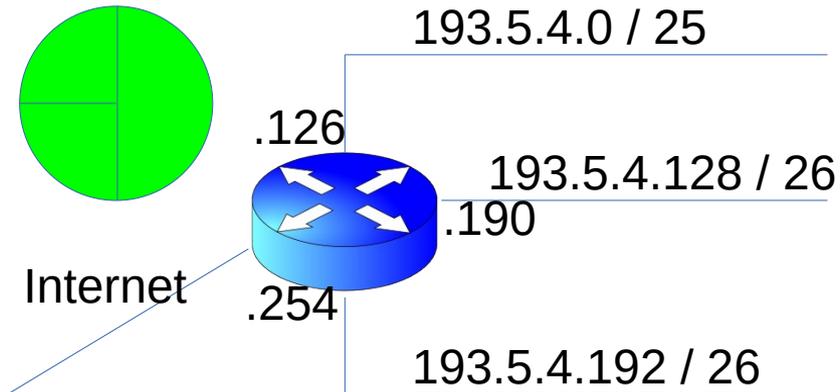
- Quelles masques et quels placements sont possibles si l'entreprise souhaite diviser le réseau précédent (9.1.59.0/24) en :

- quatre parties égales A, B, C et D
- une moitié A et deux quarts B et C



# Plan d'adressage IP

- Plan IP d'un site = schéma avec toutes les informations IP du site :
  - 1) dessiner le réseau
  - 2) spécifier les adresses de réseau et leurs masques
  - 3) spécifier les adresses des routeurs
- **Les sous-réseaux ne doivent pas se chevaucher**



# Exemple de changement de plan IP

Activité / Mission 3 réalisée - Apprenti.e

A

## Ismael BOUDEBZA a répondu :

Migration du plan d'adressage IP et du Serveur sur un site de l'entreprise à Héricourt, conformément à la nouvelle nomenclature réseau de toute l'infrastructure.

Plan d'adressage IP anciennement en 192.168.X.Y ( X étant le numéro du site et Y l'adresse machine ), ce qui limitait le nombre d'adresse disponibles sur le réseau, et rendait compliqué la segmentation du réseau (VLAN).

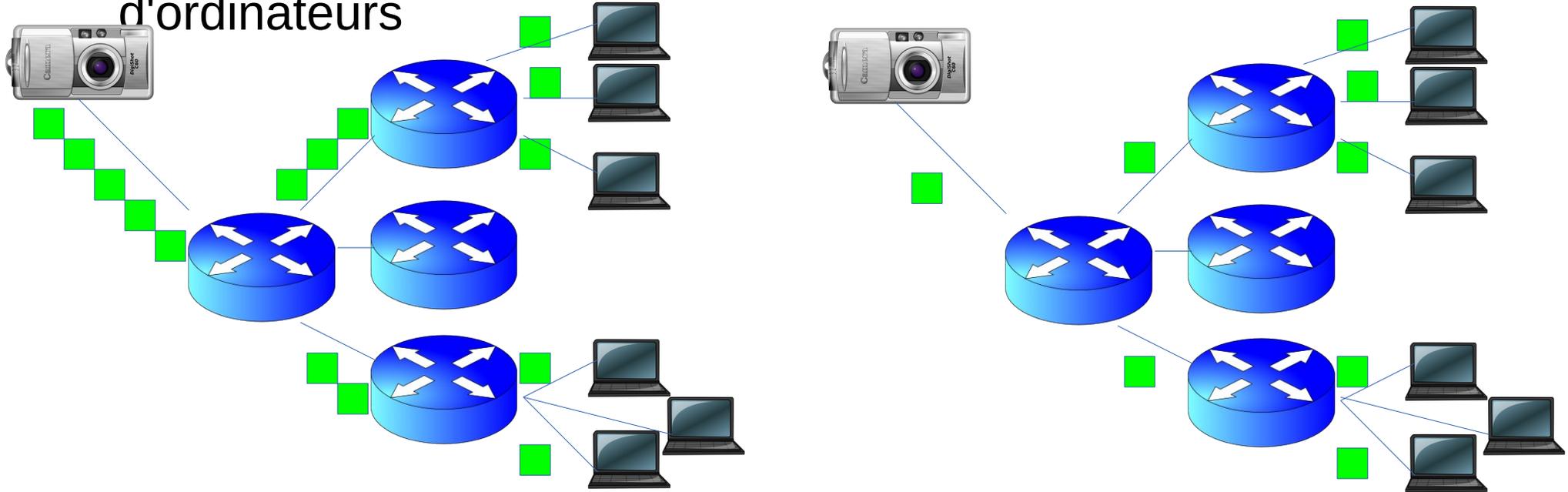
Passage en 10.X.Y.Z (X numéro du site, Y identifiant du VLAN, Z l'adresse machine), ce qui comprend :

- Désactivation du serveur DHCP de l'ancien serveur
- Redémarrage des postes ordinateurs du site (pour renouveler des requêtes DHCP vers le "nouveau" DHCP)
- Modification des IP des imprimantes et du VLAN du port de chacune d'elles (Modification effectués sur les Switchs) pour les migrer vers le VLAN imprimante
- Serveur physique éteins et desactivé car les serveurs sont désormais héberger dans un datacenter à Besançon.

# Multicast

# Multicast

- Imaginons un film sur Arte *streamé* en direct sur des milliers d'ordinateurs



Unicast

Multicast

Un paquet par lien

=> Utilisation inefficace de la bande passante

# Multicast

- Rappel : diffusion (*broadcast*) = toutes les machines d'un réseau reçoivent le paquet
- Multicast (diffusion sélective) = que certaines machines reçoivent le paquet
  - les machines peuvent se trouver sur des réseaux distincts
  - utilisé principalement pour la transmission vidéo en direct
- En multicast un paquet est transmis une seule fois par lien
  - il est **multiplié** lorsqu'il est retransmis sur plusieurs liens
  - => grande réduction de la bande passante utilisée

# Multicast, enregistrement/désabonnement

- Inscription à une adresse :
  - un hôte annonce son inscription à une certaine adresse de classe D à son routeur, qui à son tour annonce les autres routeurs
- Désabonnement à une adresse :
  - tout routeur scrute régulièrement ses RL ; si aucun hôte ne répond, le routeur cesse d'émettre des données multicast
- IGMP, protocole utilisé entre les routeurs multicast
- Peu de routeurs sur Internet ont le multicast activé

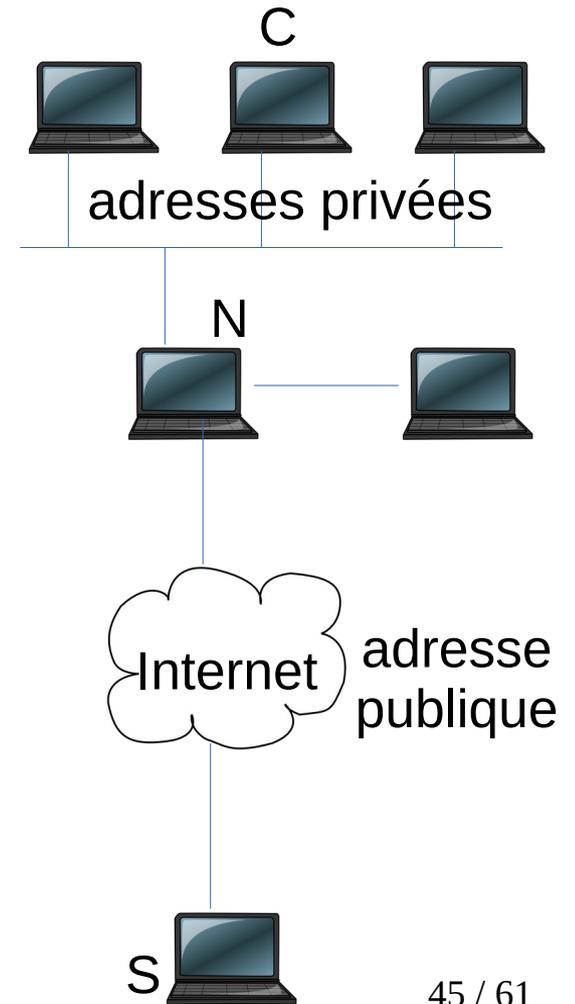
NAT

# Besoin de la NAT

- Problème : nombre d'ordinateurs supérieur au nombre d'adresses IP publiques disponibles
  - on donne des adresses privées à certains ordinateurs, mais alors comment les faire communiquer sur Internet ?
- Solution : la NAT
- Omniprésente

# Principe de la NAT

- C (d'adresse privée) envoie un paquet à S (d'adresse publique)
- Le paquet passe par N (car N est sur le chemin)
- N retransmet le paquet vers S en lui changeant l'adresse IP source (C, privée) par sa propre adresse (N, publique)
- C répond à N
- N change l'adresse IP destination de N à C et le retransmet
- Le paquet arrive à C

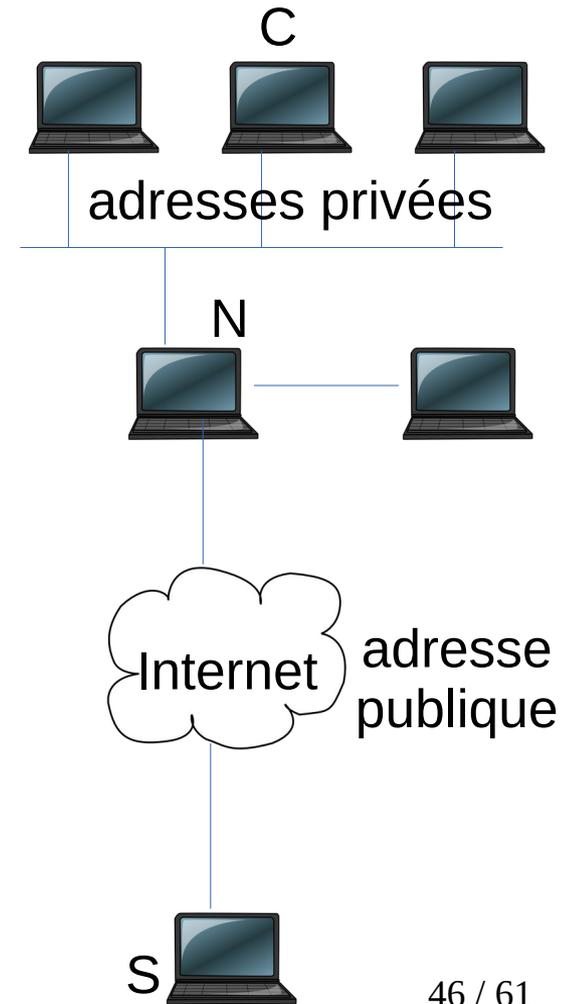


# Détails de la NAT

- Pour différencier deux machines d'adresse privée, on utilise le **port** source/destination
- Pour accepter deux machines avec le même port source, N crée aussi un **port spécifique** à cette connexion
- N utilise un tableau de correspondance :

Srcaddr	Srcport	Natport
10.1.2.3	9845	45320
10.1.2.3	9856	47099
10.1.2.4	7455	32456

- création de ligne, utilisation d'une ligne, effacement de ligne



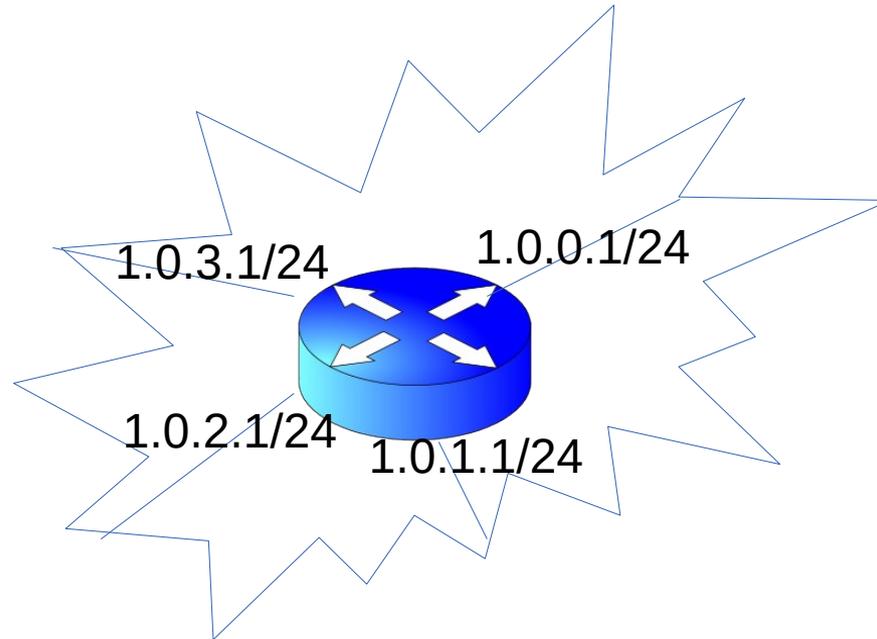
# Formes de NAT

- NAT de port (PAT, NAPT) : N:1
  - la forme la plus utilisée
- NAT dynamique : N:M ( $N > M$ )
  - timeout pour la libération d'une adresse IP publique
- NAT statique : 1:1
  - ex. : serveur Web dans le réseau privé, mais accessible depuis l'extérieur

# NAT, caractéristiques

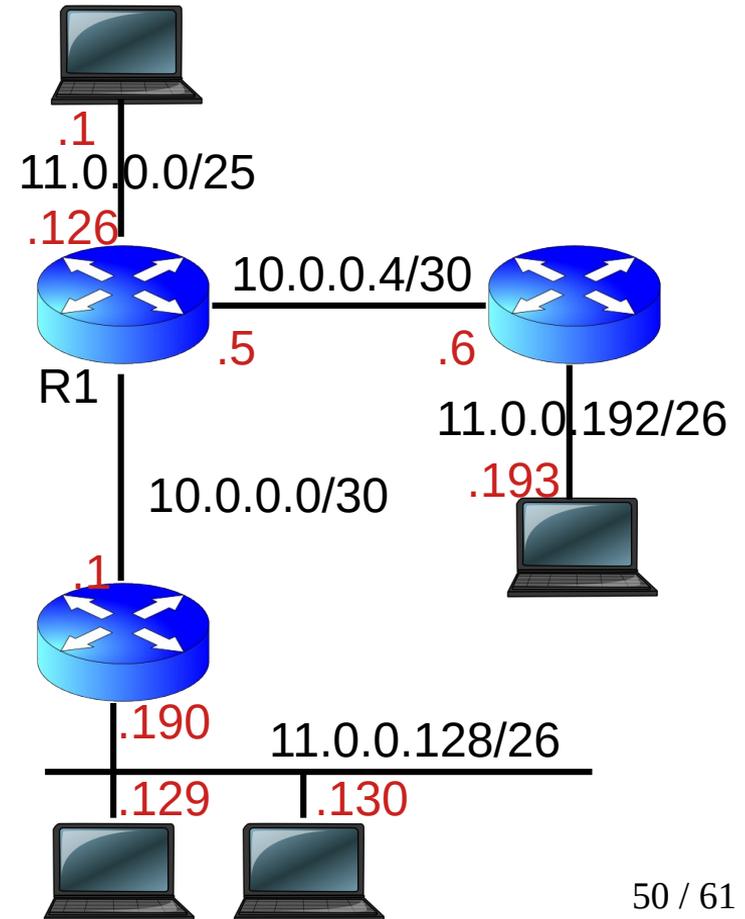
- Impossible d'initier une connexion depuis l'extérieur
  - mais sécurité pour l'ordinateur
- Inconvénients :
  - certains protocoles (FTP, SIP, ...) mettent des adresses IP dans leur en-tête
- Avantages :
  - load balancing (sur plusieurs serveurs)
  - transparent HTTP proxy
- Beaucoup de machines peuvent faire la NAT :
  - PC, routeur ou autres machines, comme le PIX (firewall)

# Tables de routage



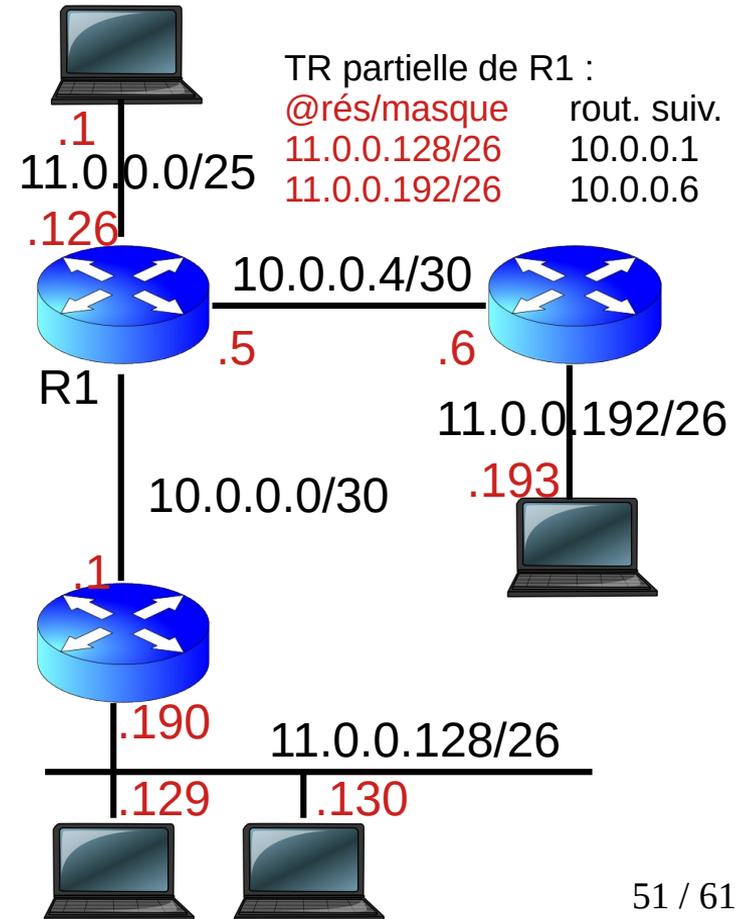
# Routage de saut en saut

- .1 app : j'envoie un paquet à .129
- .1 IP : la table de routage (TR) me dit que pour atteindre .129 je dois l'envoyer à mon voisin .126
- .126 (R1) reçoit le paquet
- .126 (R1) IP : la TR me dit que pour atteindre .129 je dois l'envoyer à mon voisin .1
- ...
- .129 reçoit le paquet
- .129 IP : je suis la destination, je ne consulte pas ma TR, le routage s'arrête, j'envoie à l'app



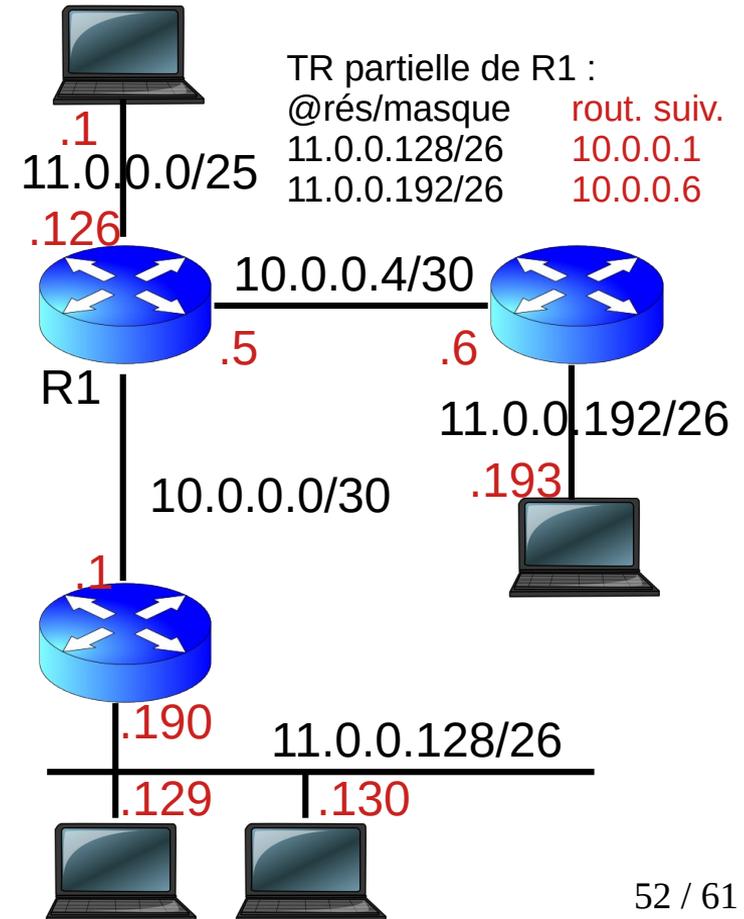
# Contenu des tables de routage, @rés/masque

- R1 a un paquet avec @dest 11.0.0.129 à envoyer (soit c'est elle qui l'a générée, d'en haut, soit elle l'a reçu, d'en bas, et doit le forwarder)
- Si c'est une de ses adresses IP, il traite le paquet
- Sinon, il consulte sa TR pour choisir la ligne :
  - il vérifie toutes les lignes une par une pour voir lesquelles correspondent
    - 11.0.0.129 appartient-il à 11.0.0.128/26 ?
    - 11.0.0.129 appartient-il à 11.0.0.192/26 ?
  - 0, 1 ou plusieurs lignes peuvent correspondre
    - si 0 lignes, il rejette le paquet, crée un paquet ICMP Network is unreachable et l'envoie à la source
    - si plusieurs lignes, il choisit la ligne la plus spécifique (de masque le plus long)
- Le paquet est transmis au routeur suivant spécifié dans la ligne choisie, donc 10.0.0.1
- Refaire l'exercice avec 11.0.0.200 et 10.10.10.10



# Contenu des tables de routage, routeur suivant

- Le routeur **suivant** doit obligatoirement être **directement** connecté
- Exemples d'erreurs :
  - 10.0.0.5 n'est pas un routeur suivant
  - 11.0.0.190 n'est pas directement connectée





# Table de routage complète (quand tout le réseau est accessible)

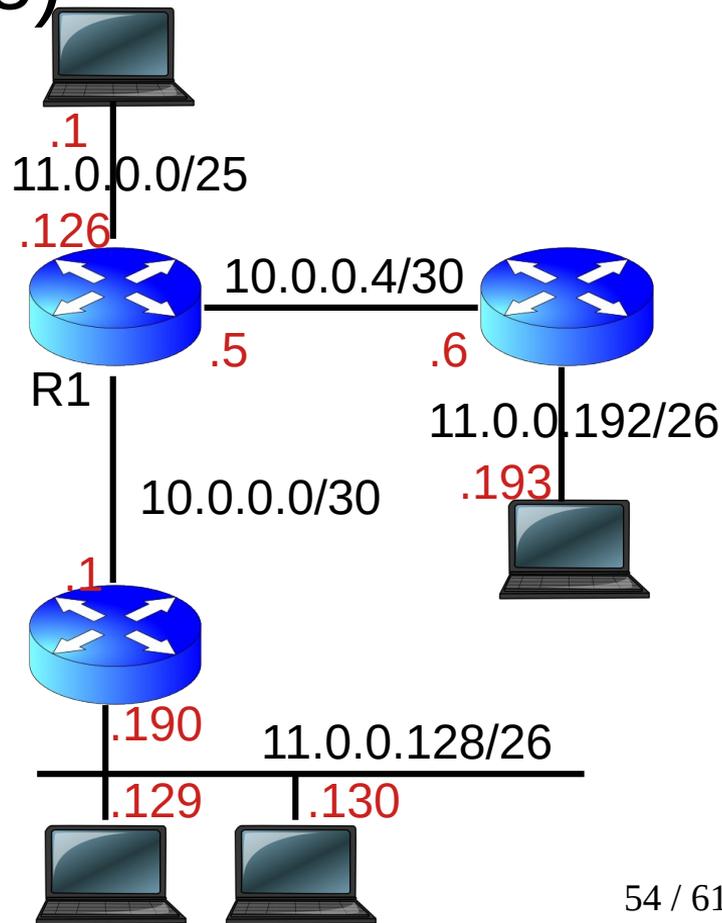
TR de R1 :

@rés/masque	routeur suivant
11.0.0.0/25	-
10.0.0.4/30	-
10.0.0.0/30	-
11.0.0.128/26	10.0.0.1
11.0.0.192/26	10.0.0.6

TR de PC1 :

@rés/masque	routeur suivant
11.0.0.0/25	-
0.0.0.0/0	11.0.0.126

- La TR d'un hôte a en général juste 2 lignes
- La TR d'un RC n'a pas de route par défaut et a 500 000 lignes (connaissance globale des réseaux)
- Q1 : y a-t-il une différence si on remplace la dernière ligne de TR de R1 par :  
0.0.0.0/0      10.0.0.6 ?
- Q2 : que fait R1 s'il reçoit un paquet à destination 100.0.0.1 dans les deux cas ?



# Sur-adressage : CIDR

- CIDR, *Classless Inter-Domain Routing* (routage sans classe)
- Soit la table de routage d'un routeur :

192.0.48.0/25	x.y.z.t
192.0.48.128/25	x.y.z.t
- Table de routage équivalente :

192.0.48.0/24	x.y.z.t
---------------	---------
- Le sur-adressage permet de **réduire la taille des tables de routage**
- Les réseaux destination doivent être contiguës

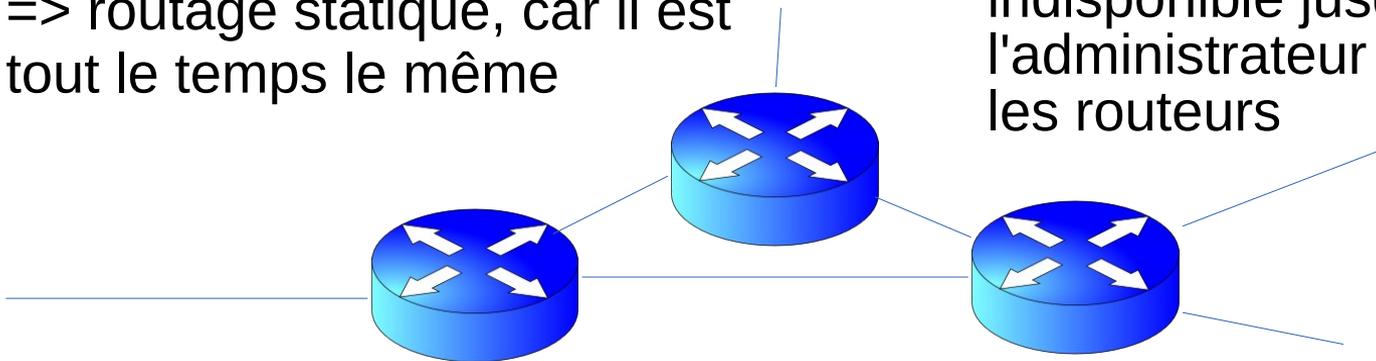
Comment réduire les tables de routage suivantes ?

- Exemple 1 :

192.0.4.0/24	x.y.z.t
192.0.5.0/24	x.y.z.t
192.0.6.0/24	x.y.z.t
192.0.7.0/24	x.y.z.t
- Exemple 2 :
  - pareil, mais la dernière ligne a comme routeur suivant a.b.c.d

# Routage statique vs dynamique

- Pour un réseau petit, l'administrateur peut configurer le(s) routeur(s) une fois pour toutes
  - => routage manuel, car il spécifie lui-même tous les réseaux
  - => routage statique, car il est tout le temps le même
- Un routage statique ne convient pas :
  - pour un grand réseau qui change souvent (ajout/effacement de liens/routeurs), par ex. les RC
  - si un problème de réseau apparaît, le réseau devient indisponible jusqu'à ce que l'administrateur reconfigure les routeurs



DHCP

# DHCP, besoins

- Pour configurer un ordinateur en réseau, les paramètres suivants sont nécessaires :
  - son adresse IP
  - le masque de sous-réseau
  - adresse IP du routeur par défaut (nécessaire pour sa table de routage)
  - l'adresse IP du serveur de nom (DNS)
  - (l'administrateur peut en rajouter d'autres)
- Grand réseau, e.g. 100 ordinateurs :
  - difficile de configurer/modifier les adresses IP, e.g. pour une machine nouvelle
- Machine qui change souvent d'adresse, e.g. ordinateur portable
- Connexions temporaires à Internet
  - ex. : étudiants qui viennent avec leurs portables à l'école : 1000 étudiants, mais seulement 200 en même temps
- WAN party : l'ordi de chaque étudiant est automatiquement configuré

# DHCP, idée

- DHCP, *Dynamic Host Configuration Protocol*
- Idée : à partir de son adresse MAC, un ordinateur trouve tous ses paramètres réseau
- Types de paquets : DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK

# DHCP, gestion sur le client

- Chaque fois que la machine démarre, un client DHCP (dhclient sous linux) est lancé qui récupère tous les paramètres
- L'adresse IP obtenue est temporaire
  - le temps est spécifié par le serveur
    - généralement, entre quelques heures et plusieurs jours
  - résiliation avant expiration possible
- Le client DHCP continue à s'exécuter et, vers la fin de la période, demande au serveur s'il peut continuer à utiliser son adresse IP

# DHCP, gestion sur le serveur

- L'administrateur demande à l'utilisateur son adresse MAC
  - mais ce n'est pas obligatoire
- Il la met dans le serveur, avec tous les paramètres associés
- 3 types de configurations sur le serveur :
  - manuelle : adresse spécifique pour chaque ordinateur
  - automatique : adresse nouvelle définitive lors de la 1ère connexion du client
  - dynamique : adresse temporaire quelconque
- Plus d'informations, voir cours 2ème année