

# Panorama des Protocoles Internet

Marion Gilson



# Panorama des protocoles internet

- I. Introduction
- II. Modèles TCP/IP – principe de l'encapsulation
- III. Modèles TCP/IP -- couche 3 (réseau)
  - a. rôle
  - b. adressage
  - c. introduction au routage
- IV. Modèles TCP/UDP – couche 4 (transport)
  - b. rôle et fonctionnement
  - c. TCP et UDP
  - d. NAT et PAT
- V. Modèles TCP/IP – couche 5 (application)
  - a. rôle et fonctionnement
  - b. DNS et DHCP
  - c. Services et protocoles (HTTP, FTP, Telnet,...)

# Panorama des protocoles internet

- I. Introduction
- II. Modèles TCP/IP – principe de l'encapsulation

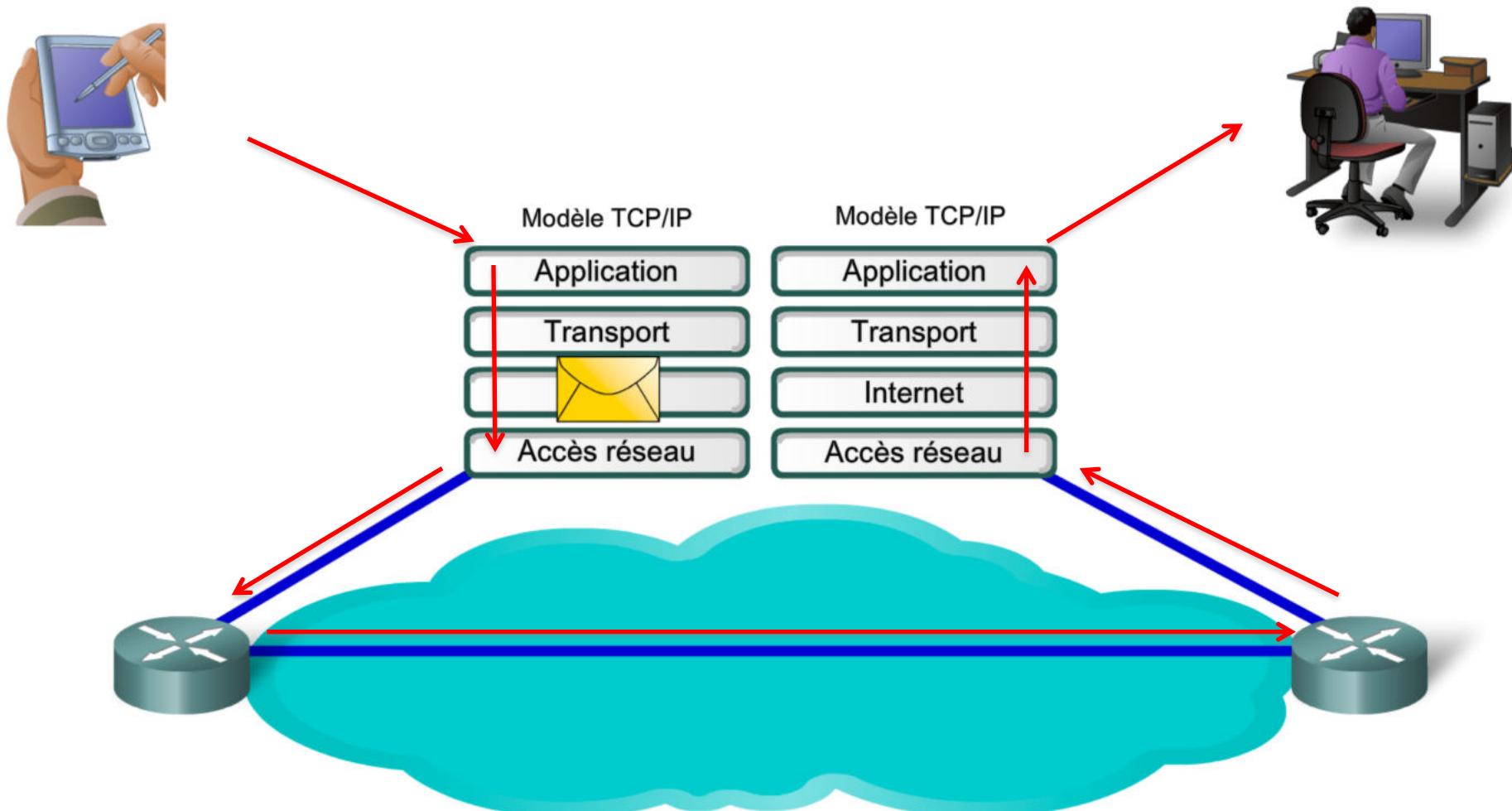
- II. Modèles TCP/IP -- couche 3 (réseau)
  - a. rôle
  - b. adressage
  - c. introduction au routage

- II. Modèles TCP/UDP – couche 4 (transport)
  - b. rôle et fonctionnement
  - c. TCP et UDP
  - d. NAT et PAT

- IV. Modèles TCP/IP – couche 5 (application)
  - a. rôle et fonctionnement
  - b. DNS et DHCP
  - c. Services et protocoles (HTTP, FTP, Telnet,...)

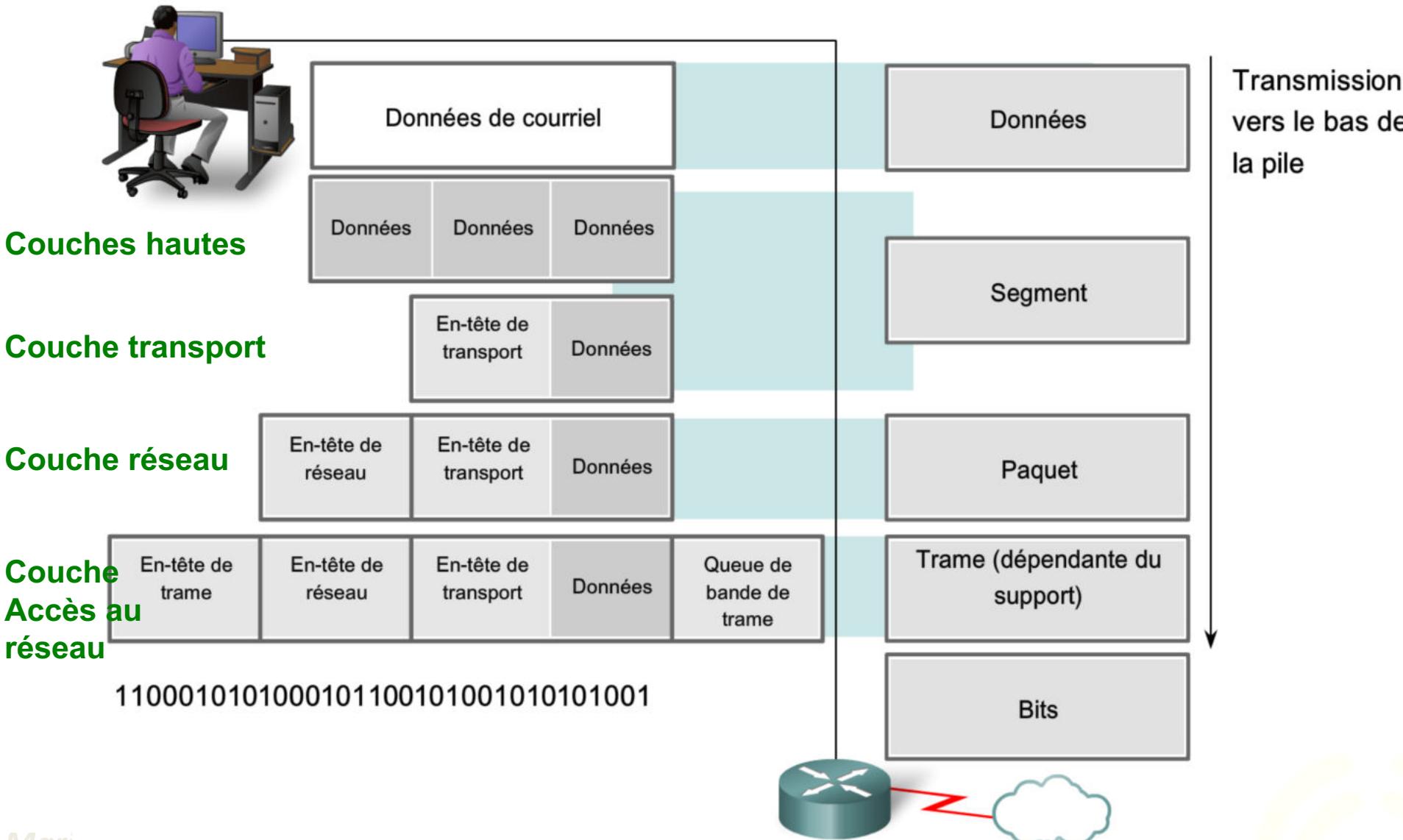
# L'encapsulation

De l'envoi d'un message à sa réception, au travers du réseau : passage à travers les différentes couches du modèles TCP/IP (communication d'égal à égal)



# L'encapsulation

## Encapsulation



# Couche 2 : Ethernet

Utilisation d'Ethernet sur des périphériques physiques

**Deux fonctions principales :**

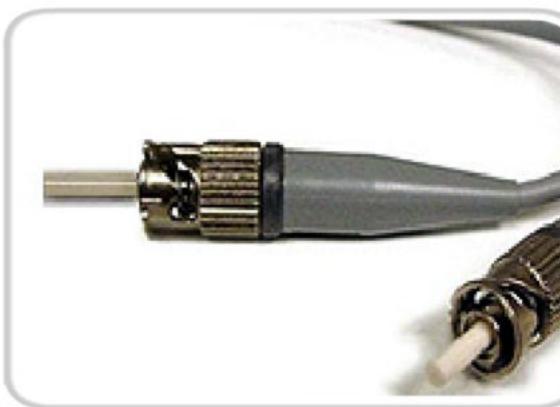
- encapsulation des données
- contrôle de l'accès aux supports



Panneaux de brassage à paires torsadées non blindées dans une baie



Commutateurs Ethernet



Connecteurs pour fibre optique Ethernet



Commutateur Ethernet

# Panorama des protocoles internet

I. Modèles TCP/IP – principe de l'encapsulation

II. Modèles TCP/IP -- couche 3 (réseau)

- a. rôle
- b. adressage
- c. introduction au routage

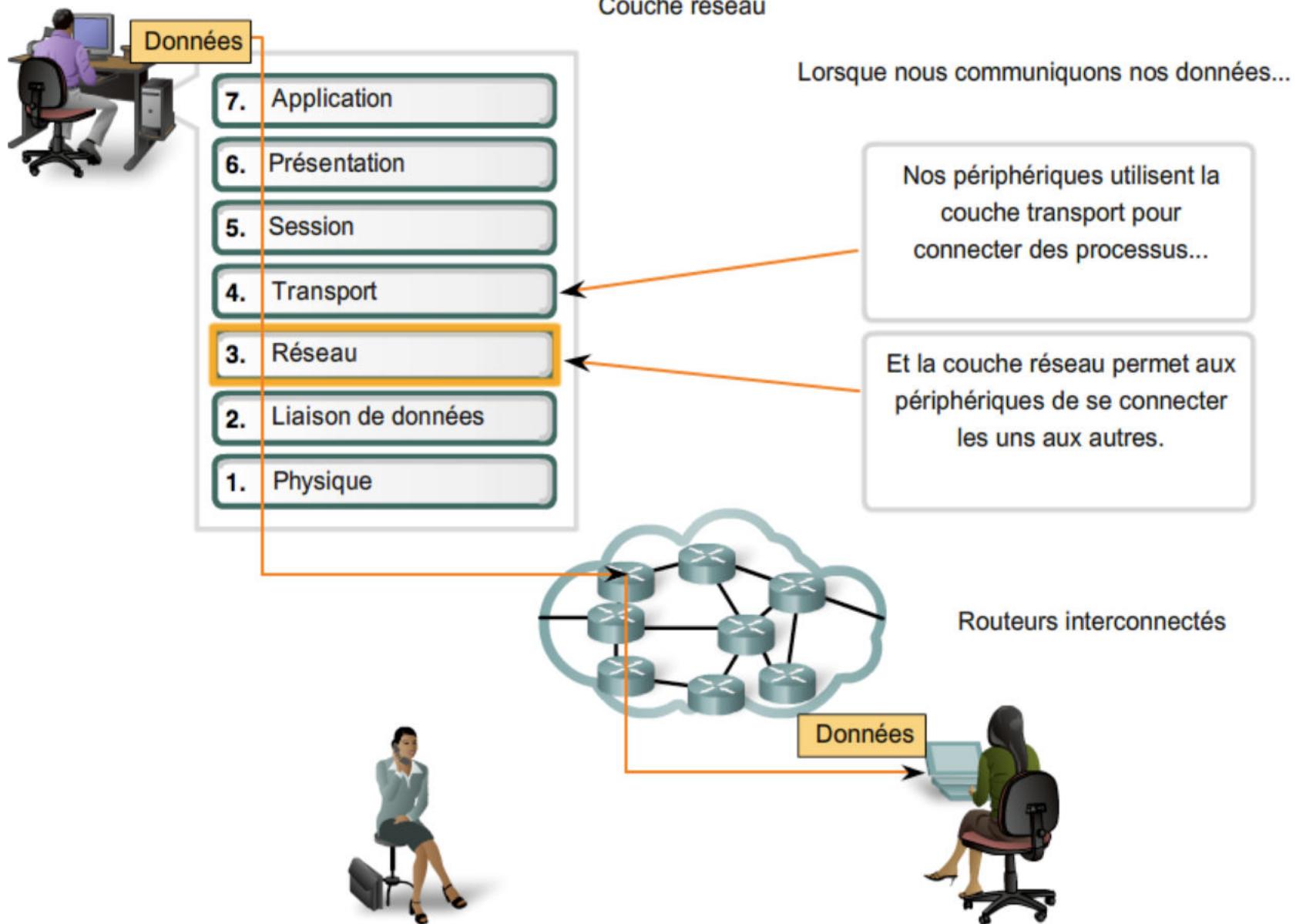
II. Modèles TCP/UDP – couche 4 (transport)

- b. rôle et fonctionnement
- c. TCP et UDP
- d. NAT et PAT

IV. Modèles TCP/IP – couche 5 (application)

- a. rôle et fonctionnement
- b. DNS et DHCP
- c. Services et protocoles (HTTP, FTP, Telnet,...)

# Couche 3 : IP



# Couche 3 : IP

Rôle de la couche réseau :

Fournir des services aux périphériques finaux pour échanger des données sur le réseau, en se fondant sur :

**L'adressage des périphériques finaux** : une adresse IP unique doit être configurée sur les périphériques finaux pour les identifier sur le réseau

**L'encapsulation** : la couche réseau encapsule l'unité de données de protocole (PDU) de la couche transport dans un paquet : ajout des informations d'en-tête IP, telles que *l'adresse IP* des hôtes source (expéditeurs) et de destination (destinataires).

**Le routage** : la couche réseau fournit des services permettant de *diriger* les paquets vers un hôte de destination sur un autre réseau. Pour voyager vers d'autres réseaux, le paquet doit être traité par un *routeur*. Le rôle du routeur est de sélectionner le meilleur chemin et de diriger les paquets vers l'hôte de destination (routage).

**La désencapsulation** : lorsque le paquet arrive au niveau de la couche réseau de l'hôte de destination, l'hôte *vérifie* l'en-tête IP du paquet. Si l'adresse IP de destination correspond à l'adresse IP de l'hôte qui effectue la vérification, l'en-tête IP est supprimée du paquet puis *transmission* à la couche 4

# Couche 3 : IP

## Périphériques réseaux : les routeurs

La gamme de routeurs à services intégrés (ISR) Cisco



Routeur à services intégrés de la gamme Cisco  
800



Routeur à services intégrés de la gamme Cisco  
3800



Routeur à services intégrés de la gamme Cisco  
1800

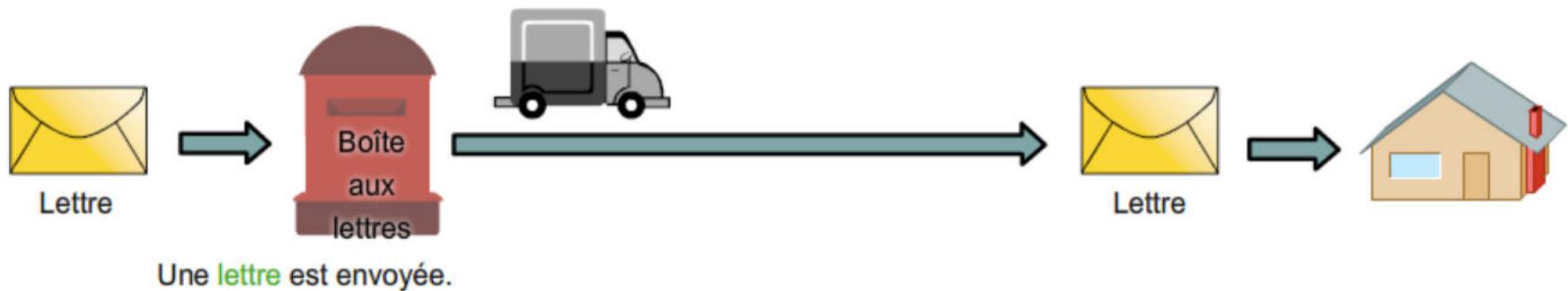


Routeur à services intégrés de la gamme Cisco  
2800

# Couche 3 : IP

Protocole sans connexion (analogie avec le réseau postal)

Communication sans connexion



L'expéditeur ne sait pas :

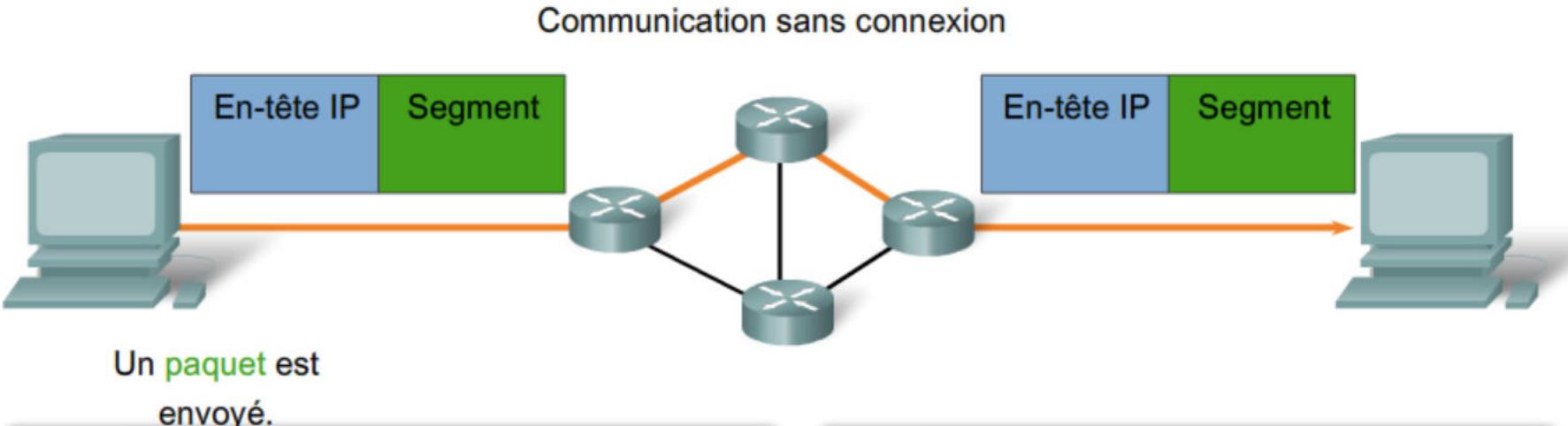
- si le destinataire est présent
- si la lettre est arrivée
- si le destinataire peut lire la lettre

Le destinataire ne sait pas :

- quand elle arrive

# Couche 3 : IP

Protocole sans connexion (réseau de données) :



L'expéditeur ne sait pas :

- si le destinataire est présent
- si le paquet est arrivé
- si le destinataire peut lire le paquet

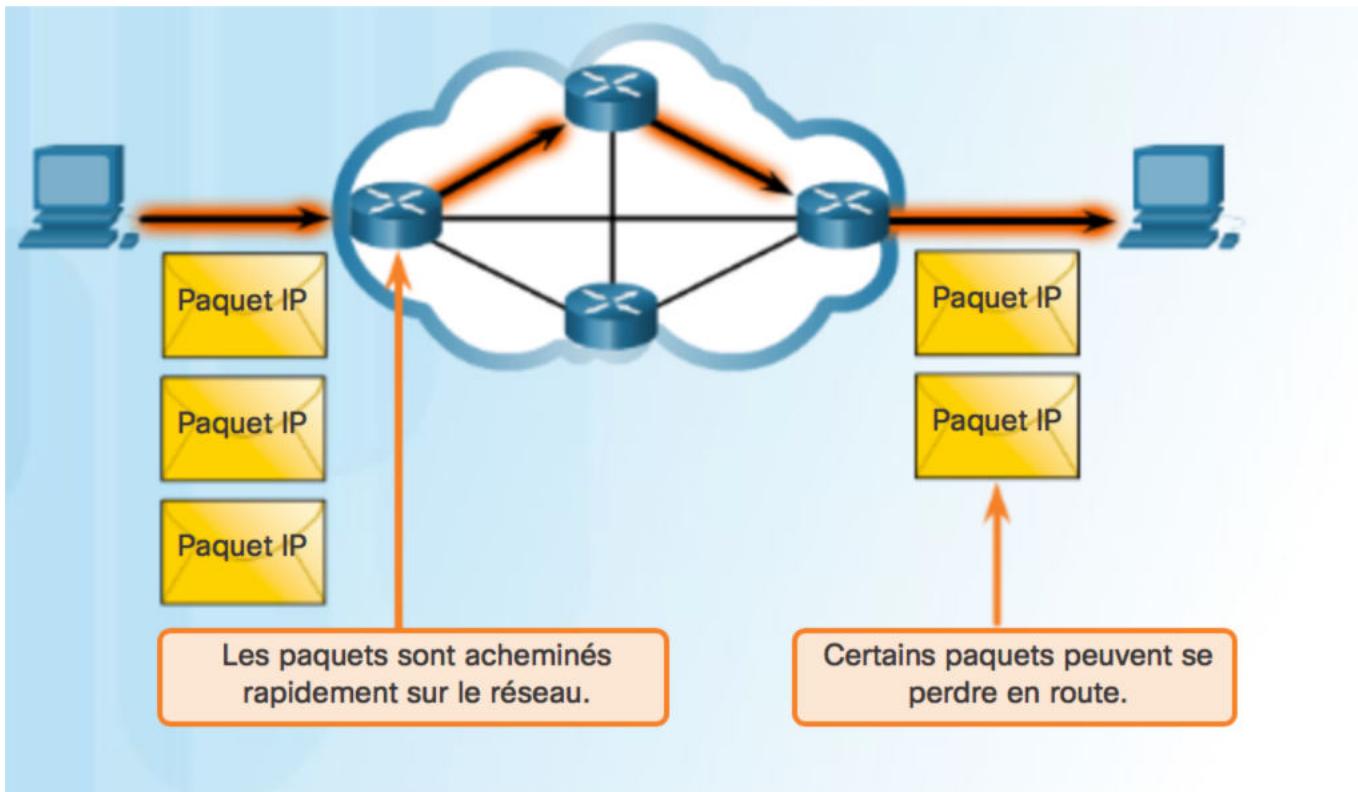
Le destinataire ne sait pas :

- quand il arrive

Protocole peu fiable : d'autres protocoles gèrent le suivi des paquets et garantissent leur acheminement

# Couche 3 : IP

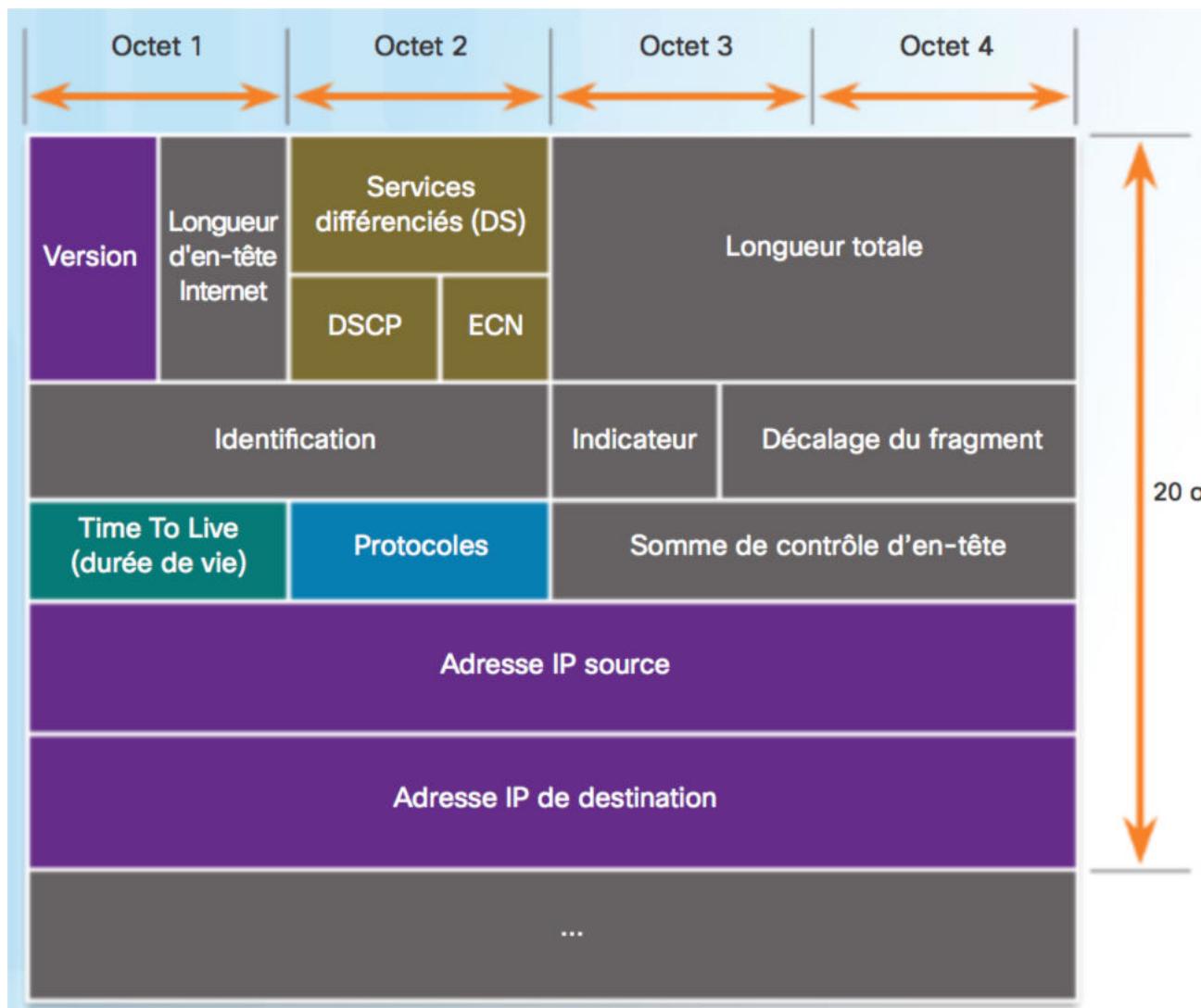
Protocole sans connexion : acheminement au mieux



IP est un protocole de couche réseau peu fiable, il ne garantit donc pas que tous les paquets envoyés seront reçus. D'autres protocoles gèrent le processus de suivi des paquets et garantissent leur livraison.

# Couche 3 : IP

## IP : entête de paquet IPV4



- Version = 0100
- TTL = limite la durée de vie du paquet (Time To Live)
- Protocole = protocole de la couche supérieure tel que TCP ou UDP (couche 4)
- Adresse IP source = source du paquet
- Adresse IP de destination = destination du paquet

# Couche 3 : IP

IP : entête de paquet IPV4 : exemple d'une trame capturée sous Wireshark

Cas d'une trame http

Microsoft: \Device\NPF\_{7B83C130-30C5-4419-B79E-C0868085ABED} [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

| No. | Time        | Source                     | Destination   | Protocol | Length | Info   |
|-----|-------------|----------------------------|---------------|----------|--------|--|
| 1   | 0.000000000 | fe80::b1ee:c4ae:a11ff02::c |               | SSDP     | 208    | M-SEARCH * HTTP/1.1  |
| 2   | 0.30588900  | 192.168.1.109              | 192.168.1.1   | TCP      | 66     | 56081 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=4 SACK_PIE       |
| 3   | 0.30723400  | 192.168.1.109              | 192.168.1.1   | TCP      | 66     | 56082 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=4 SACK_PIE       |
| 4   | 0.31007200  | 192.168.1.1                | 192.168.1.109 | TCP      | 66     | http > 56081 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PIE |
| 5   | 0.31018800  | 192.168.1.109              | 192.168.1.1   | TCP      | 54     | 56081 > http [ACK] Seq=1 Ack=1 Win=66780 Len=0                       |
| 6   | 0.31092800  | 192.168.1.1                | 192.168.1.109 | TCP      | 66     | http > 56082 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PIE |
| 7   | 0.31103000  | 192.168.1.109              | 192.168.1.1   | TCP      | 54     | 56082 > http [ACK] Seq=1 Ack=1 Win=66780 Len=0                       |
| 8   | 0.35044400  | 192.168.1.109              | 192.168.1.1   | HTTP     | 425    | GET / HTTP/1.1   |

+ Frame 8: 425 bytes on wire (3400 bits), 425 bytes captured (3400 bits) on interface 0

+ Ethernet II, Src: IntelCor\_45:d5:c4 (24:77:03:45:d5:c4), Dst: Cisco-Li\_a0:d1:be (00:18:39:a0:d1:be)

- Internet Protocol version 4, Src: 192.168.1.109 (192.168.1.109), Dst: 192.168.1.1 (192.168.1.1)

    Version: 4

    Header length: 20 bytes

+ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

    Total Length: 411

    Identification: 0x3200 (12800)

+ Flags: 0x02 (Don't Fragment)

    Fragment offset: 0

    Time to live: 128

    Protocol: TCP (6)

+ Header checksum: 0x439e [correct]

    Source: 192.168.1.109 (192.168.1.109)

    Destination: 192.168.1.1 (192.168.1.1)

    [Source GeoIP: Unknown]

    [Destination GeoIP: Unknown]

+ Transmission Control Protocol, Src Port: 56081 (56081), Dst Port: http (80), Seq: 1, Ack: 1, Len: 371

+ Hypertext Transfer Protocol

| Hex  | Dec                     | ASCII                           |
|------|-------------------------|---------------------------------|
| 0000 | 00 18 39 a0 d1 be 24 77 | . . . 9 . . \$W . E ] . . . E . |
| 0010 | 01 9b 32 00 40 00 80 06 | . . 2 . @ . . C . . . m . .     |
| 0020 | 41 01 db 11 00 50 a0 cc | ..... P . . D . . [ O4P .       |
| 0030 | 41 37 b0 3d 00 00 47 45 | A7 . . . GE T / HTTP            |
| 0040 | 2f 31 2e 31 0d 0a 48 6f | / 1 . 1 . Ho st: 192 .          |
| 0050 | 31 26 29 20 21 20 21 0d | 160 1 . 1 . Connect             |

Internet Protocol Version 4 (ip), 20 bytes

Packets: 16 Displayed: 16 Marked: 0 Dropped: 0

Profile: Default

# Couche 3 : IP

IP : entête de paquet IPV4 : exemple d'une trame capturée sous Wireshark

Cas d'une trame ICMP

Microsoft: \Device\NPF\_{7B83C130-30C5-4419-B79E-C0868085ABED} [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

| No. | Time       | Source                                 | Destination     | Protocol | Length | Info   |
|-----|------------|--|-----------------|----------|--------|--|
| 16  | 3.64050300 | 192.168.1.109                          | 192.168.1.1     | ICMP     | 74     | Echo (ping) request id=0x0001, seq=5/1280, ttl=128 |
| 17  | 3.64506800 | 192.168.1.1                            | 192.168.1.109   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=5/1280, ttl=64    |
| 18  | 3.68215500 | 192.168.1.109                          | 38.112.107.53   | TCP      | 54     | 55502 > https [ACK] Seq=1 Ack=134 Win=16661 Len=0  |
| 19  | 4.19945400 | fe80::15ff:98d8:d28ff02::c             |                 | SSDP     | 208    | M-SEARCH * HTTP/1.1                                |
| 20  | 4.60748800 | fe80::15ff:98d8:d28fe80::b1ee:c4ae:a11 |                 | SSDP     | 453    | HTTP/1.1 200 OK                                    |
| 21  | 4.64229900 | 192.168.1.109                          | 192.168.1.1     | ICMP     | 74     | Echo (ping) request id=0x0001, seq=6/1536, ttl=128 |
| 22  | 4.64509200 | 192.168.1.1                            | 192.168.1.109   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=6/1536, ttl=64    |
| 23  | 4.73605200 | 192.168.1.109                          | 255.255.255.255 | DB-LSP-  | 154    | Dropbox LAN sync Discovery Protocol                |

Frame 16: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

Ethernet II, Src: IntelCor\_45:5d:c4 (24:77:03:45:5d:c4), Dst: Cisco-Li\_a0:d1:be (00:18:39:a0:d1:be)

Internet Protocol Version 4, Src: 192.168.1.109 (192.168.1.109), Dst: 192.168.1.1 (192.168.1.1)

Version: 4  
Header length: 20 bytes  
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
Total Length: 60  
Identification: 0x3704 (14084)  
Flags: 0x00  
Fragment offset: 0  
Time to live: 128  
Protocol: ICMP (1)  
Header checksum: 0x7ffe [correct]  
Source: 192.168.1.109 (192.168.1.109)  
Destination: 192.168.1.1 (192.168.1.1)  
[Source GeoIP: Unknown]  
[Destination GeoIP: Unknown]  
Internet Control Message Protocol

| 0000 | 00 | 18 | 39 | a0 | d1 | be | 24 | 77 | 03 | 45 | 5d | c4 | 08 | 00 | 45 | 00                  | . . . . . \$W . E ] . . . E .   |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------|---------------------------------|
| 0010 | 00 | 3c | 37 | 04 | 00 | 00 | 80 | 01 | 7f | fe | c0 | a8 | 01 | 6d | c0 | a8                  | . . <7. . . . . . . m . .       |
| 0020 | 01 | 01 | 08 | 00 | 4d | 56 | 00 | 01 | 00 | 05 | 61 | 62 | 63 | 64 | 65 | 66                  | . . . . . M V . . . . . abcdef  |
| 0030 | 67 | 68 | 69 | 6a | 6b | 6c | 6d | 6e | 6f | 70 | 71 | 72 | 73 | 74 | 75 | 76                  | g h i j k l m n o p q r s t u v |
| 0040 | 77 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |    |    |    |    |    | w a b c d e f g h i |                                 |

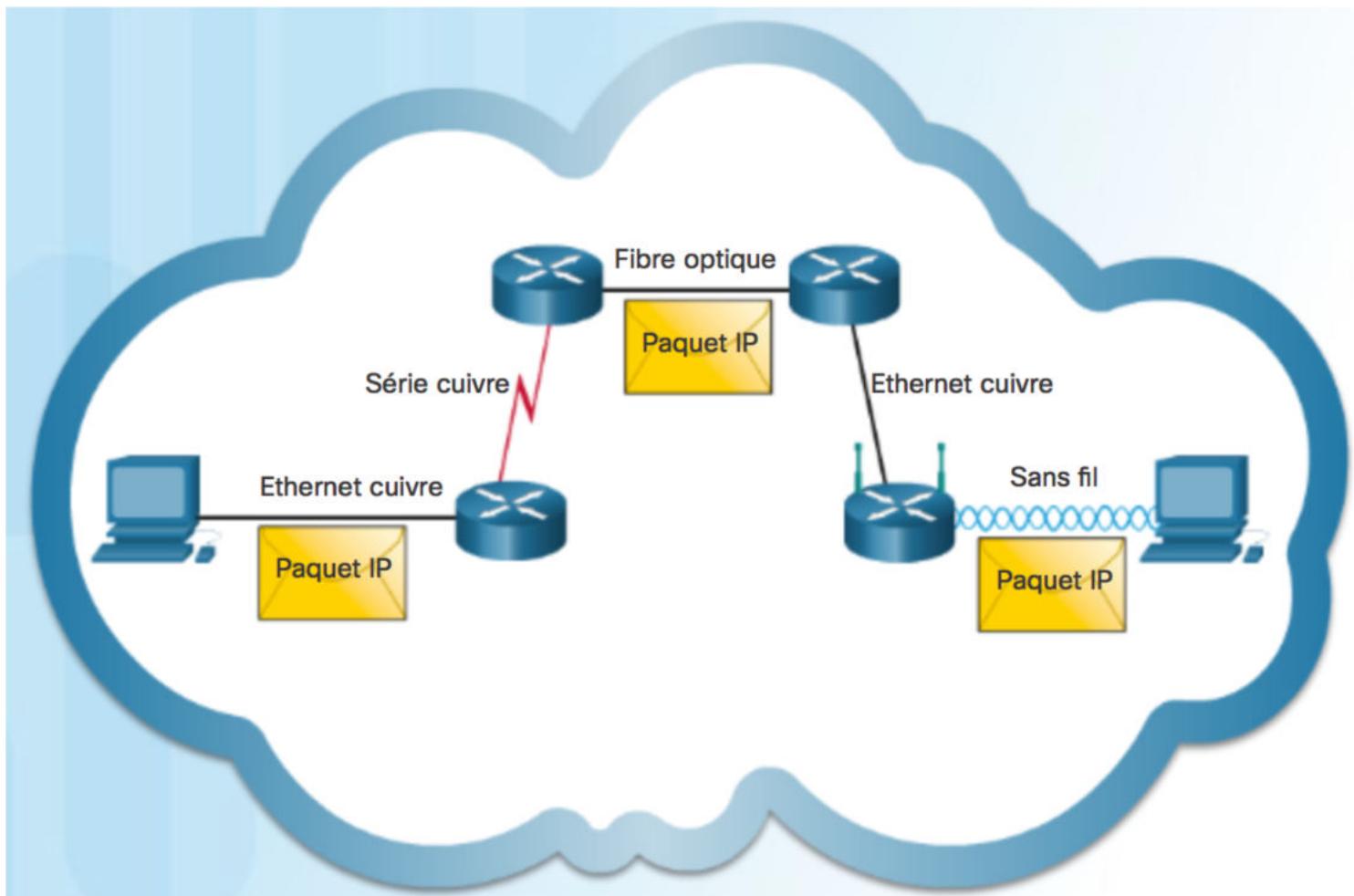
Internet Protocol Version 4 (ip), 20 bytes

Packets: 35 Displayed: 35 Marked: 0 Dropped: 0

Profile: Default

# Couche 3 : IP

Protocole sans connexion : indépendant du support



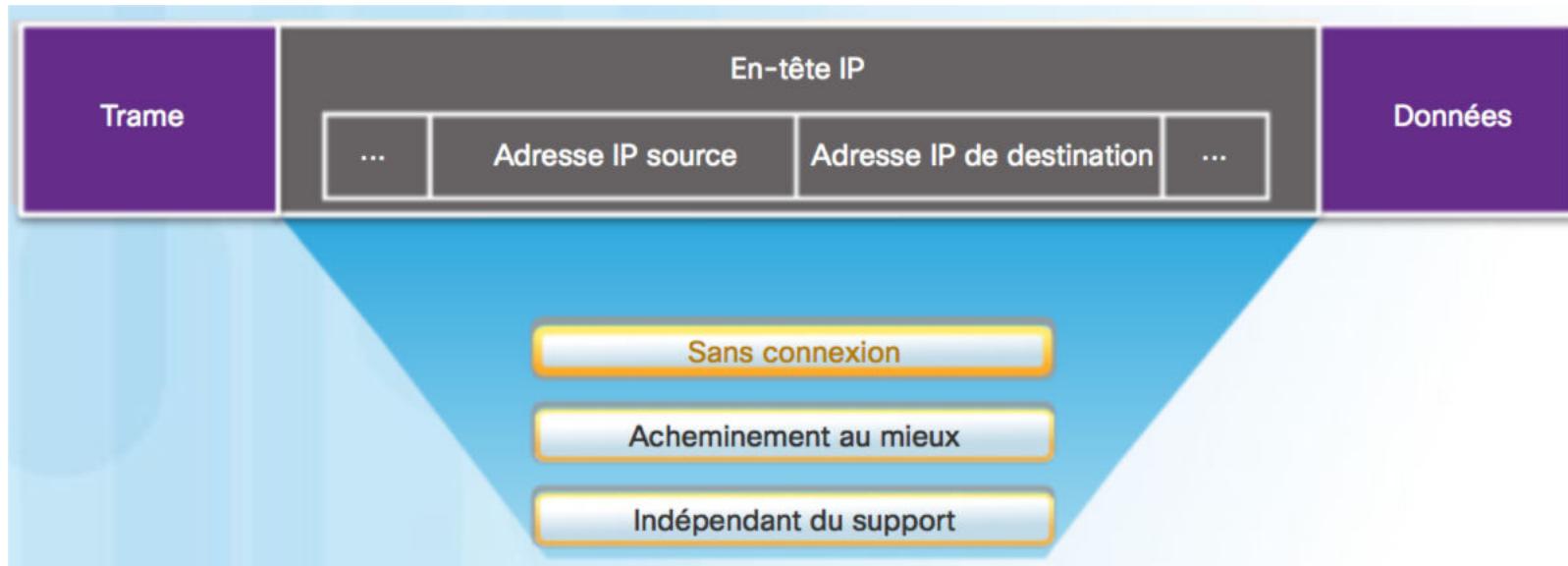
Les paquets IP peuvent transiter par différents supports.

# Couche 3 : IP

En résumé, le protocole IP ne nécessite :

- **aucun échange initial** d'informations de contrôle pour établir une connexion de bout en bout avant le transfert des paquets,
- **ni de champs supplémentaires** dans l'en-tête d'unité de données de protocole pour maintenir cette connexion.

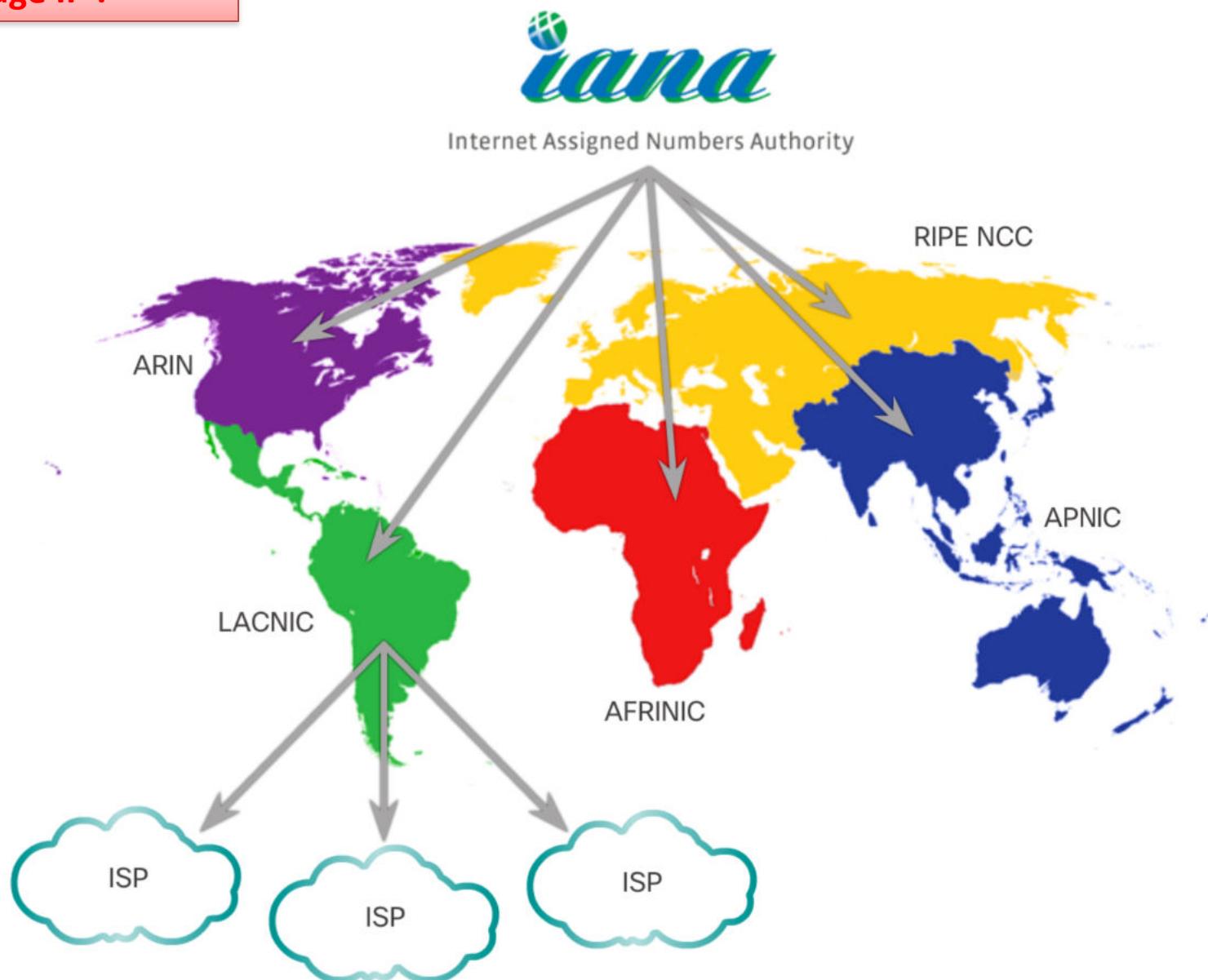
**Intérêt** : réduction considérable de la surcharge d'IP.



**Protocole peu fiable** : d'autres protocoles gèrent le suivi des paquets et garantissent leur acheminement

# Couche 3 : IP

L'adressage IP :



# Couche 3 : IP

## L'adressage IP :

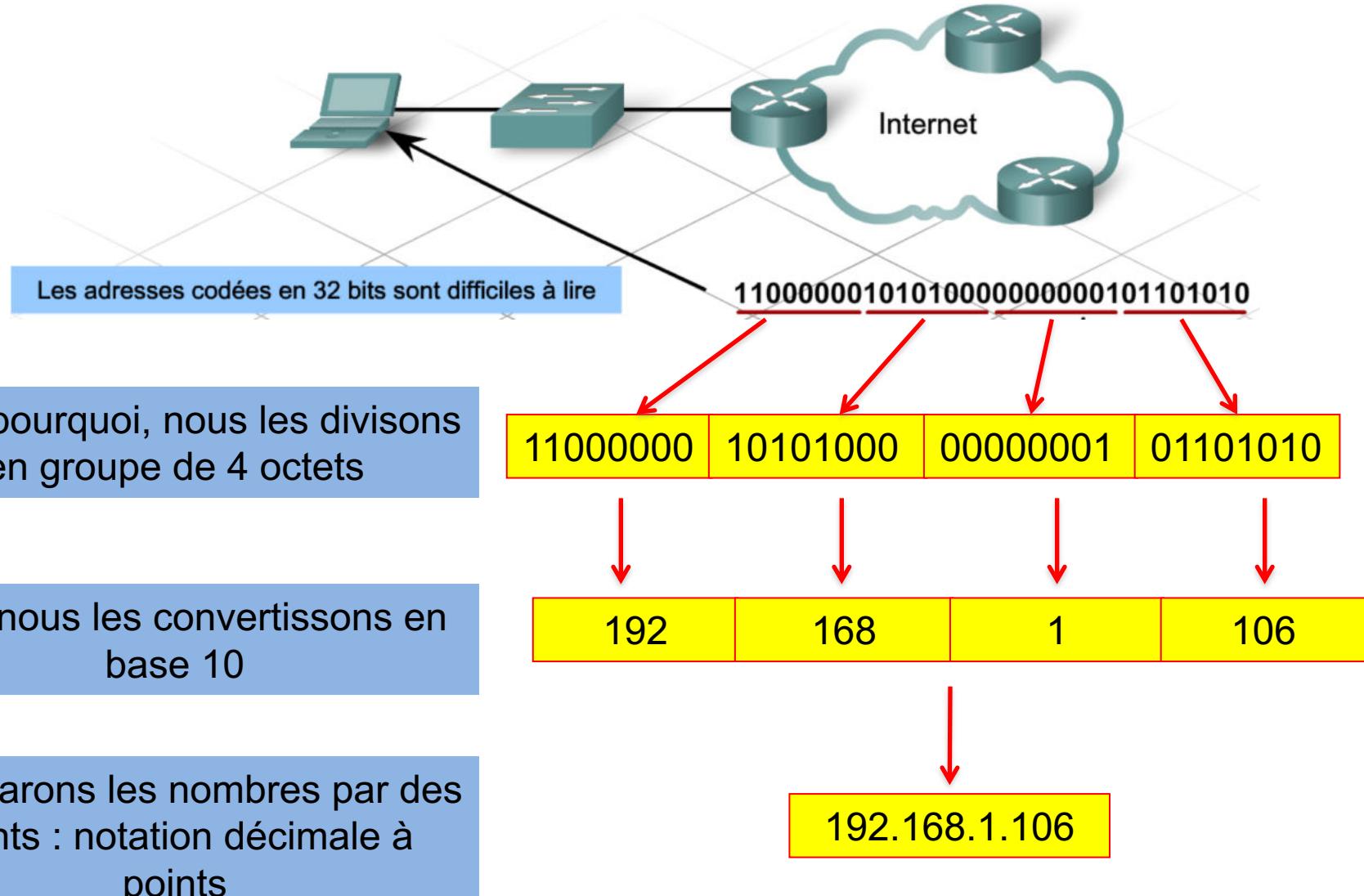
- un des aspects ***les plus importants*** de la communication sur un inter-réseau.
- méthode utilisée pour ***identifier*** les hôtes et les périphériques réseau.
- face au développement d'Internet et à l'augmentation du nombre d'hôtes connectés à ce réseau, les schémas d'adressage IP ont dû s'adapter.
- mais la ***structure*** de base des adresses IP utilisée pour IPv4 est restée identique.

- **Principe** : adresse IP 32 bits ***unique*** attribuée à chaque hôte du réseau !
- Format ***binaire***
- Problème : grand nombre difficile à lire en binaire :  
11000000.10101000.00000001.01101010
- Adoption d'une ***notation en décimale à point***.
  - chacun des quatre octets est converti en un nombre décimal.
  - par exemple, l'adresse IP suivante :

11000000.10101000.00000001.01101010

est représentée sous la forme 192.168.1.106 en décimale à point.

# Couche 3 : IP

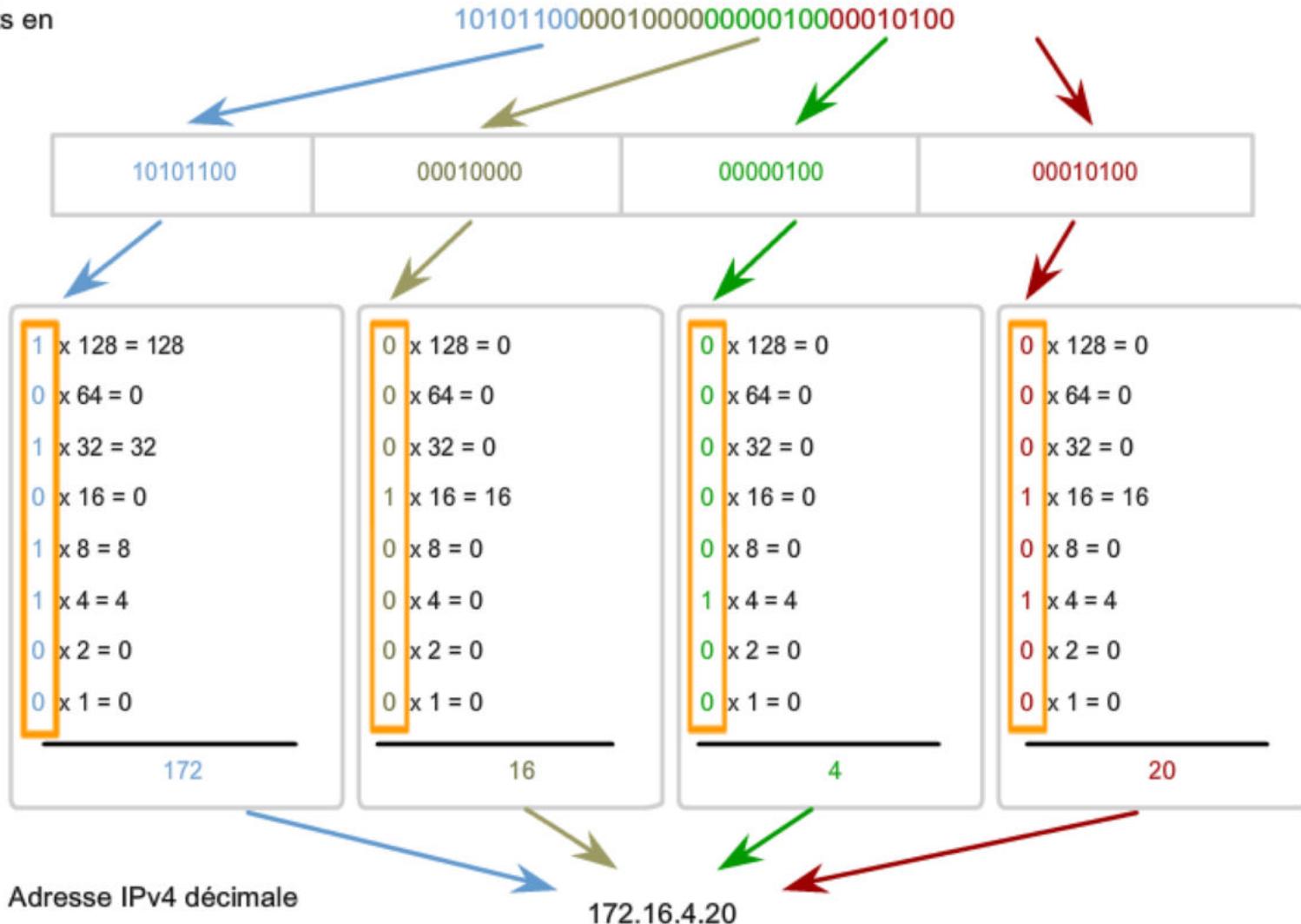


# Couche 3 : IP

Conversion d'une adresse IPv4 de la notation binaire en notation en décimale à point

Adresse IPv4 binaire 10101100000100000000010000010100

Divisez les 32 bits en  
4 octets.



# Couche 3 : IP

## Au tout début d'Internet :

- organisations connectées à Internet très peu nombreuses
- les réseaux étaient attribués seulement par les huit premiers bits (premier octet)
- les 24 bits (3 octets) restants étaient utilisés pour les adresses d'hôtes locaux

D'où : 8 bits seulement pour le réseau,

- donc 256 réseaux distincts, plus de 16 millions d'hôtes chacun
- logique car Internet = grandes Universités et org. gouvernementales et militaires
- Puis : augmentation exponentielle du nombre d'hôtes
- Nécessite de convenir ***d'une nouvelle méthode d'attribution*** avec un plus grand nombre de réseaux distincts

|           | 1 octet<br>8 bits | 1 octet<br>8 bits | 1 octet<br>8 bits | 1 octet<br>8 bits |
|-----------|-------------------|-------------------|-------------------|-------------------|
| Classe A: | N                 | H                 | H                 | H                 |
| Classe B: | N                 | N                 | H                 | H                 |
| Classe C: | N                 | N                 | N                 | H                 |



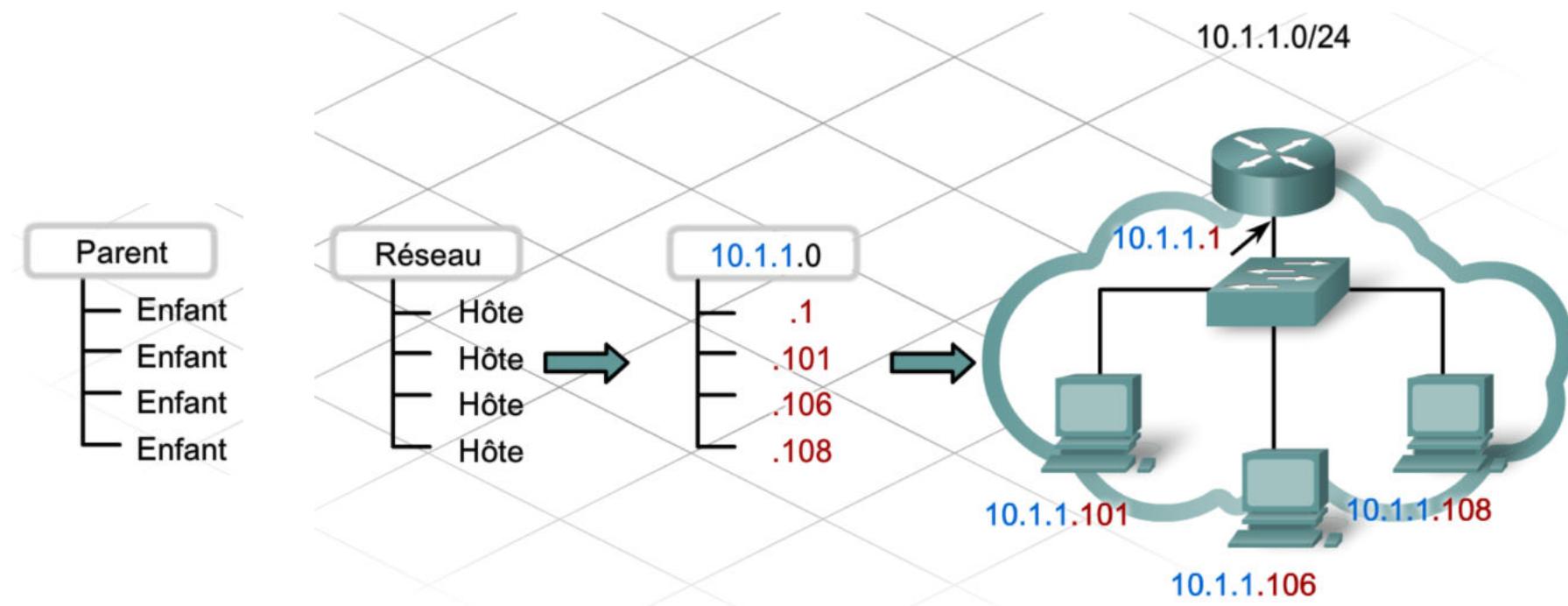
N = n° de réseau attribué pour l'ARIN

H = n° d'hôte attribué par l'administrateur (Host)

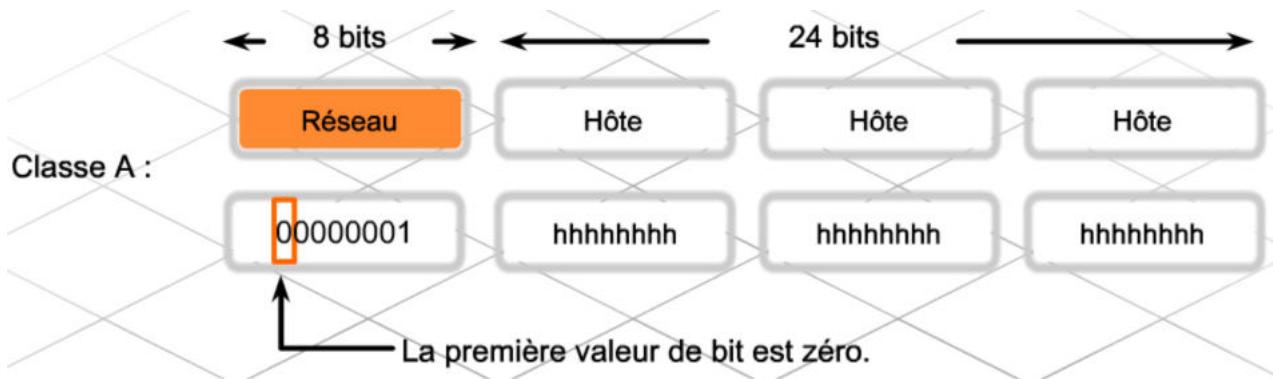
# Couche 3 : IP

Les adresses IP sont **hiérarchiques** (arbre généalogique):

- une partie du nombre : réseau (parent)
- le reste : hôte (enfant)

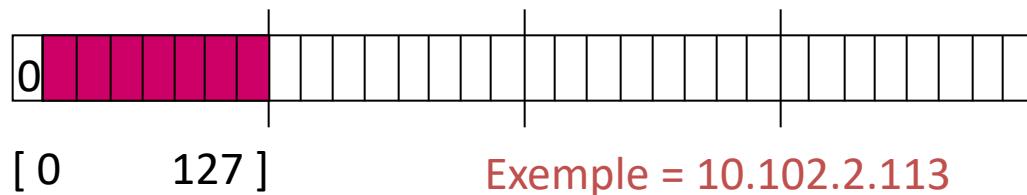


# Couche 3 : IP

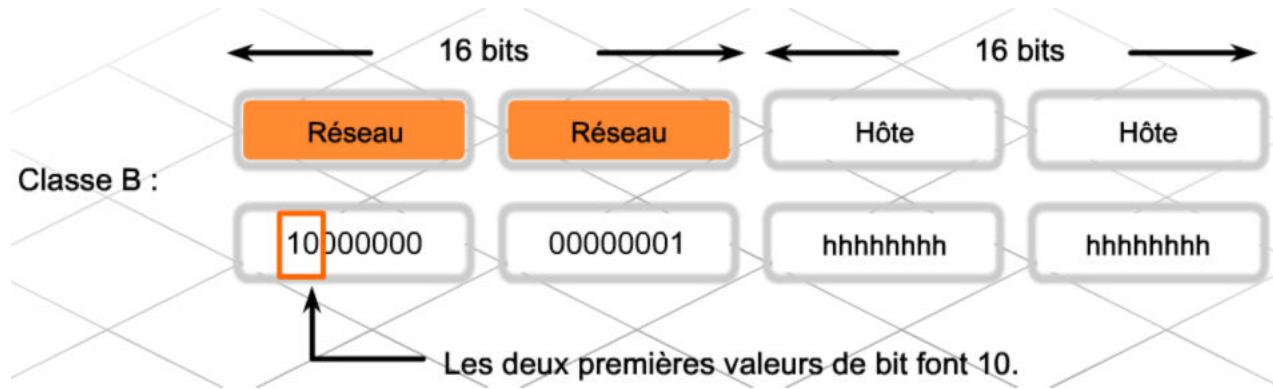


## Classe A :

- **1<sup>er</sup> octet pour le réseau, 3 autres pour les hôtes**
- 1<sup>er</sup> octet : compris entre 1 et 126
- l'adresse de classe A **127** est réservée au test de bouclage
- ces adresses sont réservées aux très grands réseaux (plus de 65 534 hôtes)

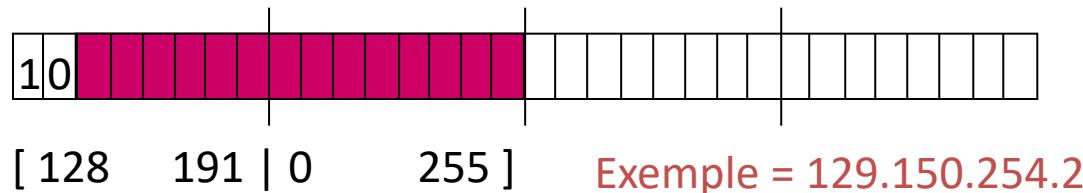


# Couche 3 : IP

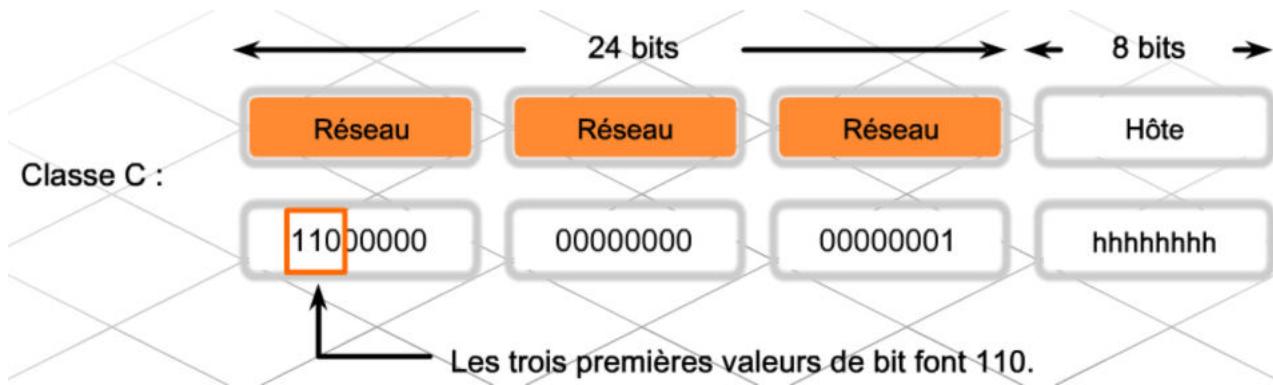


## Classe B :

- **les 2 premiers octets pour le réseau, 2 autres pour les hôtes**
- adresses de classe B : comprises entre 128 et 191
- l'adresse de classe A **127** est réservée au test de bouclage
- ces adresses sont réservées aux grands réseaux (entre 255 et 65 534 hôtes)

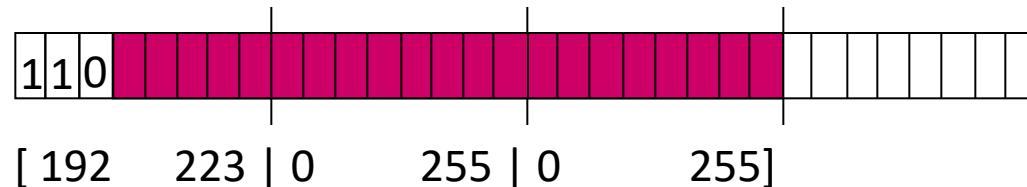


# Couche 3 : IP

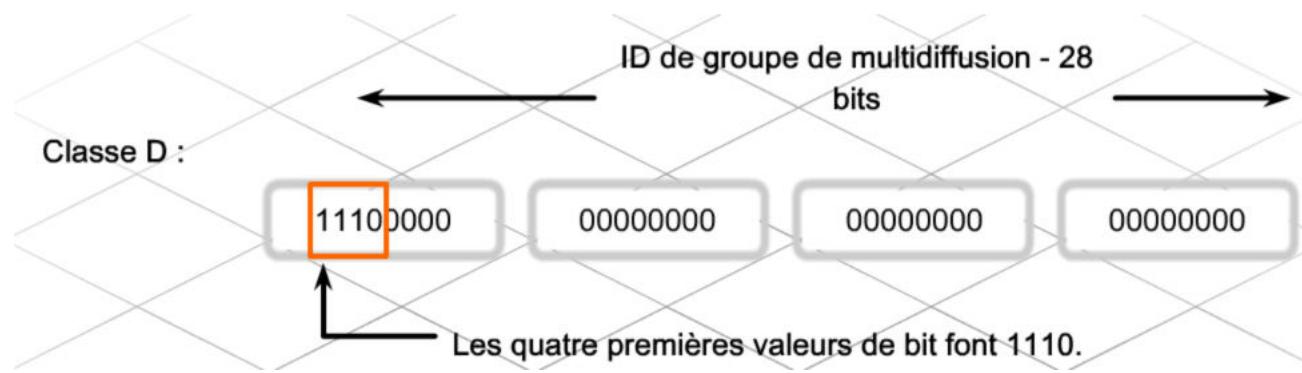


## Classe C :

- *les 3 premiers octets pour le réseau, le dernier pour les hôtes*
- adresses de classe C : comprises entre 192 et 223
- ces adresses sont réservées aux petits réseaux (255 hôtes ou moins)

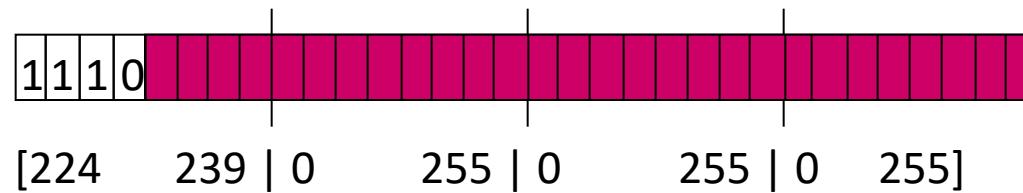


# Couche 3 : IP



## Classe D :

- *Adressage de multidiffusion*
- adresses de classe D : comprises entre 224 et 239
- Video en streaming, ou Irs, fog, système de sauvegarde/restauration



# Couche 3 : IP

Classes d'adresses IP

| Classe de l'adresse | Plage du premier octet (décimal) | Bits du premier octet ( <b>les bits verts ne changent pas</b> ) | Parties réseau (N) et hôte (H) de l'adresse | Masque de sous-réseau par défaut (décimal et binaire)        | Nombre de réseaux et d'hôtes possibles par réseau         | Notes et plage d'adresses d'hôte**                          |
|---------------------|----------------------------------|---|---|--|---|---|
| A                   | 1 - 127*                         | 00000000 - 01111111   | N.H.H.H                                     | 255.0.0.0<br>1111111.000000<br>00.00000000.0000<br>0000      | 128 réseaux (2^7)<br>16 777 214 hôtes par réseau (2^24-2) | Commercial<br>1.0.0.1 -<br>126.255.255.254                  |
| B                   | 128 - 191                        | 10000000 - 10111111   | N.N.H.H                                     | 255.255.0.0<br>11111111.111111<br>11.00000000.0000<br>0000   | 16 384 réseaux (2^14)<br>65 534 hôtes par réseau (2^16-2) | Commercial<br>128.0.0.1 -<br>191.255.255.254                |
| C                   | 192 - 223                        | 11000000 - 11011111   | N.N.N.H                                     | 255.255.255.0<br>11111111.111111<br>11.11111111.0000<br>0000 | 2 097 152 réseaux (2^21)<br>254 hôtes par réseau (2^8-2)  | Commercial<br>192.0.0.1 -<br>223.255.255.254                |
| D                   | 224- 239                         | 11100000 - 11101111   | Non destiné à une utilisation commerciale   |  |   | Multidiffusion (réservée)<br>224.0.0.1 -<br>239.255.255.255 |

\* L'adresse de classe A 127.0.0.0 est réservée au test de bouclage.

\*\* Les adresses ne contenant que des zéros (0) et que des uns (1) sont des adresses d'hôte

# Couche 3 : IP

## Particularités

Adresse réseau

Adresse de diffusion

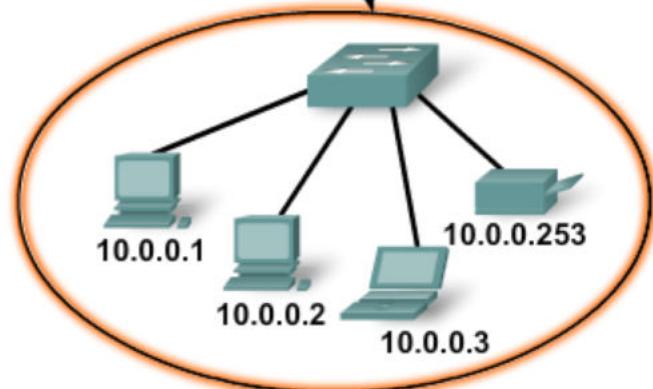
Adresse d'hôte

Placez le pointeur sur un élément pour en savoir plus

10.0.0.0 fait référence au réseau dans son ensemble. Tous les périphériques de ce réseau partagent les mêmes bits d'adresse réseau.

## Types d'adresse

| Réseau                     | Hôte     |
|----------------------------|----------|
| 10 0 0                     | 0        |
| 00001010 00000000 00000000 | 00000000 |
| 10 0 0                     | 255      |
| 00001010 00000000 00000000 | 11111111 |
| 10 0 0                     | 1        |
| 00001010 00000000 00000000 | 00000001 |



# Couche 3 : IP

## Types d'adresse

Adresse réseau

| Réseau   |          |          | Hôte     |
|----------|----------|----------|----------|
| 10       | 0        | 0        | 0        |
| 00001010 | 00000000 | 00000000 | 00000000 |

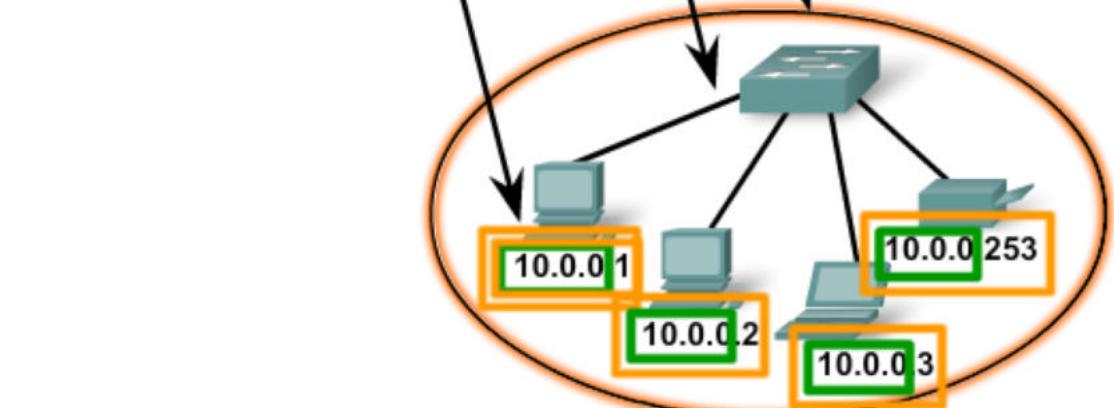
Adresse de diffusion

|          |          |          |          |
|----------|----------|----------|----------|
| 10       | 0        | 0        | 255      |
| 00001010 | 00000000 | 00000000 | 11111111 |

Adresse d'hôte

|          |          |          |          |
|----------|----------|----------|----------|
| 10       | 0        | 0        | 1        |
| 00001010 | 00000000 | 00000000 | 00000001 |

Masque : 255.255.255.0



# Couche 3 : IP

## Groupe IETF :

- Création de classes séparées (A, B, C, D, E) publiques
- Création d'un espace d'adressage à l'usage de réseaux privés

## Réseaux privés :

- Ils ne sont pas reliés aux réseaux publics
- Ces adresses ne sont pas acheminées sur le réseau Internet
- Création d'un espace d'adressage à l'usage de réseaux privés
- donc : utilisation d'un même schéma d'adressage privé sans conflit
- **Objectif** : réduire le nombre d'adresses IP enregistrées uniques

| Classe | Adresses IP privées<br>(RFC 1918) | Masque de sous-réseau par défaut | Nombre de réseaux | Hôtes par réseau | Nombre total d'hôtes |
|--------|-----------------------------------|----------------------------------|-------------------|------------------|----------------------|
| A      | De 10.0.0.0 à 10.255.255.255      | 255.0.0.0                        | 1                 | 16,777,214       | 16,777,214           |
| B      | De 172.16.0.0 à 172.31.255.255    | 255.255.0.0                      | 16                | 65,534           | 1,048,544            |
| C      | De 192.168.0.0 à 192.168.255.255  | 255.255.255.0                    | 256               | 254              | 65,024               |

# Couche 3 : IP

## Masque réseaux :

- Ce n'est **pas une adresse**  $\Rightarrow$  préfixe réseau étendu
- Détermination de la partie réseau et de la partie hôte
- Longueur : 32 bits (comme adresse IP)
- Détermination du masque :
  - Exprimer l'adresse du réseau au format binaire
  - Remplacer les bits de la portion réseau par des 1
  - Remplacer les bits de la portion hôte par des 0
  - Convertir l'adresse binaire au format décimal

## Masque par défaut :

- classe A : 255.0.0.0 ou /8
- classe B : 255.255.0.0 ou /16
- classe C : 255.255.255.0 ou /24

1111 1111 . 1111 1111 . 0000 0000 . 0000 0000

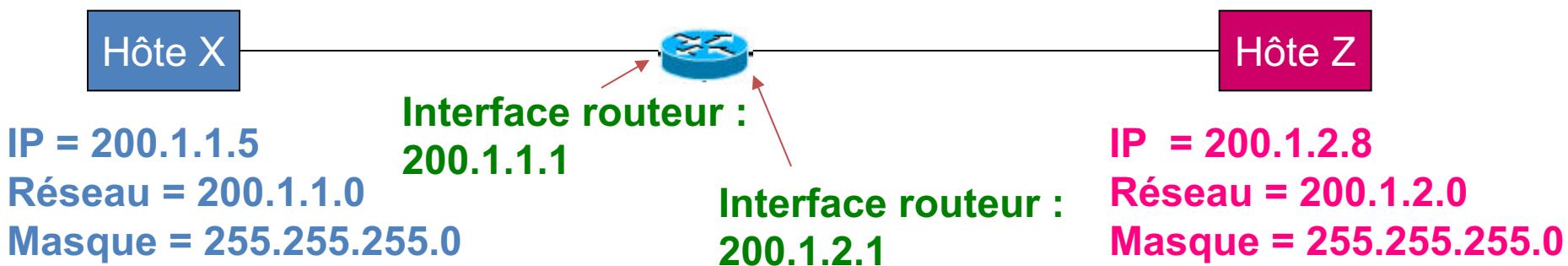
16 bits pour  
le réseau

16 bits pour  
l'hôte

# Couche 3 : IP

**Exemple :** utilisation du masque par défaut de classe C

- *Objectif :*
  - L'hôte X veut envoyer un message à l'hôte Z
  - Déterminer le réseau dont fait partie l'hôte Z
- Utilisation du *masque* par défaut
- Configuration



# Couche 3 : IP

## Exemple (suite) : solution

- A. comparaison de l'adresse IP de X avec son masque de sous-réseau à l'aide d'une opération AND

- IP de X = 1100 1000 . 0000 0001 . 0000 0001 . 0000 0101 (200.1.1.5)  
– Masque = 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000 (255.255.255.0)  
– Résultat 1 = 1100 1000 . 0000 0001 . 0000 0001 . 0000 0000 (200.1.1.0)



200.1.1.0 = adresse réseau de X

- B. comparaison de l'adresse IP de Z avec son masque :

- IP de Z = 1100 1000 . 0000 0001 . 0000 0010 . 0000 1000 (200.1.2.8)  
– Masque = 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000 (255.255.255.0)  
– Résultat 2 = 1100 1000 . 0000 0001 . 0000 0010 . 0000 0000 (200.1.2.0)



200.1.2.0 = adresse réseau de Z

- C. Comparaison des 2 résultats

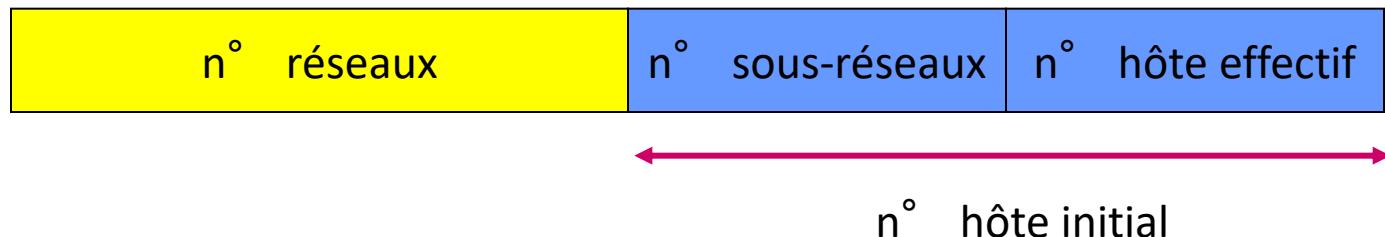
- Résultat 1 ≠ Résultat 2 donc Z n'est pas au LAN de X
- X envoie son paquet à la passerelle, c'est à dire à l'interface routeur 200.1.1.1
- Opération AND effectuée par le routeur pour déterminer l'interface par laquelle il doit transmettre le paquet

# Couche 3 : IP

## Sous -- réseaux :

- **Découpage** du réseau en entités plus petites
- Décision de l'administrateur du site
- Adresse de sous-réseau **prélevée sur la partie hôte**
- Longueur calculée en bits décidée par l'administrateur

Adresse IP d'un élément du réseau (ordinateur) :



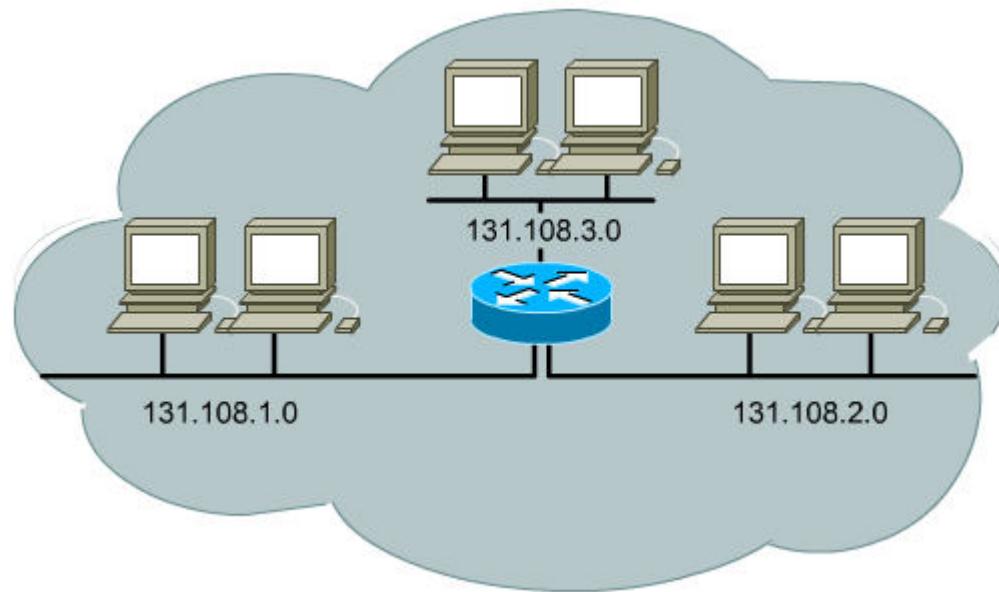
- Découpage inconnu de l'extérieur
- Réalisation par un masque de sous-réseau
- Interconnexion par des **routeurs**

Raison principale = meilleure **structuration** de site

# Couche 3 : IP

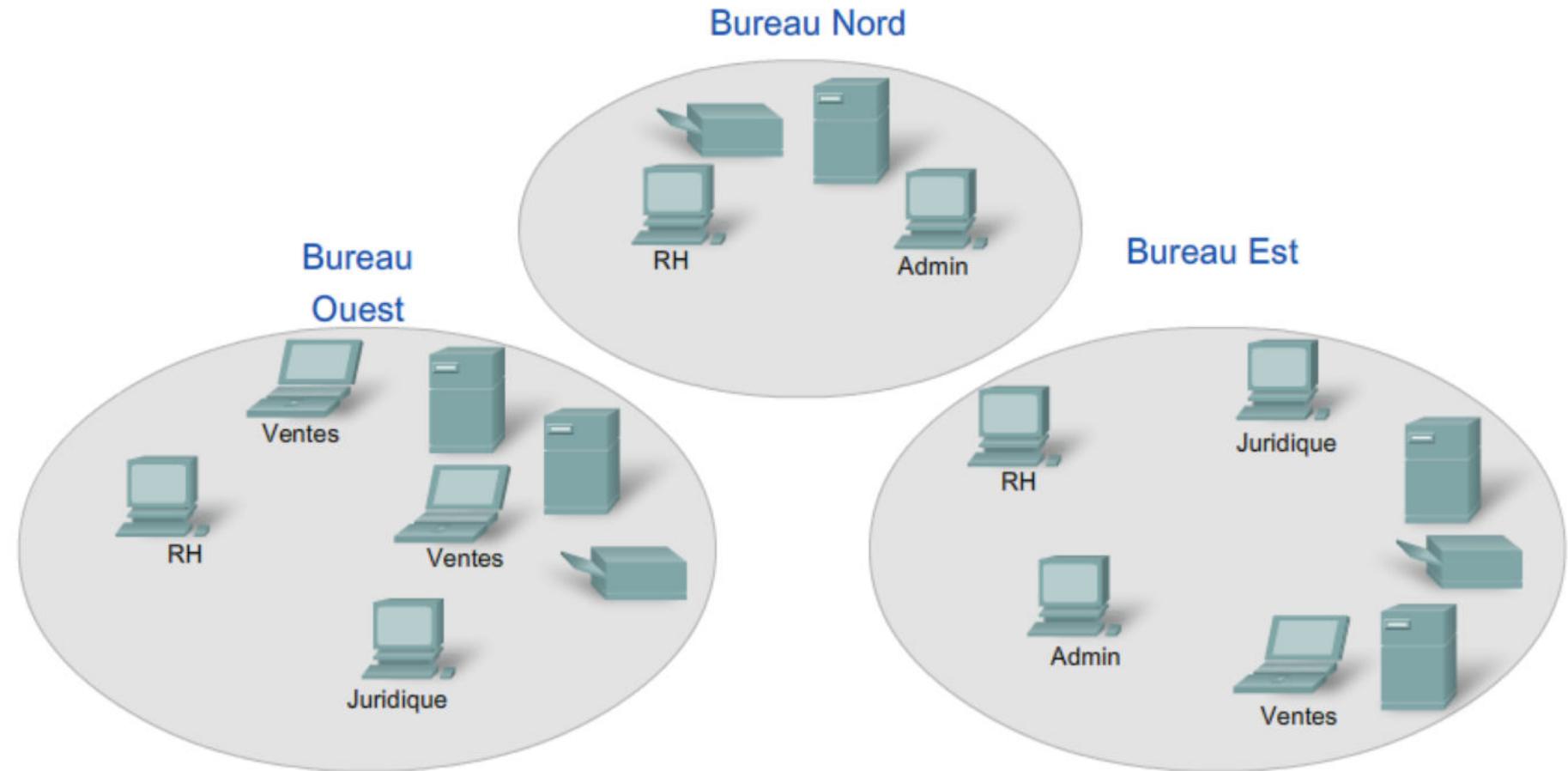
## Exemple de sous -- réseaux :

- Réseau de classe B de numéro IP : **131.108.0.0/16**
- On peut créer, par exemple 3 sous-réseaux d'adresse :
  - 131.108.1.0 /24
  - 131.108.2.0/24
  - 131.108.3.0/24



# Couche 3 : IP

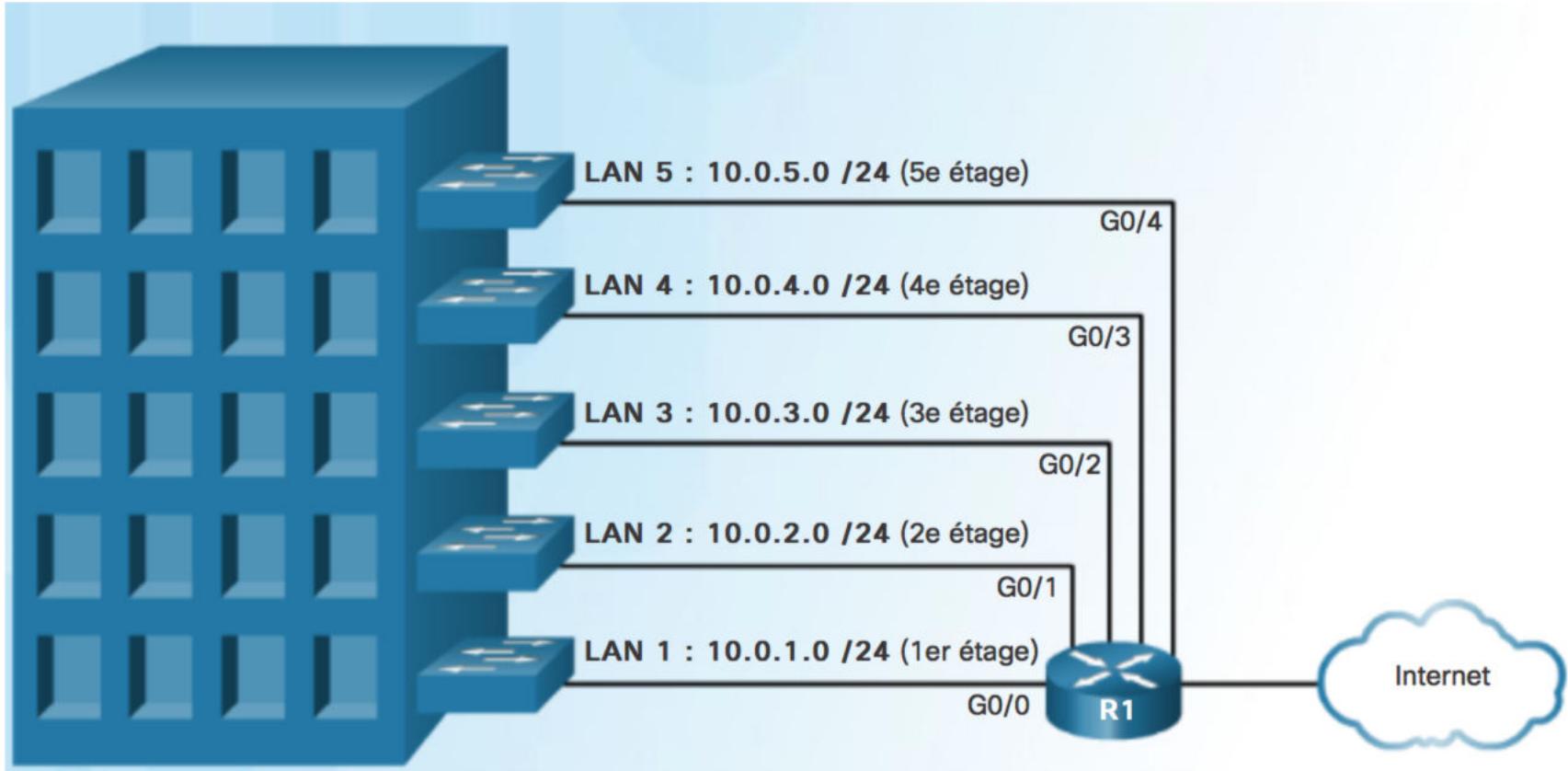
## Raison géographique



Le simple fait de câbler le réseau physique peut faire de l'emplacement géographique un début logique pour la segmentation d'un réseau.

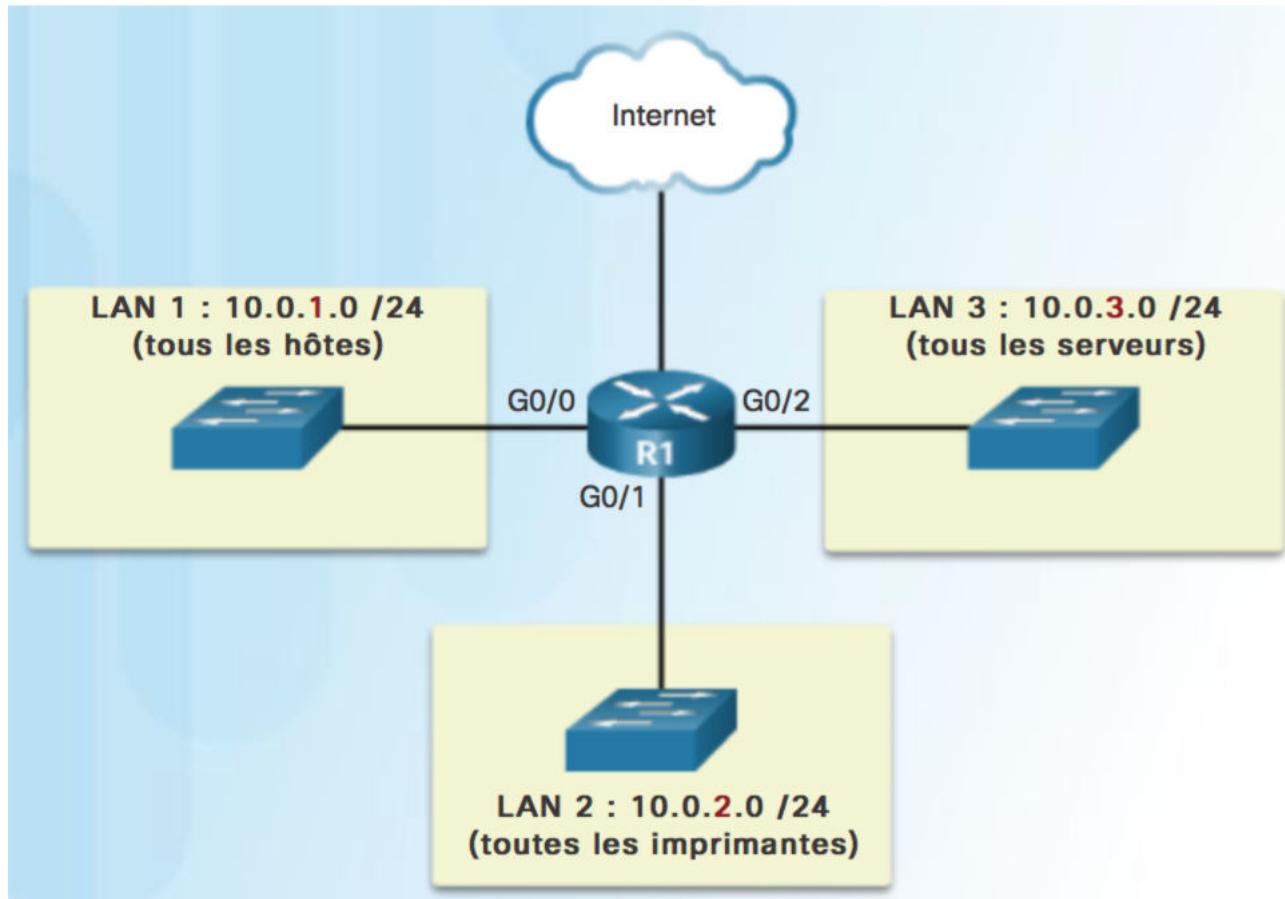
# Couche 3 : IP

## Raison géographique



# Couche 3 : IP

Par type d'appareils

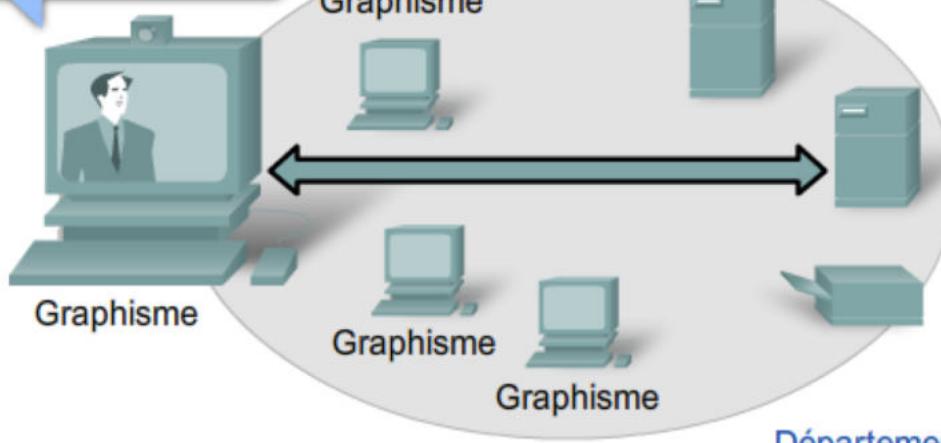


# Couche 3 : IP

## Selon les objectifs

Le volume et le type de données générées par une classe d'utilisateurs peuvent justifier le groupement d'utilisateurs similaires dans un réseau.

Les graphistes ont besoin d'une bande passante élevée pour créer des vidéos.

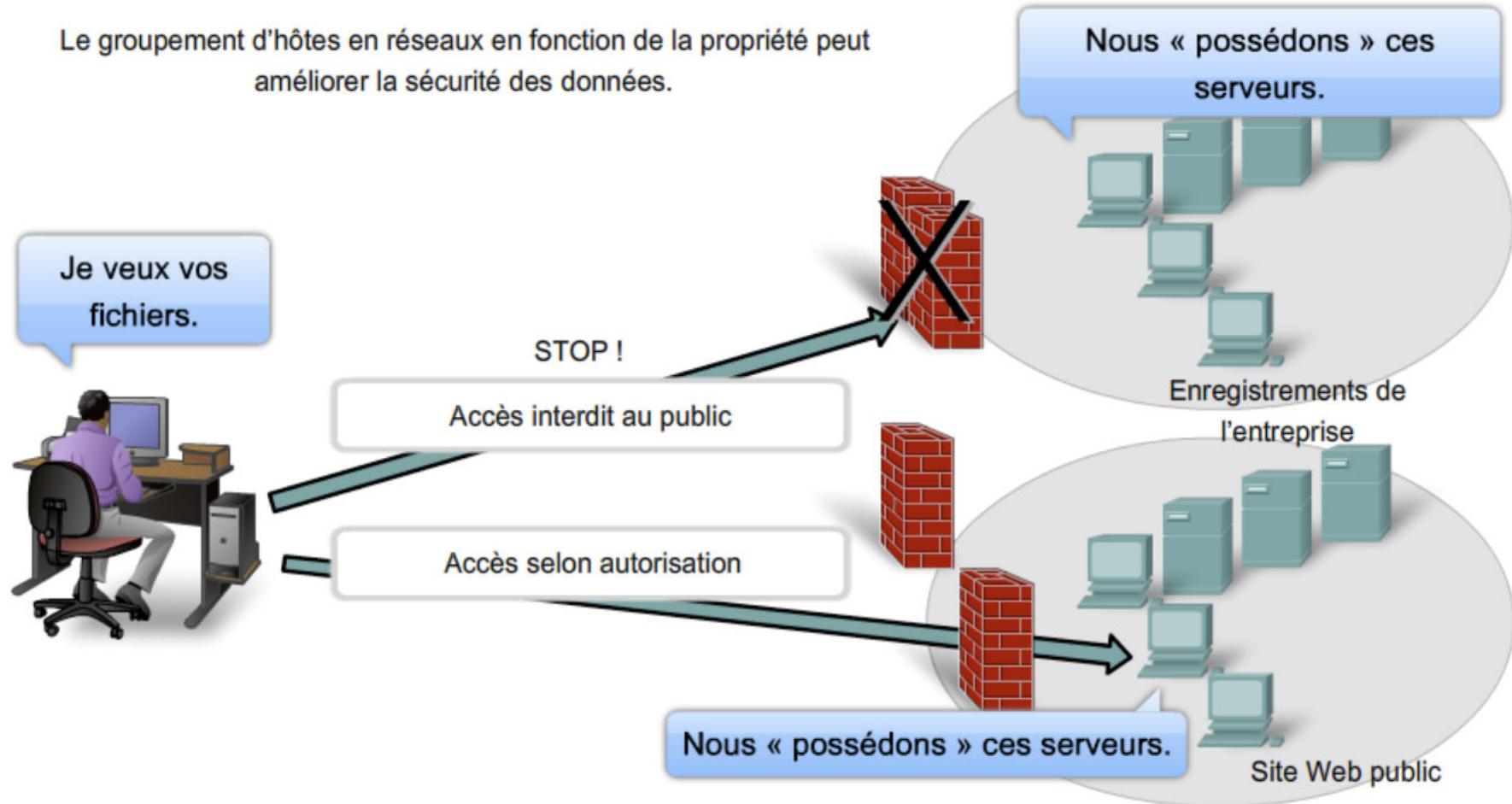


Les vendeurs ont besoin de vitesse et d'une fiabilité à 100 %.

# Couche 3 : IP

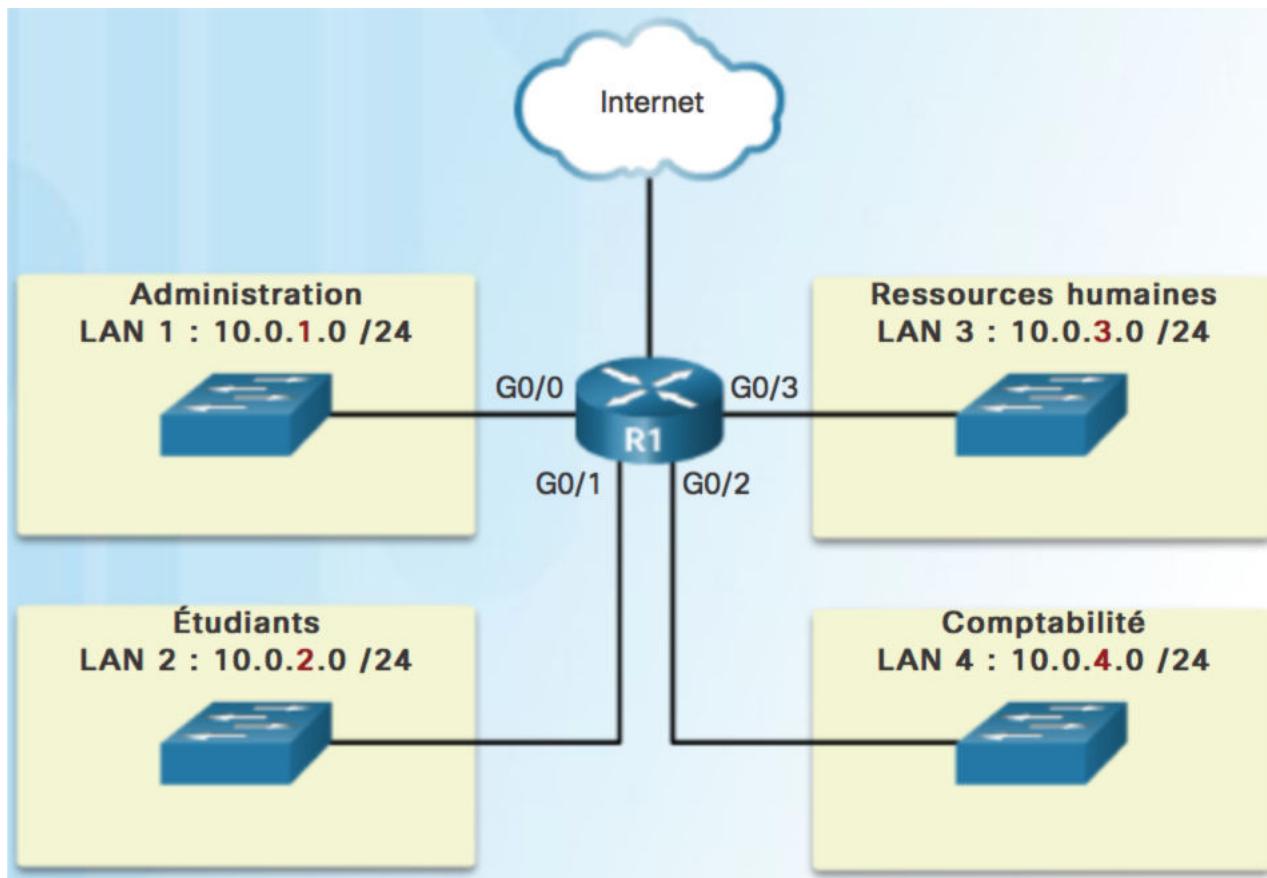
## Raison de sécurité

Le groupement d'hôtes en réseaux en fonction de la propriété peut améliorer la sécurité des données.

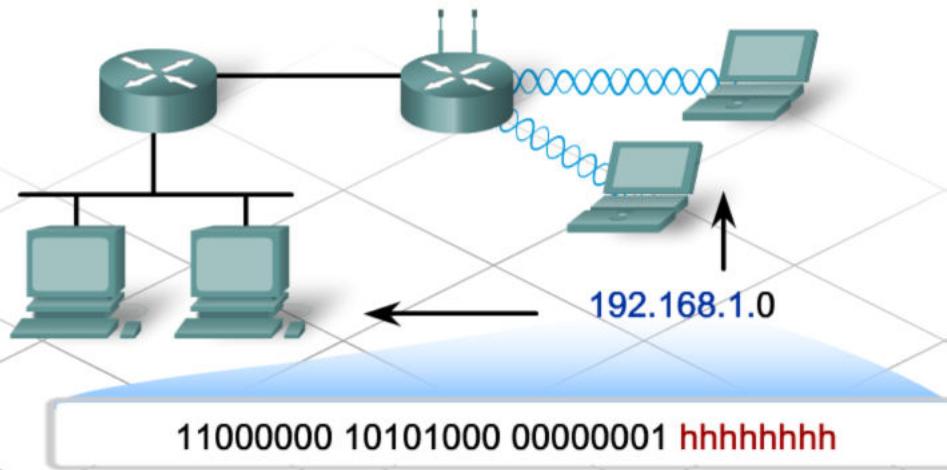


# Couche 3 : IP

## Communication entre les réseaux par routeur



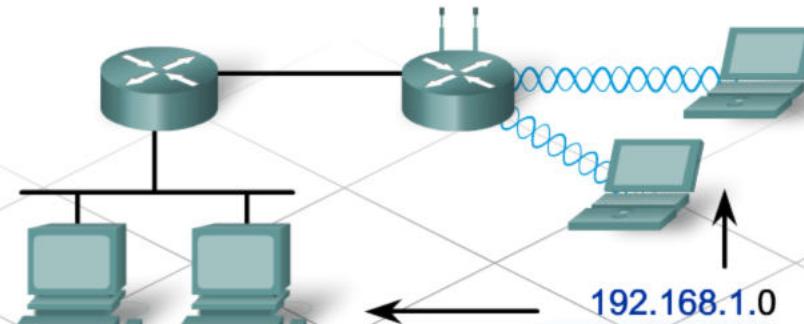
# Couche 3 : IP



| Bits d'ID de sous-réseau | Bits d'ID d'hôte | Nombre de sous-réseaux | Nombre d'hôtes | Configuration binaire |
|--------------------------|------------------|------------------------|----------------|-----------------------|
| 0                        | 8                | 1                      | 254            | <b>hhhhhhhh</b>       |

Bits d'ID de sous-réseau = 0, le réseau dispose d'un sous-réseau.

# Couche 3 : IP

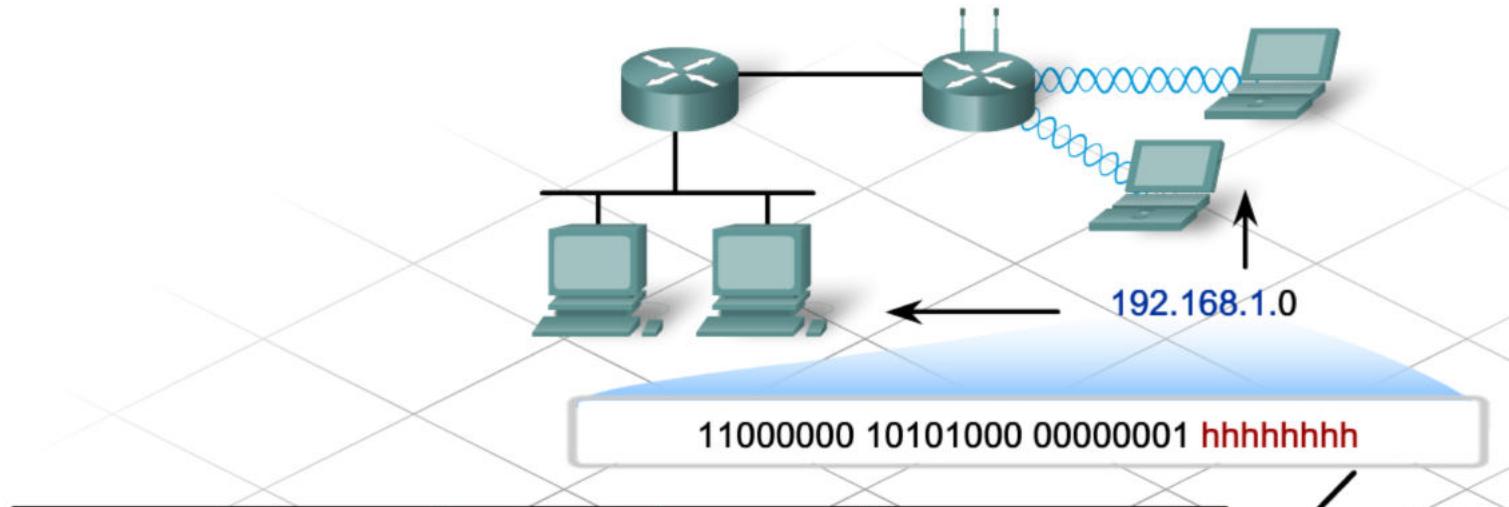


11000000 10101000 00000001 **hhhhhhhh**

| Bits d'ID de sous-réseau | Bits d'ID d'hôte | Nombre de sous-réseaux | Nombre d'hôtes | Configuration binaire |
|--------------------------|------------------|------------------------|----------------|-----------------------|
| 0                        | 8                | 1                      | 254            | <b>hhhhhhhh</b>       |
| 1                        | 7                | 2                      | 126            | <b>shhhhhhh</b>       |

Dès qu'un des bits d'hôte est désigné comme bit de sous-réseau, le réseau dispose de deux sous-réseaux. Souvenez-vous qu'en mode binaire, un bit peut afficher l'état 1 ou 0. Le nombre de sous-réseaux est donc  $2^s$ .

# Couche 3 : IP



| Bits d'<br>ID de sous-<br>réseau | Bits d'ID<br>d'hôte | Nombre<br>de sous-<br>réseaux | Nombre<br>d'hôtes | Configuration binaire |
|----------------------------------|---------------------|-------------------------------|-------------------|-----------------------|
| 0                                | 8                   | 1                             | 254               | <b>hhhhhhhh</b>       |
| 1                                | 7                   | 2                             | 126               | <b>shhhhhhh</b>       |
| 2                                | 6                   | 4                             | 62                | <b>sshhhhhh</b>       |
| 3                                | 5                   | 8                             | 30                | <b>ssshhhhh</b>       |
| 4                                | 4                   | 16                            | 14                | <b>sssshhh</b>        |
| 5                                | 3                   | 32                            | 6                 | <b>sssssbbb</b>       |
| 6                                | 2                   | 64                            | 2                 | <b>ssssssbb</b>       |

# Couche 3 : IP

Une seule adresse réseau est disponible.

192.168.1.0 (/24)

255.255.255.0

Adresse :

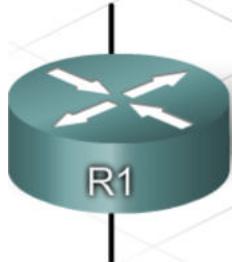
Masque :

11000000.10101000.00000001.00000000

11111111.11111111.11111111.00000000

← Partie réseau de →  
l'adresse

Réseau 0



Grâce au découpage en sous-réseaux, deux adresses réseau sont disponibles.

0 192.168.1.0 (/25)      Adresse : 11000000.10101000.00000001.00000000  
                                Masque : 11111111.11111111.11111111.10000000

1 192.168.1.128 (/25)      Adresse : 11000000.10101000.00000001.10000000  
                                Masque : 11111111.11111111.11111111.10000000

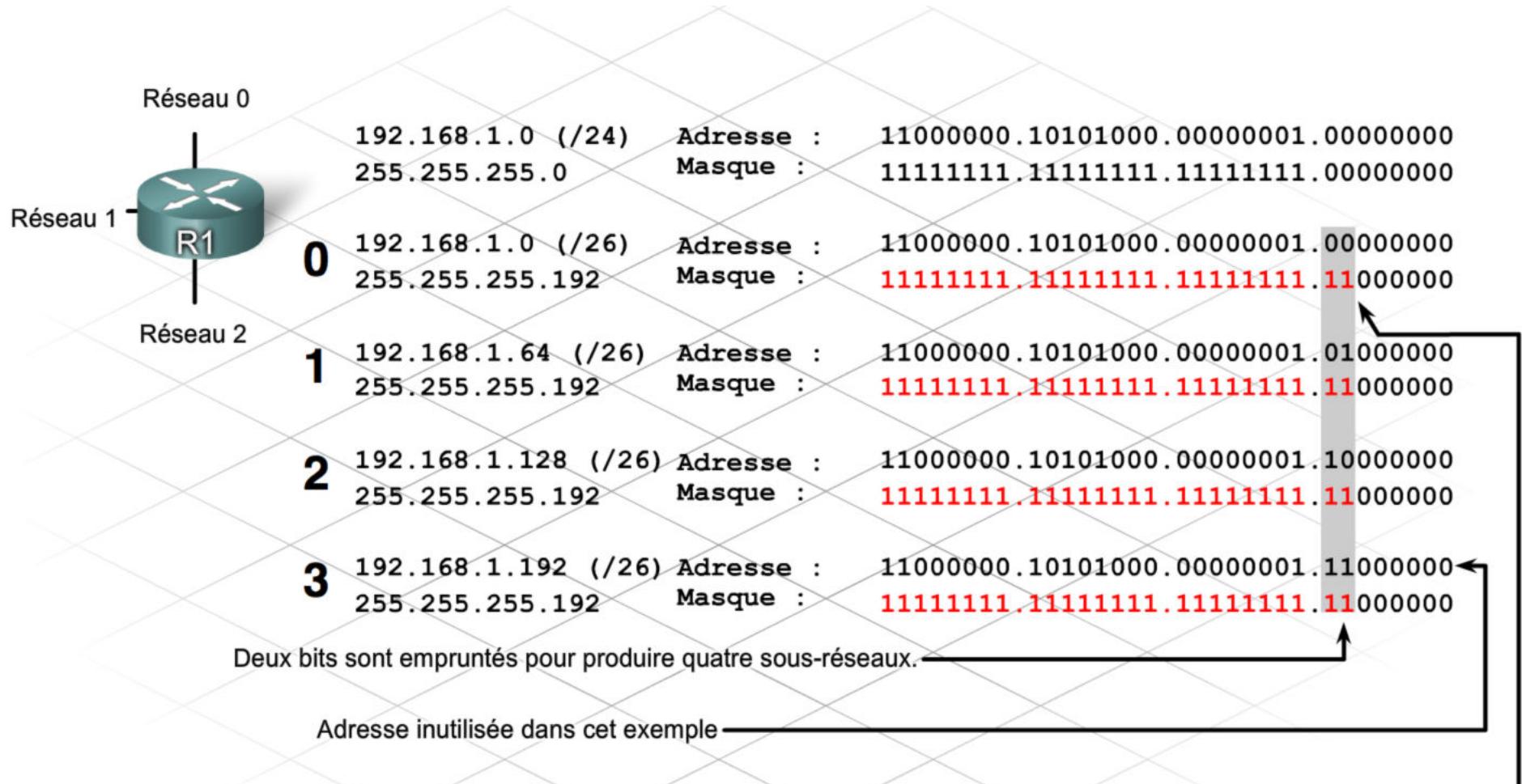
← Augmenter la partie →  
réseau de l'adresse

Emprunter un bit à la  
partie hôte

Schéma d'adressage : exemple de 2 réseaux

| Sous-réseau | Adresse réseau   | Plage d'hôtes                 | Adresse de diffusion |
|-------------|------------------|-------------------------------|----------------------|
| 0           | 192.168.1.0/25   | 192.168.1.1 - 192.168.1.126   | 192.168.1.127        |
| 1           | 192.168.1.128/25 | 192.168.1.129 - 192.168.1.254 | 192.168.1.255        |

# Couche 3 : IP



Lorsque « 1 » apparaît dans ces positions au sein du masque, ces valeurs font partie de l'adresse réseau.

# Couche 3 : IP

Schéma d'adressage : exemple de 4 réseaux

| Sous-réseau | Adresse réseau   | Plage d'hôtes                 | Adresse de diffusion |
|-------------|------------------|-------------------------------|----------------------|
| 0           | 192.168.1.0/26   | 192.168.1.1 – 192.168.1.62    | 192.168.1.63         |
| 1           | 192.168.1.64/26  | 192.168.1.65 – 192.168.1.126  | 192.168.1.127        |
| 2           | 192.168.1.128/26 | 192.168.1.129 – 192.168.1.190 | 192.168.1.191        |
| 3           | 192.168.1.192/26 | 192.168.1.193 – 192.168.1.254 | 192.168.1.255        |

D'autres sous-réseaux sont disponibles, mais un nombre plus limité d'adresses est disponible dans chaque sous-réseau.

# Couche 3 : IP

---

## Exercice : calcul d'adressage

Un ordinateur est connecté à un réseau qui a pour adresse IP

**150.10.32.0 / 19**

1. Quel est le **masque** de sous-réseau ?
2. Quelle est la **première adresse IP** disponible pour les hôtes du réseau ?
3. Quelle est la **dernière adresse IP** disponible pour les hôtes du réseau ?
4. Quelle est l'adresse de **broadcast** ?

# Couche 3 : IP

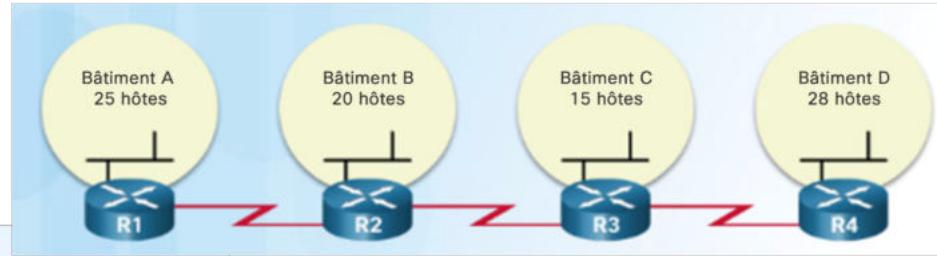
## Vers IPv6

- **Limites du protocole IPv4**
  - Manque d'adresses IP (pénurie notamment due à l'IoT)
  - Croissance de la table de routage Internet
  - Absence de connectivité de bout en bout (cause : @IP privées)
- **Présentation de l'IPv6**
  - Espace d'adressage plus important
  - Amélioration du traitement des paquets
  - Élimination du besoin d'adresses réseau (NAT)
- **Encapsulation IPv6**
  - Format d'en-tête simplifié
  - Processus de somme de contrôle non obligatoire
  - Mécanisme plus efficace d'options d'en-tête
  - Amélioration de l'efficacité par le champ Étiquetage

# Couche 3 : IP

Les avantages des masques de sous-réseau de longueur variable

La segmentation traditionnelle en sous-réseaux entraîne un gaspillage d'adresses



|   | Partie réseau              | Partie hôte |       |
|---|----------------------------|-------------|-------|
|   | 11000000.10101000.00010100 | .000        | 00000 |
| 0 | 11000000.10101000.00010100 | .000        | 00000 |
| 1 | 11000000.10101000.00010100 | .001        | 00000 |
| 2 | 11000000.10101000.00010100 | .010        | 00000 |
| 3 | 11000000.10101000.00010100 | .011        | 00000 |
| 4 | 11000000.10101000.00010100 | .100        | 00000 |
| 5 | 11000000.10101000.00010100 | .101        | 00000 |
| 6 | 11000000.10101000.00010100 | .110        | 00000 |
| 7 | 11000000.10101000.00010100 | .111        | 00000 |

LAN des bâtiments A, B, C et D

WAN site à site

Non utilisé/disponible

Partie sous-réseau  
 $2^3 = 8$  sous-réseaux

Partie hôte  
 $2^5 - 2 = 30$  adresses IP hôtes par sous-réseau

|   | Partie réseau              | Partie hôte | Décimale à point  |
|---|----------------------------|-------------|-------------------|
| 4 | 11000000.10101000.00010100 | .100        | 192.168.20.128/27 |
| 5 | 11000000.10101000.00010100 | .101        | 192.168.20.160/27 |
| 6 | 11000000.10101000.00010100 | .110        | 192.168.20.192/27 |

Partie hôte  
 $2^5 - 2 = 30$  adresses IP hôtes par sous-réseau

$30 - 2 = 28$   
Chaque sous-réseau WAN gaspille 28 adresses

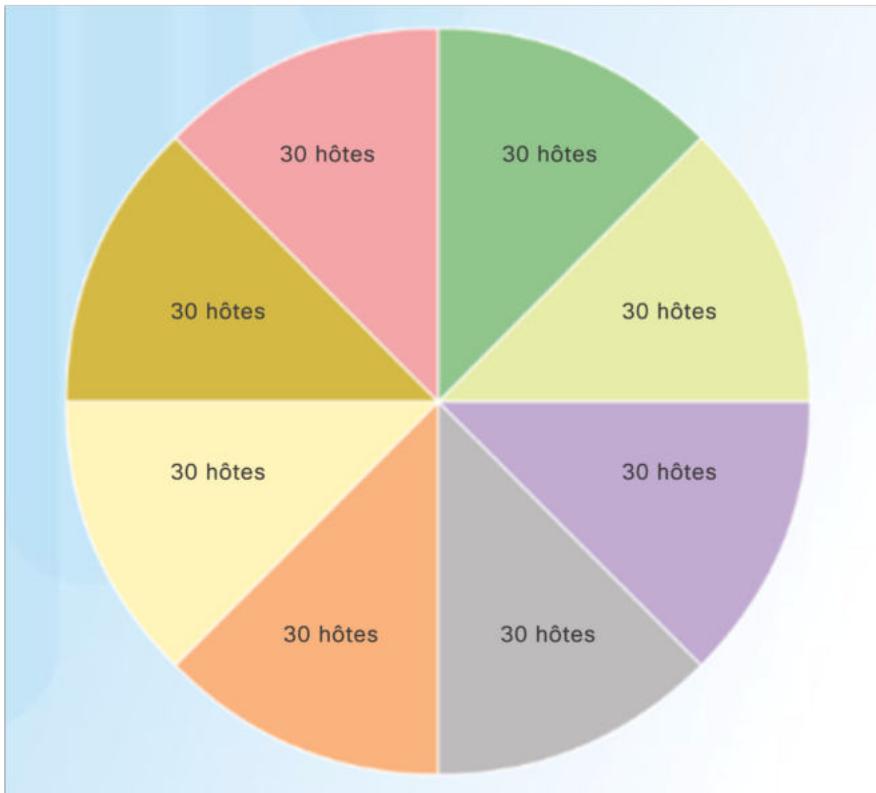
$28 \times 3 = 84$   
84 adresses sont inutilisées

# Couche 3 : IP

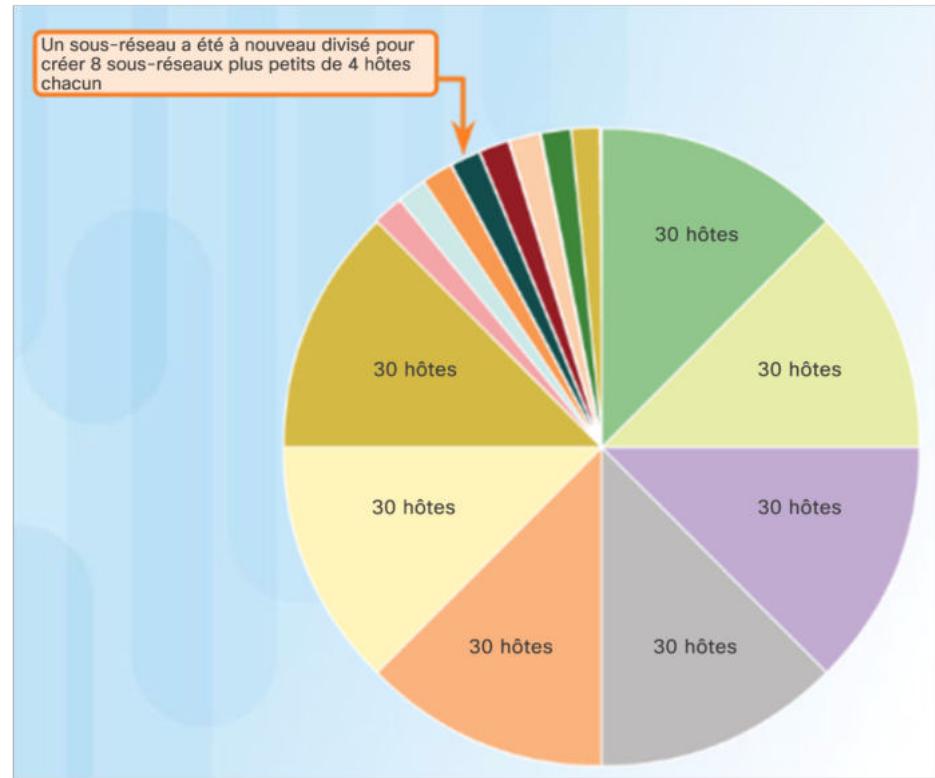
Les avantages des masques de sous-réseau de longueur variable

## Les masques de sous-réseau de longueur variable (VLSM)

Traditionnel



Sous-réseaux de tailles variables



# Couche 3 : IP

Les avantages des masques de sous-réseau de longueur variable

## VLSM de base

### Création de sous-réseaux de base

|   | Partie réseau              | Partie hôte | Décimale à point  |
|---|----------------------------|-------------|-------------------|
|   | 11000000.10101000.00010100 | .00000000   | 192.168.20.0/24   |
| 0 | 11000000.10101000.00010100 | .000        | 192.168.20.0/27   |
| 1 | 11000000.10101000.00010100 | .001        | 192.168.20.32/27  |
| 2 | 11000000.10101000.00010100 | .010        | 192.168.20.64/27  |
| 3 | 11000000.10101000.00010100 | .011        | 192.168.20.96/27  |
| 4 | 11000000.10101000.00010100 | .100        | 192.168.20.128/27 |
| 5 | 11000000.10101000.00010100 | .101        | 192.168.20.160/27 |
| 6 | 11000000.10101000.00010100 | .110        | 192.168.20.192/27 |
| 7 | 11000000.10101000.00010100 | .111        | 192.168.20.224/27 |

Le sous-réseau 7 est à nouveau segmenté en sous-réseaux.

|     | Partie réseau                            | Partie hôte | Décimale à point  |
|-----|--|-------------|-------------------|
| 7   | 11000000.10101000.00010100               | .111        | 192.168.20.224/27 |
|     | 3 autres bits empruntés au sous-réseau 7 |             |                   |
| 7:0 | 11000000.10101000.00010100               | .111000     | 192.168.20.224/30 |
| 7:1 | 11000000.10101000.00010100               | .111001     | 192.168.20.228/30 |
| 7:2 | 11000000.10101000.00010100               | .111010     | 192.168.20.232/30 |
| 7:3 | 11000000.10101000.00010100               | .111011     | 192.168.20.236/30 |
| 7:4 | 11000000.10101000.00010100               | .111100     | 192.168.20.240/30 |
| 7:5 | 11000000.10101000.00010100               | .111101     | 192.168.20.244/30 |
| 7:6 | 11000000.10101000.00010100               | .111110     | 192.168.20.248/30 |
| 7:7 | 11000000.10101000.00010100               | .111111     | 192.168.20.252/30 |

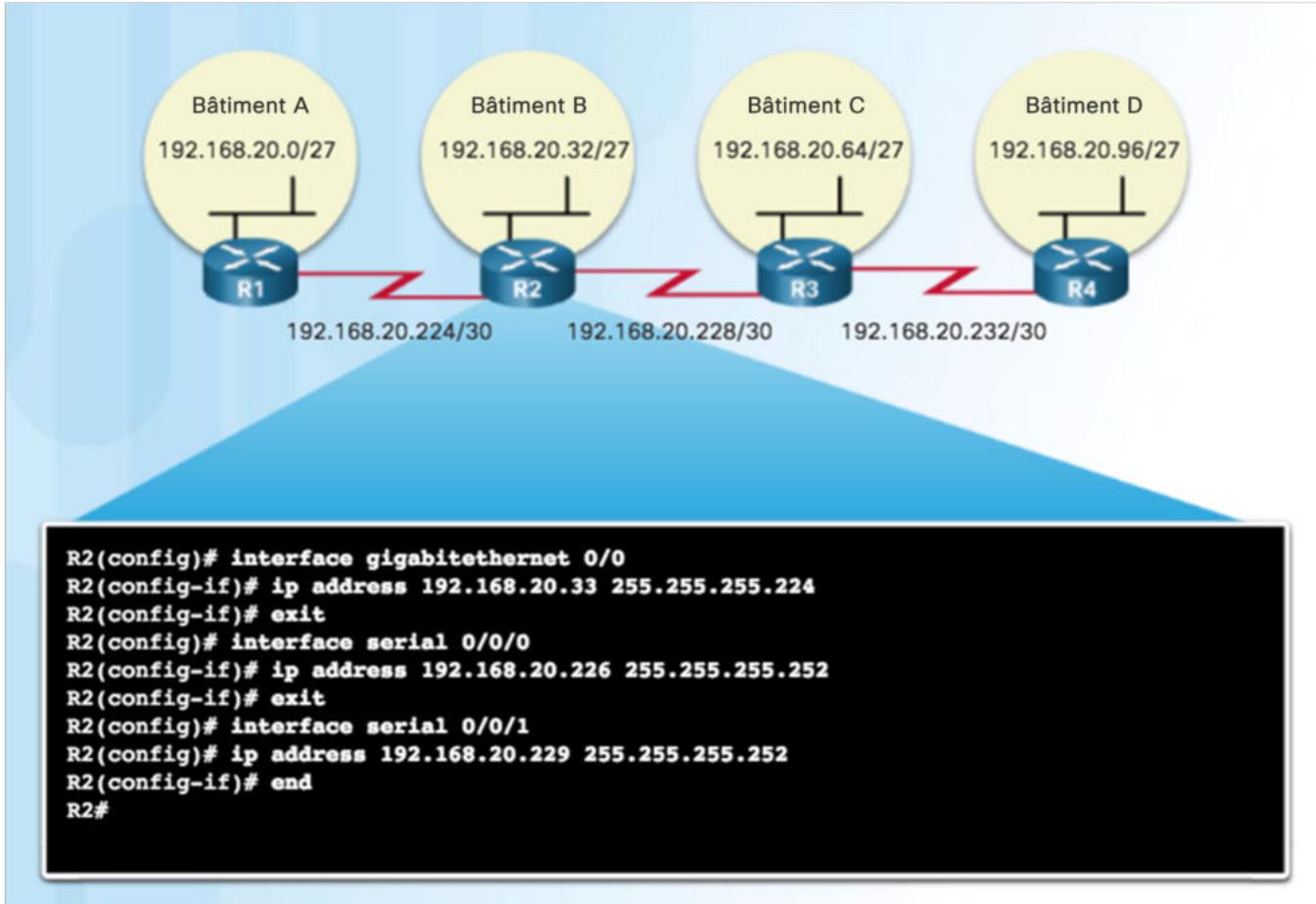
Segmentation d'un sous-réseau

Réseaux locaux  
A, B, C, D

Non utilisé/  
disponible

# Couche 3 : IP

Les avantages des masques de sous-réseau de longueur variable  
Le VLSM dans la pratique



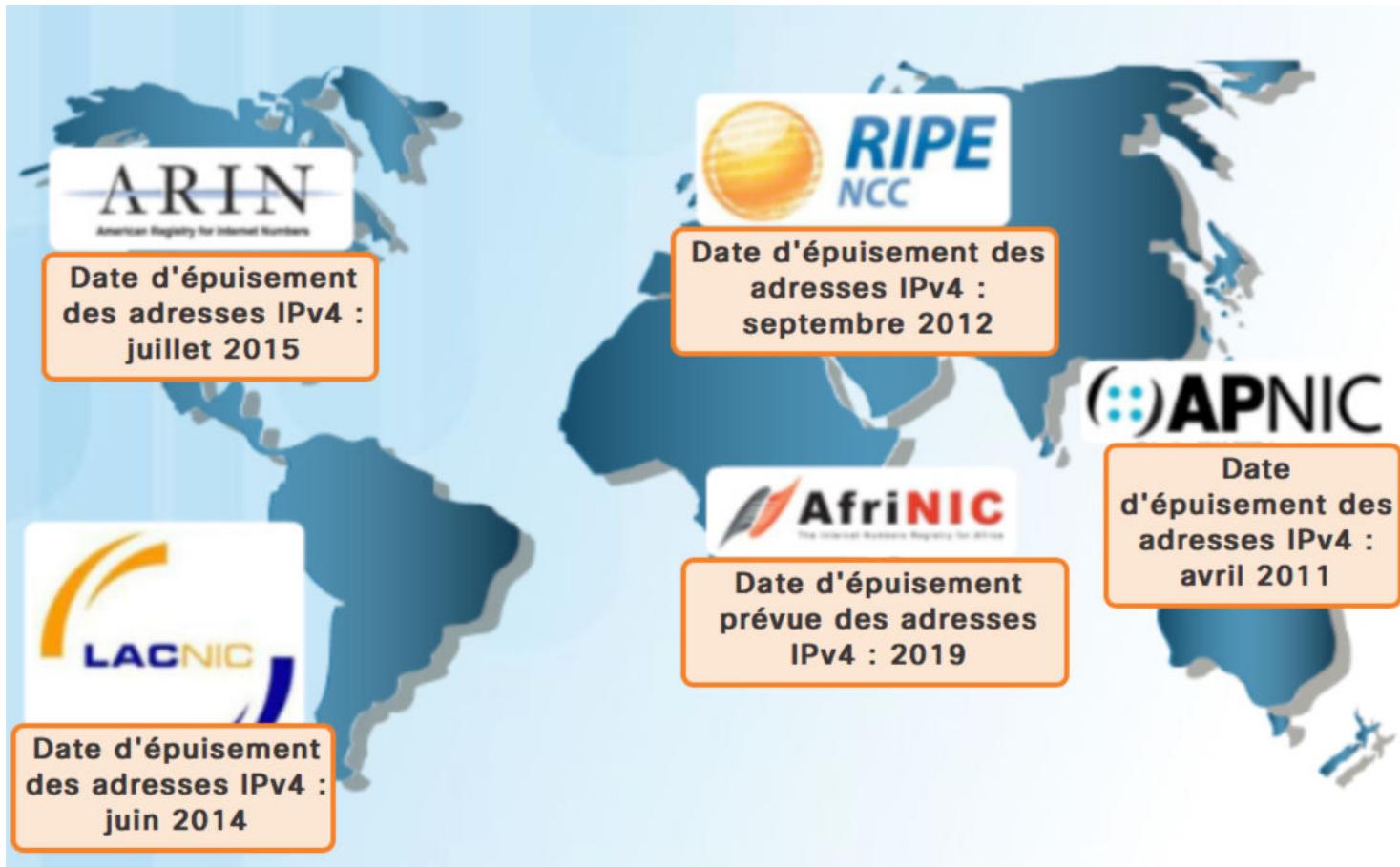
# Couche 3 : IP

Les avantages des masques de sous-réseau de longueur variable  
Diagramme VLSM

| Segmentation en sous-réseaux VLSM de 192.168.20.0/24 |            |             |
|--|------------|-------------|
|  | Réseau /27 | Hôtes       |
| Bât. A   | .0         | .1 - .30    |
| Bât. B   | .32        | .33 - .62   |
| Bât. C   | .64        | .65 - .94   |
| Bât. D   | .96        | .97 - .126  |
| Non utilisé  | .128       | .129 - .158 |
| Non utilisé  | .160       | .161 - .190 |
| Non utilisé  | .192       | .193 - .222 |
|  | .224       | .225 - .254 |
|  |            |             |
|  |            |             |
|  |            |             |
|  |            |             |
|  | Réseau /30 | Hôtes       |
| WAN R1-R2  | .224       | .225 - .226 |
| WAN R2-R3  | .228       | .229 - .230 |
| WAN R3-R4  | .232       | .233 - .234 |
| Non utilisé  | .236       | .237 - .238 |
| Non utilisé  | .240       | .241 - .242 |
| Non utilisé  | .244       | .245 - .246 |
| Non utilisé  | .248       | .249 - .250 |
| Non utilisé  | .252       | .253 - .254 |

# Couche 3 : IP

## Vers IPV6 : date d'épuisement des adresses IPV4



## Couche 3 : IP

# IPV6 : combien d'IP disponibles ?

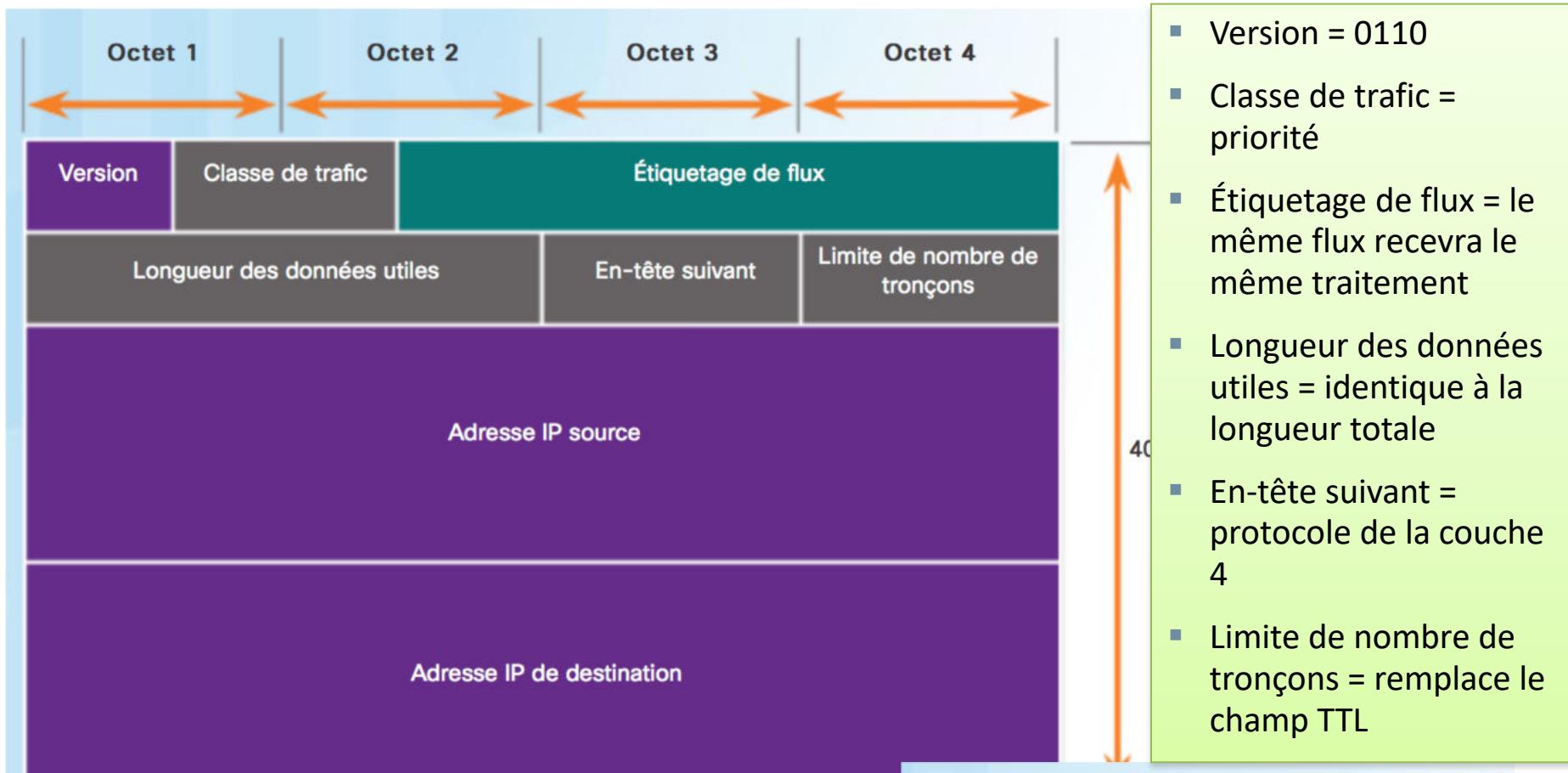
## Légende

- Il existe 4 milliards d'adresses IPv4
  - Il existe 340 undécillions d'adresses IPv6

- **IPv4** de 32 bits fournit environ 4 294 967 296 adresses uniques
  - **IPv6** fournit 340 282 366 920 938 463 463 374 607 431 768 211 456 adresses, soit 340 undécillions d'adresses, ce qui correspond à peu près au nombre de grains de sable sur Terre ! ;-)

# Couche 3 : IP

## IPV6 : simplification de l'en-tête



### Légende

- Noms des champs conservés de IPv4 à IPv6
- Nom et position modifiés dans IPv6
- Champs non conservés dans IPv6

# Couche 3 : IP

IPV6 : capture de trame avec wireshark

Cas d'une trame http

The screenshot shows a Wireshark capture window with several frames listed in the packet list pane. Frame 49 is selected, which is an IPv6 packet for a GET request. The details pane shows the following information:

- Frame 49: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits)
- Ethernet II, Src: HsingTec\_e3:e8:de (00:d0:09:e3:e8:de), Dst: IBM\_82:95:b5 (00:11:25:82:95:b5)
- Internet Protocol version 6, Src: 2001:6f8:102d:0:2d0:9ff:fee3:e8de (2001:6f8:102d:0:2d0:9ff:fee3:e8de), Dst: 2001:6f8:900:7c0::2
- Version: 6
- Traffic class: 0x00000000
- Flowlabel: 0x00000000
- Payload length: 260
- Next header: TCP (6)
- Hop limit: 64
- Source: 2001:6f8:102d:0:2d0:9ff:fee3:e8de (2001:6f8:102d:0:2d0:9ff:fee3:e8de)  
[Source SA MAC: HsingTec\_e3:e8:de (00:d0:09:e3:e8:de)]
- Destination: 2001:6f8:900:7c0::2 (2001:6f8:900:7c0::2)  
[Source GeoIP: Unknown]  
[Destination GeoIP: Unknown]
- Transmission Control Protocol, Src Port: 59201 (59201), Dst Port: http (80), Seq: 1, Ack: 1, Len: 240
- Hypertext Transfer Protocol

The bytes pane at the bottom shows the raw hex and ASCII data for the selected frame.

# Couche 3 : IP

IPV6 : capture de trame avec wireshark

Cas d'une trame icmp

The screenshot shows a Wireshark capture window with the following details:

- Table Headers:** No., Time, Source, Destination, Protocol, Length, Info.
- Packets:** There are 55 total packets displayed, with 55 selected.
- Selected Packet (Frame 6):**
  - Source:** fe80::211:25ff:fe82:95b5
  - Destination:** ff02::1:ff82:95b5
  - Protocol:** ICMPv6
  - Length:** 86
  - Info:** Neighbor Solicitation for 2001:6f8:102d:0:1033:c4c:7e'ff02::fb
- Selected Packet (Frame 6) Details:**
  - Ethernet II:** Src: Ibm\_82:95:b5 (00:11:25:82:95:b5), Dst: IPv6mcast\_ff:82:95:b5 (33:33:ff:82:95:b5)
  - Internet Protocol Version 6:** Src: fe80::211:25ff:fe82:95b5 (fe80::211:25ff:fe82:95b5), Dst: ff02::1:ff82:95b5 (ff02::1)
  - ICMPv6 Fields:**
    - Version: 6
    - Traffic class: 0x00000000
    - Flowlabel: 0x00000000
    - Payload length: 32
    - Next header: ICMPv6 (58)
    - Hop limit: 255
    - Source: fe80::211:25ff:fe82:95b5 (fe80::211:25ff:fe82:95b5)  
[Source MAC: Ibm\_82:95:b5 (00:11:25:82:95:b5)]
    - Destination: ff02::1:ff82:95b5 (ff02::1:ff82:95b5)  
[Source GeoIP: Unknown]  
[Destination GeoIP: Unknown]
  - Internet Control Message Protocol v6**
- Hex and ASCII panes:** Show the raw bytes and their corresponding ASCII characters for the selected packet.
- Status Bar:** Internet Protocol Version 6 (ipv6), 40 bytes | Packets: 55 Displayed: 55 Mark... | Profile: Default

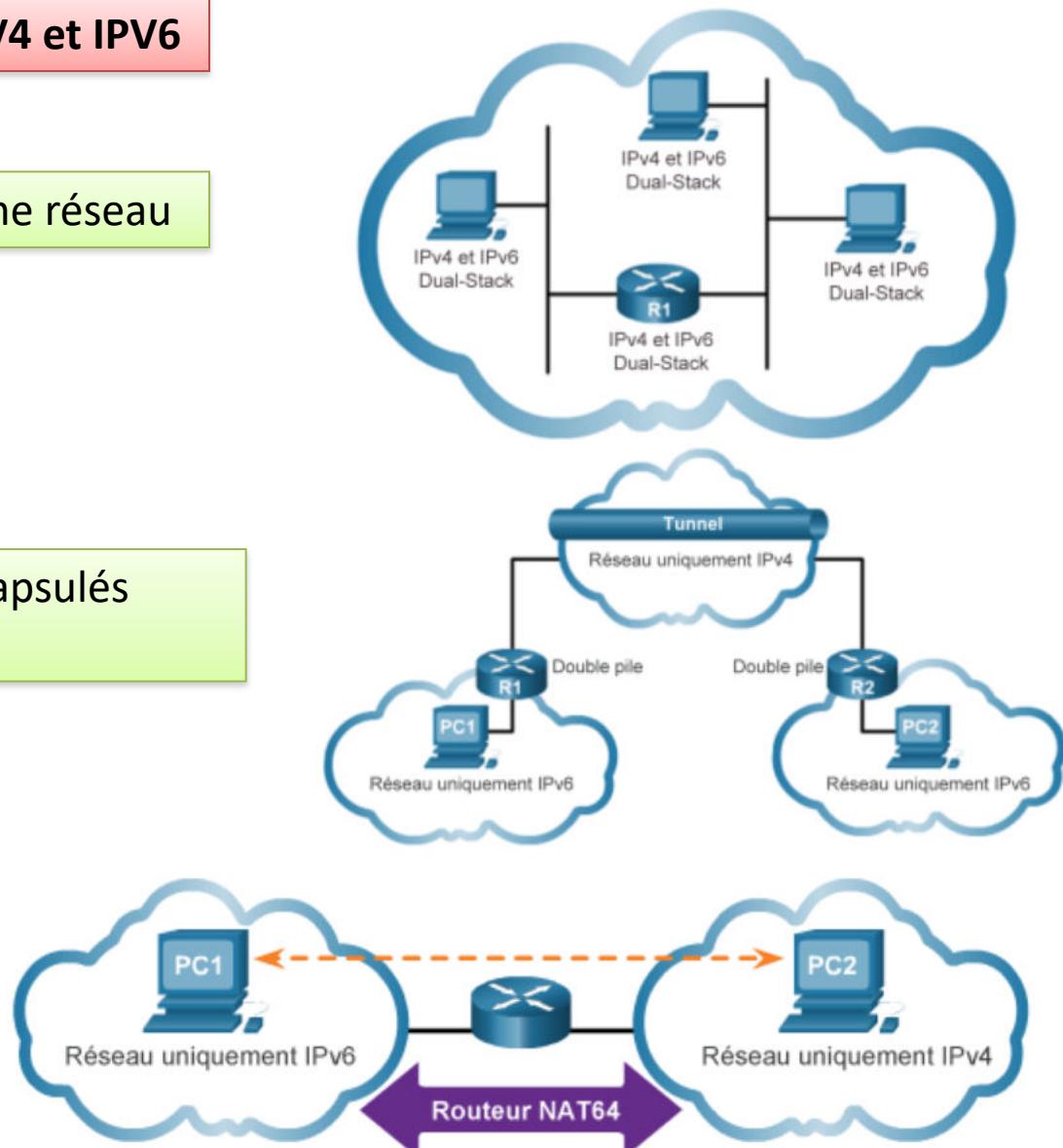
# Couche 3 : IP

IPV6 : coexistence des protocoles IPV4 et IPV6

**Double pile** : IPV4 et IPV6 sur un même réseau

**Tunnelisation** : des paquets IPV6 encapsulés dans des paquets IPV4

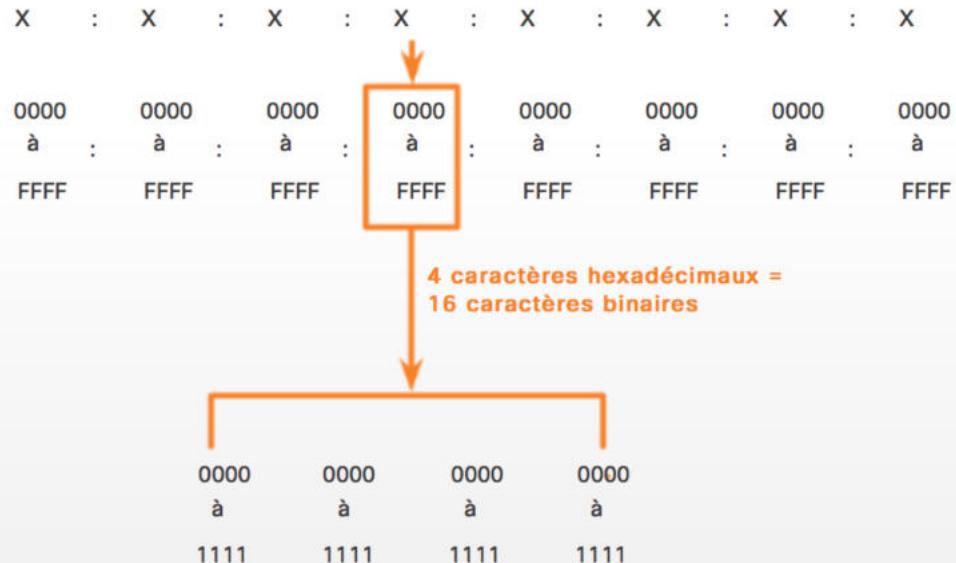
**Traduction** : un paquet IPV6 est traduit en un paquet IPV4 et inversement



# Couche 3 : IP

## IPV6 : représentation de l'adresse (hextet)

Format de l'adresse :  
128 bits  
8 hextets  
Format hexadécimal



|      |   |      |   |      |   |      |   |      |   |      |   |      |   |      |
|------|---|------|---|------|---|------|---|------|---|------|---|------|---|------|
| 2001 | : | 0DB8 | : | 0000 | : | 1111 | : | 0000 | : | 0000 | : | 0000 | : | 0200 |
| 2001 | : | 0DB8 | : | 0000 | : | 00A3 | : | ABCD | : | 0000 | : | 0000 | : | 1234 |
| 2001 | : | 0DB8 | : | 000A | : | 0001 | : | 0000 | : | 0000 | : | 0000 | : | 0100 |
| 2001 | : | 0DB8 | : | AAAA | : | 0001 | : | 0000 | : | 0000 | : | 0000 | : | 0200 |
| FE80 | : | 0000 | : | 0000 | : | 0000 | : | 0123 | : | 4567 | : | 89AB | : | CDEF |
| FE80 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0001 |
| FF02 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0001 |
| FF02 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0001 | : | FF00 | : | 0200 |
| 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0001 |
| 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 | : | 0000 |

Quelques exemples... pas simple à retenir d'où 2 solutions de simplification

# Couche 3 : IP

## IPV6 : simplification

Règle 1 : omettre les 0 en **début** de segment

|                                |  |
|--------------------------------|--|
| Recommandé                     | 2 0 0 1 : 0 DB 8 : 0 0 0 0 : 1 1 1 1 : 0 0 0 0 : 0 0 0 0 : 0 0 0 0 : 0 2 0 0 |
| Sans zéros en début de segment | 2 0 0 1 : DB 8 : 0 : 1 1 1 1 : 0 : 0 : 0 : 2 0 0                             |

Règle 2 : omettre les segments uniquement composés de 0

|                                |  |
|--------------------------------|--|
| Recommandé                     | 2 0 0 1 : 0 DB 8 : 0 0 0 0 : 1 1 1 1 : 0 0 0 0 : 0 0 0 0 : 0 0 0 0 : 0 2 0 0 |
| Sans zéros en début de segment | 2 0 0 1 : DB 8 : 0 : 1 1 1 1 : 0 : 0 : 0 : 2 0 0                             |
| Compressé                      | 2 0 0 1 : DB 8 : 0 : 1 1 1 1 :: 2 0 0  |

Attention ! Les « : » ne peuvent être utilisés qu'une seule fois par adresse  
Ainsi : 2001:0DB8::ABCD::1234  
n'est pas une adresse IPV6 valide !

# Couche 3 : IP

## IPV6 : simplification

Règle 2 (suite) : omettre les segments uniquement composés de 0

|                                |   |
|--------------------------------|---|
| Recommandé                     | 2 0 0 1 : 0 D B 8 : 0 0 0 0 : 0 0 0 0 : A B C D : 0 0 0 0 : 0 0 0 0 : 0 1 0 0 |
| Sans zéros en début de segment | 2 0 0 1 : D B 8 : 0 : 0 : A B C D : 0 : 0 : 1 0 0                             |
| Compressé                      | 2 0 0 1 : D B 8 :: A B C D : 0 : 0 : 1 0 0                                    |
| ou                             |   |
| Compressé                      | 2 0 0 1 : D B 8 : 0 : 0 : A B C D :: 1 0 0                                    |

Diagramme d'explication : deux flèches orange pointent vers les deux occurrences de '::'. Une boîte orange contenant la phrase ':: peut être utilisé une seule fois.' est placée sous la seconde flèche.

|                                |   |
|--------------------------------|---|
| Recommandé                     | F F 0 2 : 0 0 0 0 : 0 0 0 0 : 0 0 0 0 : 0 0 0 0 : 0 0 0 1 : F F 0 0 : 0 2 0 0 |
| Sans zéros en début de segment | F F 0 2 : 0 : 0 : 0 : 0 : 1 : F F 0 0 : 2 0 0                                 |
| Compressé                      | F F 0 2 :: 1 : F F 0 0 : 2 0 0  |

# Couche 3 : IP

## IPV6 : type d'adresse

### ➤ **Monodiffusion :**

- Adresses uniques routables sur Internet
- Configurées de manière statique ou attribuées dynamiquement

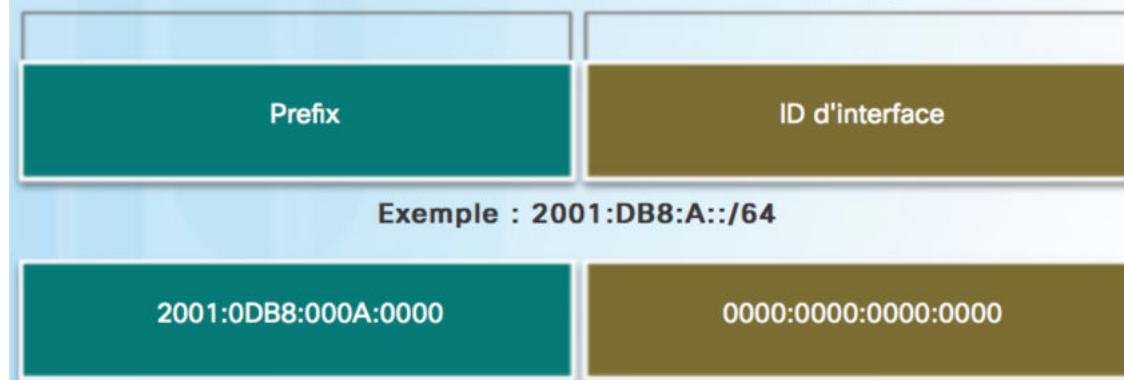
### ➤ **Multidiffusion :**

- Plusieurs destinations

### ➤ **Anycast** : cas particulier

- adresse de monodiffusion IPv6 peut être attribuée à plusieurs périphériques. Un paquet envoyé à une adresse anycast est acheminé vers le périphérique le plus proche ayant cette adresse.

## IPV6 : préfixe /64

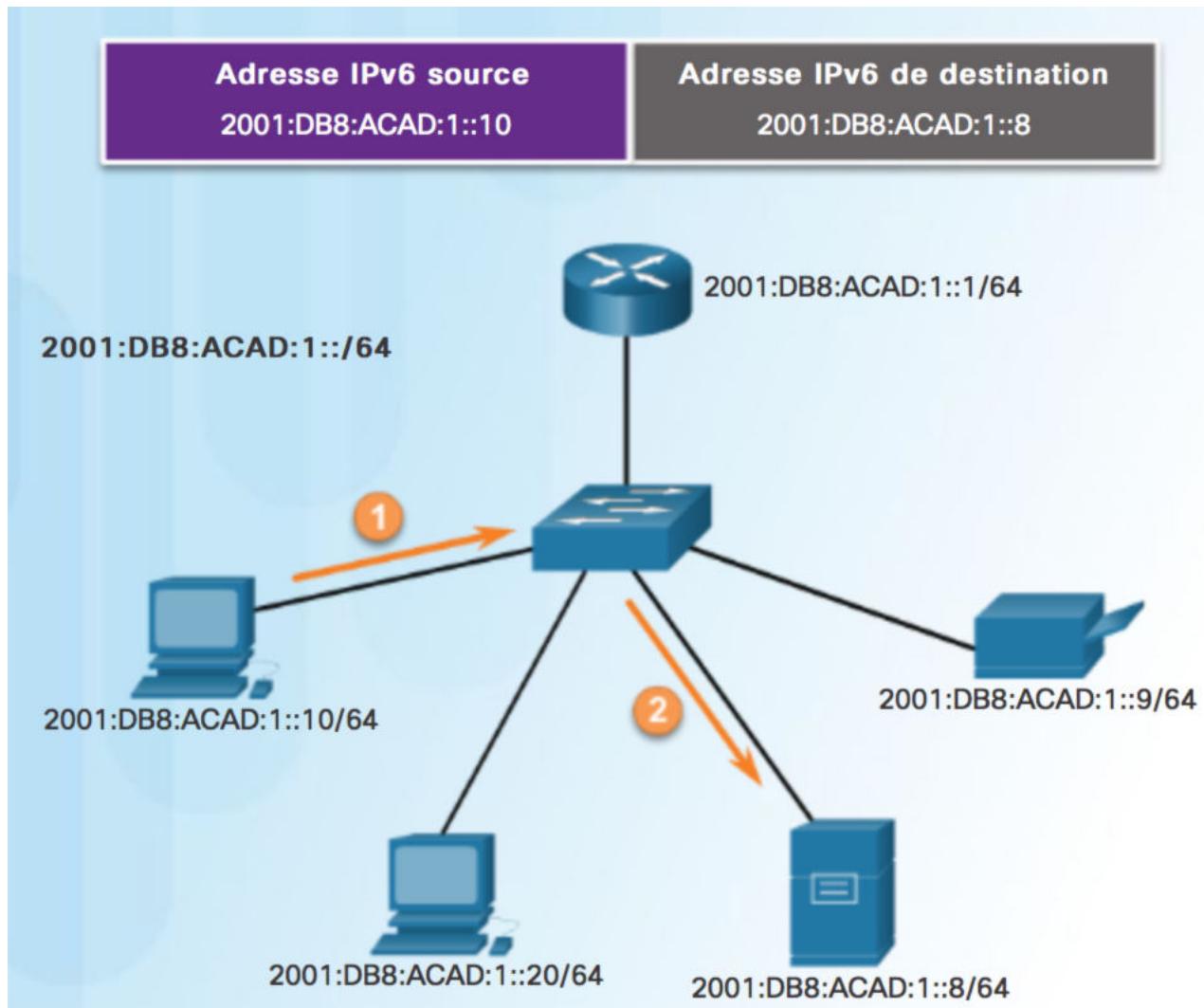


### Longueur de préfixe IPv6

- Indique la partie réseau
- Format : adresse IPv6 /longueur de préfixe
- La longueur de préfixe est comprise entre 0 et 128
- La longueur est généralement de /64

# Couche 3 : IP

## IPv6 : Exemple d'adressage pour un petit réseau



# Couche 3 : IP

## IPV6 : construction du préfixe de routage

Préfixe de routage : préfixe réseau et préfixe sous-réseau

Exemple : **2001:0820:9511::/48** correspond à un sous-réseau avec

- 1<sup>ère</sup> IP : **2001:0820:9511:0000:0000:0000:0000**
- Dernière IP : **2001:0820:9511:FFFF:FFFF:FFFF:FFFF**.

Construction classique :

| Préfixe de routage                              | Identifiant d'interface (Interface ID)   |
|---|--|
| <b>2001:0620:0000</b>                           | <b>:0000</b>   |
| Préfixe de réseau /<br>Topologie publique       | Préfixe sous-réseau /<br>Topologie du site   |
| 48 bits   | 16 bits  |
| 64 bits   | 64 bits  |
| Le préfixe caractérise le réseau ou sous-réseau | L'ID d'interface caractérise un appareil donné<br>avec une carte réseau au sein d'un réseau. |

# Couche 3 : IP

## IPV6 : construction de l'identifiant d'interface

L'ID d'interface permet l'identification claire d'un appareil donné connecté au réseau : il est généré soit manuellement ou sur la base de l'adresse MAC comme montré ci-dessous

Adresse MAC (48 bits) découpée en 2 parties longues de 24 bits (1 exemple) : ces parties constituent le début et la fin des 64 bits de l'identifiant d'interface complet

**Adresse MAC :** **00-11-24-80-C1-2C**

**Adresse MAC découpée :** **0011:24\_\_:\_80:C12C**

Deuxièmement, les 16 bits restants sont alloués au milieu par défaut avec la suite

**1111 1111 1111 1110** qui correspond au code hexadécimal **FFFE**.

**Adresse MAC complète :** **0011:24FF:FE80:C12C**

L'adresse MAC est maintenant au **format EUI-64 modifié**.

Enfin, le septième bit, appelé également bit universel ou local, est inversé. Cela indique si une adresse est unique globale ou locale.

**Suite avant l'inversion :** 0000 0000

**Suite après l'inversion :** 0000 0010

**ID d'interface avant l'inversion :** **0011:24FF:FE80:C12C**

**ID d'interface après l'inversion :** **0211:24FF:FE80:C12C**

# Panorama des protocoles internet

I. Modèles TCP/IP – principe de l'encapsulation

II. Modèles TCP/IP -- couche 3 (réseau)

- a. rôle
- b. adressage
- c. introduction au routage

II. Modèles TCP/UDP – couche 4 (transport)

- b. rôle et fonctionnement
- c. TCP et UDP
- d. NAT et PAT

IV. Modèles TCP/IP – couche 5 (application)

- a. rôle et fonctionnement
- b. DNS et DHCP
- c. Services et protocoles (HTTP, FTP, Telnet,...)

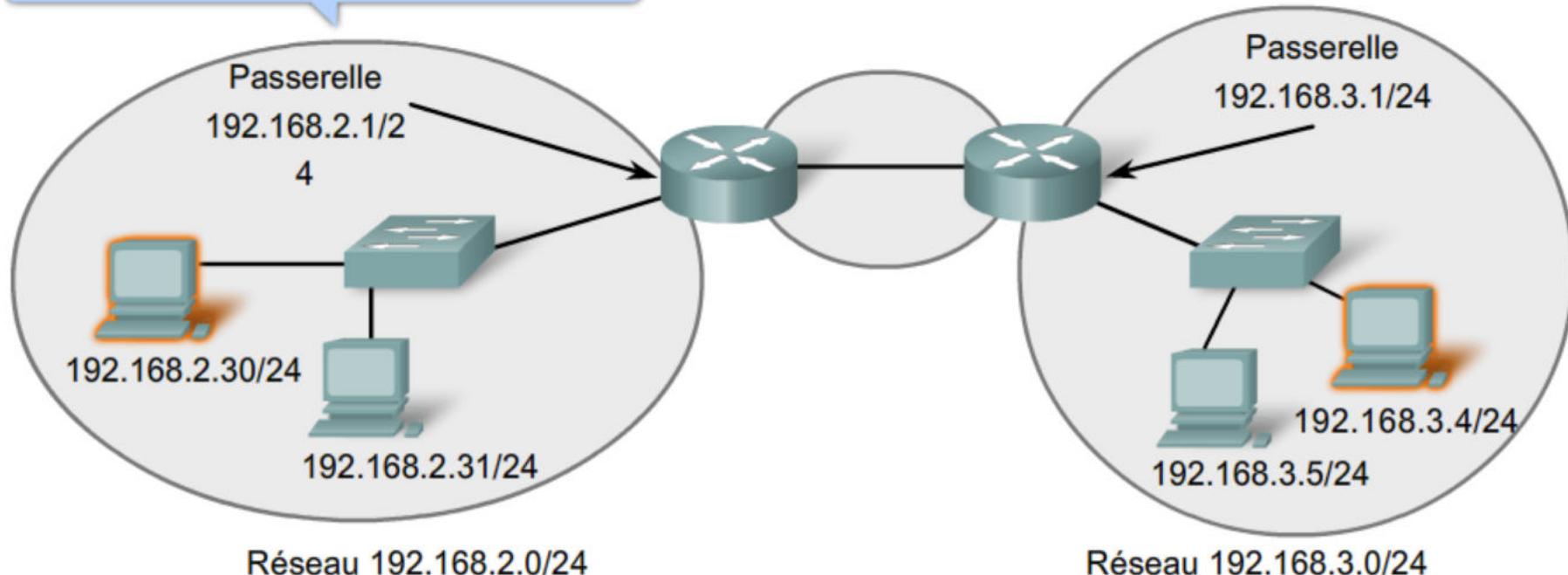
# Couche 3 : IP

## Rôle de la passerelle

Les passerelles permettent les communications entre réseaux

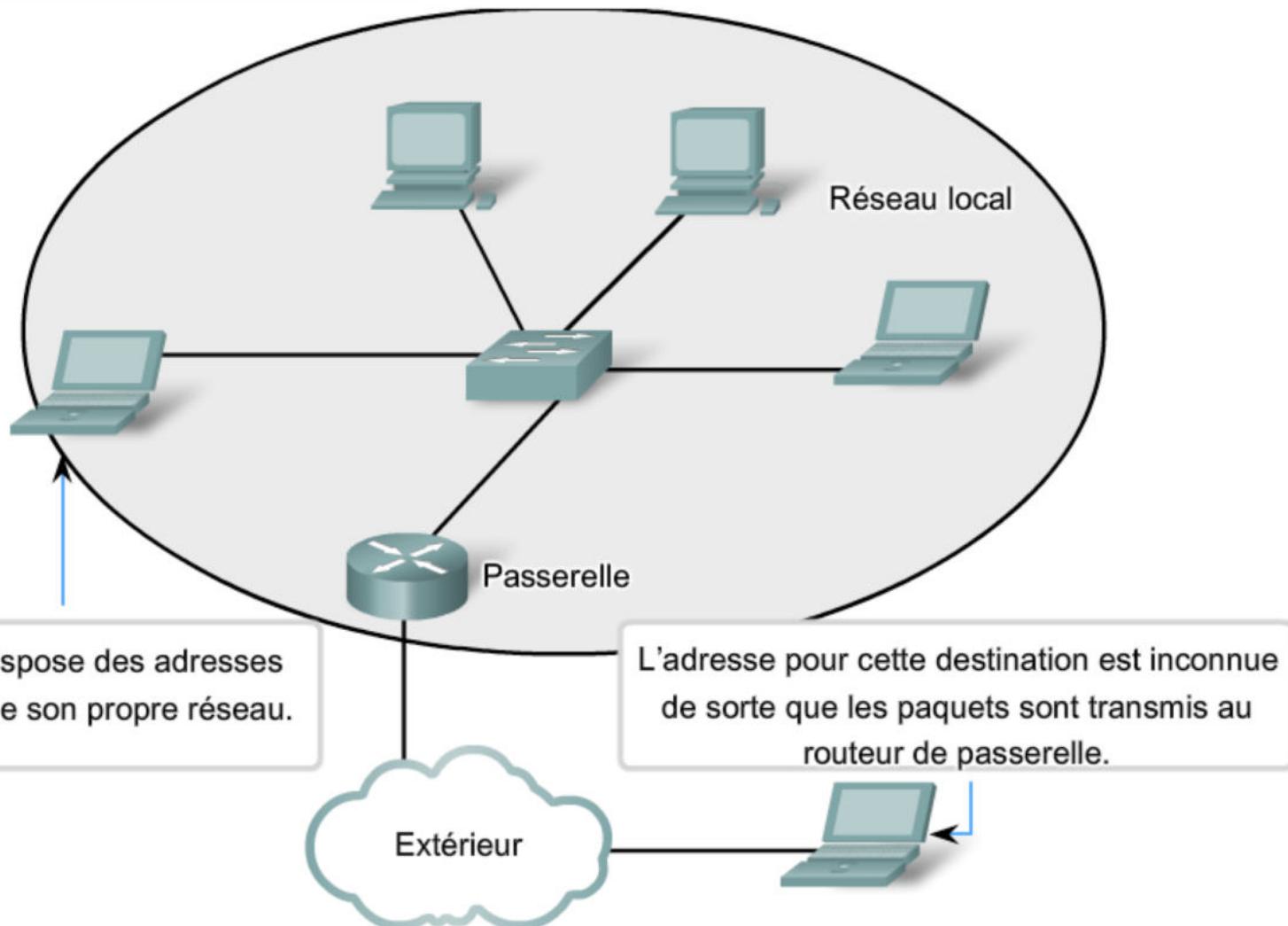
Je connais uniquement les adresses des périphériques de mon réseau.

Si je ne connais pas l'adresse du périphérique de destination, j'envoie le paquet à l'adresse de passerelle par défaut.



# Couche 3 : IP

## Rôle de la passerelle (cas de la salle TD)

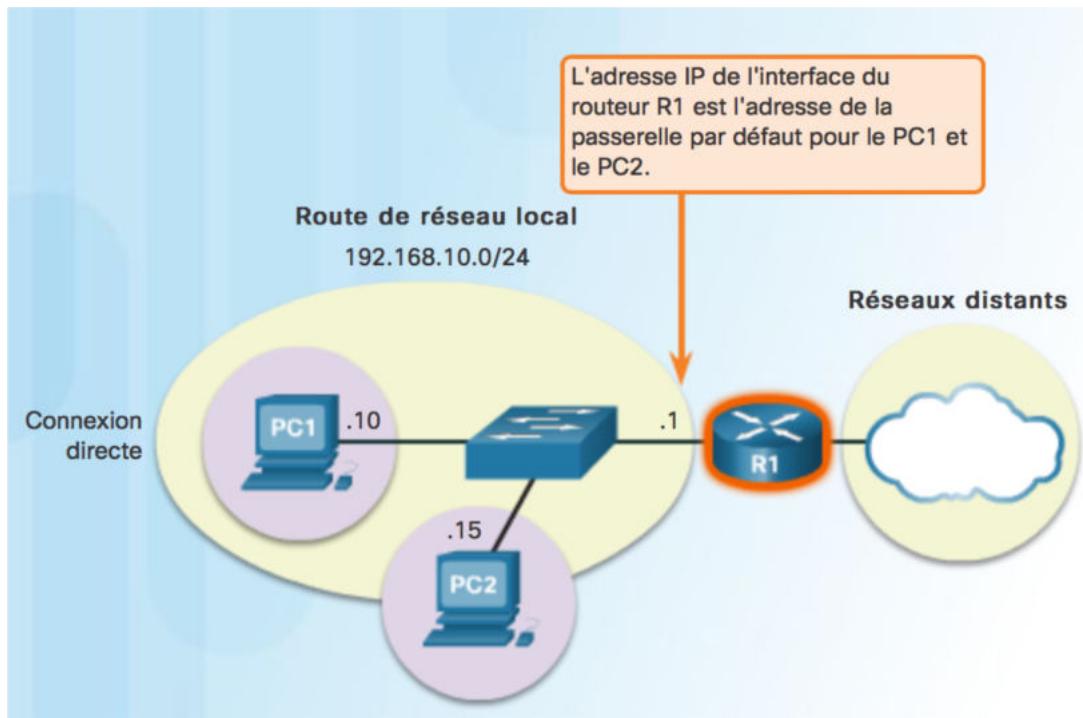


# Couche 3 : IP

## Rôle de la passerelle

En résumé, une passerelle par défaut...

- Achemine le trafic vers d'autres réseaux
- Possède une adresse IP locale située dans la même plage d'adresses que les hôtes du réseau
- Peut recevoir des données et en transmettre



# Couche 3 : IP

## Notions de base sur le routage

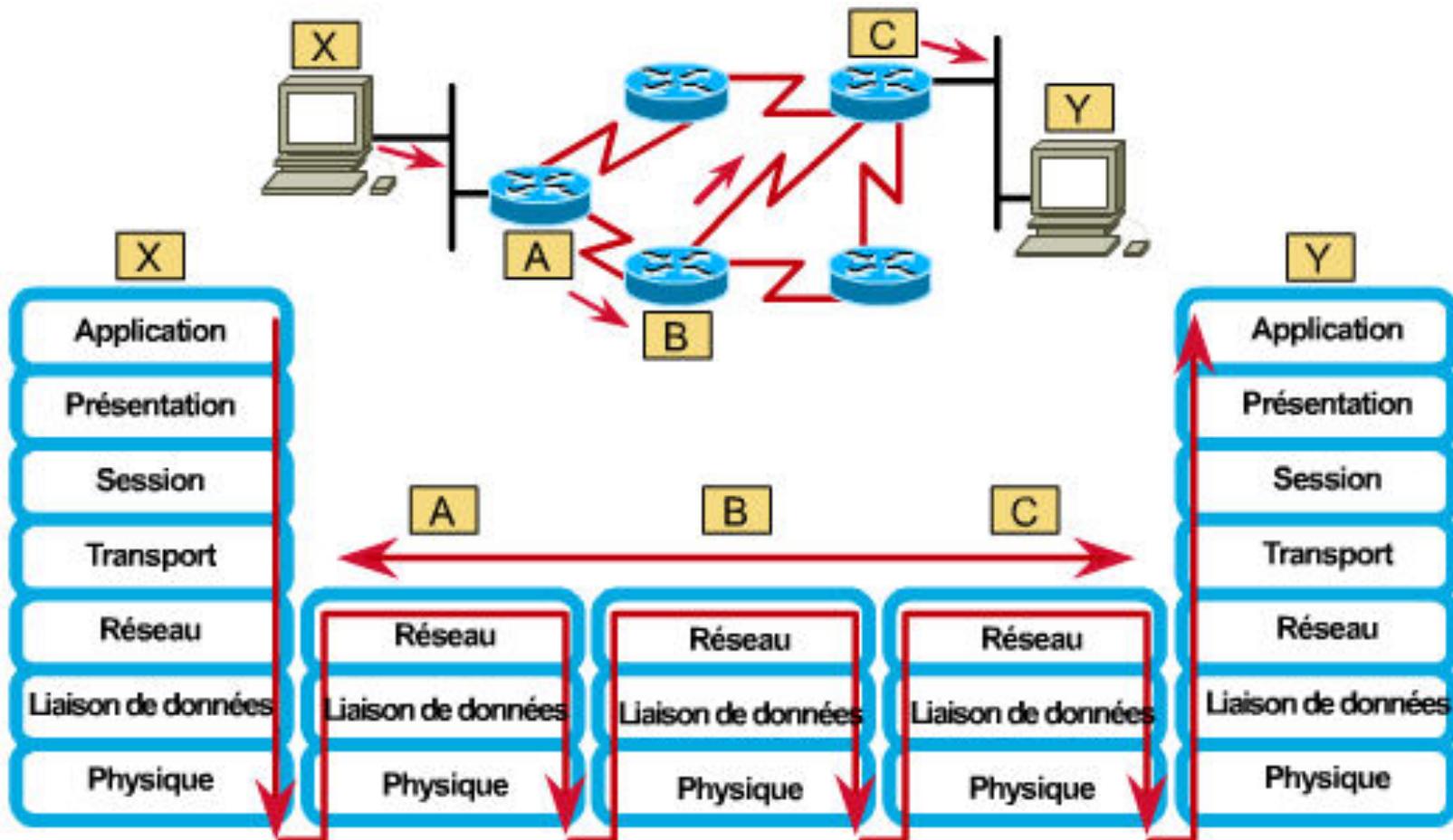
Routeur :

- pour interconnecter des réseaux entre eux
- ***table de routage***
- le routeur dispose de plusieurs ***interfaces***, chacune appartient à un réseau IP différent
- Détermination de la ***route*** vers le destinataire à l'aide de sa table
- Route ***statique*** : entrée par l'administrateur réseau
- Route ***dynamique*** gérée par un protocole de routage (RIP, OSPF, EIGRP,...)

Une route possède quatre composants principaux :

- le réseau de destination ;
- le masque de sous-réseau ;
- l'adresse de passerelle ou d'interface ;
- le coût de la route ou la mesure.

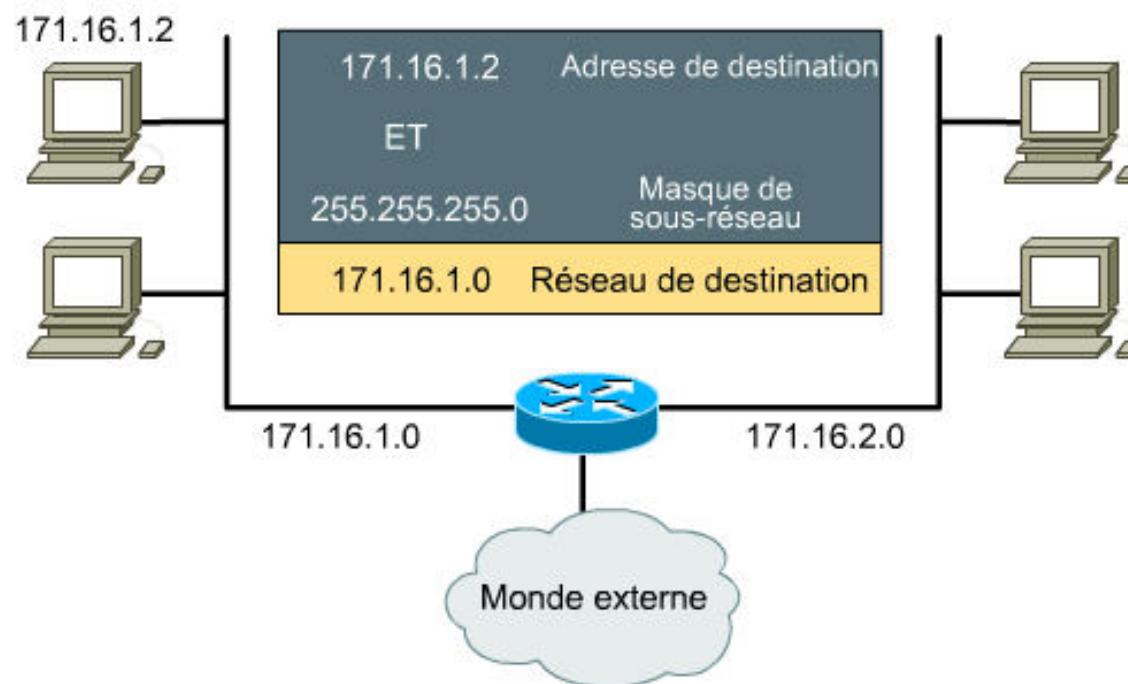
# Couche 3 : IP



# Couche 3 : IP

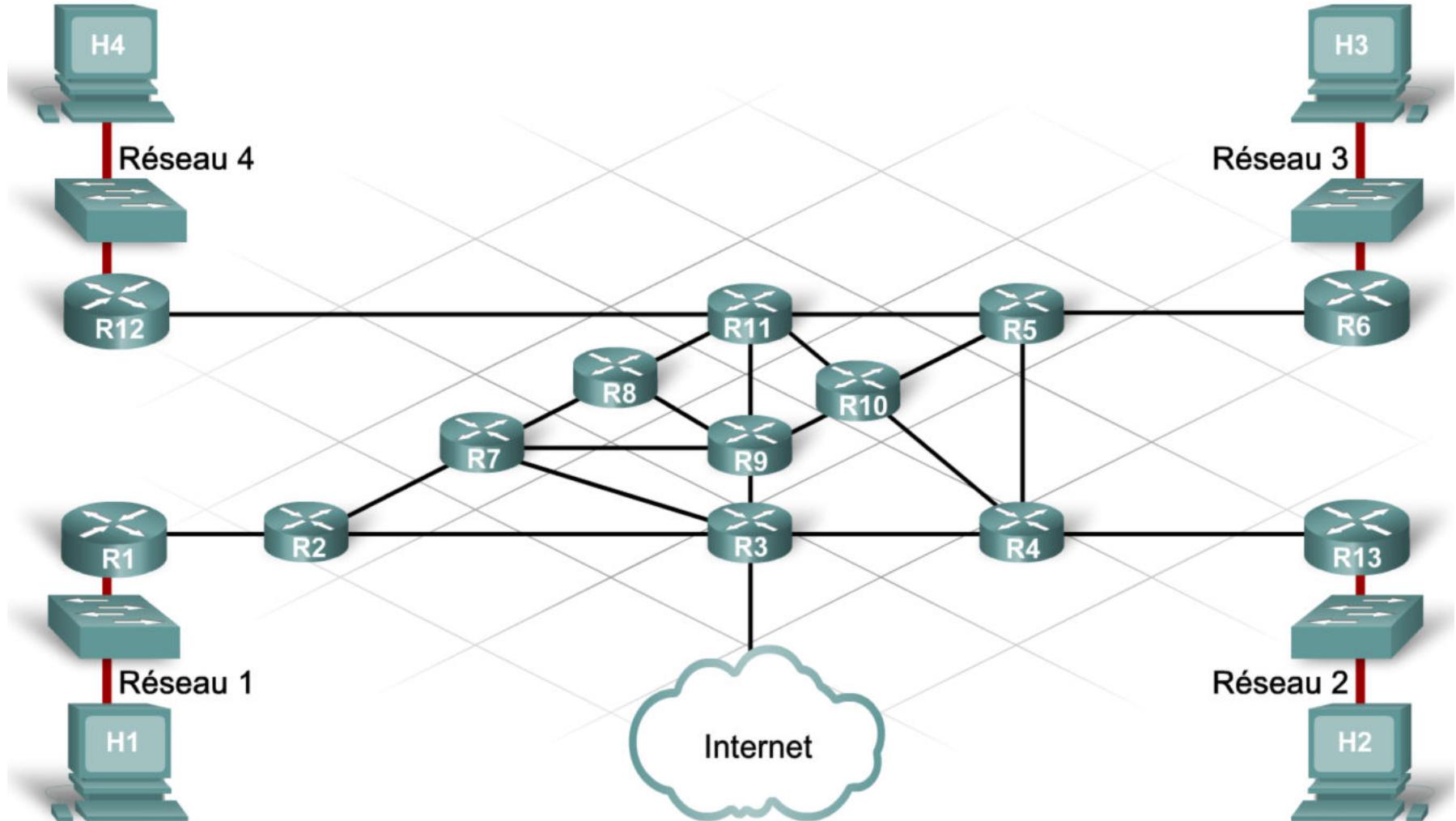
## Détermination d'une adresse réseau :

- extraction de l'**adresse de destination IP** du paquet entrant par le routeur
- récupération du **masque** de réseau interne
- **opération ET logique** pour en déduire le numéro de réseau
- recherche du **numéro de réseau de destination** et de l'**interface** correspondante dans la table de routage

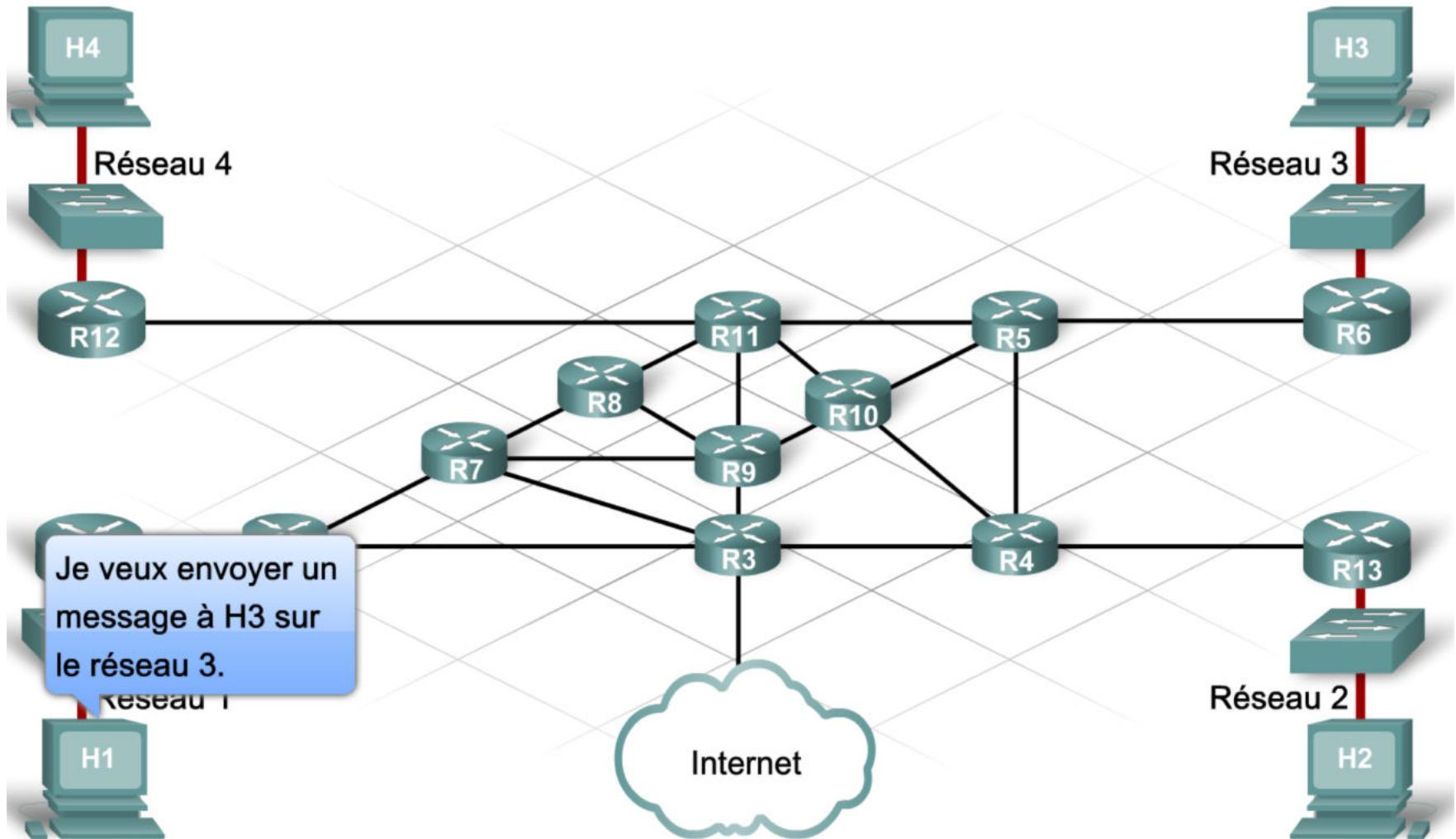


# Couche 3 : IP

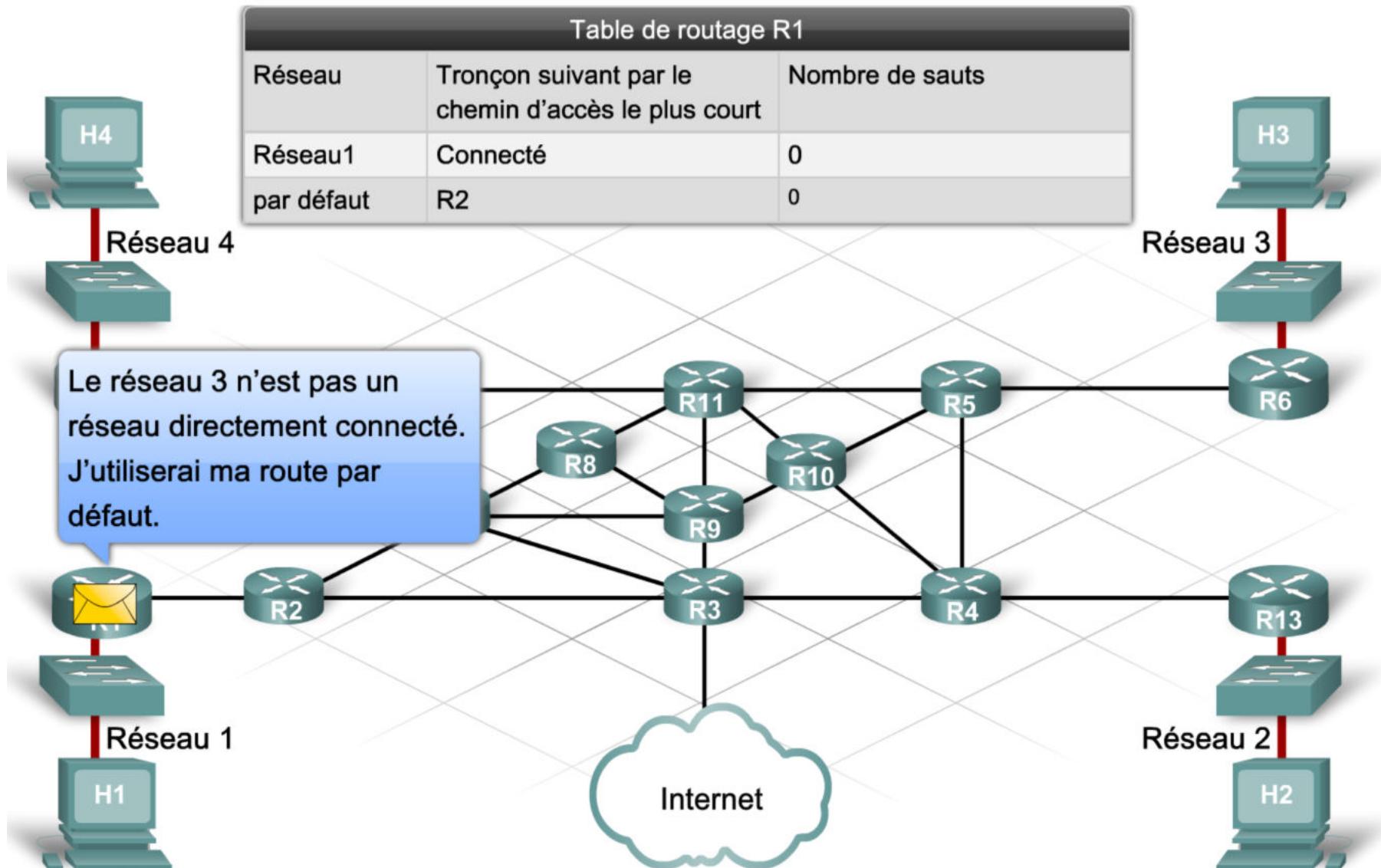
## Notion de base sur le routage



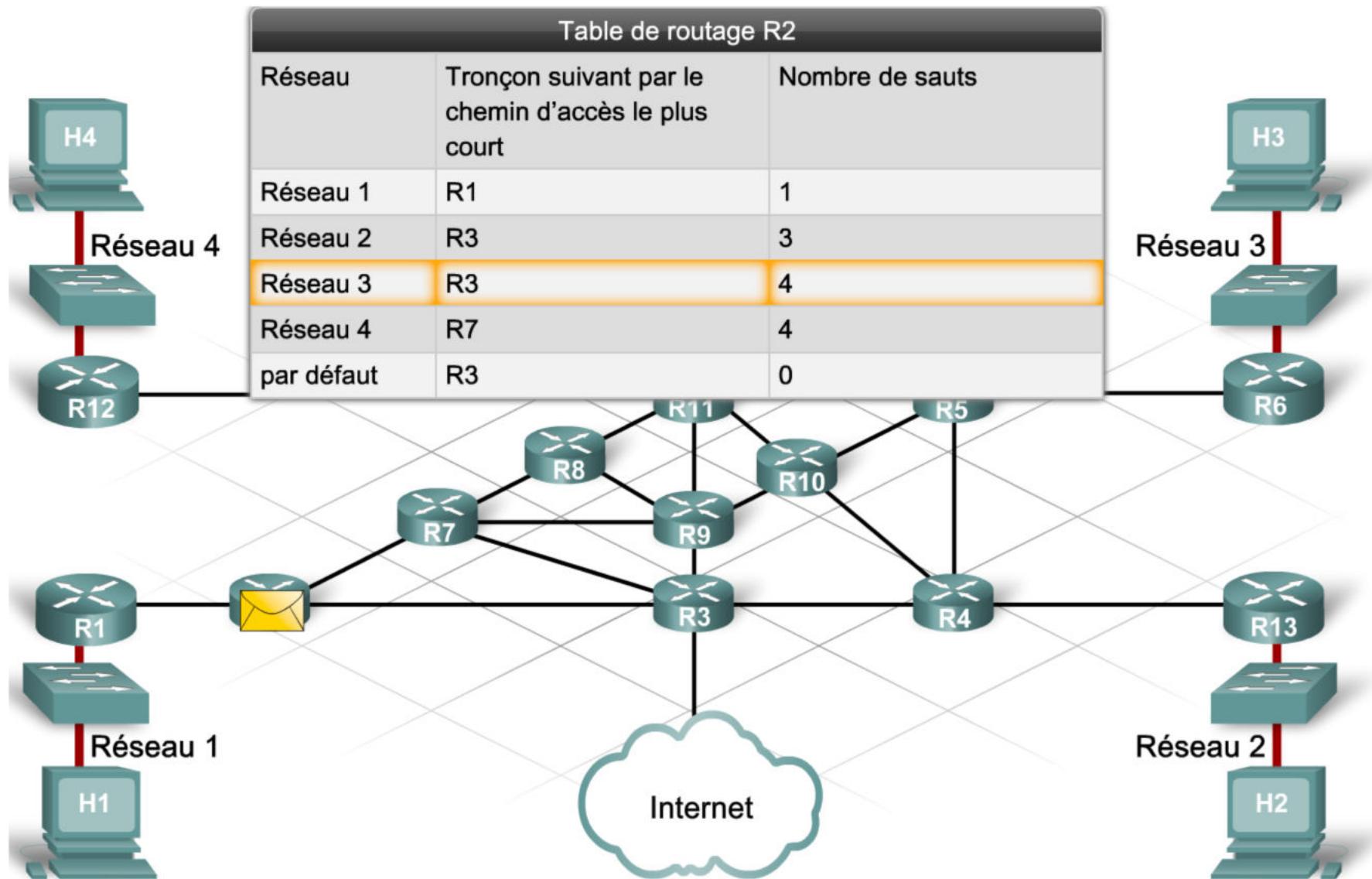
# Couche 3 : IP



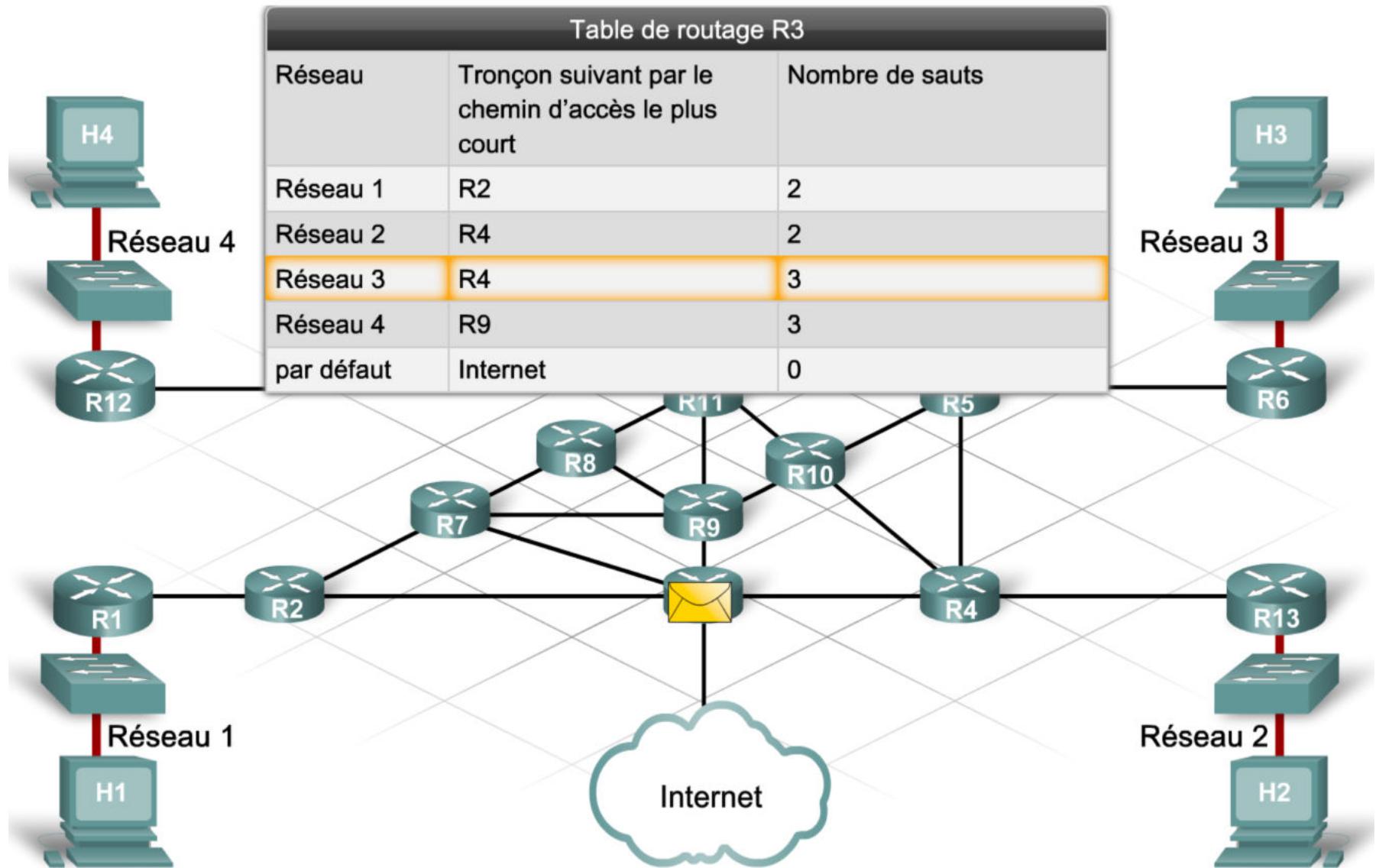
# Couche 3 : IP



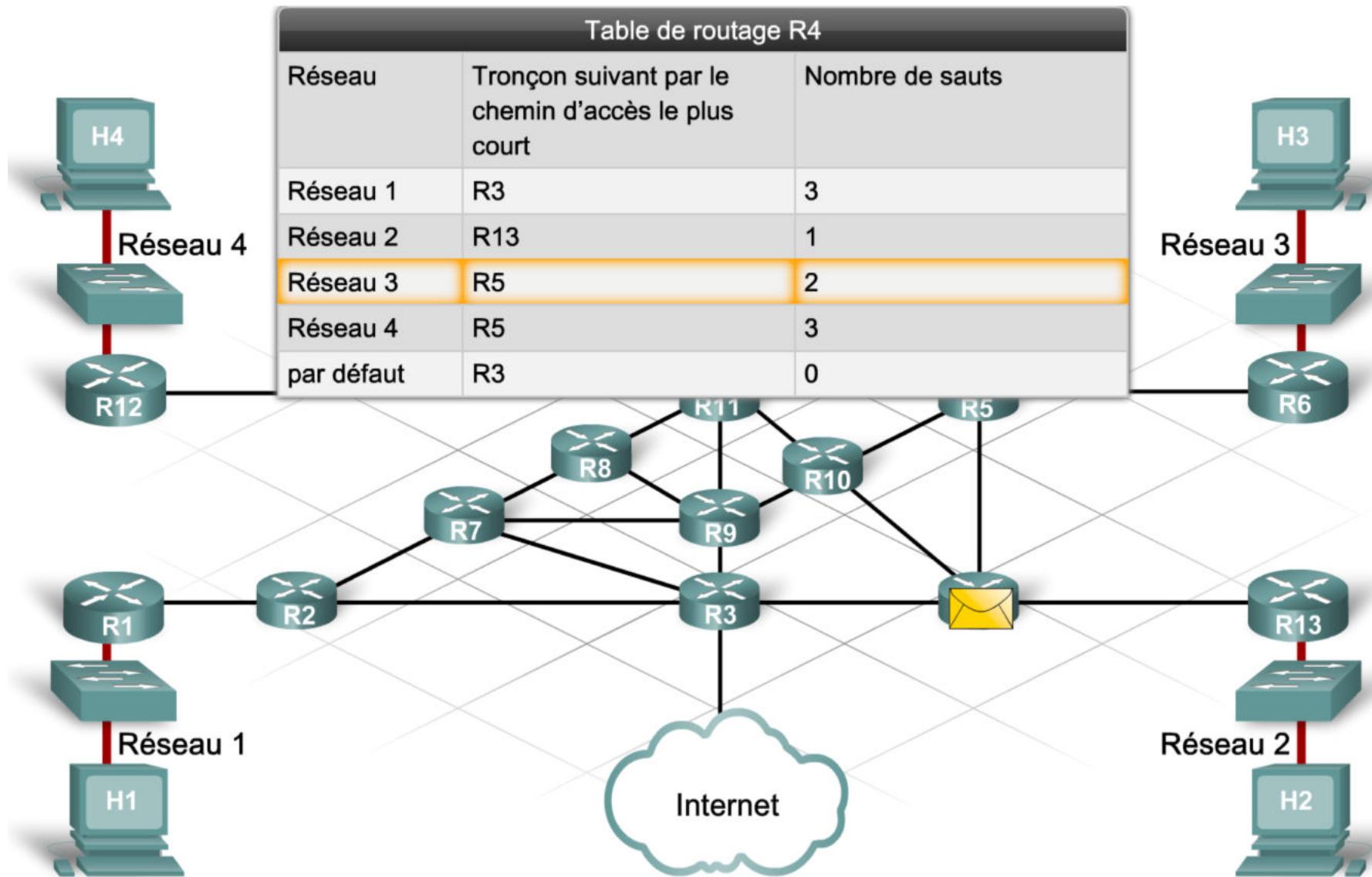
# Couche 3 : IP



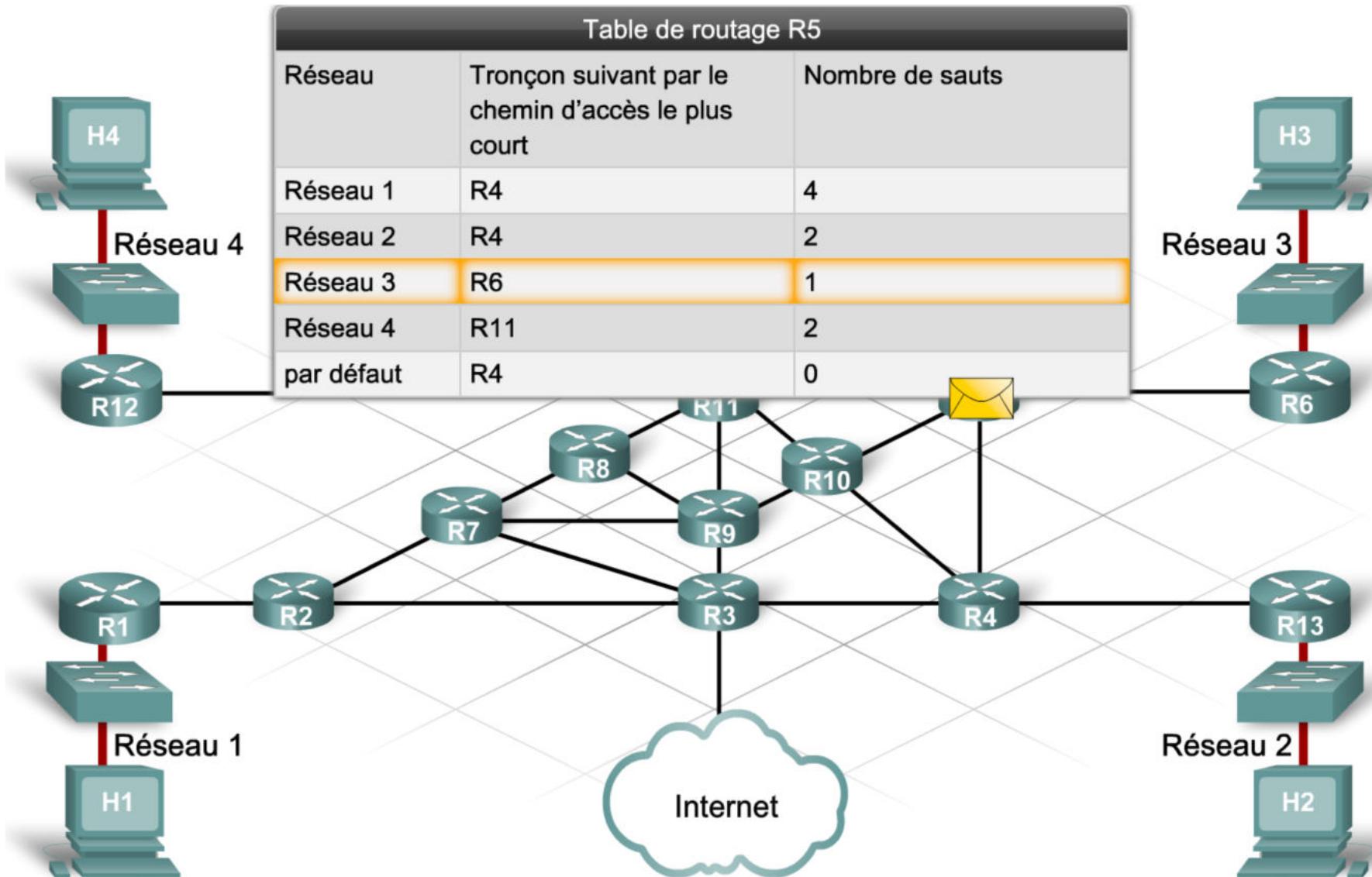
# Couche 3 : IP



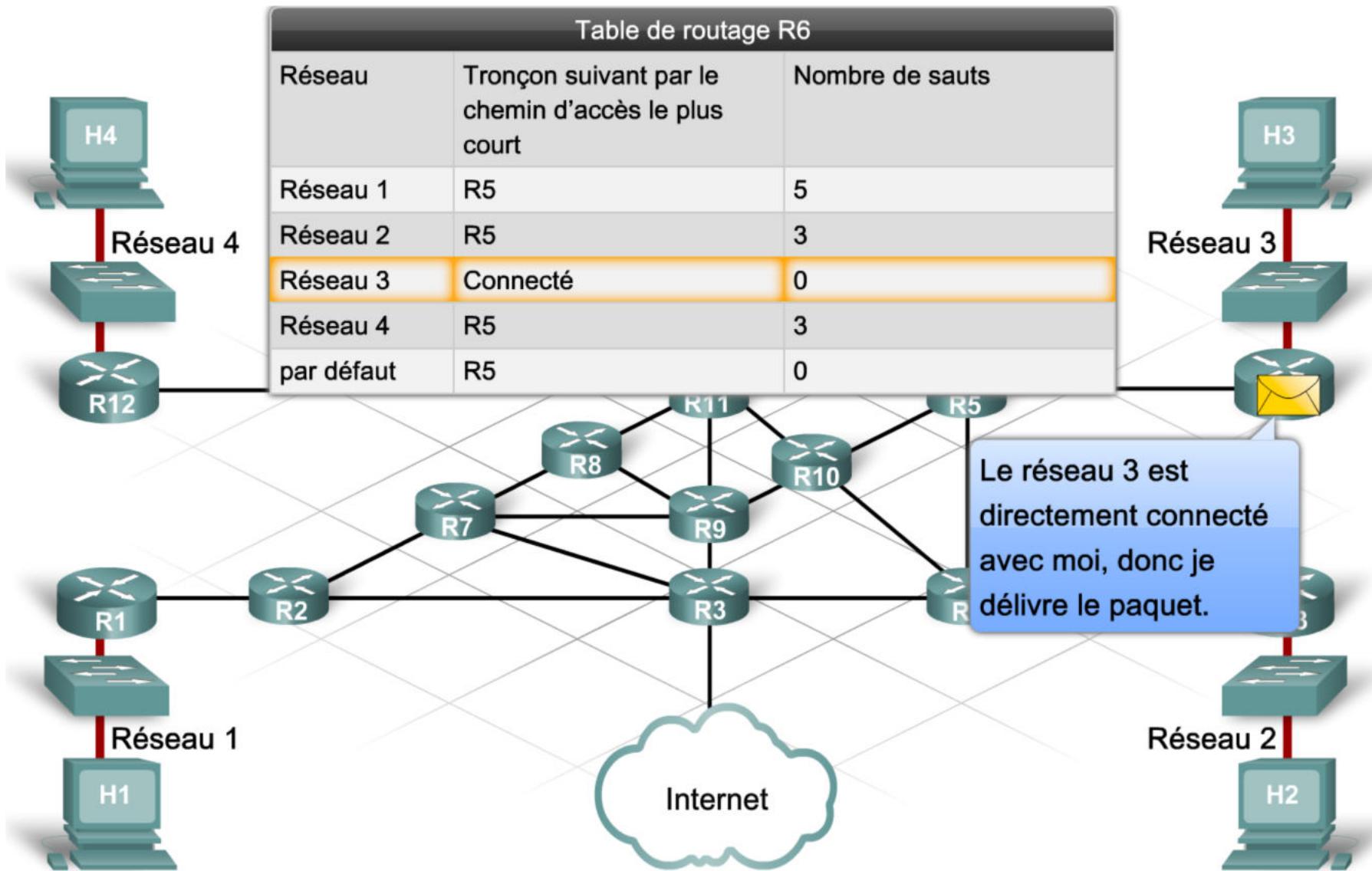
# Couche 3 : IP



# Couche 3 : IP



# Couche 3 : IP



# Couche 3 : IP

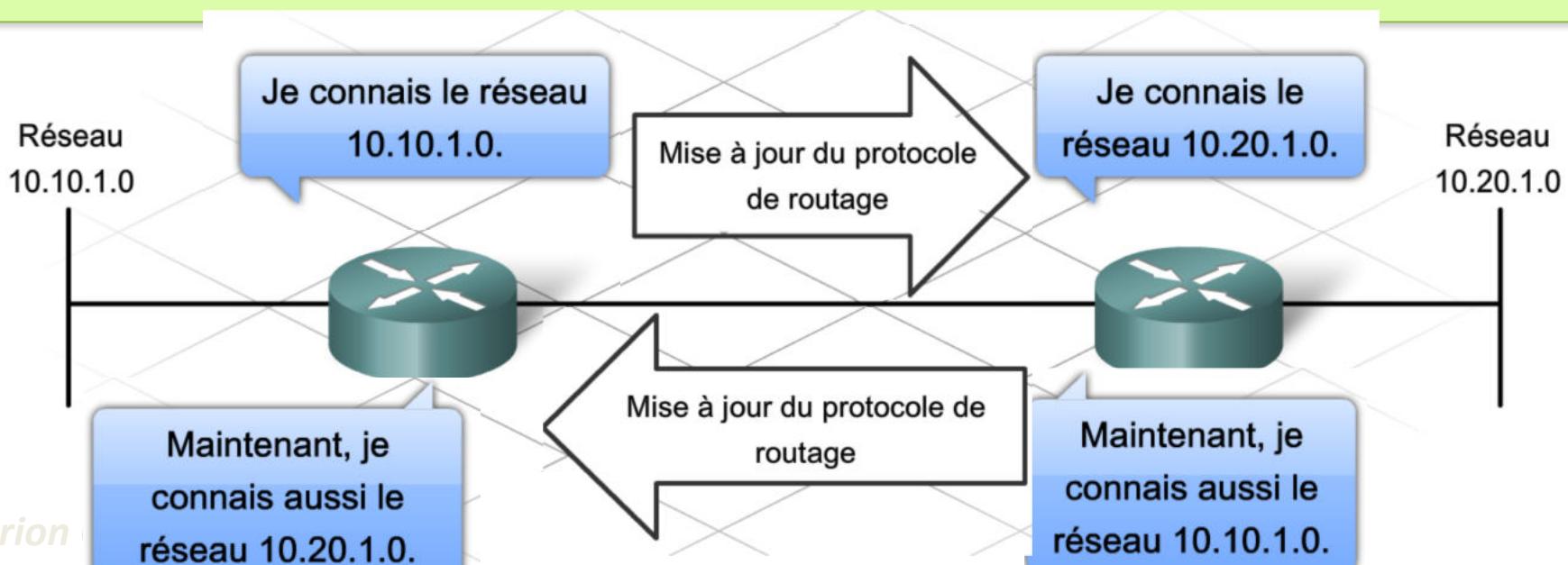
## Protocole de routage

### Protocoles de routage:

- pour gérer dynamiquement les informations reçues des autres routeurs
- pour gérer dynamiquement les changements dans le réseau
- utilisation ***des tables de routage***

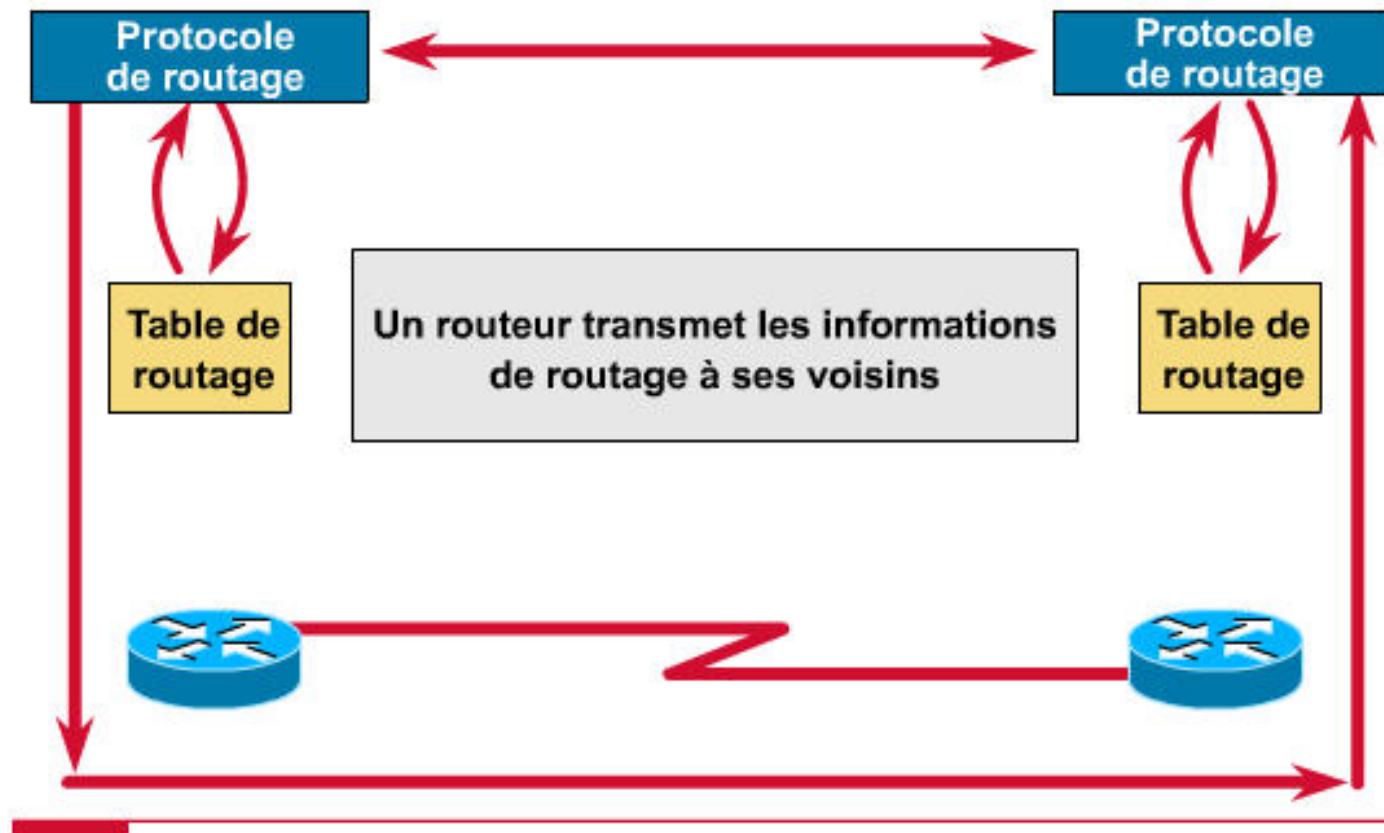
### Algorithme de routage :

- choix du meilleur chemin (optimisation)
- deux classes principales :
  - ***vecteur de distance***
  - ***états de liens***

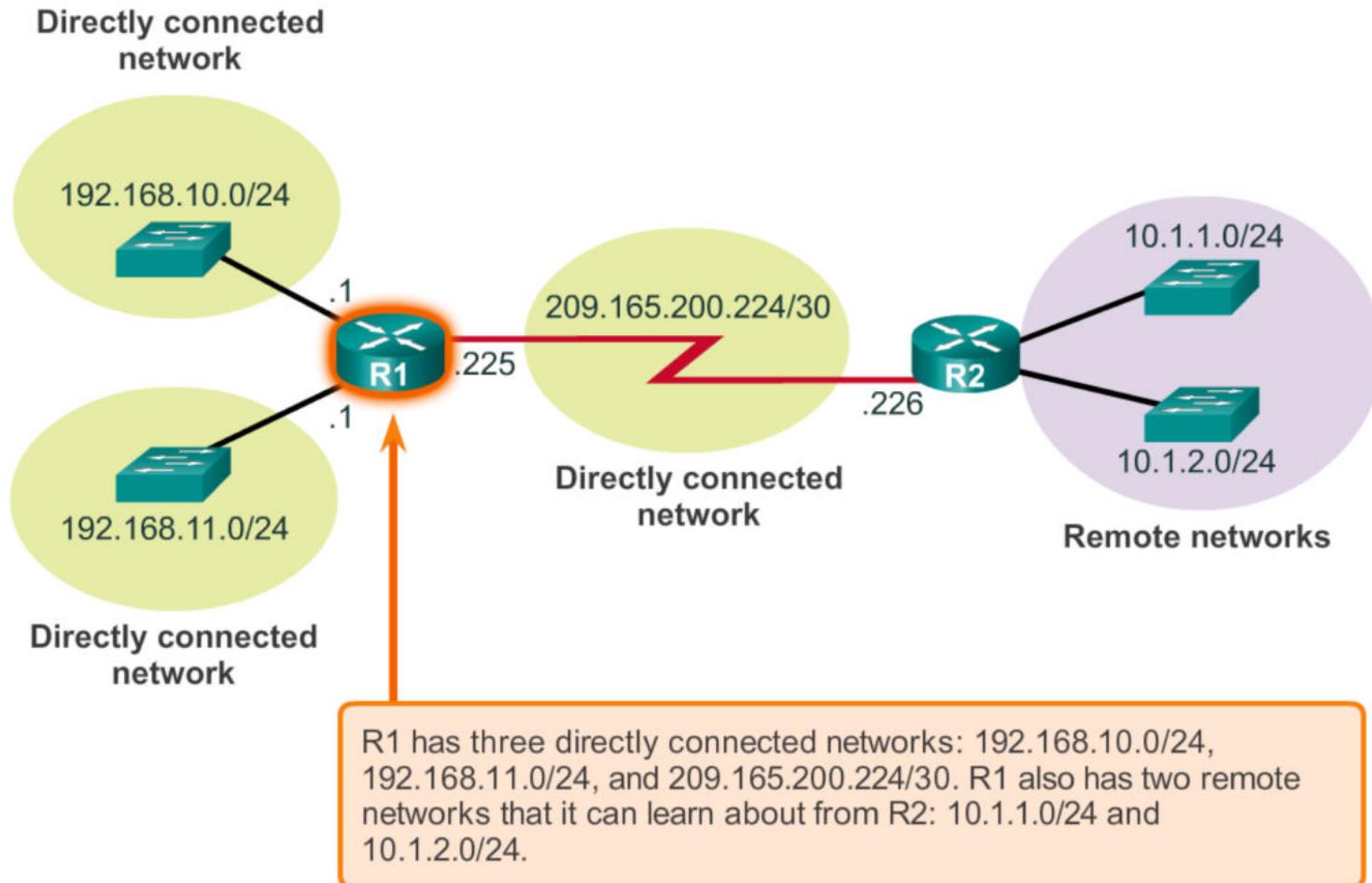


# Couche 3 : IP

En résumé, le protocole de routage tient à jour et distribue les informations de routage



# Couche 3 : IP



# Couche 3 : IP

---

## Points faibles de l'adressage IP :

- *Si une machine change de réseau, son adresse doit changer (IPV4)*
  - Administration quelquefois lourde
- *Problème de routage :*
  - Le routage utilise la partie réseau de l'adresse IP
  - Mais un hôte peut avoir plusieurs adresses IP
  - Le routage dépend de l'adresse utilisée
- *Les plages d'adresses commencent à s'épuiser*
  - Surtout les adresses de classe B...
  - **Vers IPv6...**

# Panorama des protocoles internet

I. Modèles TCP/IP – principe de l'encapsulation

II. Modèles TCP/IP -- couche 3 (réseau)

- a. rôle
- b. adressage
- c. introduction au routage

II. Modèles TCP/UDP – couche 4 (transport)

- b. rôle et fonctionnement
- c. TCP et UDP
- d. NAT et PAT

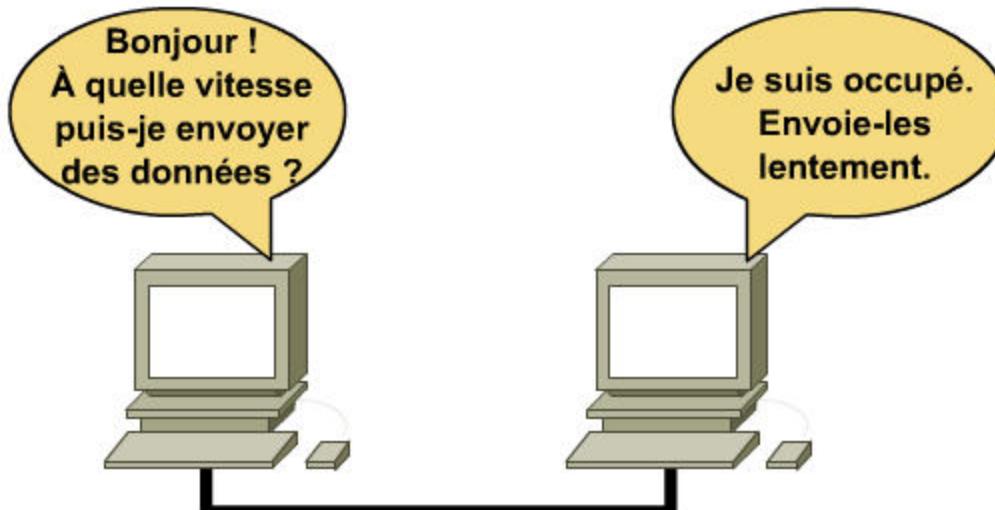
IV. Modèles TCP/IP – couche 5 (application)

- a. rôle et fonctionnement
- b. DNS et DHCP
- c. Services et protocoles (HTTP, FTP, Telnet,...)

# Couche 4 : TCP/UDP

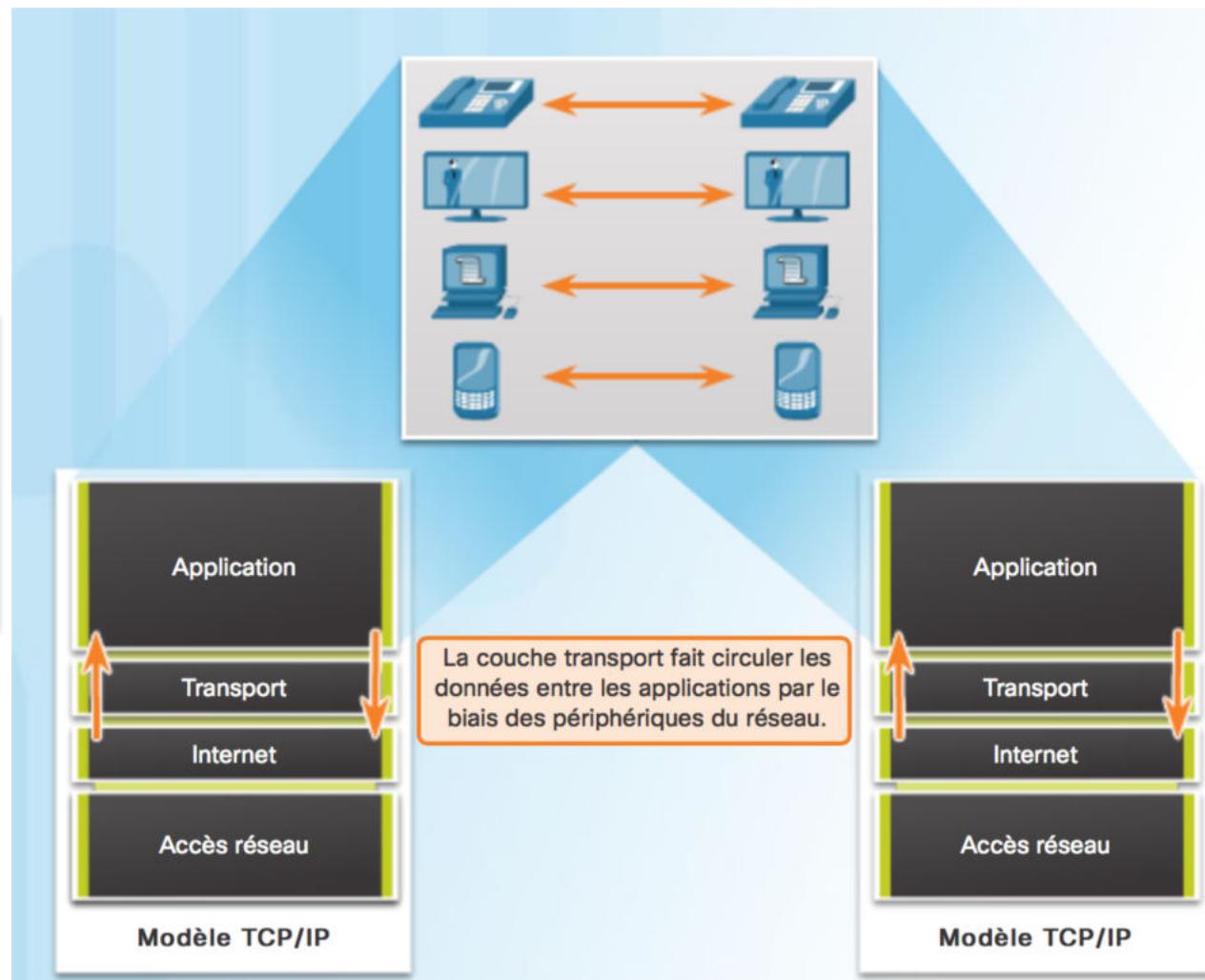
## Couche transport :

- la couche transport se sert du réseau comme d'un « nuage » pour envoyer ces paquets
  - elle ne s'occupe pas du chemin emprunté par les données
- assurer la qualité de service :
  - transporter et contrôler le flux d'informations de la source à la destination de manière fiable et précise
  - contrôle de bout en bout



# Couche 4 : TCP/UDP

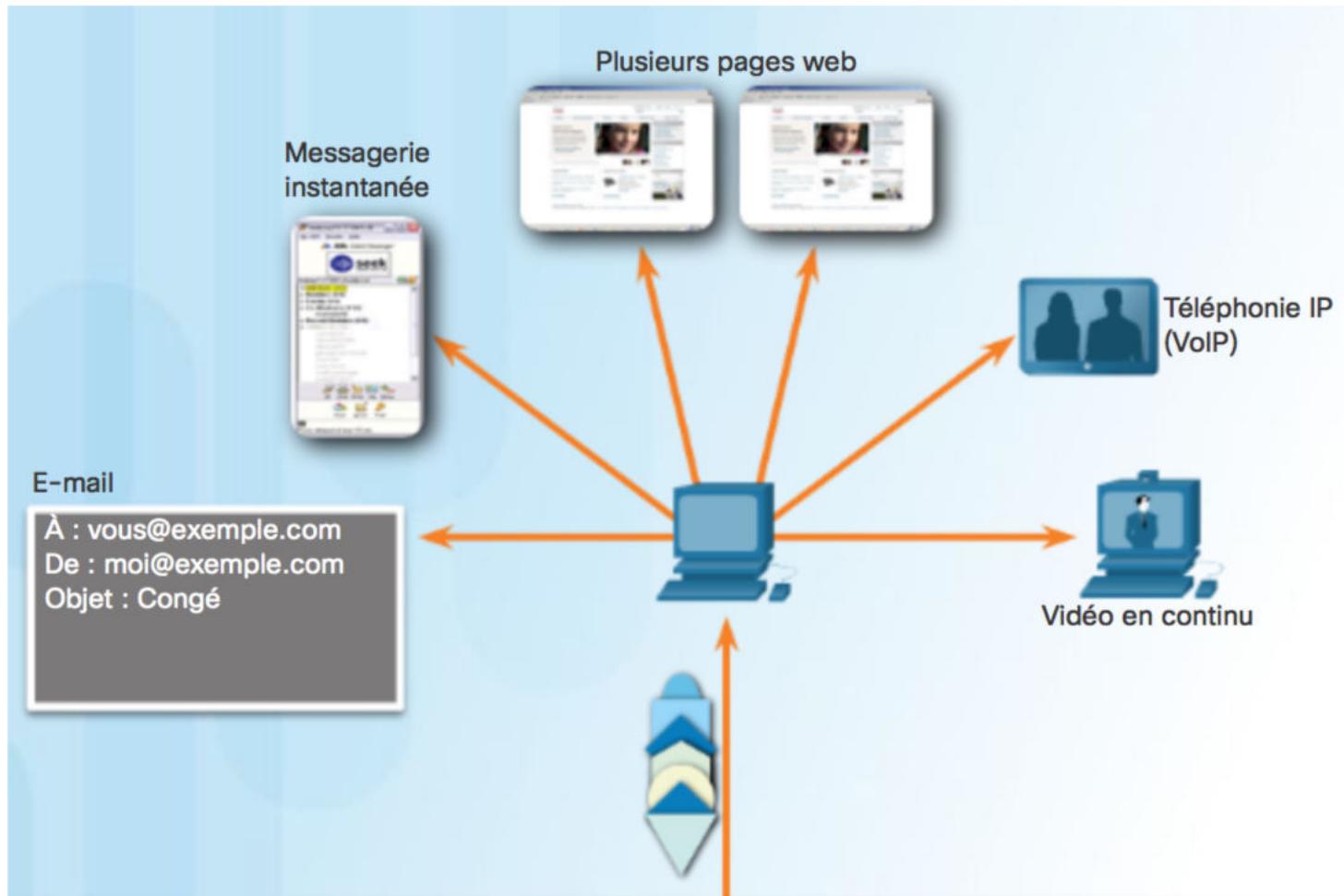
**La couche transport :**  
établit une session de communication temporaire entre deux applications et l'achemine les données entre ces deux applications



# Couche 4 : TCP/UDP

La couche transport : 2 rôles majeurs

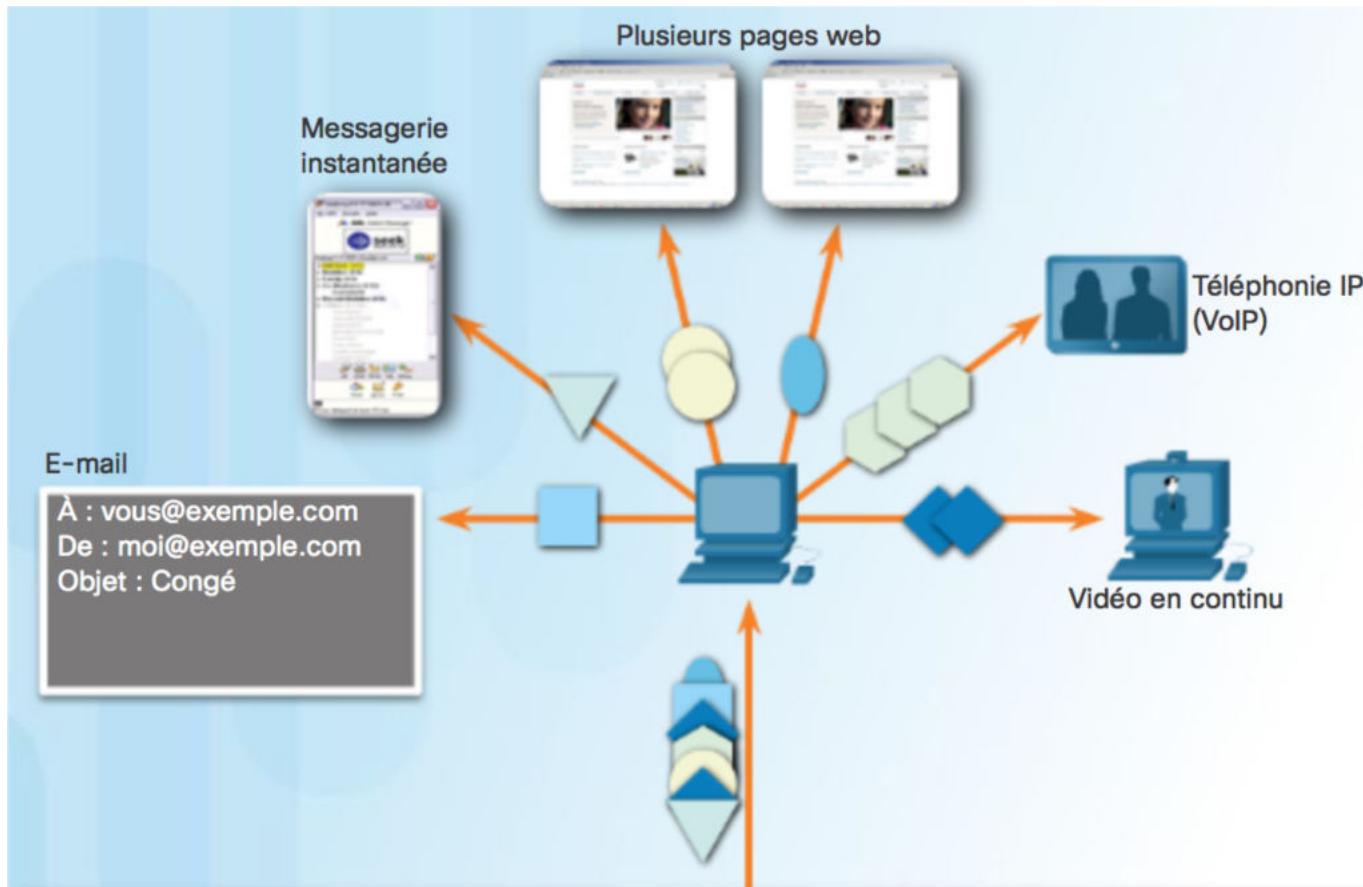
La segmentation des données  
(envoi et réception)



# Couche 4 : TCP/UDP

La couche transport : 2 rôles majeurs

Identification de l'application



La couche transport s'assure que même lorsque plusieurs applications s'exécutent sur un périphérique, toutes les applications reçoivent les données correctes.

# Couche 4 : TCP/UDP

UDP

Protocoles de la couche transport

TCP



- Téléphonie sur IP
- Lecture vidéo en continu

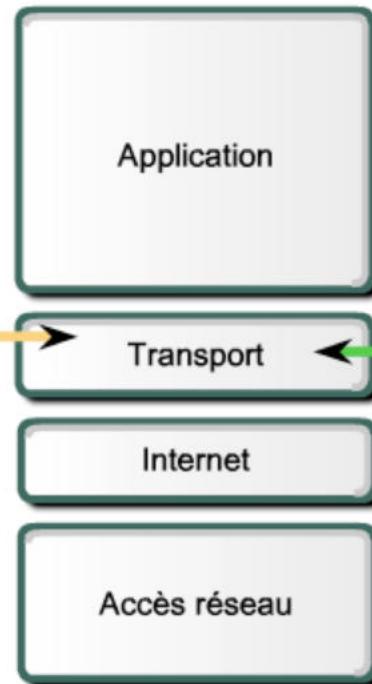
Propriétés requises du protocole

- Rapide
- Faibles frais généraux
- Ne nécessite pas de reçu
- Ne renvoie pas les données perdues
- Livre les données selon leur arrivée

Modèle OSI



Modèle TCP/IP



- SMTP/POP (Courriel)
- HTTP

Propriétés requises du protocole

- Fiable
- Accuse réception des données
- Renvoie les données perdues
- Livre les données dans l'ordre de leur envoi

Les concepteurs d'applications choisissent le protocole de couche transport approprié en fonction de la nature de l'application.

# Couche 4 : TCP/UDP

**Couche transport** : applications ont différents besoins en matière de transport

## Protocole UDP:

- Protocole simple, *sans connexion*
- Protocole de couche transport, dit *au mieux*
- Les datagrammes peuvent arriver dans le désordre, ou se perdre
  - les applications utilisant UDP peuvent tolérer la perte de petites quantités de données.
  - exemple : webradio. Si quelques données manquent à l'arrivée, la qualité de la diffusion ne s'en trouvera que modérément affectée.

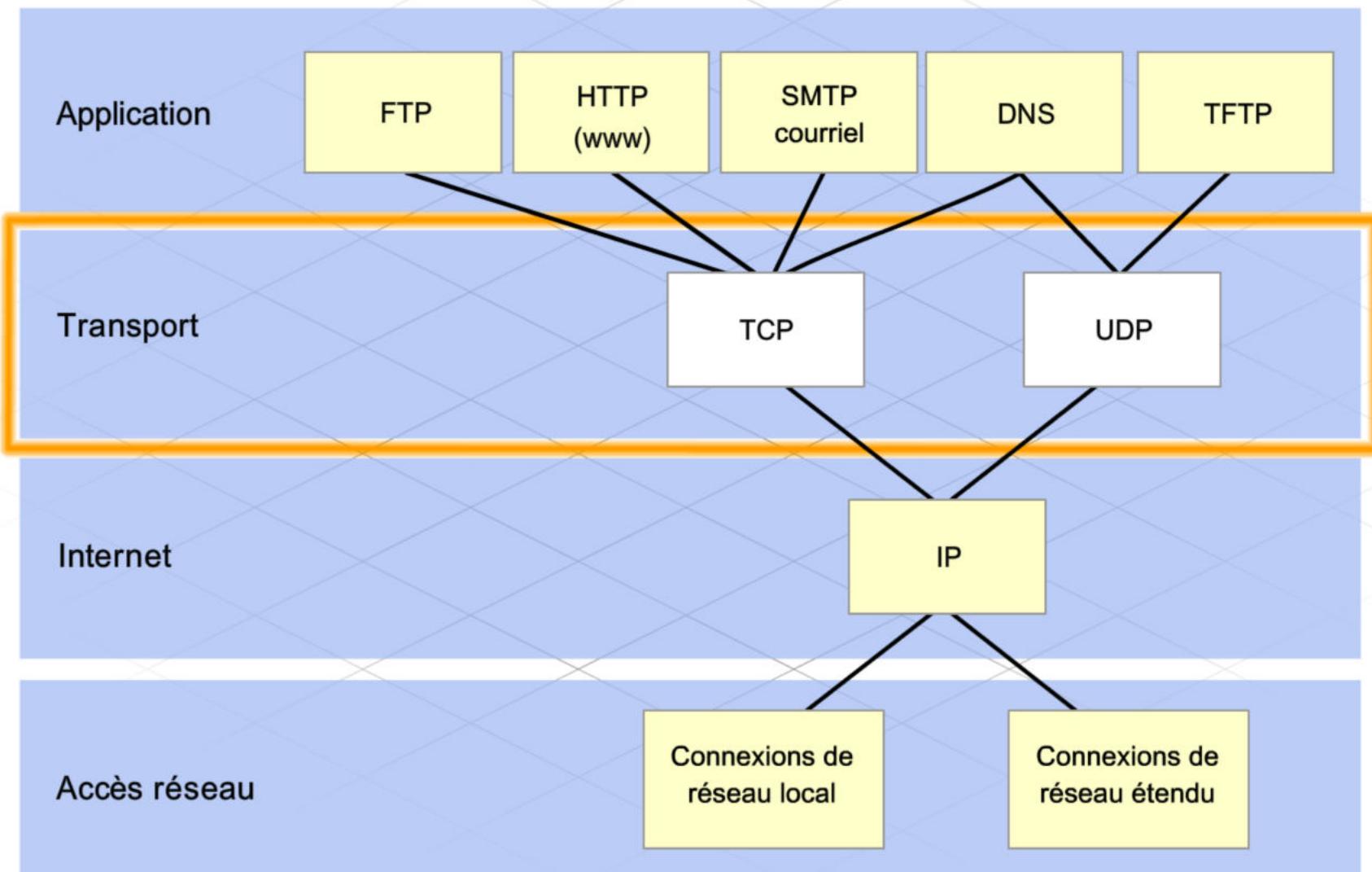
## Protocole TCP :

- protocole *fiable*, avec *livraison garantie*.
- *accusés de réception*
- renvoie des paquets non reçus (réception non confirmée)
- création d'une *session* de communication.

C'est pourquoi TCP est appelé protocole orienté *connexion*.

- surcharge du réseau
- retard possible non admis par certaines applications

# Couche 4 : TCP/UDP



# Couche 4 : TCP/UDP

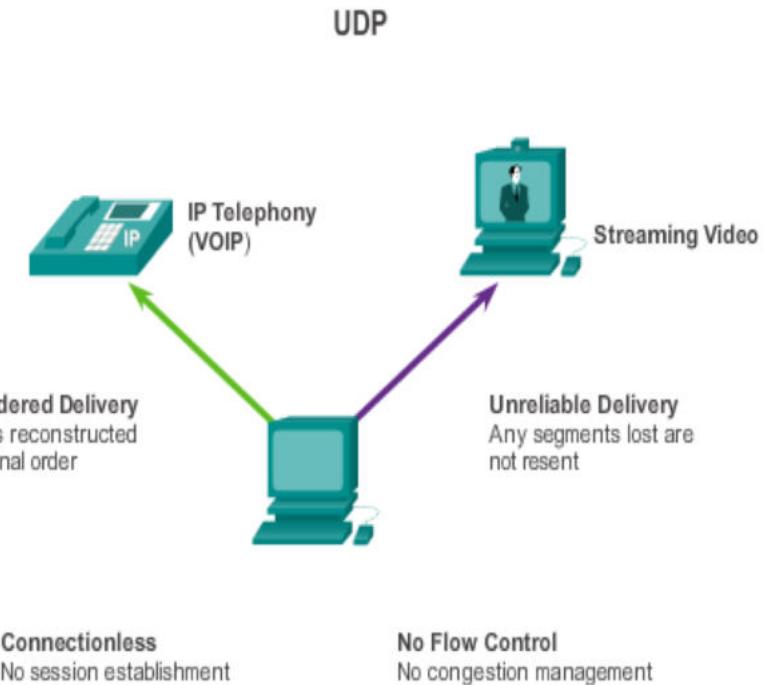
## Protocole UDP

### User Datagram Protocol (UDP)

- RFC 768
- Sans connexion
- non fiable
- Pas d'info sur l'ordre de reconstruction des paquets
- Pas de contrôle de flux

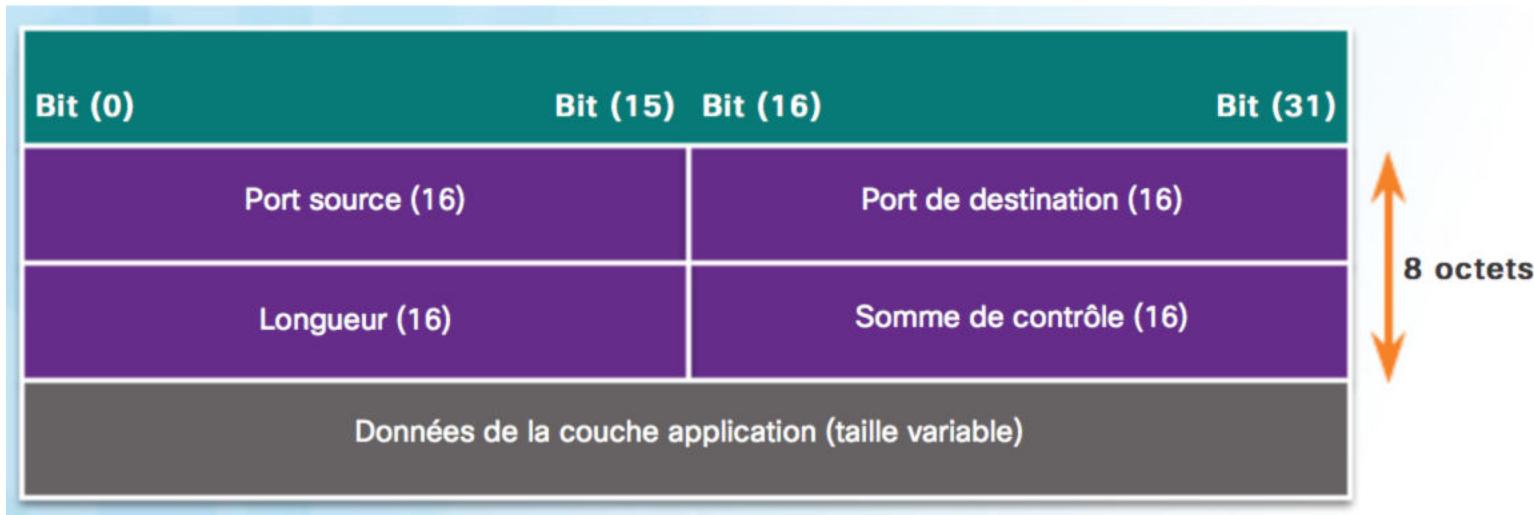
### Applications utilisant UDP:

- Domain Name System (DNS)
- Video Streaming
- Voix sur IP (VoIP)



# Couche 4 : TCP/UDP

## Protocole UDP : entête

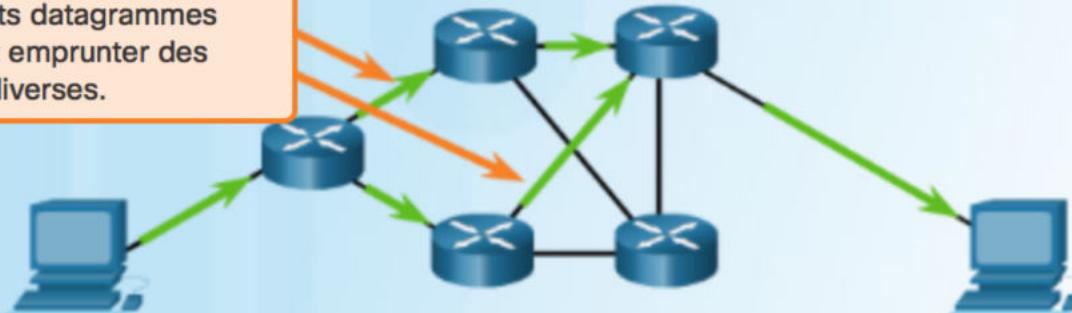


# Couche 4 : TCP/UDP

## Protocole UDP

### UDP : sans connexion et peu fiable

Différents datagrammes peuvent emprunter des routes diverses.



**Données**  
Les données sont divisées en datagrammes.

- Datagramme 1
- Datagramme 2
- Datagramme 3
- Datagramme 4
- Datagramme 5
- Datagramme 6

Comme ils ont suivi des routes différentes jusqu'à leur destination, les datagrammes arrivent dans le désordre.

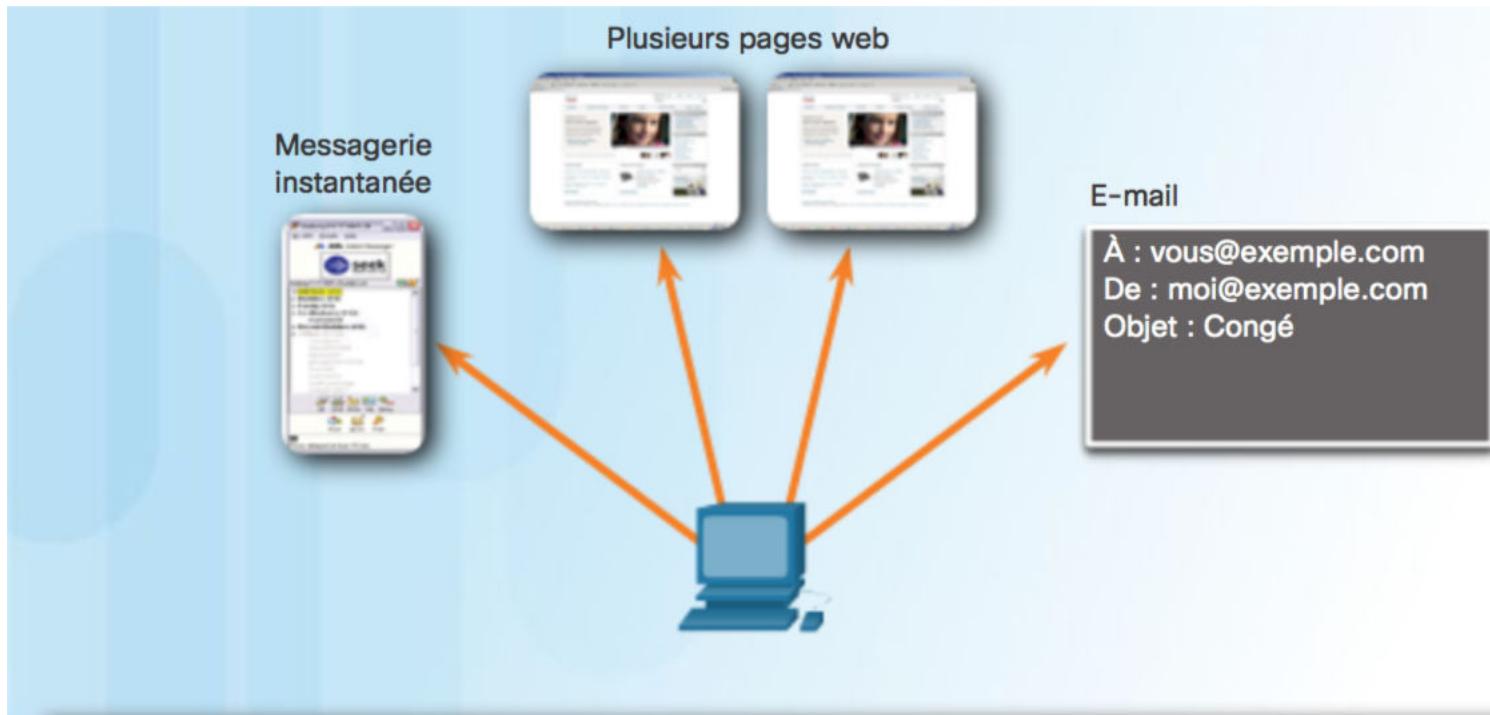
- Datagramme 1
- Datagramme 2
- Datagramme 6
- Datagramme 5
- Datagramme 4

Les datagrammes dans le désordre ne sont pas remis dans l'ordre.

Les datagrammes perdus ne sont pas renvoyés.

# Couche 4 : TCP/UDP

## Protocole TCP

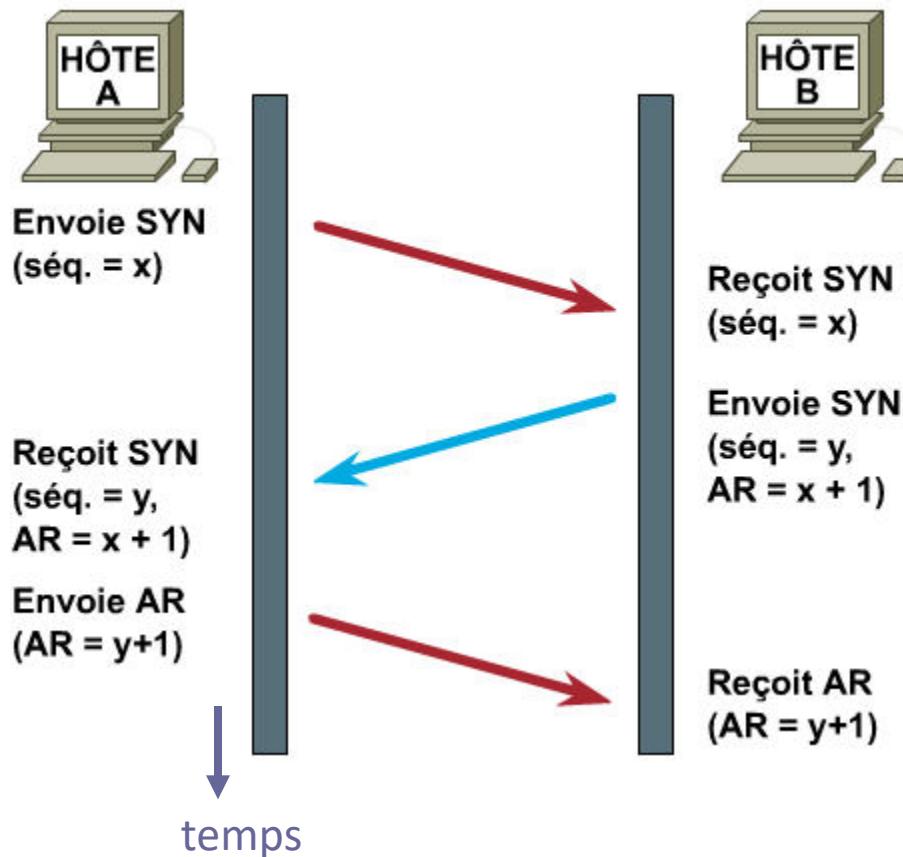


- L'établissement d'une **session** permet de s'assurer que l'application est prête à recevoir les données.
- La livraison dans un ordre défini permet de s'assurer que les segments sont remis dans le bon ordre.
- L'acheminement fiable signifie que les segments perdus sont renvoyés afin que les données soient reçues dans leur intégralité.

# Couche 4 : TCP/UDP

## Protocole TCP : Handshake

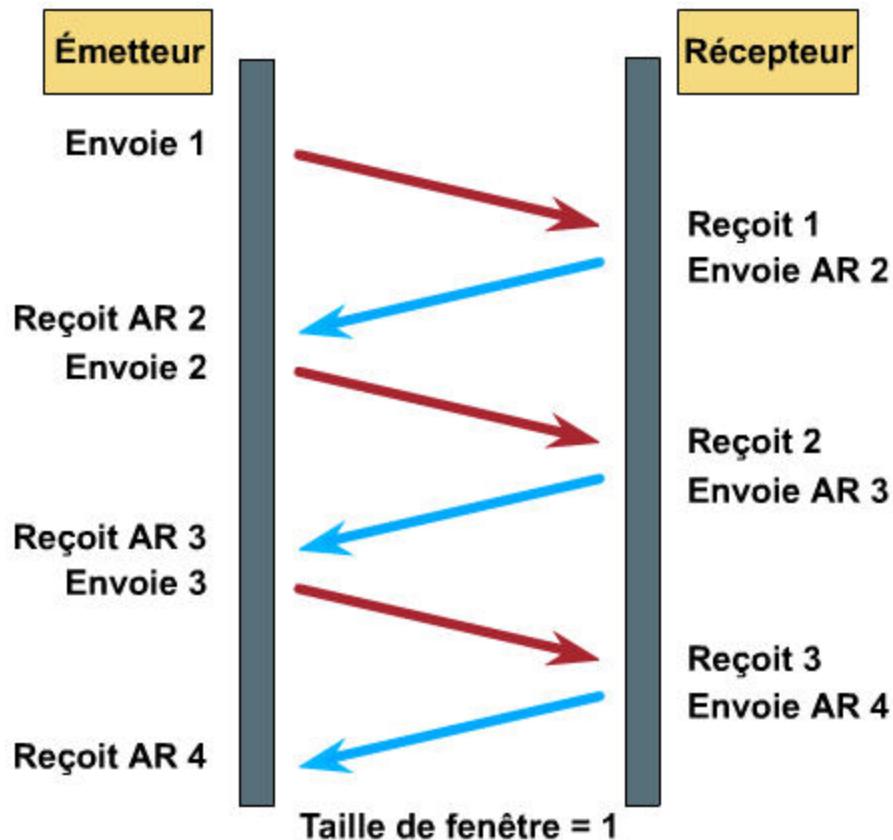
Etablissement de la connexion avec un échange à 3 étapes



# Couche 4 : TCP/UDP

## Protocole TCP : mécanisme send – wait

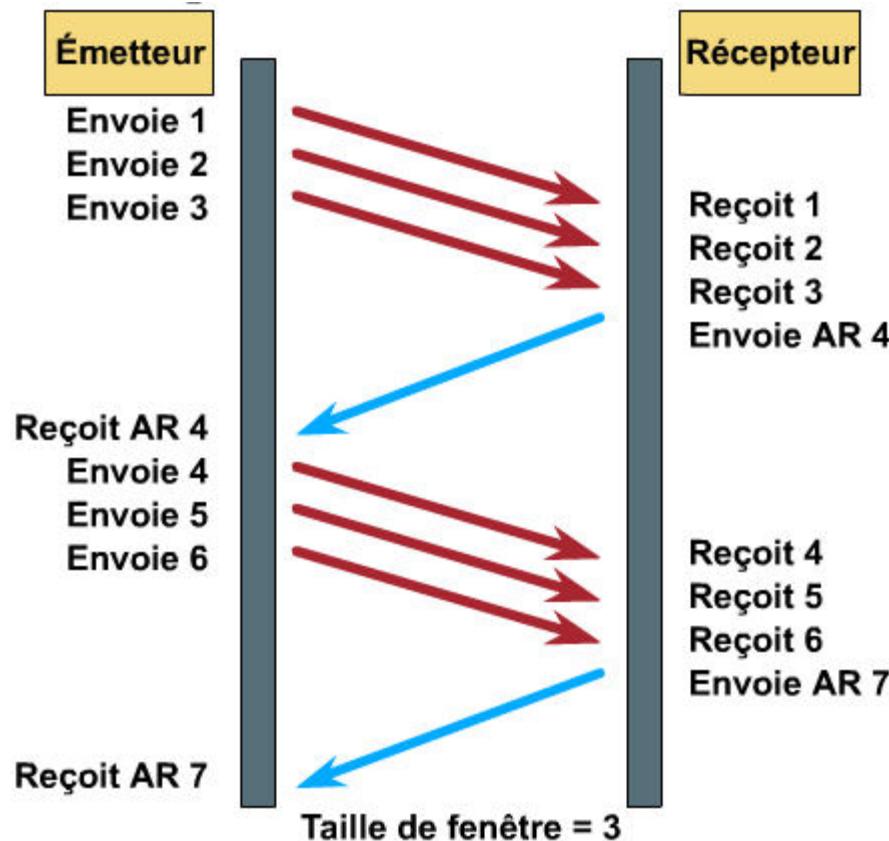
Accusé de réception simple



# Couche 4 : TCP/UDP

## Protocole TCP : mécanisme send – wait

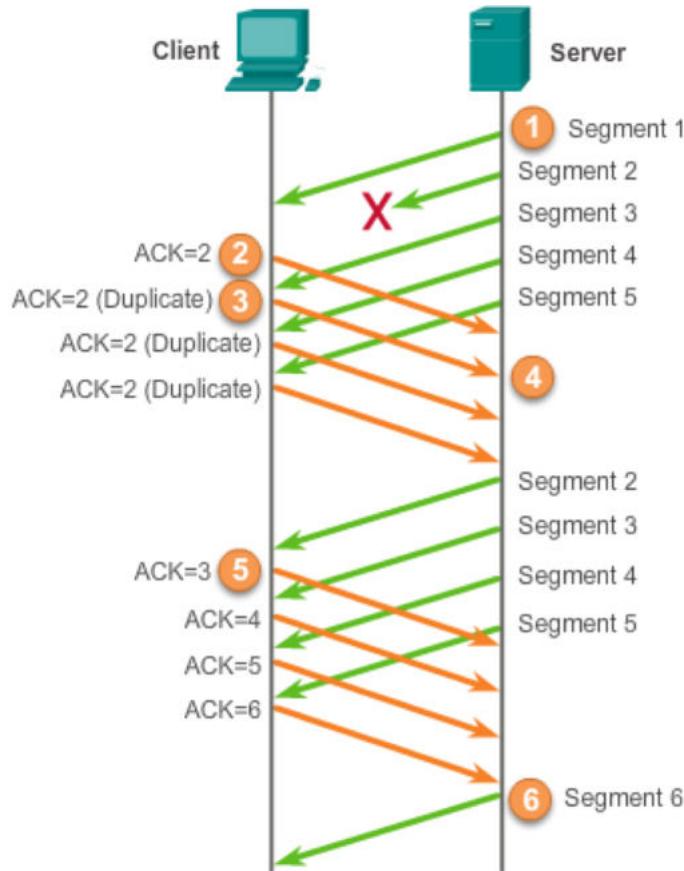
Fenêtre de 3 messages



# Couche 4 : TCP/UDP

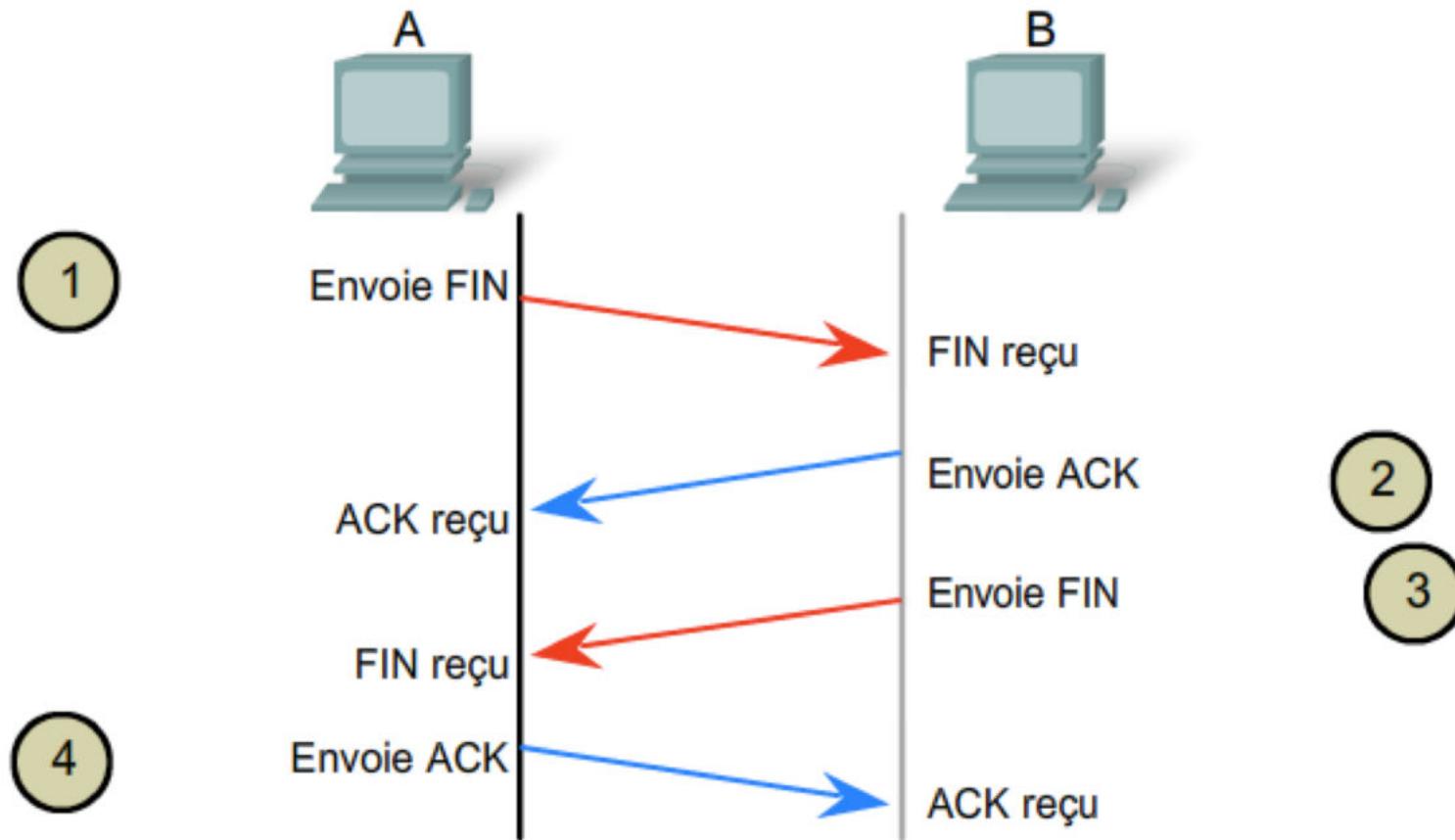
## Protocole TCP : mécanisme send – wait

Si erreur, retransmission



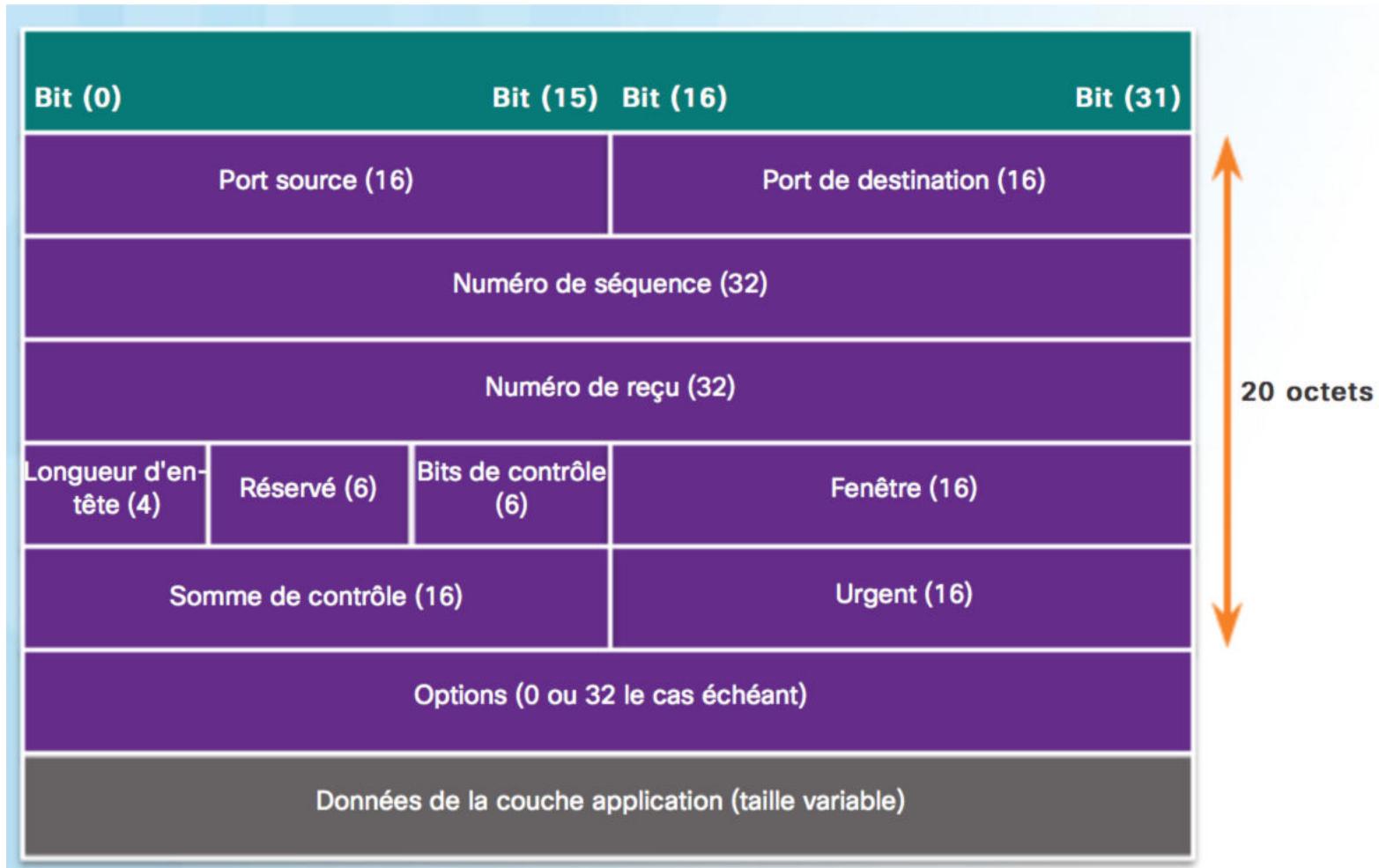
# Couche 4 : TCP/UDP

## Protocole TCP : fermeture d'une session TCP



# Couche 4 : TCP/UDP

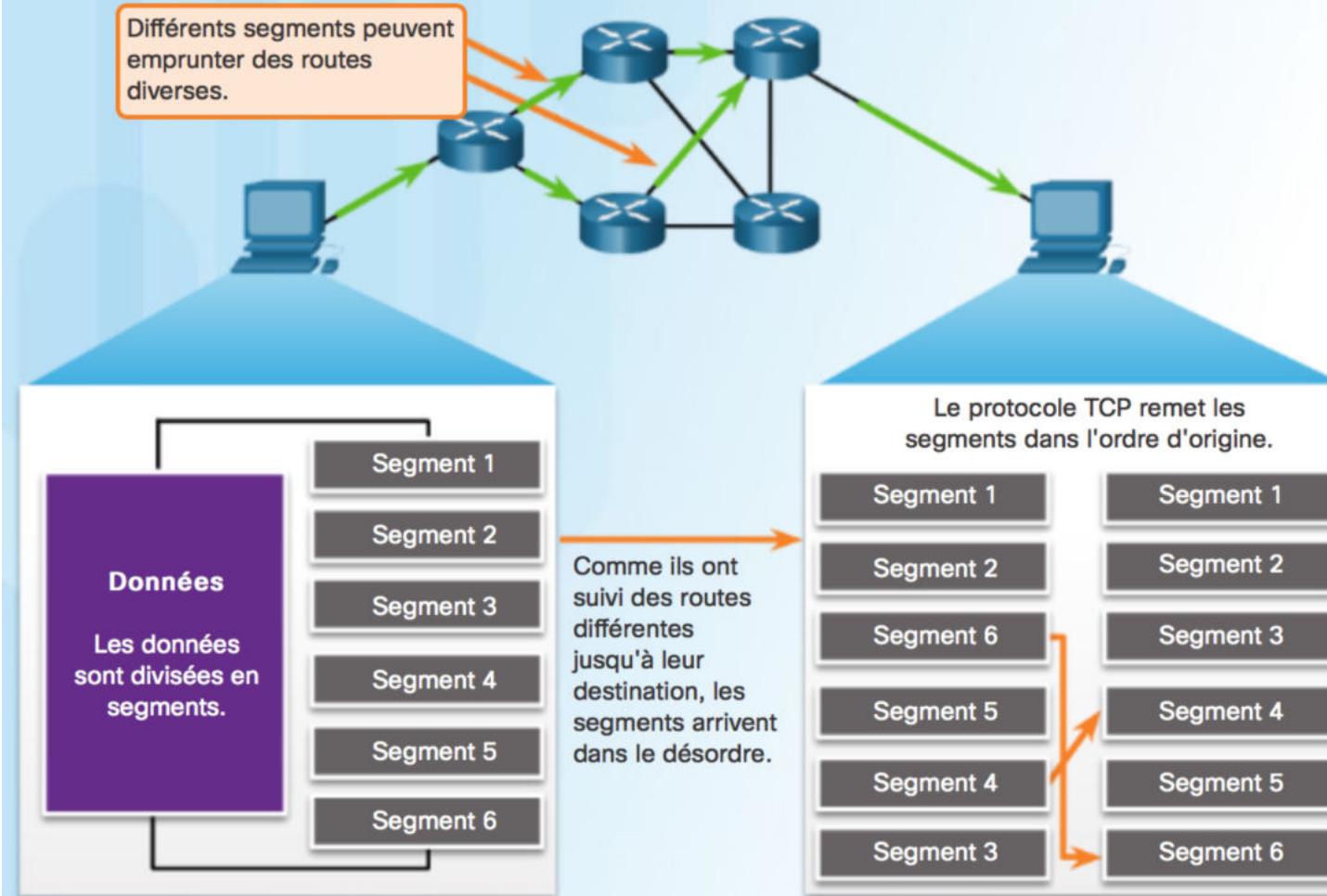
## Protocole TCP : entête



# Couche 4 : TCP/UDP

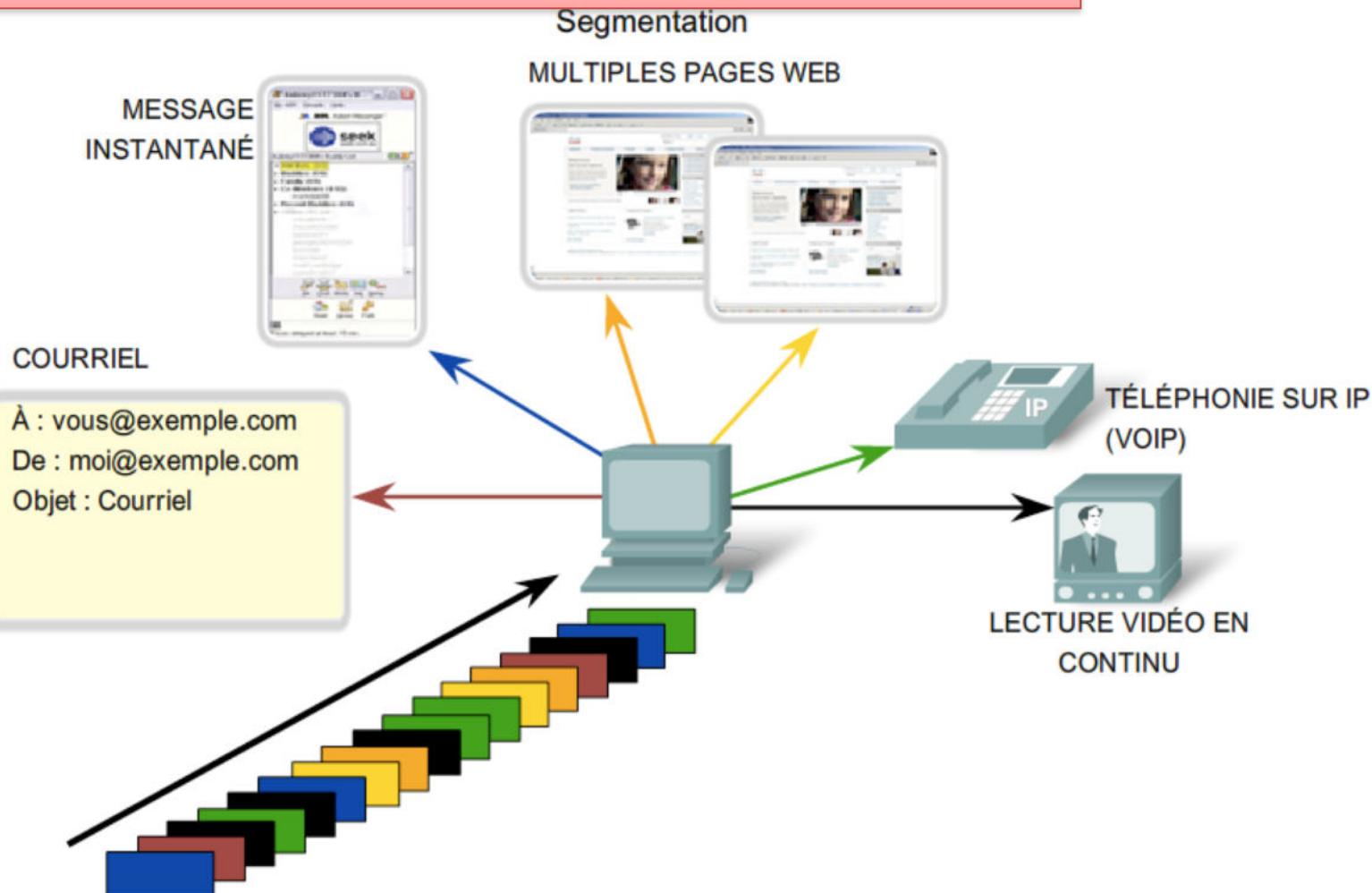
## Protocole TCP

Les segments TCP sont réorganisés au niveau de la destination.



# Couche 4 : TCP/UDP

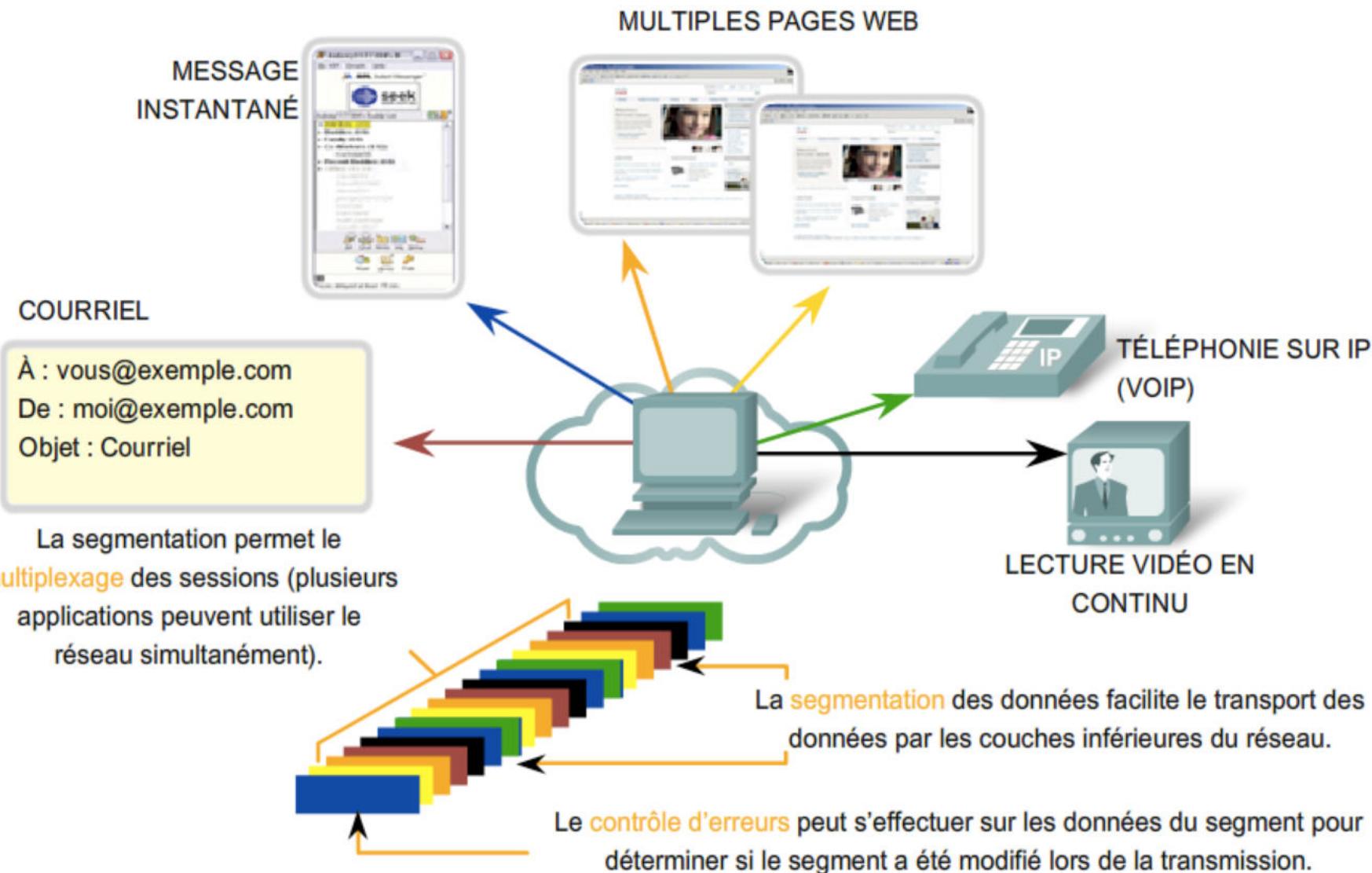
## Protocoles TCP et UDP : prise en charge de services multiples



La couche transport divise les données en segments qui sont plus faciles à gérer et à transporter.

# Couche 4 : TCP/UDP

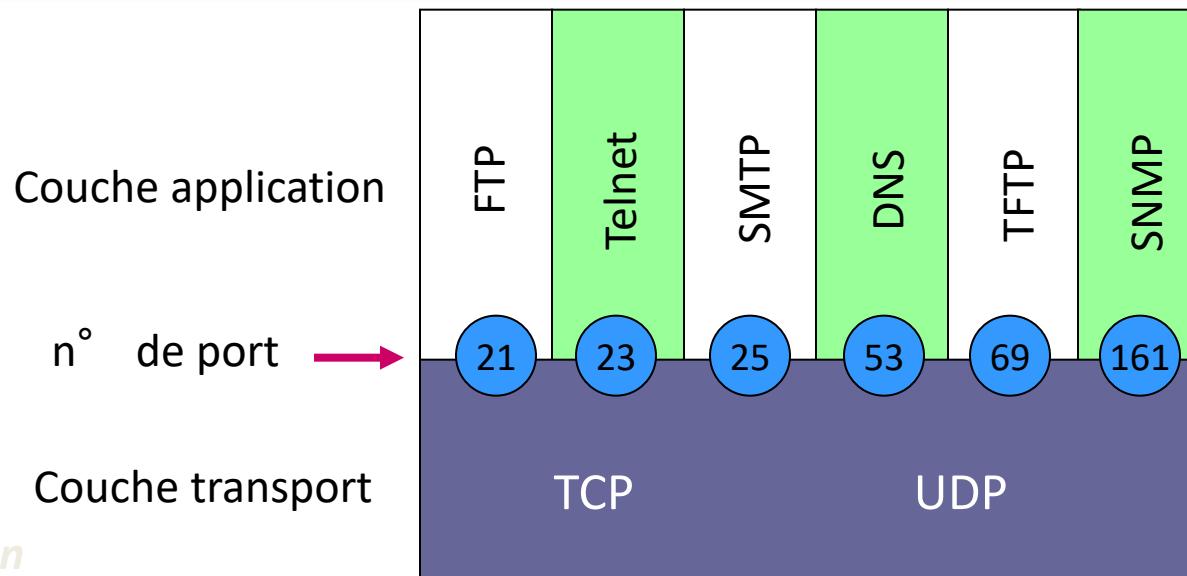
Services de la couche transport



# Couche 4 : TCP/UDP

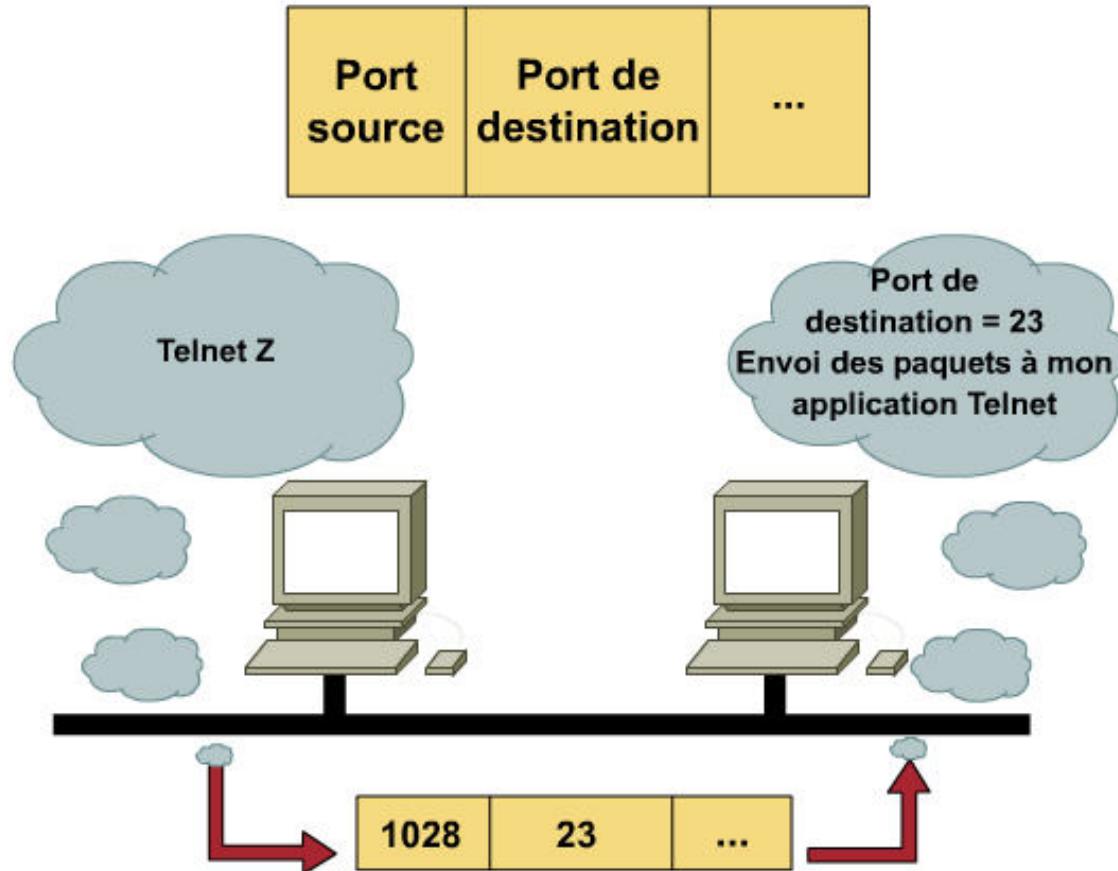
## Les ports

- **Utilisation de numéro de ports (ou sockets pour IP + port)** par les protocoles TCP/IP pour transmettre les infos à la couche supérieure
- **Rôle** : distinguer les différentes informations circulant simultanément sur le réseau
- **Deux plages** d'attribution de ces numéros :
  - De 1 à 1023 : réservés aux applications (FTP, telnet ...)
  - > 1023 : non attribués
- **Exemple 1** : conversation destinée à l'appli FTP utilise le n° de port 21
- **Exemple 2** : conversation ne visant pas une appli ayant un n° de port reconnu se voit attribuer un n° > 1023



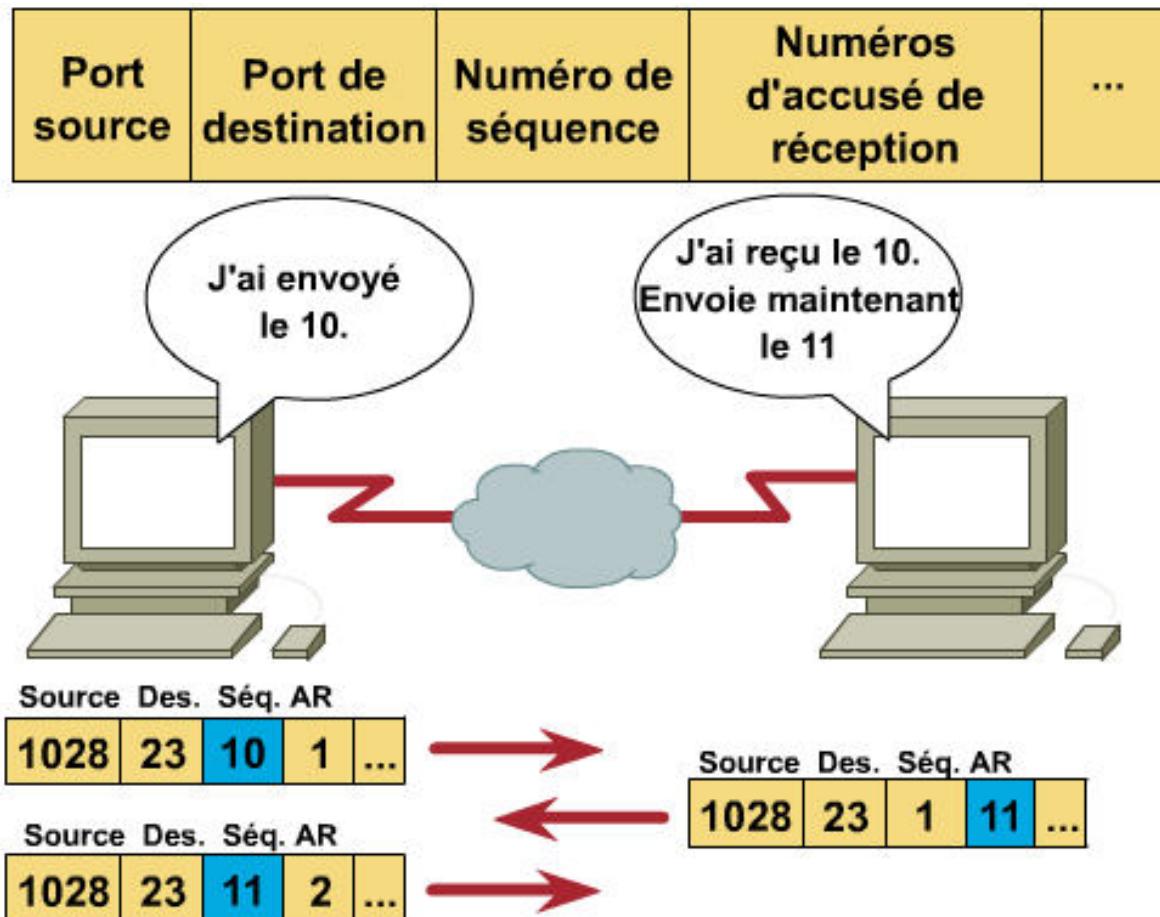
# Couche 4 : TCP/UDP

## Cas d'un telnet



# Couche 4 : TCP/UDP

## Cas d'un telnet



# Couche 4 : TCP/UDP

## Adressage de ports

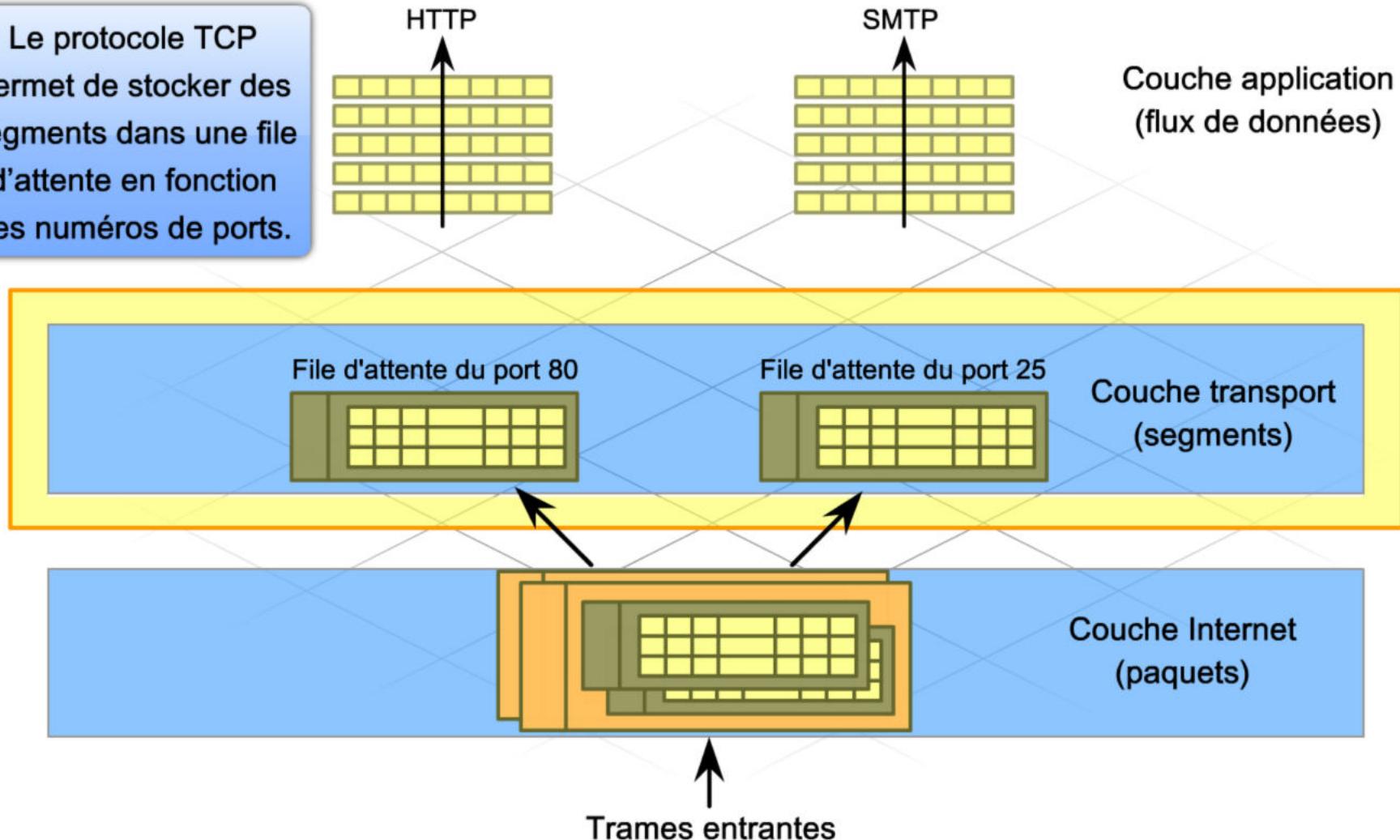
| Différentes applications |                                  |                                  |                                  |
|--------------------------|----------------------------------|----------------------------------|----------------------------------|
| Messagerie électronique  | Page HTML                        | Conversation sur Internet (Chat) |                                  |
| Protocoles               | POP3                             | HTTP                             | MI                               |
| Transport                | Port des applications<br>Données | Port des applications<br>Données | Port des applications<br>Données |
| Numéros de port          | 110                              | 80                               | 531                              |

Les données des différentes applications sont dirigées vers l'application adéquate car chaque application dispose d'un numéro de port unique.

# Couche 4 : TCP/UDP

## Prise en charge de services multiples

Le protocole TCP permet de stocker des segments dans une file d'attente en fonction des numéros de ports.



# Couche 4 : TCP/UDP

## Prise en charge de services multiples

**Socket** = combinaison du ***numéro de port*** de la couche transport et de ***l'adresse IP*** de la couche réseau de l'hôte

- pour ***identifier*** de manière unique un processus d'application d'un hôte individuel
- interface de connexion, ou socket
- ***paire de sockets*** = composée des adresses IP et numéros de port source et de destination, identifie la conversation spécifique entre les deux hôtes.

Exemple de socket client :

- 192.168.1.1:7151

Le socket d'un serveur Web peut avoir la forme suivante :

- 10.10.10.101:80

Ensemble, ces deux sockets constituent une paire de sockets :

- 192.168.1.1:7151, 10.10.10.101:80

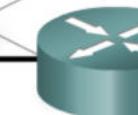
# Couche 4 : TCP/UDP

## Prise en charge de services multiples

Requête

|             |              |      |
|-------------|--------------|------|
| Source      | 192.168.1.1  | 7151 |
| Destination | 10.10.10.101 | 80   |

Source



Internet



Destination

Réponse

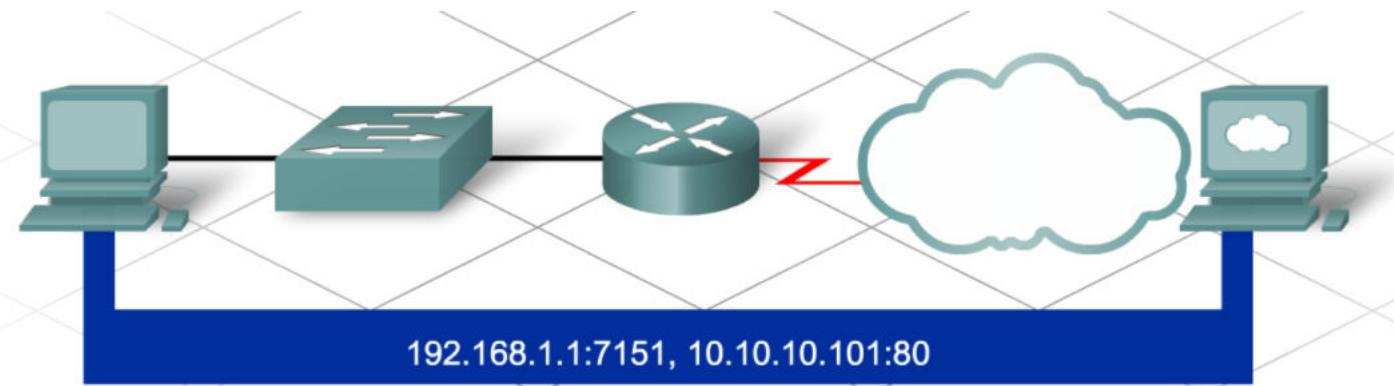
|             |              |      |
|-------------|--------------|------|
| Source      | 10.10.10.101 | 80   |
| Destination | 192.168.1.1  | 7151 |

**Paire de sockets :** 192.168.1.1:7151, 10.10.10.101:80

Conversation spécifique entre ces 2 hôtes et applications

# Couche 4 : TCP/UDP

## Prise en charge de services multiples



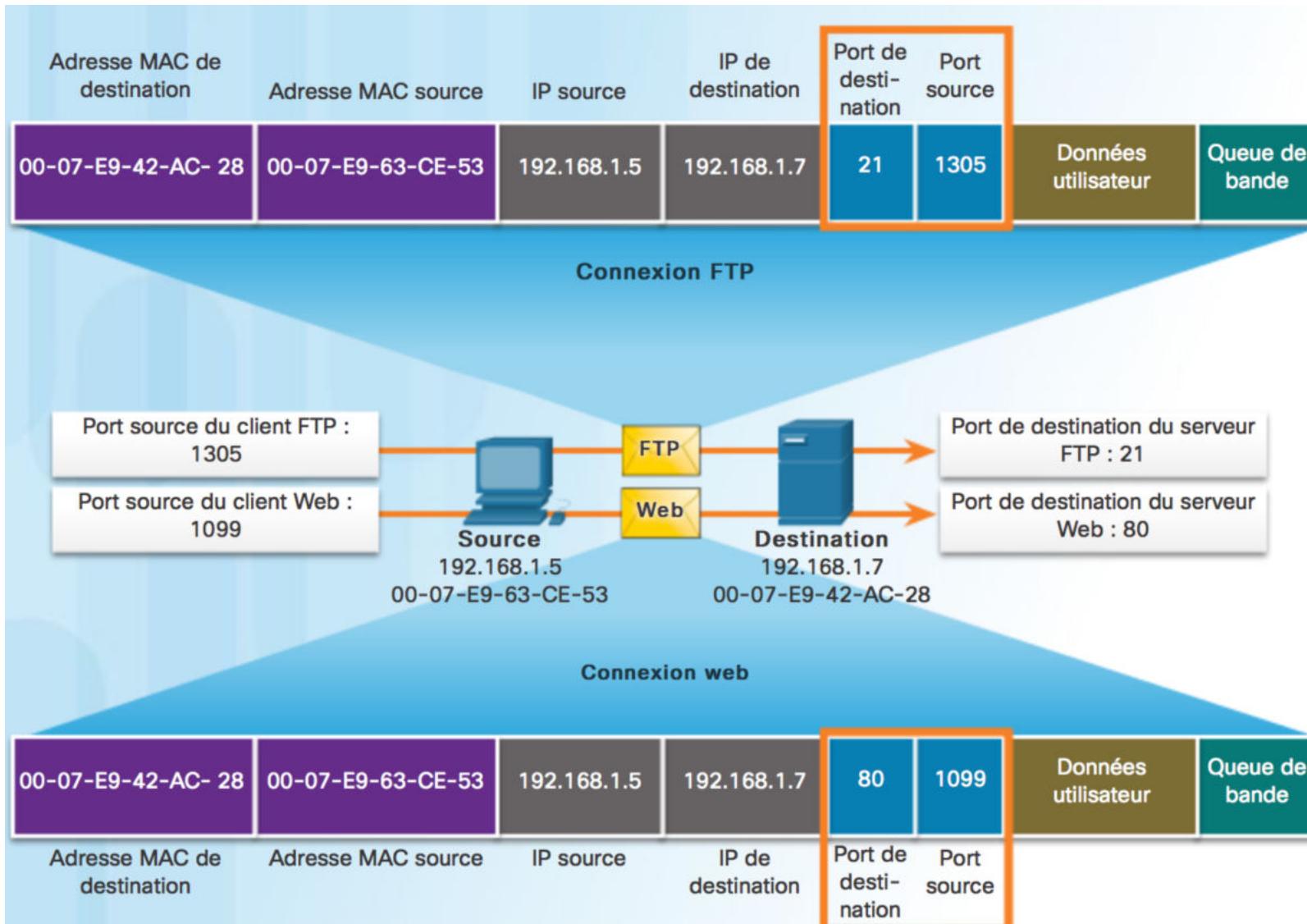
Une paire d'interfaces de connexion permet de relier l'hôte local au service de destination.

**sockets** = *points de communication* permettant le passage d'une application sur un hôte à une application sur un autre.

**sockets** = différentiation des *processus* exécutés sur un client et différentiation des multiples *connexions*

# Couche 4 : TCP/UDP

Prise en charge de services multiples : infos de couche 2, 3 et 4



# Couche 4 : TCP/UDP

## Commande netstat

```
C:\> netstat

Active Connections

Proto  Local Address    Foreign Address        State
TCP    kenpc:3126       192.168.0.2:netbios-ssn ESTABLISHED
TCP    kenpc:3158       207.138.126.152:http   ESTABLISHED
TCP    kenpc:3159       207.138.126.169:http   ESTABLISHED
TCP    kenpc:3160       207.138.126.169:http   ESTABLISHED
TCP    kenpc:3161       sc.msn.com:http        ESTABLISHED
TCP    kenpc:3166       www.cisco.com:http     ESTABLISHED

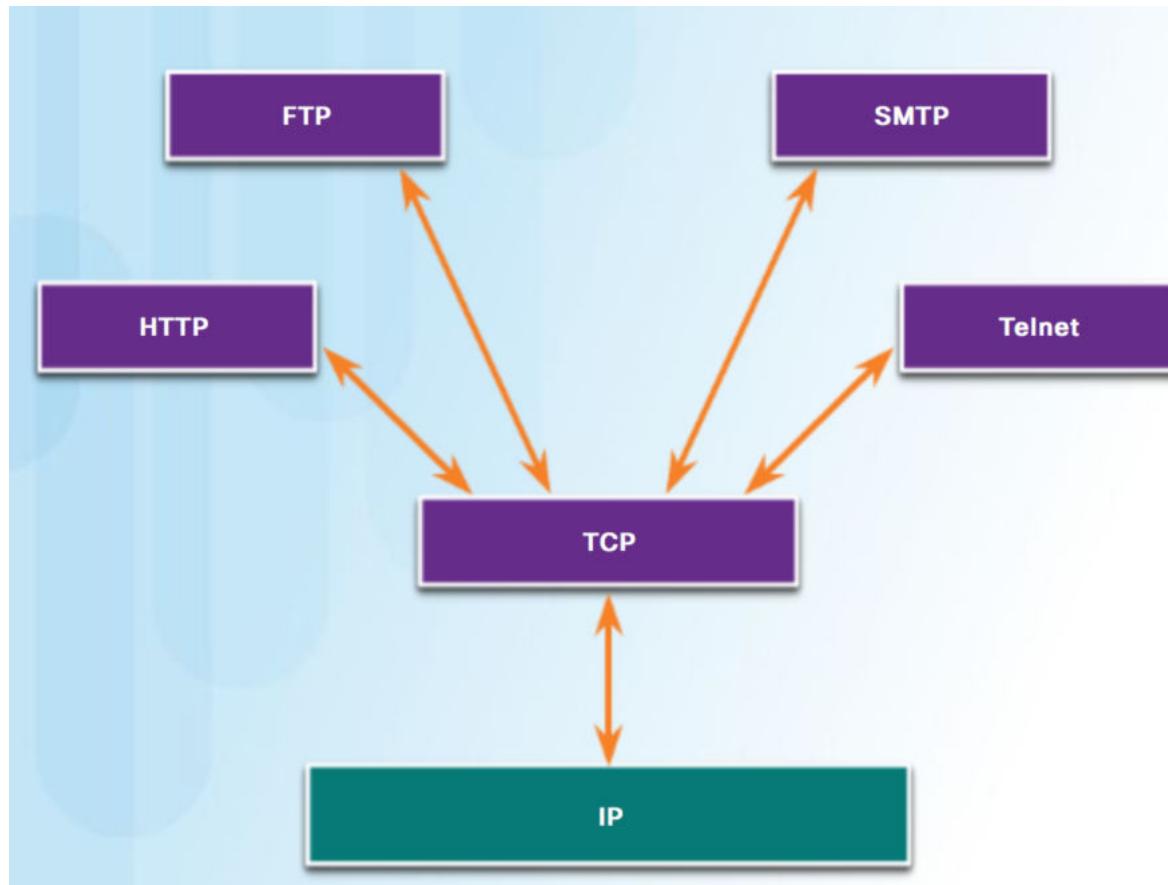
C:\>
```

! Attention aux connexions TCP inexplicées : risque de sécurité  
Pour savoir quelles connexions tcp sont ouvertes, utilitaire **netstat** répertorie :

- les protocoles utilisés,
- l'adresse et les numéros de port locaux,
- l'adresse et les numéros de port distants,
- l'état de la connexion.

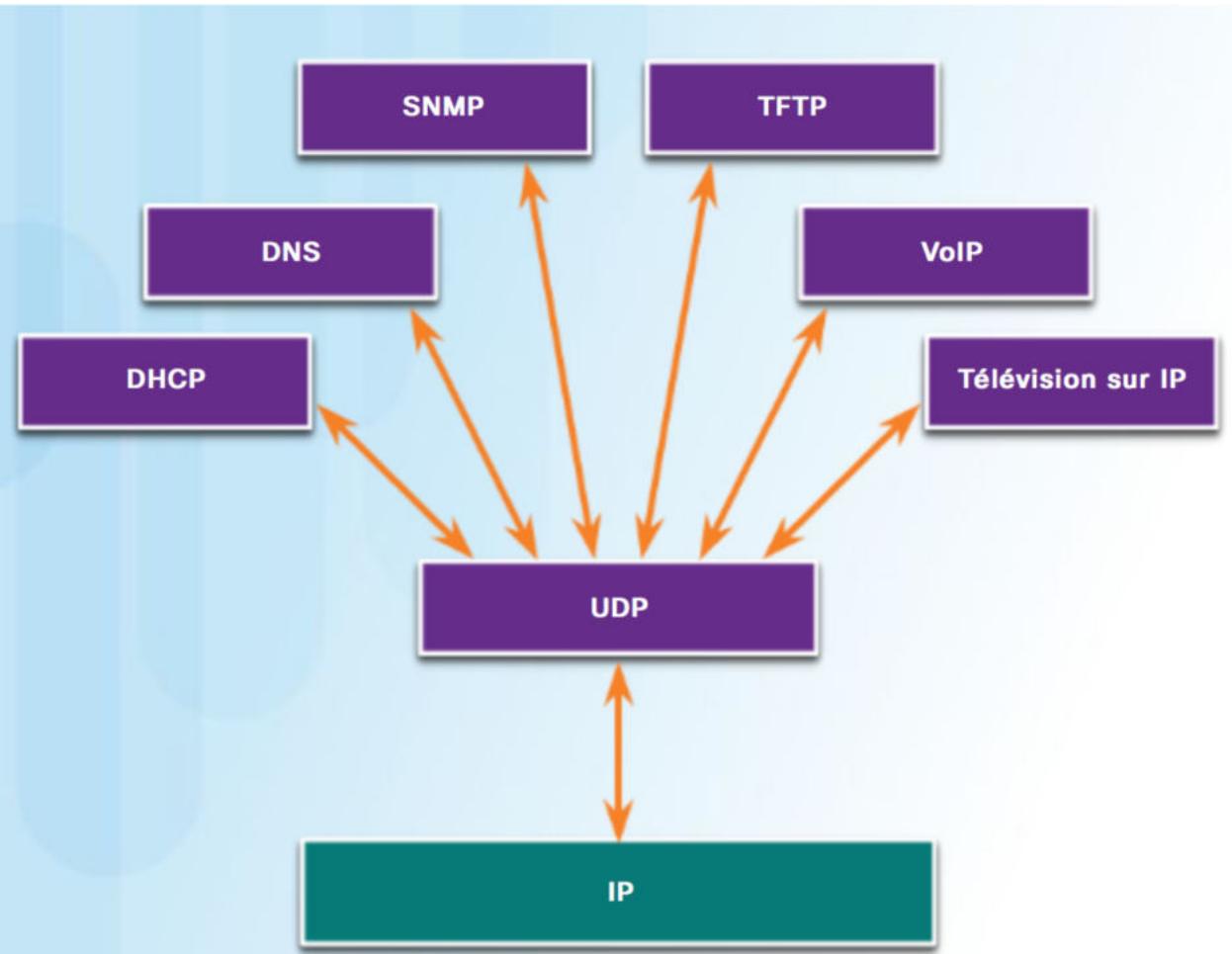
# Couche 4 : TCP/UDP

Résumé : principaux protocoles de couche application utilisant TCP en couche transport



# Couche 4 : TCP/UDP

Résumé : principaux protocoles de couche application utilisant UDP en couche transport



- Applications pouvant gérer des pertes mais pas de retard
- Simples applications de requêtes /réponses (DHCP)
- Applications gérant elles-même la fiabilité (SNMP, TFTP)

# NAT

## Définition et objectifs

- **NAT = Network Address Translation**
  - Technique de "translation" d'adresses IP
- **Objectifs**
  - Partition du réseau en différents sous-réseaux
  - Pallier le manque croissant d'adresses IPV4 libres
    - ✓ **Peu d'adresses** disponibles en comparaison du nombre croissant de machines sur Internet
    - ✓ **Décision** : réservé des intervalles d'adresses à des usages privés uniquement

| Classes réseau | Adresses réseau               |
|----------------|-------------------------------|
| Classe A       | 10.0.0.0 à 10.255.255.255     |
| Classe B       | 172.16.0.0 à 172.31.255.255   |
| Classe C       | 192.168.0.0 à 192.168.255.255 |

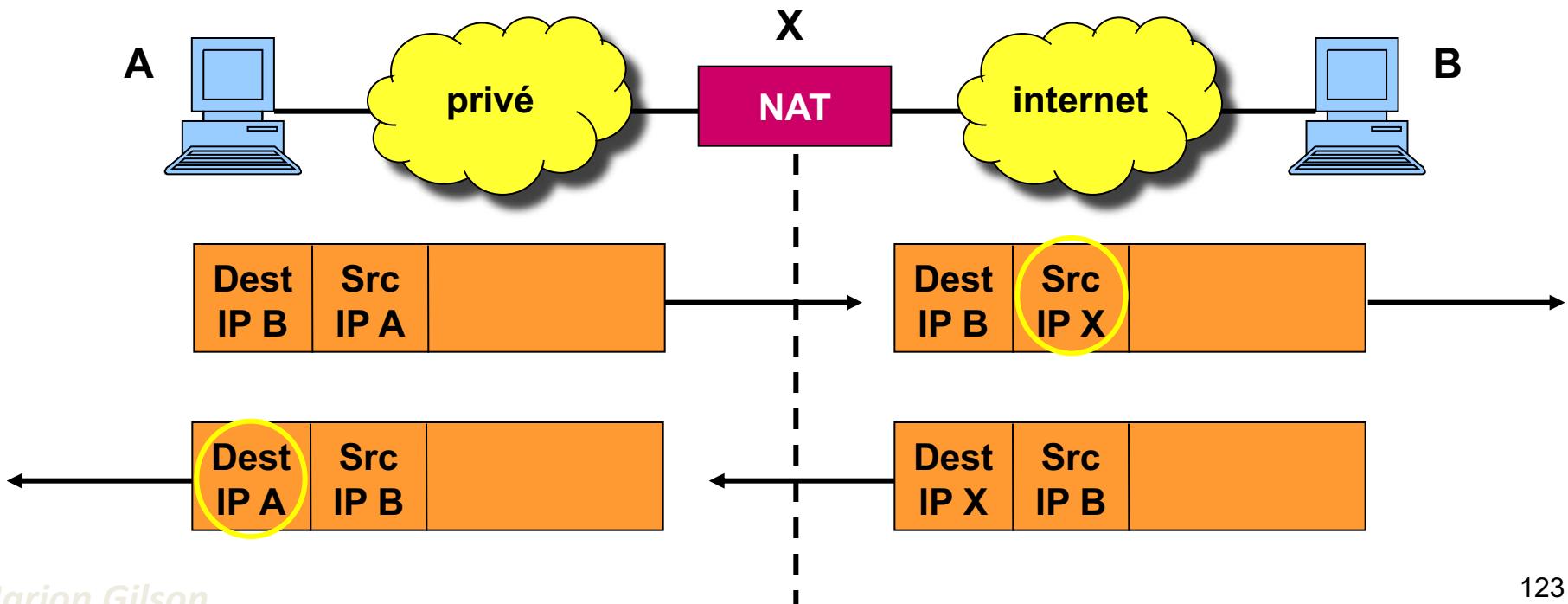
- ➔ **adresses non routables sur internet**
- ➔ **réservées aux réseaux privés**
- ➔ **Ex. salle TP : 192.168.1.0**

# NAT

## Définition et objectifs

### ▪ Principes

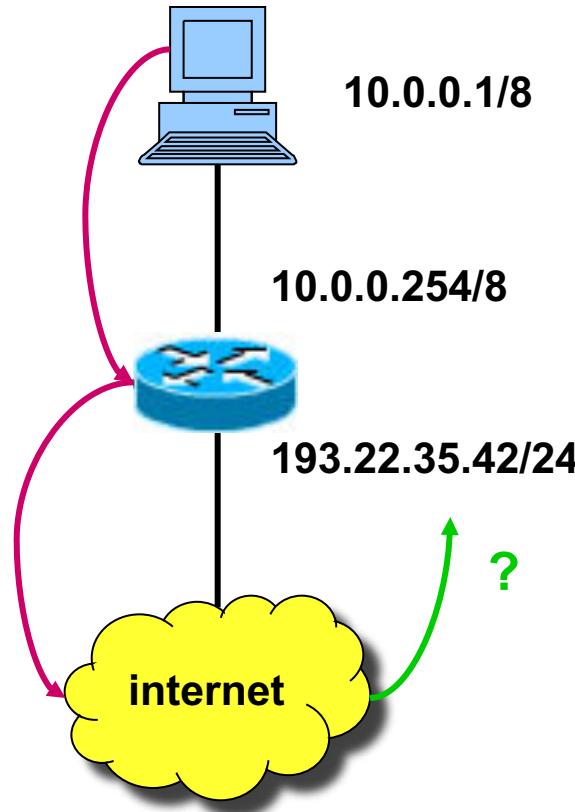
- Adresses privées : non routables donc **translation** d'adresses pour accéder à d'autres réseaux
- **Remplacement** des champs d'adresses des paquets à destination d'autres réseaux



# NAT

## Définition et objectifs

- Remarque : impossibilité de l'accès à Internet avec une adresse privée



1. PC1 : cherche à joindre www.google.fr  
→ ok, le message arrive à destination
2. www.google.fr cherche à répondre à PC1  
→ problème : 10.0.0.1 non routée sur internet  
→ impossibilité d'acheminer la réponse
3. Conclusion : aucune réponse reçue par le PC ayant une adresse privée



# NAT

---

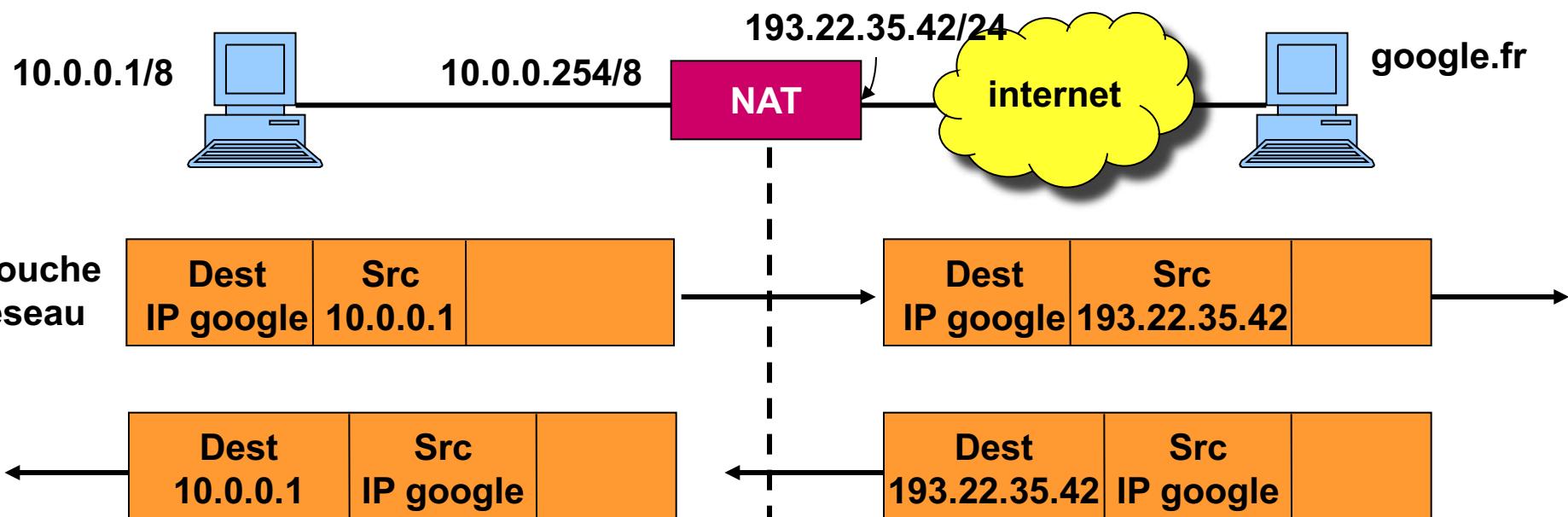
## Techniques de translation

- Choix entre plusieurs techniques suivant :
  - Topologie du réseau privé
  - Nombre de machines privées
  - Nombre d'adresses IP publiques disponibles
  - Besoins en terme de
    - ✓ Services
    - ✓ Visibilité depuis l'extérieur
    - ✓ Accessibilité depuis l'extérieur
- 2 grandes approches :
  - Translation statique
  - Translation dynamique

# NAT

## Techniques de translation : NAT statique

- **Principe :**
  - NAT de base
  - Attribution de façon automatique d'une adresse IP (privée) à une autre (publique)
  - Situation idéale : nbre adresses IP privées = nbre adresses IP publiques



# NAT

---

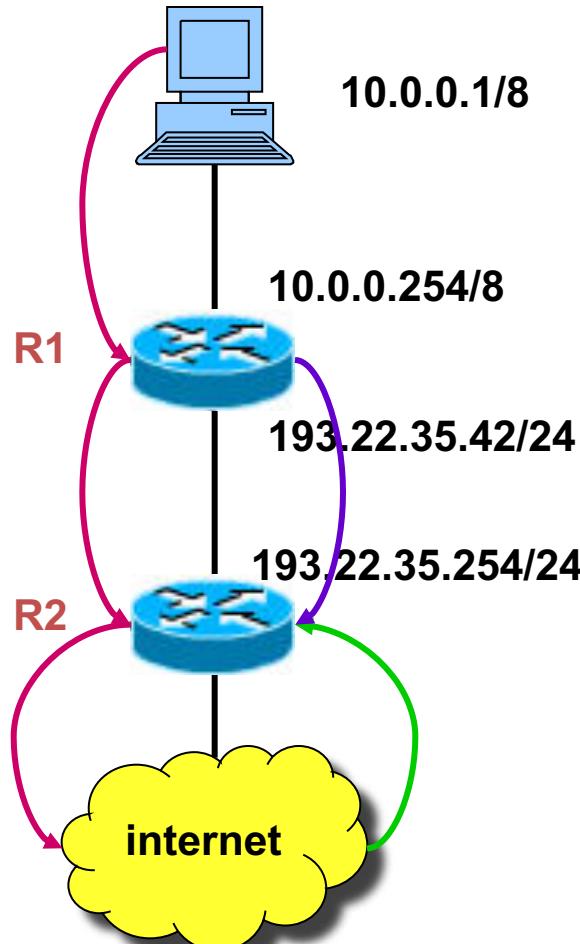
## Techniques de translation : NAT statique

- **Avantage :**
  - Rendre une machine d'un réseau privé accessible sur Internet
- **Inconvénient :**
  - Nécessite de disposer d'une adresse publique par machine accessible
- **Remarque :** pourquoi ne pas donner directement une adresse publique ?
  - Pour garder un adressage uniforme en interne
  - Administration (modifications, interventions, ajout, élimination de machine...) du réseau facilitée
- **Problème de la pénurie d'adresse IP non réglé**
  - NAT dynamique

# NAT

## Techniques de translation : NAT statique

- Problème de routage lié au NAT statique :
  - L'adresse 10.0.0.1 est remplacée par 193.22.35.43



1. PC : cherche à joindre www.google.fr
  - ok, le message arrive à destination
2. www.google.fr cherche à répondre à PC1
  - le message arrive à destination R2
  - Adresse IP 193.22.35.43 : dans son réseau
  - Envoi d'une **requête ARP** pour connaître l'adresse MAC du PC ayant pour IP 193.22.35.43
  - **Réponse** : aucune car adresse virtuelle
3. R1 doit répondre
  - **Proxy arp**

# NAT

---

## Techniques de translation : NAT statique

- **Problème de routage lié au NAT statique : solutions**
  - Mettre en place un **proxy arp** sur la machine NAT
  - Soit ajout d'une **entrée statique dans la table arp** du routeur internet (R2) :
    - ✓ `arp -s 193.22.35.43 @MAC_routeurR1` (sous windows)
  - Soit ajout d'une **route statique** pour chaque adresse virtuelle
    - ✓ `route add -p 193.22.35.43 mask 255.255.255.0 193.22.35.42` (sous windows)
- **Problème de routage lié au NAT statique : sur la passerelle (R1)**
  - R2 envoie le paquet à R1 qui reconnaît son **@MAC**
  - R1 envoie le paquet à la couche réseau mais **@IP** différente de la sienne
  - **Solution : ajout d'une route interne :**
    - ✓ `route add -p 193.22.35.43 mask 255.255.255.0 10.0.0.1` (sous windows)

# NAT

## Techniques de translation : NAT dynamique

- **Principe**

- Rôle du NAT : **associer  $m$  adresses privées à  $n$  adresses publiques**
- Généralement :  $m > n$  voire  $m \gg n$
- Intérêt : permettre à un grand nombre de machines une connexion internet à partir d'une seule IP publique
- **Inconvénient :**
  - ✓ NAT seul impossible
  - ✓ Ajout d'un **PAT (Port Address Translation)**



**NAT dynamique**  
=

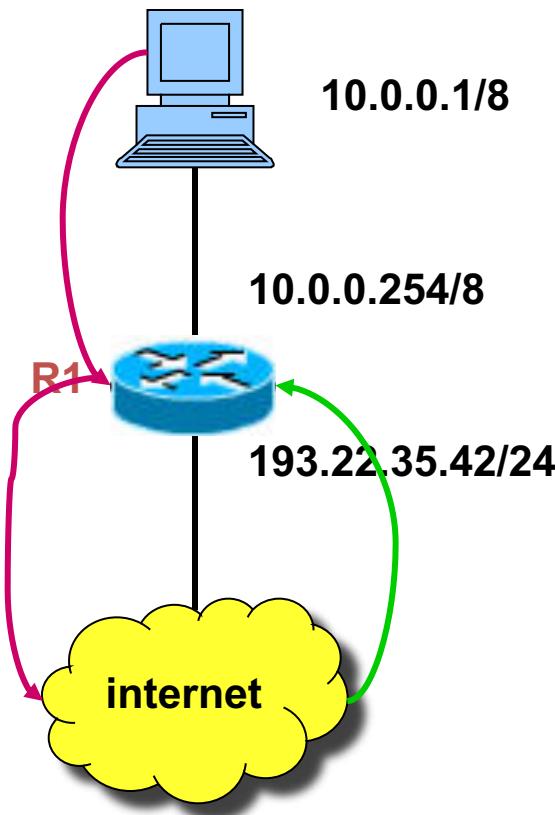
**modification d'adresses**  
+

**modification de numéro de port (TCP/UDP)**

# NAT

## Techniques de translation : NAT dynamique

- Principe (suite)



1. Soit  $n = 1 : 193.22.35.42$
2. Le PC envoie une requête à www.google.fr
3. La requête est reçue par www.google.fr qui répond
4. La réponse arrive à R1 qui reconnaît son @IP



**Question :** comment reconnaître un paquet qui lui est destiné d'un autre ?

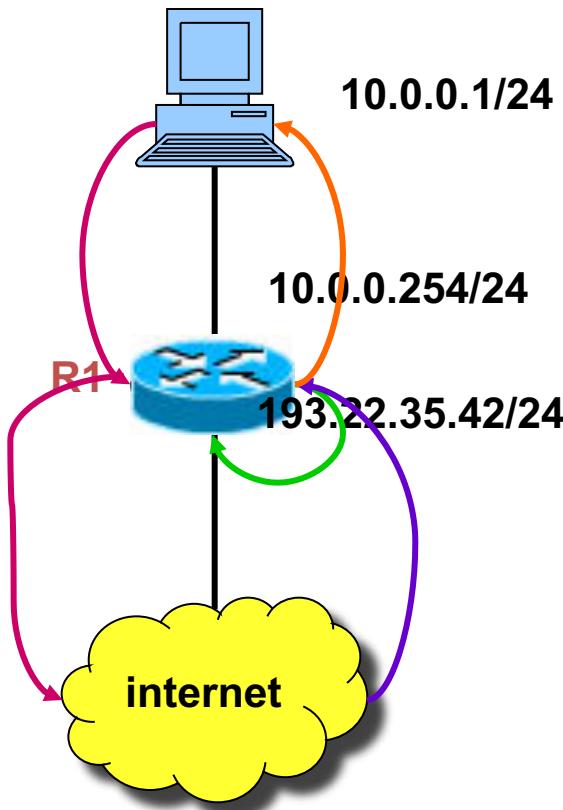


**Solution :** PAT

# NAT

## Techniques de translation : NAT dynamique

- Principe (suite)



1. Soit  $n = 1 : 193.22.35.42$
2. Le PC envoie une requête à www.google.fr
  - @IP src : 10.0.0.1, port src : 2048 (ex.)
3. R1 : NAT + PAT
  - @IP src : 193.22.35.42, port src : 10000 (ex.)
  - Informations conservées dans la table NAT
4. Le paquet modifié ("NATé" et "PATé") reçu par google qui le retourne à 193.22.35.42
5. Le paquet réponse arrive à R1
  - @IP dest : 193.22.35.42 (la sienne), traitement du paquet
  - Port dest : 10000, consultation de sa table NAT
  - Modification du paquet
    - @IP src : 10.0.0.1, port src : 2048
6. Envoi du paquet réponse modifié (déNATé et déPATé) au PC

# NAT

---

## Techniques de translation : NAT dynamique

### ▪ Résumé

- NAT dynamique : pour permettre à des machines ayant des adresses IP privées d'accéder à Internet
- MAIS : impossibilité de joindre depuis internet une machine du réseau privé (contrairement au NAT statique)

### ▪ Avantages

- Partage d'un accès internet : m machines peuvent être atteintes, cachées derrière 1 seule IP publique
  - ✓ "Résolution" du problème de pénuries d'adresse IP
- Machines non accessibles depuis l'extérieur : intéressant au niveau sécurité

### ▪ Inconvénient :

- Impossible de joindre une machine du réseau privé : pas de serveur accessible

# NAT

---

## Techniques de translation : NAT dynamique

- Problème lié au NAT dynamique et PAT
  - Système fondé sur l'utilisation de n° de **ports** (couche transport)
  - Tous les protocoles n'utilisent pas de n° de port,
    - ✓ Ex. : ICMP (commande ping par exemple),
    - ✓ Netbios...
  - **Solution** : se fonder sur l'identifiant présent dans l'entête du message

# NAT

## Techniques de translation : NAT statique ou dynamique ?

- Choix du NAT statique

- Pour rendre une ***application disponible*** sur internet
    - ✓ Ex. : serveur web, email, ftp ...

- Choix du NAT dynamique

- ***Économiser*** des adresses IP
  - ***Donner accès à Internet à des machines non joignables de l'extérieur***
  - ***Sécurité*** accrue (mais non suffisante ! ⇒ filtrage nécessaire)
  - Choix classique pour un client réseau

- Combiner les deux techniques

- ***Meilleure solution***
    - ✓ NAT dynamique pour les clients
    - ✓ NAT statique pour les serveurs

# NAT

## Remarque : NAT dynamique

- Possibilité de contourner le problème de la non accessibilité d'une machine dans le réseau privé :

- **Port forwarding :**

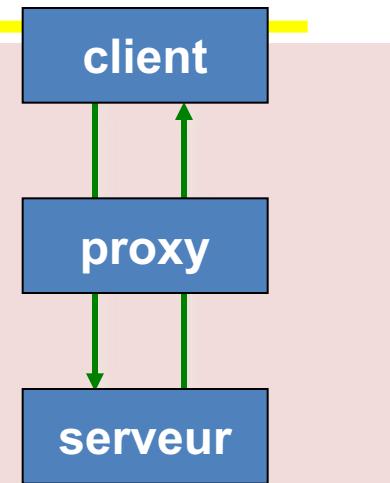
- Rediriger un paquet vers une machine précise en fonction du port de destination
    - Ex. : serveur ftp
      - ✓ Machine : 10.0.0.1 = serveur ftp, config NAT dynamique
      - ✓ Configuration de R1 pour rediriger les connexions sur port 21 vers la machine 10.0.0.1
      - ✓ La machine 10.0.0.1 devient accessible depuis l'extérieur pour une application donnée
    - Limite du port forwarding :
      - ✓ S'il existe plusieurs serveurs ftp en local ... ??
      - ✓ Autre astuce à trouver

- **Port mapping :** rediriger la requête sur un port différent de celui demandé

# NAT

## Remarque : proxy

- définition
  - *Mandataire pour une application (protocole) donnée*
    - ✓ Intermédiaire entre le client et le serveur
    - ✓ Un proxy dédié à UNE application donnée :  
ftp, http, smtp ...
    - ✓ Possibilité de *modifications des informations* circulant entre le client et le serveur (aller et retour)
    - ✓ Un proxy par application (sinon : multi-proxy)
- Fonctionnalités supplémentaires
  - *Fonctions de cache*
    - ✓ Proxy centralisant l'accès au *web* : si x personnes envoient une requête sur le même site, un seul chargement puis conservation en cache
  - *NAT* :
    - » Modification de l'adresse source du paquet pour que la réponse du serveur passe par lui



# Couche 4 : TCP/UDP

---

# Couche 4 : TCP/UDP

---