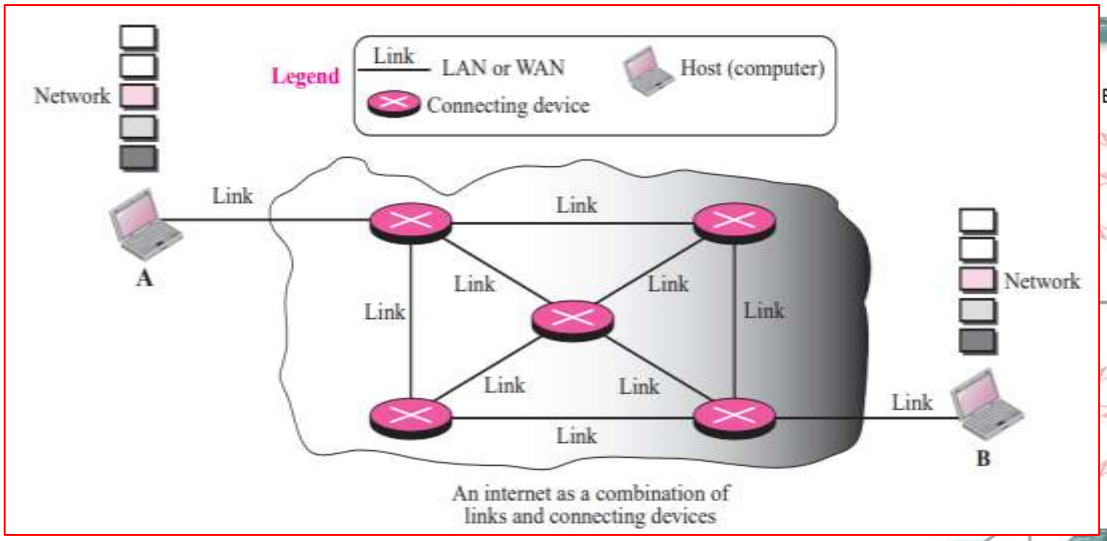
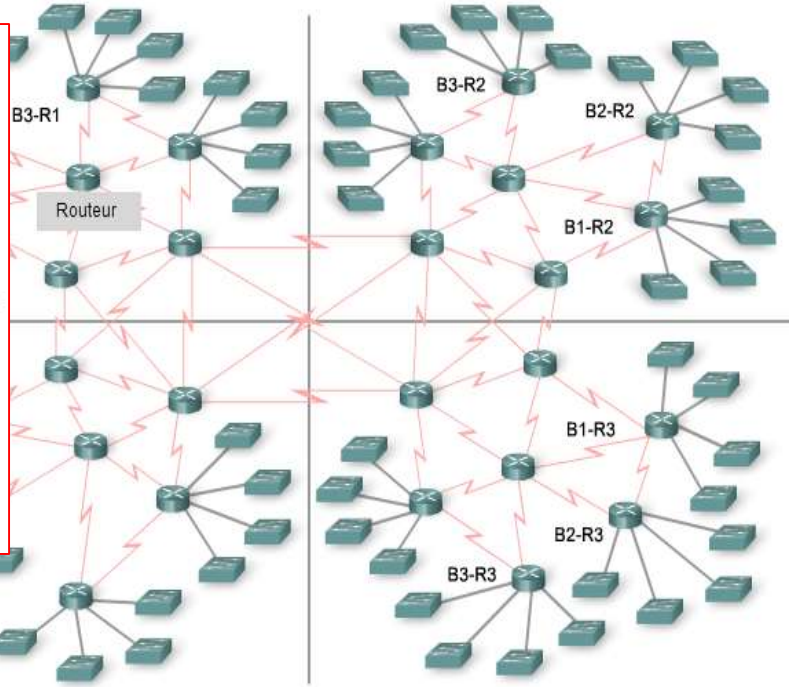
Mr A.Dekhinet
Université BATNA 2 / Département Informatique
a.dekhinet@univ-batna2.dz
http://staff.univ-batna2.dz/dekhinet-abdelhamid

Chapitre 2 : Couche Réseau

Introduction

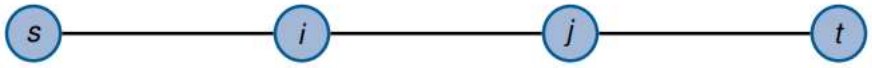
- La couche réseau (Network or Internetwork Layer) a été conçu pour résoudre le problème de livraison à travers plusieurs liens (Chemin).
- La couche réseau est responsable de l'acheminement et du routage des paquets via les routeurs.
- Le routage consiste à trouver le chemin ou la route optimale, à emprunter par un paquet.
- Les critères d'optimalité sont diverses : Distance, Débit, Surcharge, …
- Le routeur agit comme un commutateur, quand un paquet arrive sur un port (Interface) d'entrée il sera dirigé ou transféré (Forwarding), via un autre port de sortie, à l'équipement suivant : Un autre routeur ou la destination finale.
- En d'autres termes, un processus de ***commutation*** se produit au niveau du routeur.

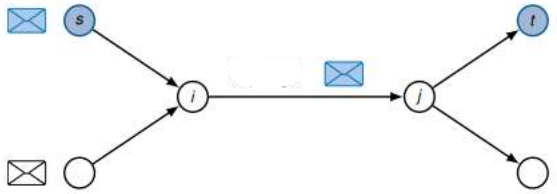


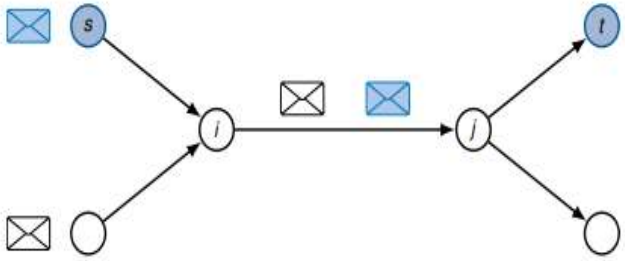


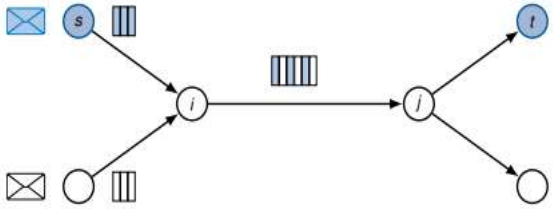
Commutation (Switching) : Trois types de commutation

- **Commutation de circuit** : Un chemin (Circuit) physique dédié, de bout en bout, est établit entre la source et la destination avant la transmission. Le chemin est maintenu pendant toute la durée de la communication. Par exemple les premiers systèmes téléphonique analogiques.
- **Commutation de message** : Les messages sont transmis intégralement de nœud en nœud, sans qu'il y ait une limite de leur taille. Chaque nœud est un commutateur de message: mémoriser – expédier (store and forward). Par exemple courrier électronique X.400
- **Commutation de paquet** : Le message est décomposé en paquets dont la taille dépend principalement des caractéristiques des technologies réseaux sous-jacentes. Les paquets constituent alors l'unité de communication et sont acheminés `a travers les nœuds vers la destination. Par exemple IP, X.25, ATM.

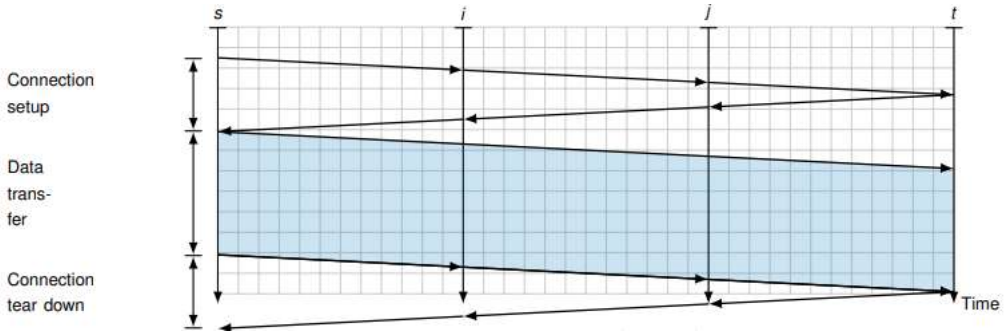


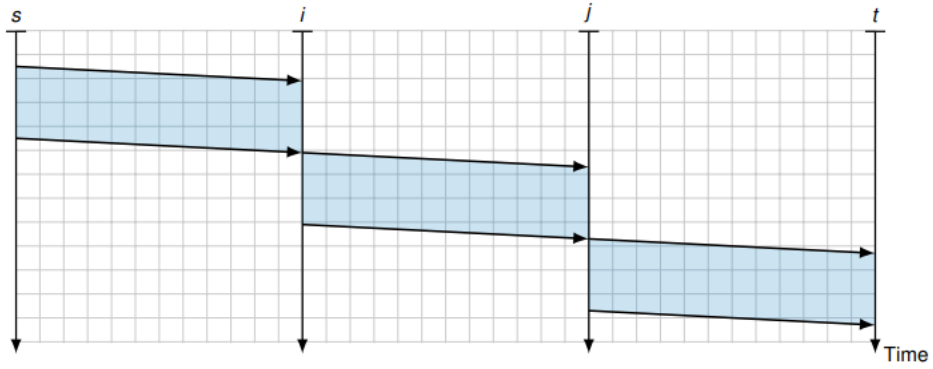


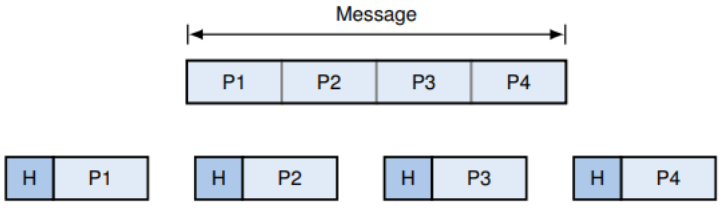


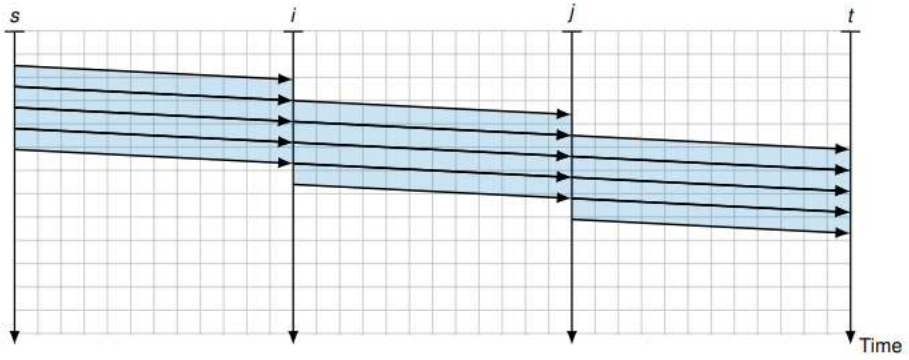


Commutation (Switching) : Trois types de commutation



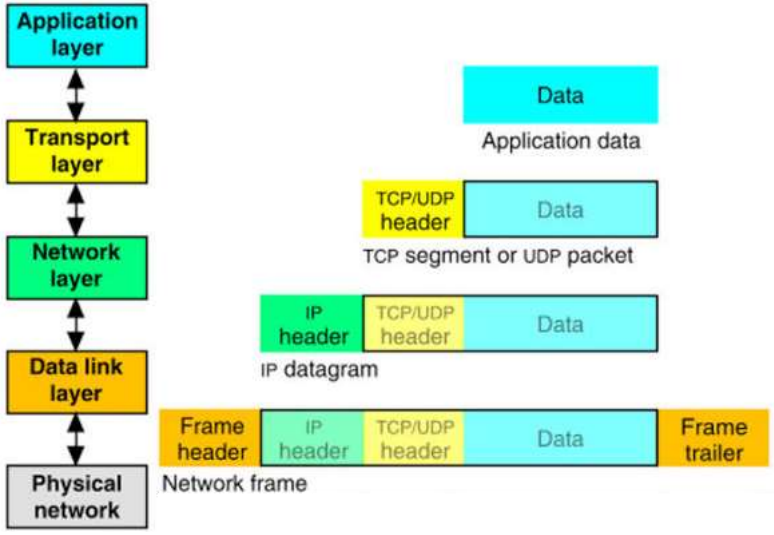


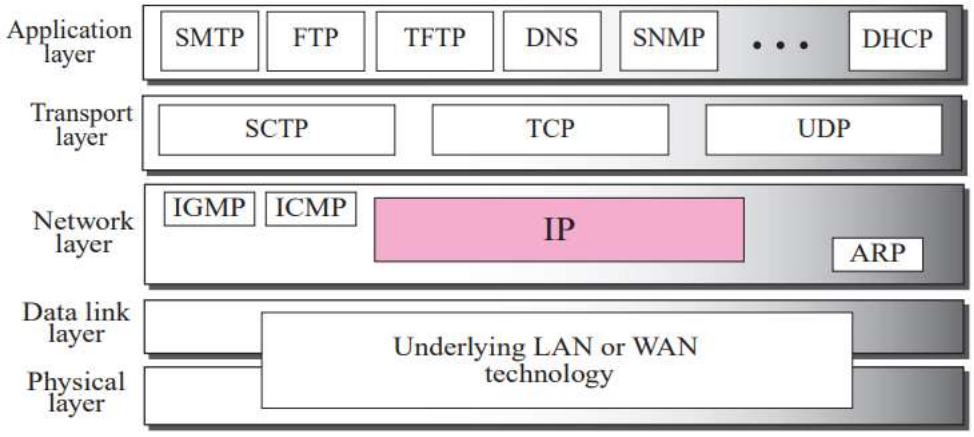




IPv4 : Internet Protocol Version 4

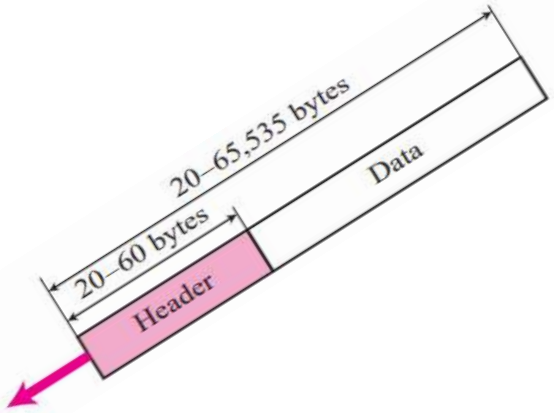
- IP est un protocole non fiable, au mieux et sans connexion.
- Les PDU est appelé Datagramme.
- Best-effort service : IP fait de son mieux pour garantir la livraison des paquets.
- En d'autres termes les paquets peuvent être corrompus, perdus, arrivent désordonnés, retardés et peuvent créer une congestion dans le réseau.
- Si la fiabilité est requise, elle doit être assurée par les couches supérieures, c'est le cas de TCP.





Datagramme

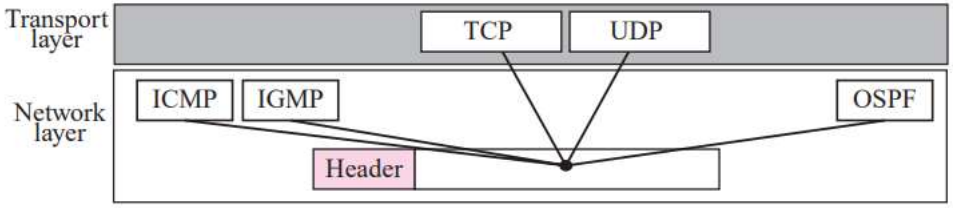
- Le Datagramme est un paquet de taille variable, composé de : Entête (Header) et Donnée (Payload)
- L'entête, dont la taille varie de 20 à –à 60 bytes, contient les informations essentielles pour le routage et la livraison.



Offset	0 1 2 3	4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0 B	Version	IHL	TOS	Total Length
4 B	Identification			Flags / Fragment Offset
8 B	TTL		Protocol	Header Checksum
12 B	Source Address			
16 B	Destination Address			
20 B	Options / Padding (optional)			

Datagramme

- **Version** (4 bit) : Version IP, 4 ou 6.
- **IHL** (IP header length, 4 bit) : Définit la taille ou la longueur totale de l'entête, exprimé en mot de 32 bit (4 bytes). Lorsqu'il n'y a pas d'options, la longueur de l'entête est de 20 octets et la valeur de ce champ est de 5 (5x4=20). Lorsque le champ d'option est à sa taille maximale, la valeur de ce champ est 15 (15x4=60).
- **TOS** (Type Of Service, 4 bit) : Définit la qualité de service, la priorité, délai, fiabilité, …
- **Total Length** (16 bit) : Définit la longueur totale du datagramme (Entete + Donnée), exprimée en byte. La longueur totale maximale est de 65535 (2^{16}-1) bytes.
- **Identification** (16 bit), **Flags** (3 bit), **Fragment Offset** (13 bit) : Ces champs sont utilisés dans la fragmentation de paquet.
- **TTL** (Time To Live, 8 bit) : Définit la durée de vie maximale du datagramme. Elle est décrémentée à chaque passage d'un routeur. Le paquet est détruit lorsque TTL = 0.
- **Protocol** (16 bit) : Spécifie le type de protocole de haut niveau, dont le PDU est encapsulé dans le datagramme : TCP, UDP, ICMP, IGMP, OSPF, …



Value	*Protocol*	*Value*	*Protocol*
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

Datagramme

- **Header Checksum** (16 bit) : Le checksum sert pour le contrôle d'érreurs et la validité du paquet.

  - En émission : Calcul — Chesksum = Complément-a-1 (∑ 16-Bit-Word)
  - En réception : Calcul — Somme = ∑ 16-Bit-Word
    Si Complément-a-1(Somme) = 0 Alors Pas d'erreurs
    Sinon Il y'à des erreurs

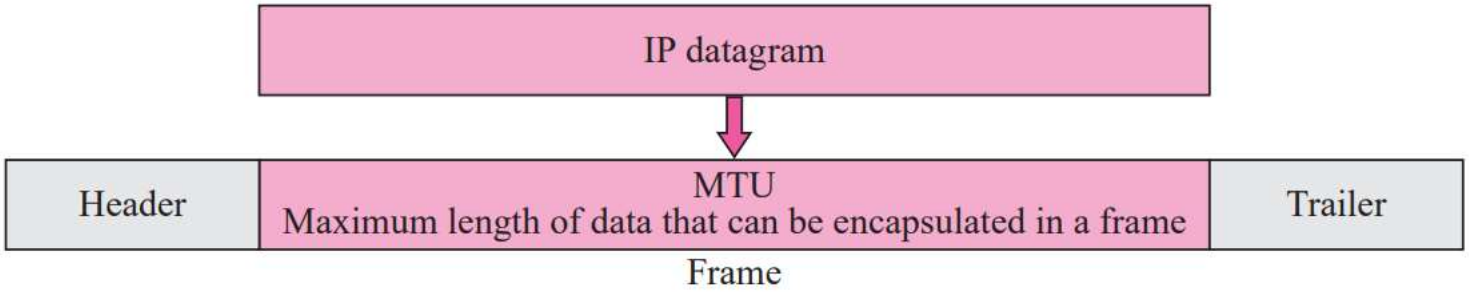
  - Exemple simple : Données à trasmettre 0110 et 1000
    Somme= 0110 + 1000 =1110
    Checksum = Complément-a-1(Somme)=0001

  Emission : 0110 1000 0001
  Réception (Sans erreur) : 0110 1000 0001
  Calcul Somme = 0110+1000+0001 = 1111
  Complément-a-1(Somme) = 0000 Alors Pas d'erreurs

- **Source Address** (32 bit) : Adresse IP source
- **Destination Addess** (32 bit) : Adresse IP destination
- **Options** : Un champ facultatif de l'entête, Il contient des paramètres liés à la sécurité, à l'itinéraire, à l'horodatage, au Débogage, etc

Datagramme : Fragmentation

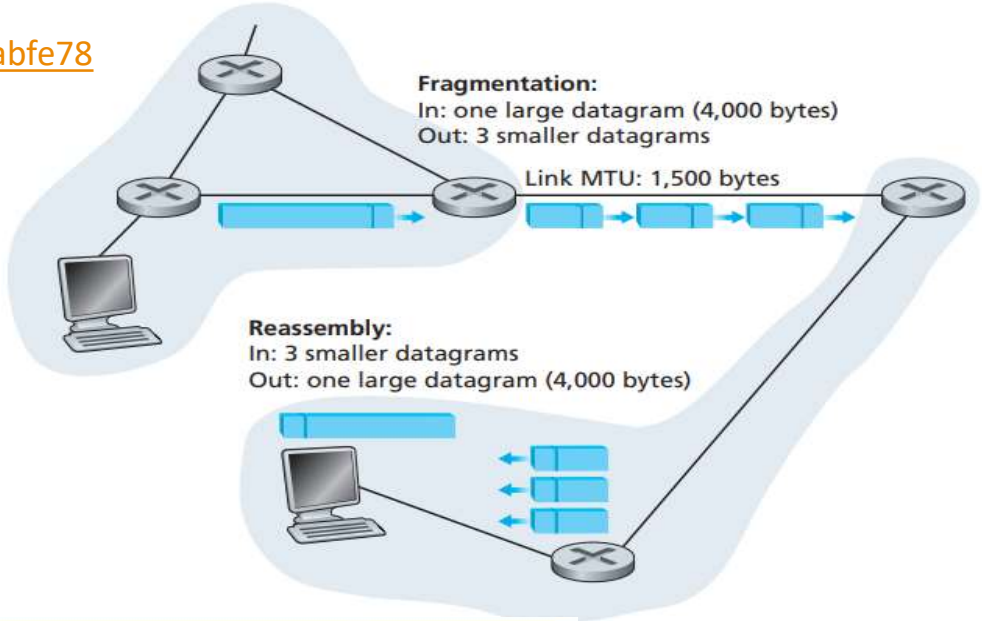
- Un datagramme, durant son acheminement, peut traverser plusieurs types de réseaux physiques (LAN, WAN, …).
- Chaque routeur décapsule la trame qu'il reçoit, la traite puis il l'encapsule dans une autre trame pour la réémettre.
- Le format et la taille de la trame dépend du protocole et de la technologie réseau sous-jacente.
- Une trame ne peut emporter qu'une quantité maximale de données. Cette quantité est appelée unité de transmission maximale (MTU : Maximum Transfert Unit).
- Chaque technologie réseau possède son propre MTU. Par exemple, MTU(Ethernet) = 1500 bytes, MTU(FDDI) = 4352 bytes, ...
- Dépendant du réseau traversé, un datagramme durant son acheminement peut être fragmenté en datagrammes de taille réduite, appelé **Fragment**.
- L'assemblage des fragments se fait dans la destination finale.



Datagramme : Fragmentation

- **Exemple** : Considérons un datagramme de 4000 bytes (20 bytes entête plus 3980 bytes payload) arrive dans un routeur et doit être transféré à un lien Ethernet (MTU=1500 bytes). Cela implique que les 3980 data bytes du datagramme original doivent être alloués à trois fragments séparés (Chacun des trois est aussi un datagramme IP). On ssuppose que le datagramme original porte le numéro d'identification 777.

https://www.cloudshark.org/captures/1c53fcabfe78



Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

Datagramme : Fragmentation

- Les champs dédiés à la fragmentation sont : **Identification**, **Flags**, **Fragment Offset**

  - **Identification** (16 bit) : Il identifie, à la source, le datagramme. Lorsqu'un datagramme est fragmenté, la valeur de d'identification est copiée dans tous les fragments. En d'autres termes, les fragments d'un même datagramme porte le même numéro d'identification, qui est celui du datagramme original. Le numéro d'identification aide la destination à réassembler le datagramme. Il sait que tous les fragments ayant la même valeur d'identification doivent être rassemblés en un seul datagramme.

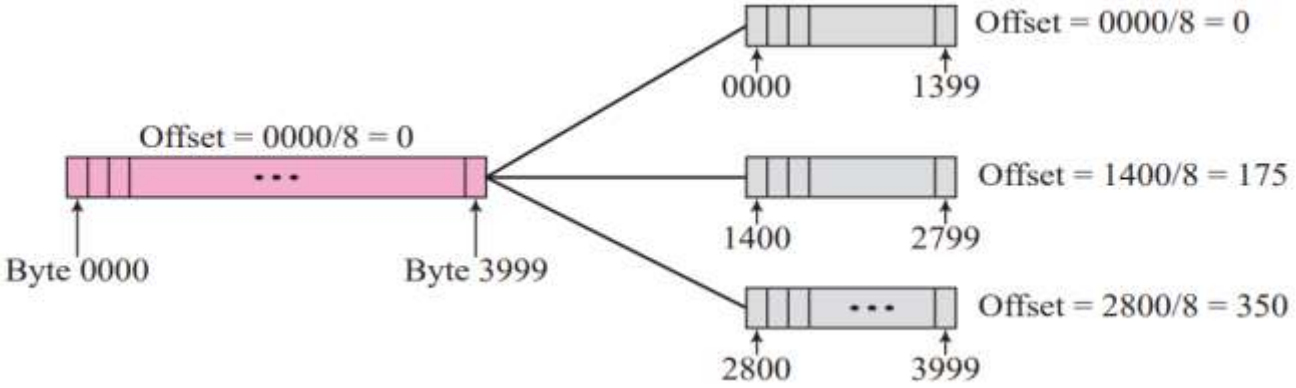
  - **Flags** (3 bit) :

    Bit 0 (Not Used, Reserved) : 0
    Bit 1 (D : Do not fragment bit) : 0 – may fragment 1- Don´t fragment
    Bit 2 (M : More fragment bit) : 0 – last fragment 1- More fragments

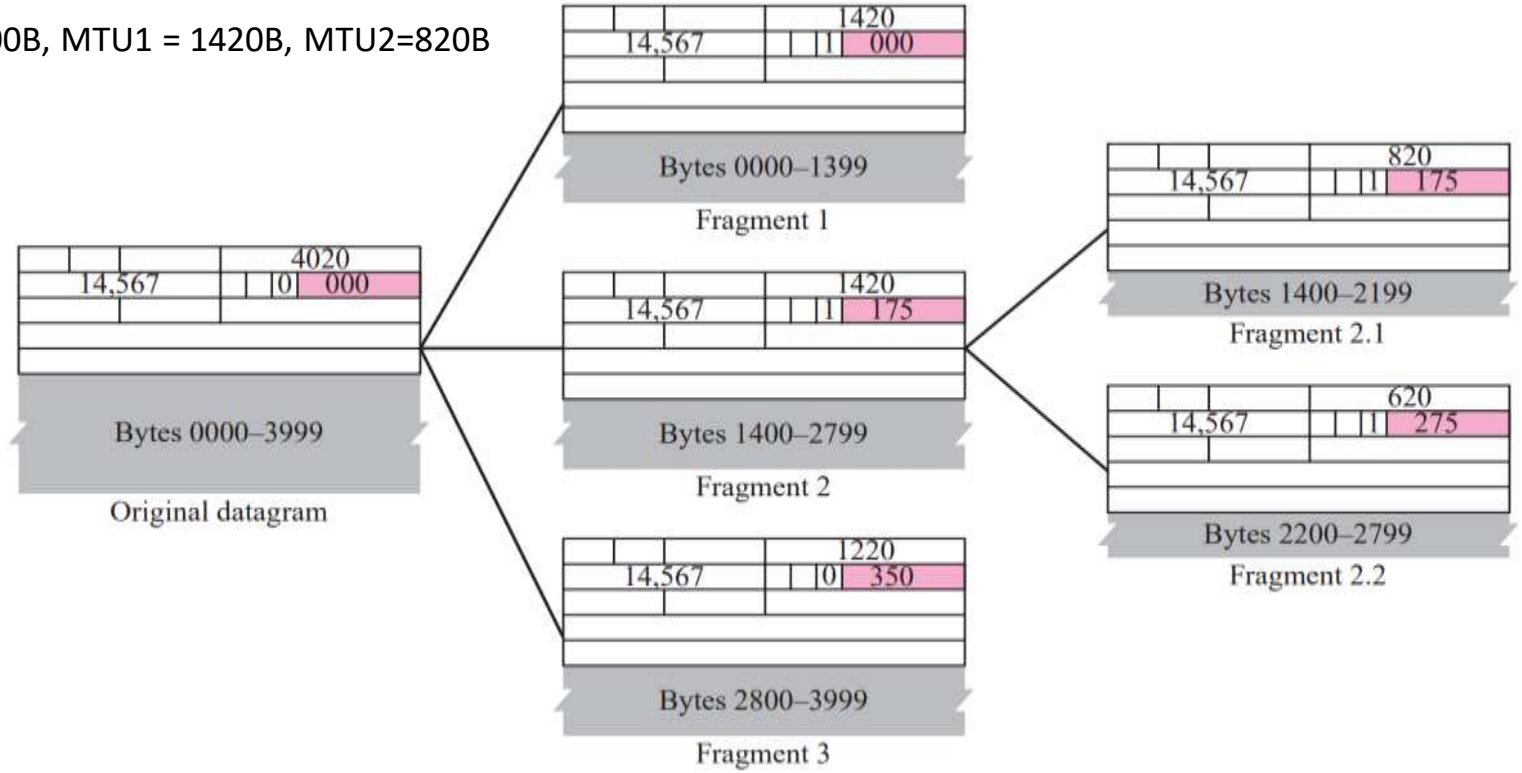
  - **Fragment Offset** (13 bit) : Il spécifie la position relative de ce fragment dans le datagramme original. C'est le déplacement par rapport à l'origine du datagramme mesuré en unités de 8 bytes.

Datagramme : Fragmentation

Data = 4000B

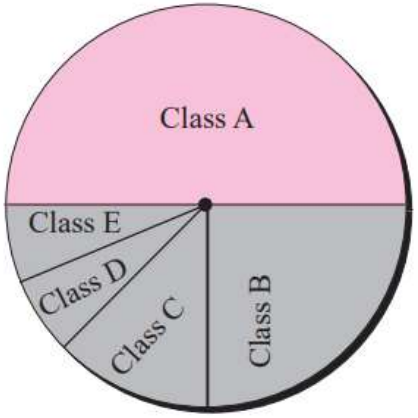


Data = 4000B, MTU1 = 1420B, MTU2=820B



Adressage IPv4

- La couche IP dans le modèle TCP/IP utilise des adresses logiques pour identifier les hôtes (Devices) connectés dans un réseau,
- Une adresse IP se présente sous forme d'une notation décimale pointée sur 32 Bits,
- Au début, les adresses IP utilisaient le concept de classes. Cette architecture est appelée adressage par classe : Classful. Au milieu des années 1990, une nouvelle architecture, appelée adressage sans classe : Classless, a été introduite pour remplacer l'architecture d'origine.
- Classful : Classification des réseaux, selon le type, en cinq classes :



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

- Classless : Pas de classification

  CIDR : SuperNeting

  VLSM : SubNeting

Adressage IPv4

Adresse IP : **32** Bit, **4** Octets ou blocs séparés par des points (Notation Décimale à Point), Contient **2** parties : Réseau et Hôte (Machine)
Masque : Délimite la partie réseau de la partie hôte
Adresse réseau : Partie hôte **0**
Adresse de Diffusion : Partie hôte 1
Nombre d'hôte par réseau = 2^n -2 ; **n**= nombre de bits de la partie hôte, **-2** = Adresse réseau et Adresse de diffusion
Préfixe : noté **/n** , **n** représente le nombre de bit de la partie réseau

Adressage par Classe (Classful)

Classe A 	**Partie réseau** = 8 bits, **Partie hôte** = 24 bits **Masque par défaut** = 255.0.0.0 (/8) **Premier MSB bit** = 0 **Nombre de réseau possible** = 2^{8-1} **Nombre de hôte par réseau** = 2^{24}-2 **Intervalle des réseaux** (Premier octet) : 1-127
Classe B 	**Partie réseau** = 16 bits, **Partie hôte** = 16 bits **Masque par défaut** = 255.255.0.0 (/16) **Premiers deux MSB bit** = 10 **Nombre de réseau possible** = 2^{16-2} **Nombre de hôte par réseau** = 2^{16}-2 **Intervalle des réseaux** : 128-191
Classe C 	**Partie réseau** = 24 bits, **Partie hôte** = 8 bits **Masque par défaut** = 255.255.255.0 (/24) **Premiers trois MSB bit** = 110 **Nombre de réseau possible** = 2^{24-3} **Nombre de hôte par réseau** = 2^{8}-2 **Intervalle des réseaux** : 192-223
Classe D 	Adresses Multidiffusion (Multicast, Multidestinataire) **Intervalle des réseaux** : 224-239
Classe E 	Adresses expérimentales (Usage future) **Intervalle des réseaux** : 240-255

Adressage sans classe (Classless)

Pas de classification, le masque est variable (Préfixe variable), le préfixe **n ≤ 30**

Typologie des adresses

Adresse de bouclage (Loopback): Adresses réservées servent pour le test local de la configuration du protocole TCP/IP
- La plage 127.0.0.0/8
- Généralement on utilise 127.0.0.1

Adresse locale-lien (Link-local) : Adresses réservées attribuée à l'hôte par le système d'exploitation, en cas d'indisponibilité d'adresse IP
- La plage 169.254.0.0/16
- Nommée adresses APIPA

Adresse TEST-NET : Adresses réservées inaccessibles depuis Internet, servent à des fins pédagogiques (Enseignements)
- 192.0.2.0 /24

Adresse privées : Adresses non routables et inaccessibles (invisible) dans Internet
- 10.0.0.0 à 10.255.255.255 (10.0.0.0 /8)
- 172.16.0.0 à 172.31.255.255 (172.16.0.0 /12)
- 192.168.0.0 à 192.168.255.255 (192.168.0.0 /16)

Adresse publique : Toute adresse différente des adresses mentionnées ci-dessus. Ce sont des adresses routables et accessibles (visible) depuis Internet.

Quelques Concepts de base

NAT (Network Adrress Translation) : Mécanisme de correspondances d'adresses IP (1 ↔ 1) . A titre d'exemple, il permet de rendre une adresse privé visible et accessible depuis Internet. Il établit une correspondance entre adresse privée et adresse publique.

PAT (Port Adrress Translation) : NAT généralisée, il permet de faire correspondre plusieurs adresses IP à d'autres (n ↔ m, n<<m). A titre d'exemple il peut être utilisé dans le partage de connexion (m=1).

Sous-Réseau : Réseaux crées à partir d'une adresse réseau de base. On emprunte quelques bits, en fonction du nombre de machines, de la partie hôte. Si on emprunte **n** bits, on peut créer 2^n sous-réseaux.

VLSM : Plusieurs masques de sous réseaux pour un même réseau de base (Granularité, Subneting). Le nombre de bits du sous réseau dépend de son nombre de machines.

CIDR : Mécanisme d'adressage et de routage basé sur le classless (Résumé et agrégation d'adresses et de routes, Superneting). CIDR peut être vu comme le VLSM appliqué à l'Internet.

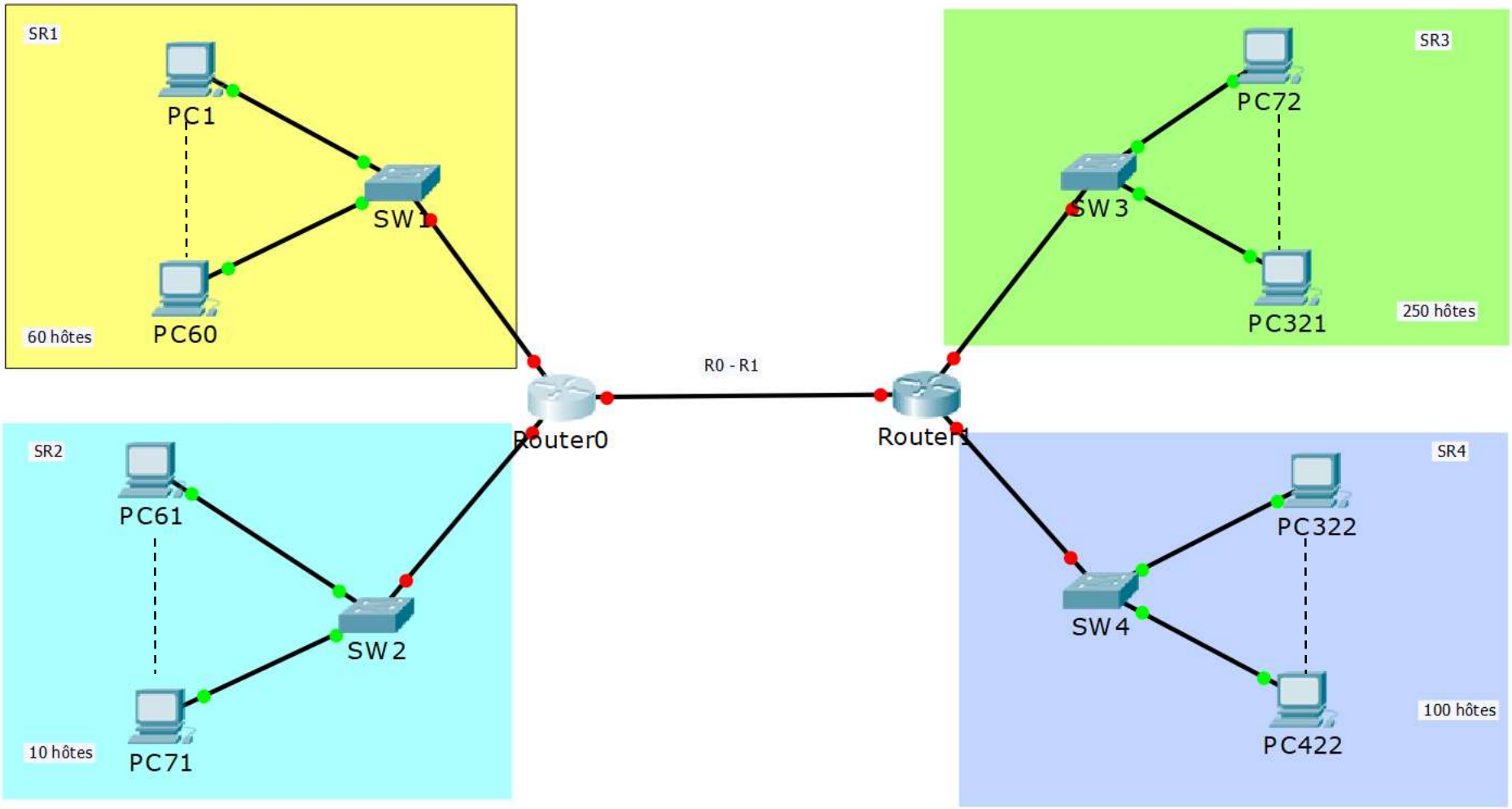
Réseaux différent : Réseaux ayant des adresses réseaux différentes.

Gateway (Passerelle, Routeur) : Equipement permettant d'interconnecter des réseaux différents.

Adressage IPv4 : VLSM

- VLSM : Variable Length Subnet Mask
- Pas de classification : Plusieurs masque de sous-réseau dans le même réseau,
- Permet une utilisation plus efficace de l'espace d'adresses IP : Réduction du gaspillage d'adresses,
- Permet aussi l'agrégation ou le résumé de routes.
- **Exemple** : L'adresse IP 193.194.4.0/22 vous a été attribuée par FAI (ISP), en tant qu'administrateur réseau, pour adresser votre réseau d'entreprise. La technique VLSM a été adopté, pour rationaliser et gérer efficacement l'espace d'adressage.

Adressage IPv4 : VLSM

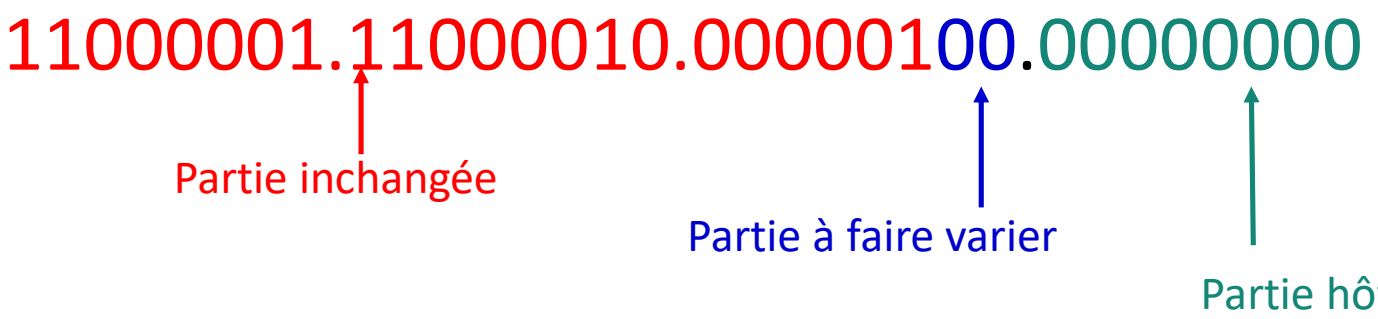


Adressage IPv4 : VLSM

- Commencer par ordonner les sous-réseaux par ordre décroissant du nombre d' hôtes.

Sous-Rréseau	Nombre d'hôtes
SR3	250
SR4	100
SR1	60
SR2	10
R0-R1	2

- L'adresse de base attribuée : 193.194.4.0/22
- On commence par le sous-réseau SR3 qui contient 250 hôtes, pour les adresser il faut 8 bits



Adressage IPv4 : VLSM

193.194.00000100.00000000 → 193.194.4.0/24 SR3
193.194.00000101.00000000 → 193.194.5.0/24
193.194.00000110.00000000 → 193.194.6.0/24
193.194.00000111.00000000 → 193.194.7.0/24

- SR4 contient 100 hôtes, pour les adresser il faut 7 bits

193.194.00000101.00000000 193.194.5.0/25 SR4
193.194.00000101.10000000 193.194.5.128/25

- SR1 contient 60 hôtes, pour les adresser il faut 6 bits

193.194.00000101.10000000 193.194.5.128/26 SR1
193.194.00000101.11000000 193.194.5.192/26

- SR2 contient 10 hôtes, pour les adresser il faut 4 bits

193.194.00000101.11000000 193.194.5.192/28 SR2
193.194.00000101.11010000 193.194.5.208/28
193.194.00000101.11100000 193.194.5.224/28
193.194.00000101.11110000 193.194.5.240/28

- SR1 contient 2 hôtes, pour les adresser il faut 2 bits

193.194.00000101.11010000 193.194.5.208/30 SR1
193.194.00000101.11010100 193.194.5.212/30
193.194.00000101.11011000 193.194.5.216/30
193.194.00000101.11011100 193.194.5.220/30

Adressage IPv4 : VLSM

- Le plan d'adressage final est récapitulé dans le tableau :

Sous-Rréseau	Nombre d'hôtes
SR3: 193.194.4.0/24	250
SR4: 193.194.5.0/25	100
SR1: 193.194.5.128/26	60
SR2: 193.194.5.192/28	10
R0-R1: 193.194.5.208/30	2

- Nombre d'adresse restant disponibles

$$NB = 2^{10} - (2^8 + 2^7 + 2^6 + 2^4 + 2^2) \quad = 556 \text{ Adresses}$$

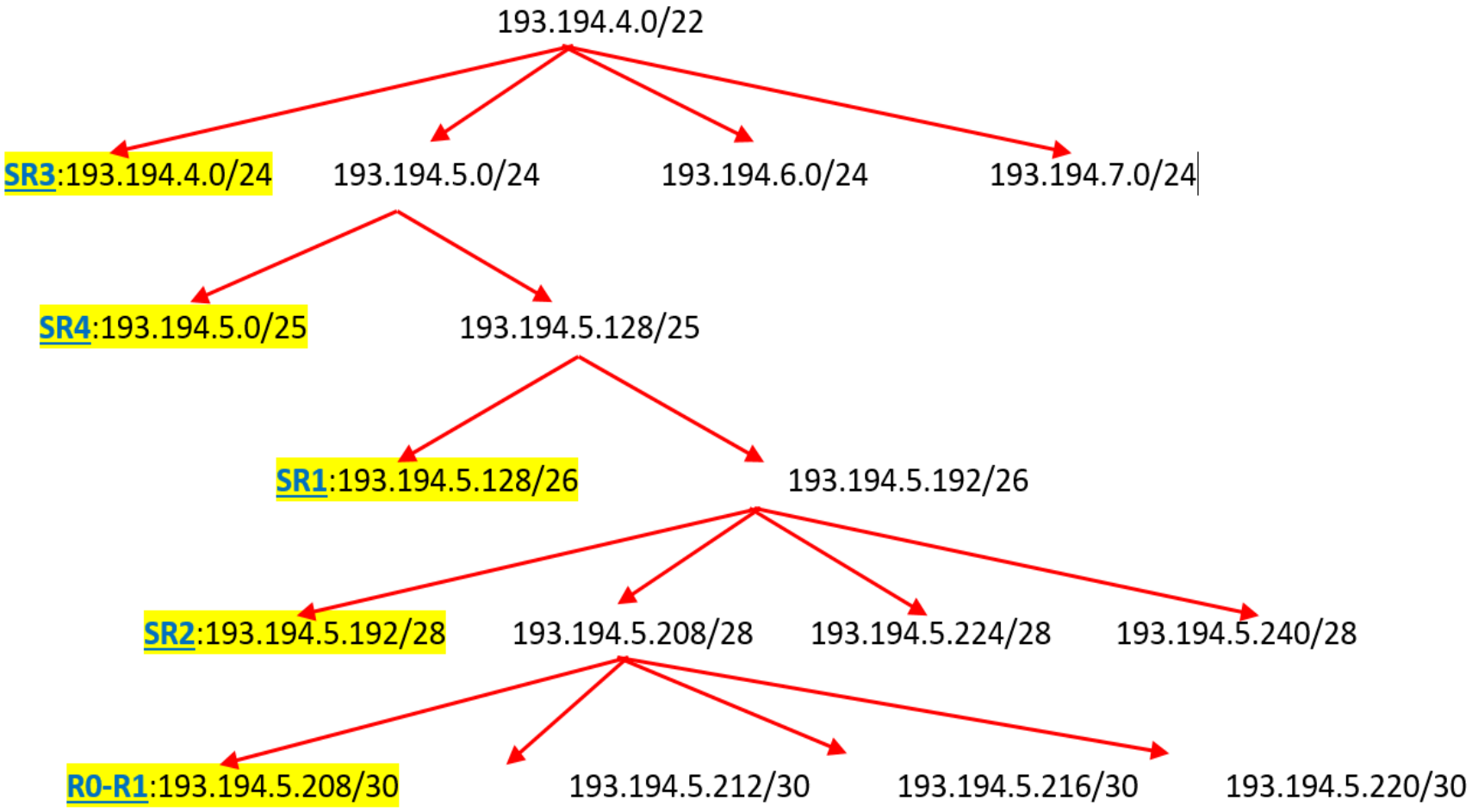
- Règle de calcul de l'adresse du sous-réseau :

Adresse SR suivant = Addition-Binaire(Adresse SR + Nombre d'adresses de SR)

SR1 = SR4 + 2^7 = 193.194.5.0 + 128 = 193.194.5.128

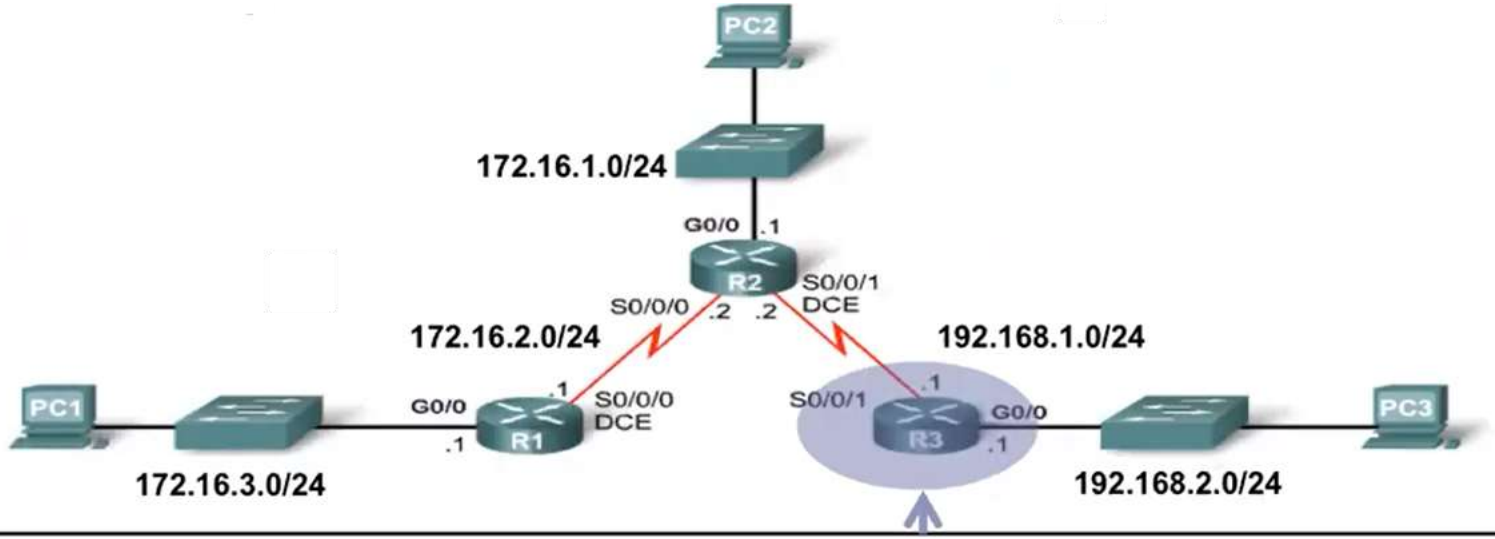
Adressage IPv4 : VLSM

- Illustration sous forme d'une arborescence



Adressage IPv4 : Supernetting

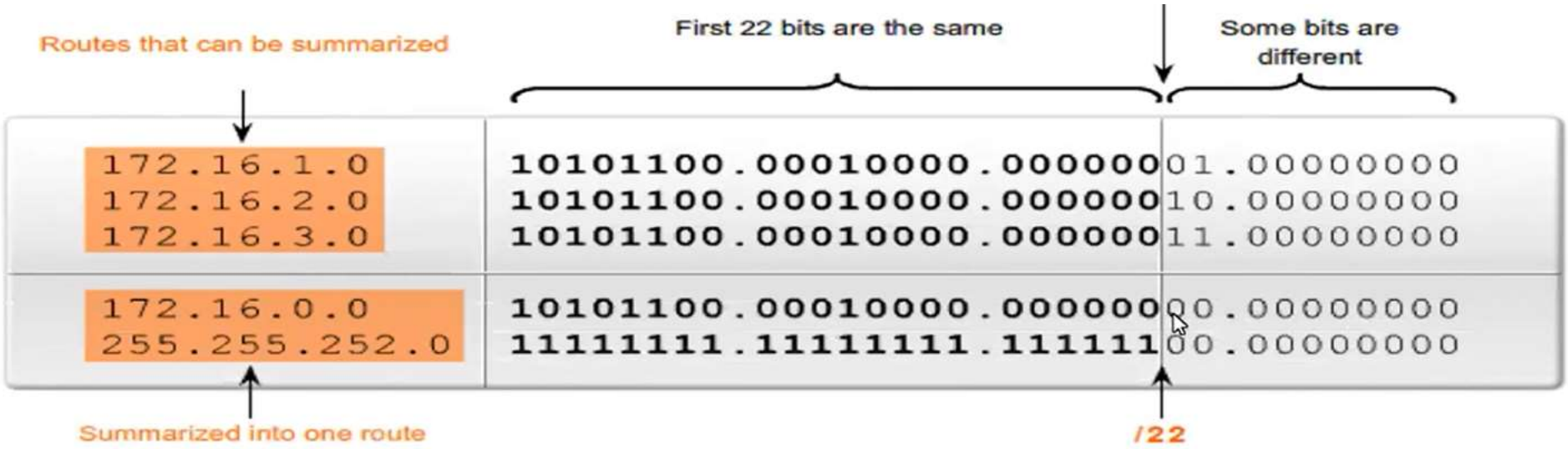
- Supernetting : Combiner plusieurs sous-réseaux pour créer un réseau plus large (SuperNet)
- Le Supernetting permet de réduire le nombre d'entrée de la table de routage
- Supernetting : Résumé de routes, Agrégation de routes
- Calcul : Processus inverse du VLSM (SuperNet vs SubNet)




```
R3#show ip route static
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
S       172.16.1.0 [1/0] via 192.168.1.2
S       172.16.2.0 [1/0] via 192.168.1.2
S       172.16.3.0 [1/0] via 192.168.1.2
```

Adressage IPv4 : Supernetting




```
R3(config)#no ip route 172.16.1.0 255.255.255.0 192.168.1.2
R3(config)#no ip route 172.16.2.0 255.255.255.0 192.168.1.2
R3(config)#no ip route 172.16.3.0 255.255.255.0 192.168.1.2
R3(config)#
R3(config)#ip route 172.16.0.0 255.255.252.0 192.168.1.2
```

```
R3# show ip route static

      172.16.0.0/22 is subnetted, 1 subnets
S        172.16.0.0 [1/0] via 192.168.1.2
```

NAT : Network Address Translation

- NAT : Traduction d'adresses réseau,
- L'espace d'adresses IP, sans NAT, est insuffisant pour pouvoir attribuer une adresse unique à chaque hôte (Device) connecté à Internet,
- Les réseaux sont généralement déployés à l'aide d'adresses IPv4 dites privées, tel que défini dans la RFC 1918

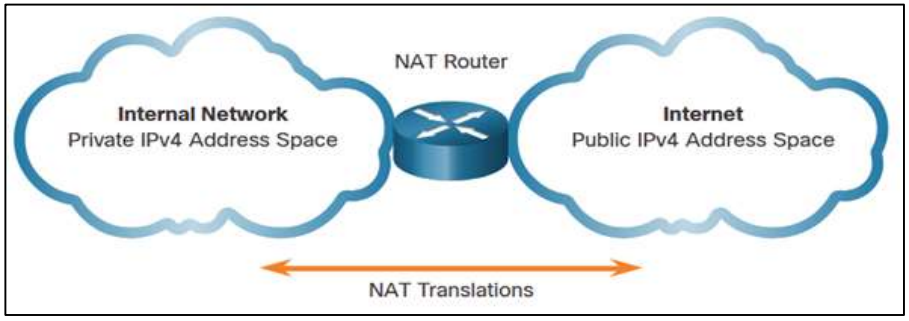
Classe	Plage d'adresses	Préfixe
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16

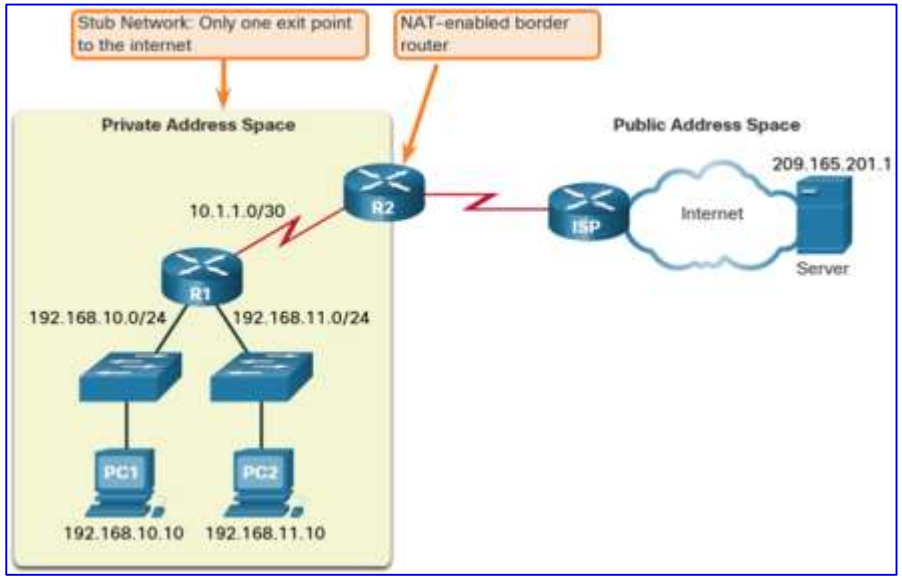
- Les adresses privées sont utilisées de manière interne, au sein d'une entreprise ou d'une organisation, et ce pour permettre aux hôtes de communiquer localement,
- Les adresses privés n'identifient aucune entreprise ou organisation, il ne sont pas routable ou visible sur Internet : Ne peuvent pas être acheminées sur Internet et faire l'objet d'un datagramme IP publique,
- Accéder à des ressource externes (Internet) depuis un réseau local ou privé nécessite :

☞ Traduction de l'adresse privée vers une adresse publique routable: **NAT**

NAT : Network Address Translation

- NAT permet de préserver et limiter la consommation des adresses IPv4 publiques,
- Une seule et même adresse IP publique peut être partagée par des centaines, voire des milliers d'hôtes,
- Le NAT permet d'ajouter un niveau de confidentialité et de sécurité à un réseau, en cachant les adresses IP internes,
- D'un point de vue pratique la fonction NAT peut être prise en charge par un Firewall, un Routeur ou un software (iptables sous Linux),





NAT : Network Address Translation

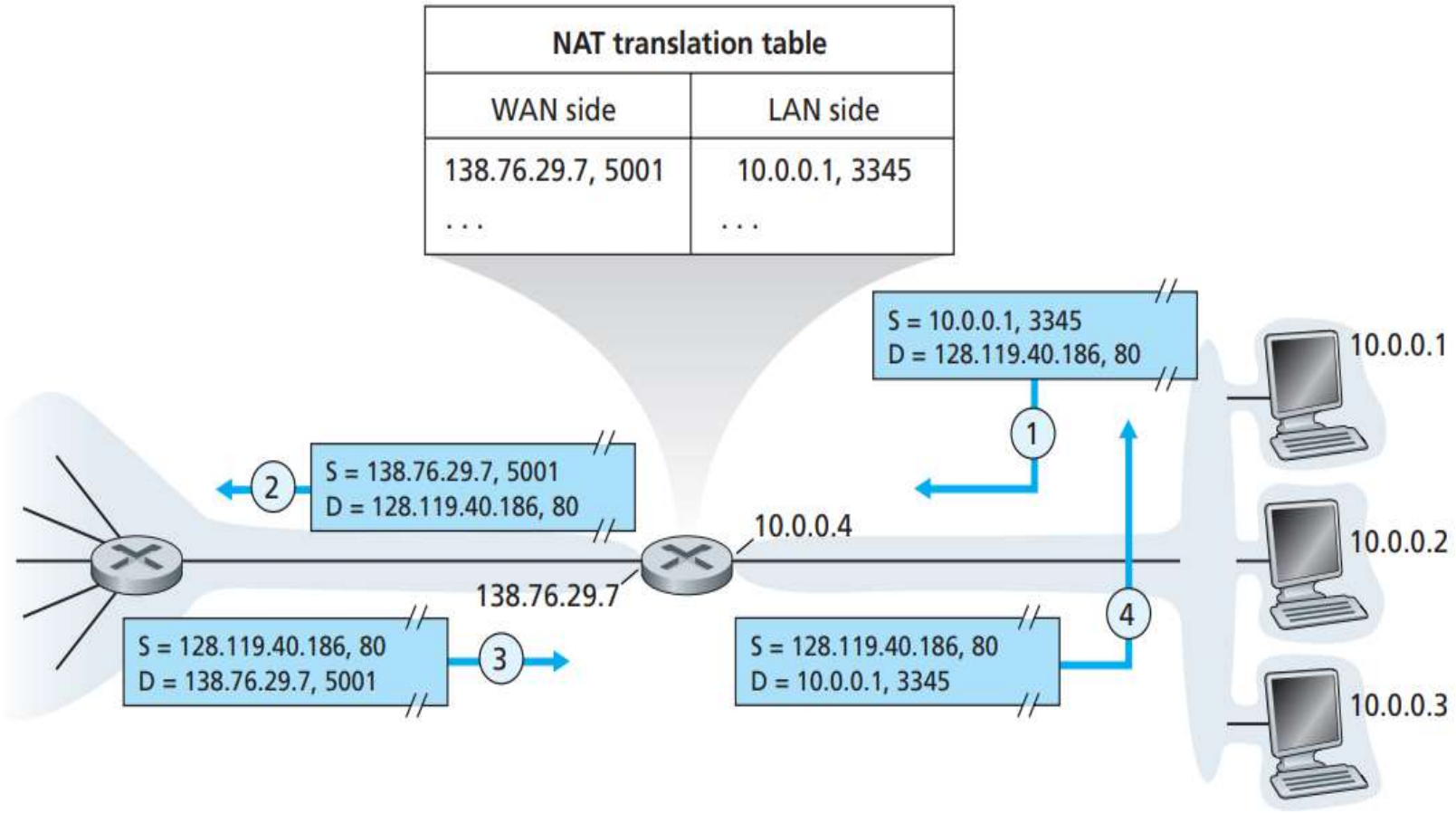
Définition : NAT est le procédé qui fait correspondre ou mapper **N ≥ 1** adresses IP à **M ≥ 1** adresses IP différentes. le cas d'utilisation le plus courant est le mappage de N adresses privées vers M adresses publiques, globalement uniques.

Types de NAT : On peut distinguer trois types de NAT

- N = M : NAT statique, chaque adresse privée est mappée à une adresse publique.
- N > M : NAT dynamique, un pool d'adresses publiques est attribué aux adresses privés selon le principe du premier arrivé, premier servi. Si toutes les adresses du pool ont été attribués, le prochain hôte doit attendre.
- M = n : PAT (Port Adress Translation), mappe plusieurs adresses privées à une seule adresse publique unique ou à quelques adresses (quelques n=1,2,..).

N.B : Selon cette classification, le NAT modifie uniquement les adresses IP, par contre le PAT peut modifier les adresses IP et les ports.

NAT : Network Address Translation



NAT : Cas de CISCO

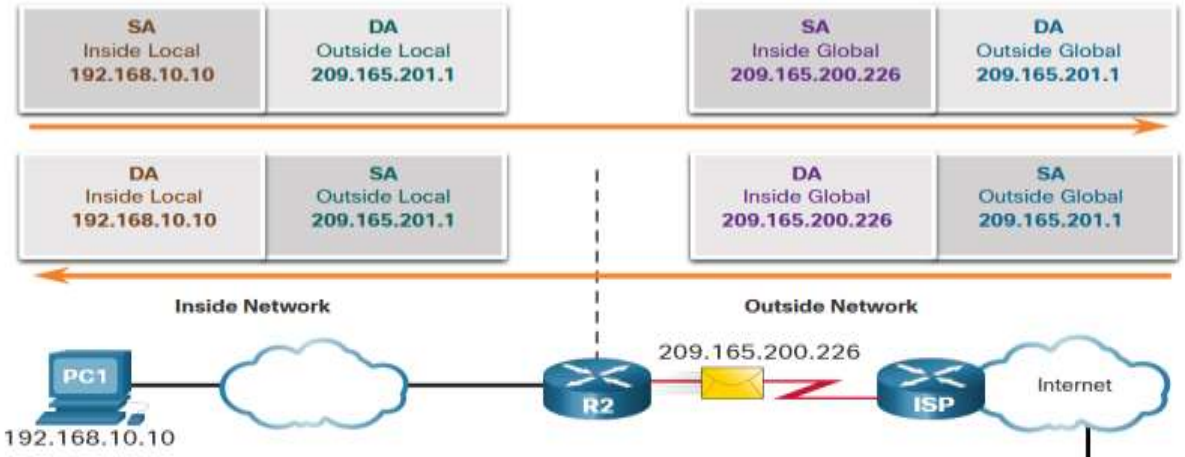
- **Terminologie** : La terminologie NAT CISCO est toujours appliquée du point de vue du hôte avec l'adresse traduite:

Adresse locale interne : L'adresse de la source vue du réseau interne. Il s'agit généralement d'une adresse IP privée. PC1 possède l'adresse locale interne 192.168.10.10.

Adresse globale interne : L'adresse de la source vue du réseau externe. L'adresse globale interne de PC1 est 209.165.200.226

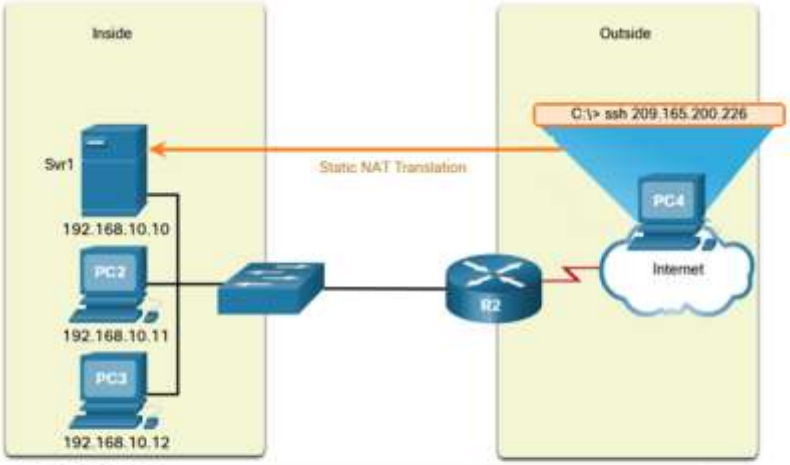
Adresse globale externe :L'adresse de destination vue du réseau externe. L'adresse globale externe du serveur Web est 209.165.201.1

Adresse locale externe : L'adresse de destination vue du réseau externe. PC1 envoie le trafic au serveur Web à l'adresse IP 209.165.201.1. Bien que cela soit rarement le cas, cette adresse peut être différente de l'adresse de destination globalement routable.



R2 NAT Table			
PC1		**Web Server**	
Inside Global Address	**Inside Local Address**	**Outside Local Address**	**Outside Global Address**
209.165.200.226	**192.168.10.10**	**209.165.201.1**	**209.165.201.1**

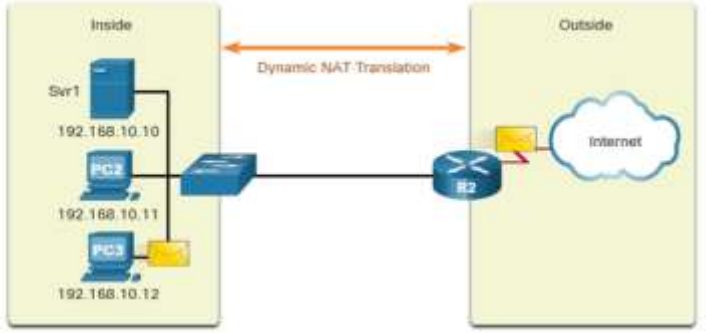
NAT : Cas de CISCO



Static NAT Table

Inside Local Address	Inside Global Address - Addresses reachable via R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

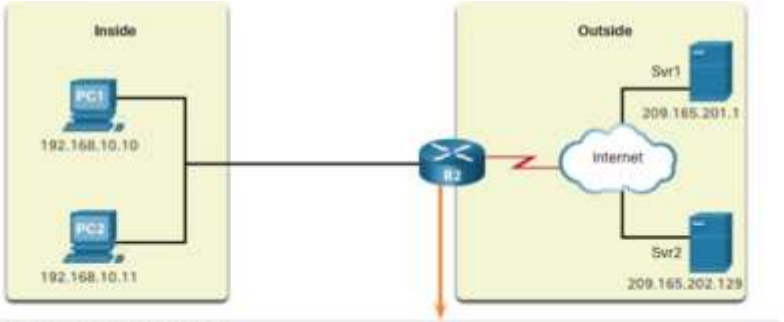
NAT



IPv4 NAT Pool

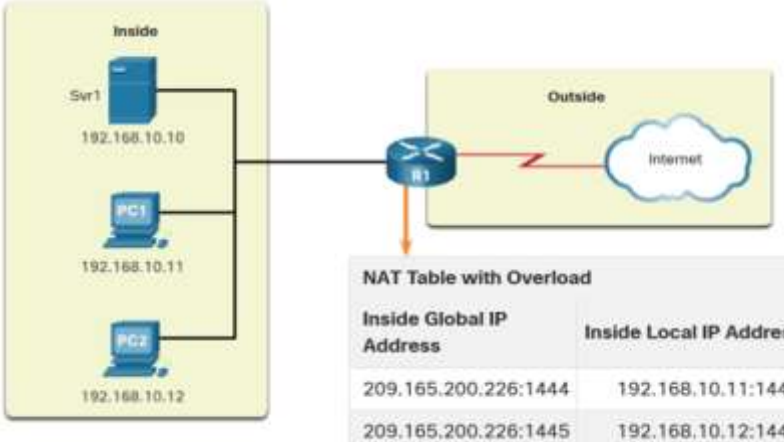
Inside Local Address	Inside Global Address Pool - Addresses reachable via R2
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

PAT



NAT Table with Overload

Inside Local IP Address	Inside Global IP Address	Outside Local IP Address	Outside Global IP Address
192.168.10.10:1555	209.165.200.226:1555	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1331	209.165.200.226:1331	209.165.202.129:80	209.165.202.129:80

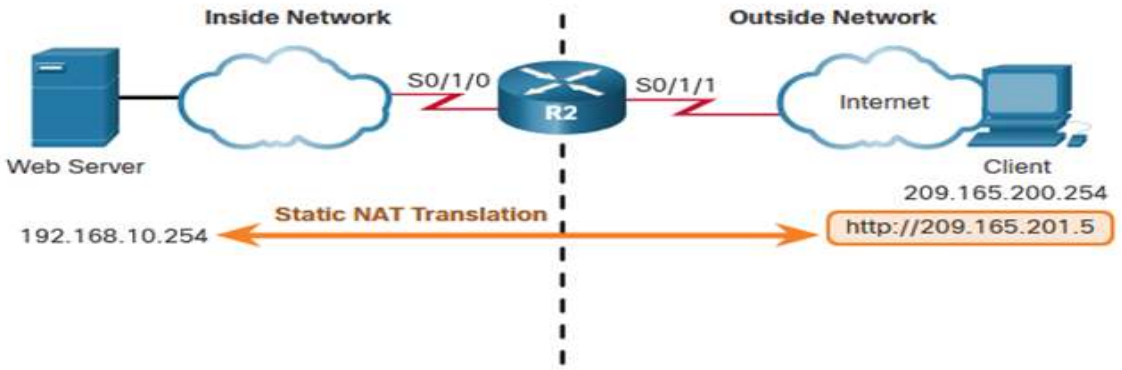


NAT Table with Overload

Inside Global IP Address	Inside Local IP Address
209.165.200.226:1444	192.168.10.11:1444
209.165.200.226:1445	192.168.10.12:1444

NAT Statique : Cas de CISCO

- Créer un mappage entre l'adresse locale interne et une adresse globales interne
- Spécifier les interfaces, interne et externe, participant au NAT




```
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5
R2(config)#
R2(config)# interface serial 0/1/0
R2(config-if)# ip address 192.168.1.2 255.255.255.252
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface serial 0/1/1
R2(config-if)# ip address 209.165.200.1 255.255.255.252
R2(config-if)# ip nat outside
```

```
R2# show ip nat translations
Pro Inside global Inside local Outside local Outside global
--- 209.165.201.5 192.168.10.254 --- ---
Total number of translations: 1
```

```
R2# show ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 209.165.201.5 192.168.10.254 209.165.200.254 209.165.200.254
--- 209.165.201.5 192.168.10.254 --- ---
Total number of translations: 2
```

NAT Dynamique : Cas de CISCO

- Définissez le pool d'adresses globales internes à utiliser dans le NAT
- Définissez une liste de contrôle d'accès (ACL) standard pour spécifier les adresses locales internes autorisées dans le NAT. Pour spécifier les adresses on utilise le masque inversé **Wildcard Mask** au lieu du masque normal
- Liez l'ACL au pool
- Spécifier les interfaces, interne et externe, participant au NAT

```
R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL1
R2(config)# interface serial 0/1/0
R2(config-if)# ip nat inside
R2(config-if)# interface serial 0/1/1
R2(config-if)# ip nat outside
```

PAT : Cas de CISCO

- Le PAT peut utiliser une seule adresse global interne pour la traduction, ou elle peut utiliser un pool d'adresses.

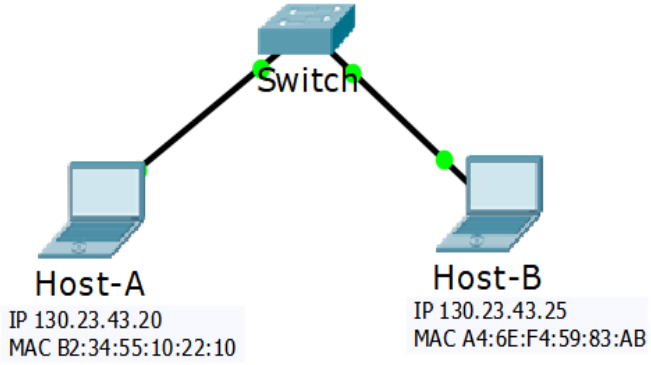
- Pour le pool, les commandes de configuration sont similaires à ceux du NAT dynamique, juste on ajoute le mot **overload** à la fin de la commande qui lie l'ACL au pool
- Dans le cas d'une seule adresse :
  - On définit la liste d'accès (ACL)
  - On crée le mappage avec la commande **ip nat inside source**, et on ajoute **overload** à la fin

```
R2(config)# ip nat inside source list 1 interface serial 0/1/0 overload
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2 (config) # interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2 (config) # interface Serial0/1/1
R2(config-if)# ip nat outside
```

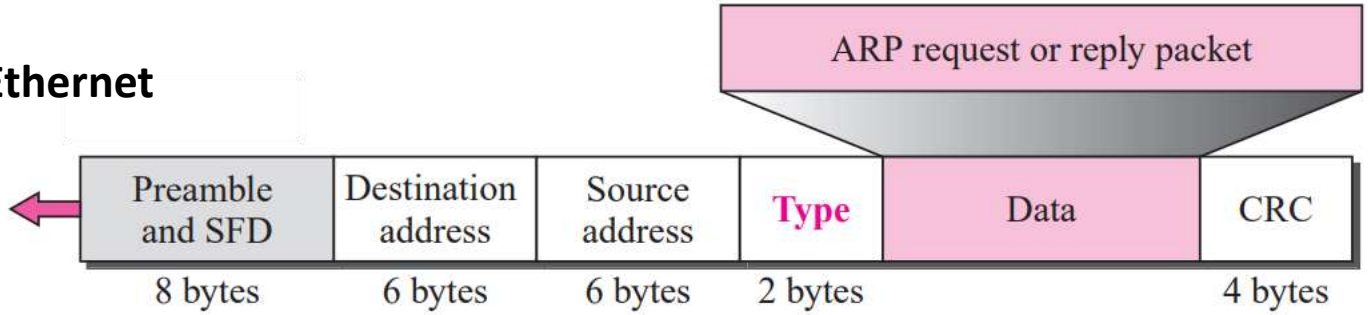
```
R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL2 overload
R2 (config) # interface serial0/1/0
R2(config-if)# ip nat inside
R2 (config-if) # interface serial0/1/0
R2(config-if)# ip nat outside
```

Protocole ARP (Address Resolution Protocol) : F(IP) = MAC

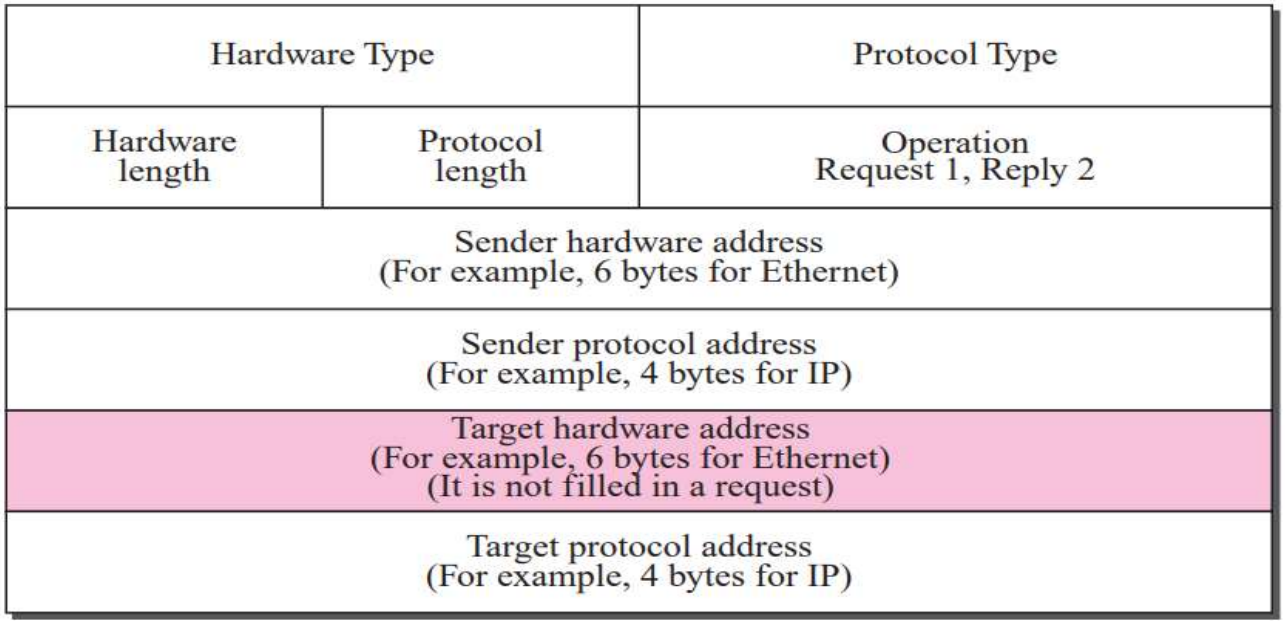
- La machine Host-A veut envoyer un paquet à Host-B (Par exemple **ping 130.23.43.25**)
- Evidemment, Host-A connait l'adresse IP de Host-B
- C'est la première communication entre les deux machines
- Host-A connait son IP, son MAC et l'IP de Host-B. Il ne connait pas la MAC de Host-B
- La tramme à envoyer comprend MAC Source et MAC Destination
- Comment Host-A obtient la MAC Destination ?
- Host-A diffuse une requete ARP request:
  "Who has 130.23.43.25? Tell 130.23.43.20 at B2:34:55:10:22:10 "
- Host-B répond avec une réponse ARP reply :
  "130.23.43.25 is at A4:6E:F4:59:83:AB"
- Cache ARP : Host-A sauvegarde la résolution dans une table ARP pour une utilisation future, et éviter une autre requête ARP.
  Sous Windows la commande **arp –a** visualise le contenu de la table ARP



Trame Ethernet



Protocole ARP : Structure du PDU



- **Hardware Type** : Type de réseau de couche inférieure sur lequel ARP s'exécute (01- Ethernet, 17-HDLC)
- **Protocol Type** : Type de protocole de la couche supérieure (0x0800 – IPv4)
- **Hardware length** : Longueur de l'adresse physique (6- Ethernet, 1-Token Ring)
- **Protocol length** : Longueur de l'adresse logique (4- IPv4)
- **Operation** (RFC 826) : Type de message ARP (1- Requête, 2- Réponse)
- **Sender hardware address** : Adresse physique de l'émetteur (MAC Source – Ethernet)
- **Target hardware address** : Adresse physique du récepteur (MAC Destination – Ethernet)
- **Sender protocol address** : Adresse logique de l'émetteur (IP Source – IPv4)
- **Target protocol address** : Adresse logique du récepteur (IP Destination – IPv4)

ARP Protocole

.