

Le protocole IP



IPv4 et IPv6



Problème adresses IPv4

✦ Raisons:

- ✦ Croissance rapide d'internet ➔ manque d'adresses IPv4 (seulement sur 4 octets)
- ✦ Tables de routage gigantesques...
- ✦ Des fonctionnalités manquantes à mettre en œuvre

✦ Migration vers IPv6:

- ✦ **Adresse sur 16 octets (128 bits, 8 * 16 bits)**
- ✦ Fonctionnalités nouvelles :
 - sécurité (chiffrement des paquets, authentification,...)
 - source routing
 - Gestion du temps réel
 - autoconfiguration (amélioration de DHCP, passage par ICMPv6)

IPv6 (1)



✦ Les différentes sortes d'adresses

◆ Unicast

-> pour un destinataire unique

◆ Multicast

-> pour désigner un groupe d'interfaces, donc un groupe de machines

◆ Anycast

-> pour désigner une interface, appartenant à un groupe de machines

-> théoriquement, la machine la plus proche doit recevoir le paquet

◆ Broadcast

-> n'existe plus.

Adressage IPv6 (1)

- ◆ Utilisation de la notation hexadécimale, en regroupant les chiffres par 4 et en les séparant par ‘:’

ex : FEDC:E323:A65A:95F5:63D4:08BB:76F5:A234

- ◆ Disparition des sous-réseaux, apparition de **la taille du préfixe ‘/’**

(nb de bits appartenant au préfixe réseau)

ex : 2F45:EE34:C23E::/48

- ◆ Elimination des 0 présents dans l’adresse:

ex : 2001:0:0:0:0:342D:342F:FF45



2001::342D:342F:FF45

Adressage IPv6 (3)

✦ Adresse Unicast

- ◆ 64 bits pour le réseau
 - 48 bits pour le préfixe global de routage (topologie publique)
 - théoriquement, 3 premiers bits à : **001**
 - 45 bits suivants dépendent de l'IANA, et donc RIPE-NCC
 - 16 bits pour le réseau d'entreprise (topologie du site)
- ◆ 64 bits pour l'hôte
 - 64 bits -> interface (unicast -> utilisation de icmpv6)

L'adresse commence en général par 2001:.... et se termine par l'adresse MAC modifiée:

3 octets (constructeur) + FFFE (16 bits) + 3 octets (unique MAC)

Adressage IPv6 (4)

✦ Adresse Multicast

- ✦ indicateur (8 b) + drapeau (4 b) + visibilité (4 b) + group Id (112 b)
 - Indicateur : FF (1 octet avec que des 1)
 - Drapeau : 000 puis 0 pour un groupe permanent, 1 sinon
 - Visibilité : sur internet, local à l'entreprise...
 - ◆ Node – Local (même médium) → visibilité =1
 - ◆ Link – Local (même domaine broadcast) → visibilité =2
 - ◆ Site – Local (même site) → visibilité =5
 - ◆ Organization – Local (même entreprise) → visibilité =8
 - ◆ Global → visibilité =E
- ✦ Group id : 1 = machines, 2 = routeurs, 4 = OSPF routers,
- ✦ Exemple : *FF01::1 -> multicast pour node local et machines (id =1)*
FF05::2 -> multicast pour site-local et routeurs (id =2)

Équipement

Adressage équipement

✦ Ordinateur, serveur :

- Obligatoire pour communiquer

✦ HUB :

- Pas d'adresse IP

✦ Switch :

- Pas d'adresse IP

sauf si switch manageable à distance

✦ Téléphone en 3G, 4G, 5G :

- Obligatoire pour communiquer



La fonction routage dans la couche Réseau La couche Transport



Le routage
Protocole TCP / UDP

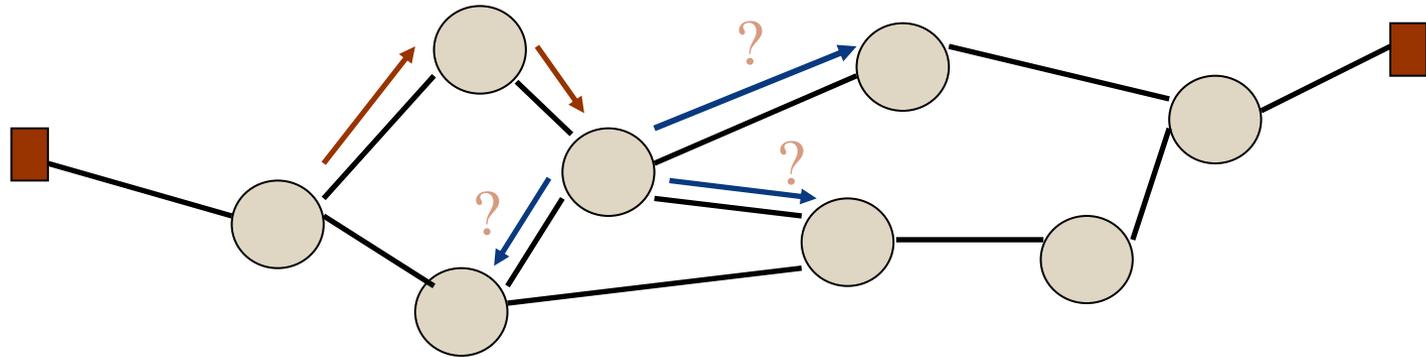




Le Routage

Routage (1)

L'acheminement des paquets vers un destinataire dans un réseau maillé est réalisé par un procédé appelé **routage**.



A chaque nœud correspond un routeur contenant une table de routage et mettant en place des algorithmes permettant l'acheminement des paquets dans Internet et éventuellement la mise à jour des tables automatiquement.

Les nœuds routent du mieux qu'ils peuvent (notion de best effort d'IP);

Routages (2)

Politiques de calcul

- Fixe (une fois pour toutes)
- Adaptative aux modifications de topologie ou de charge

Participants aux calculs

- Un seul centre (routage centralisé)
- Tout ou partie des systèmes appelés **routeurs** (routage distribué ou réparti)

Portée du calcul

- Le prochain pas (hop by hop)
- Toute la route (source routing)

Méthodes de calcul

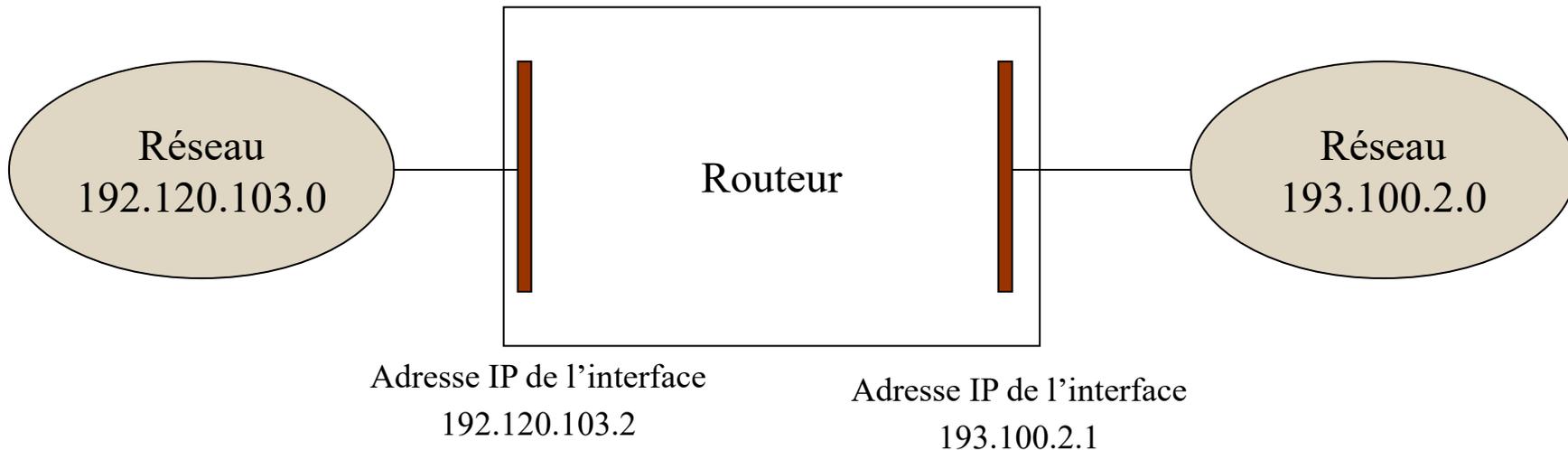
- Par Inondation
- Avec table de routage
 - Fixes
 - Adaptatives ou dynamiques
 - Nécessitent la transmission d'informations d'état de la topologie et de la charge
 - Protocoles d'échange d'informations de routage : RIP, BGP, OSPF, IS-IS...

Architecture d'un routeur

➤ Architecture du Routeur:

Station composée de plusieurs cartes réseau, lui permettant de se connecter à plusieurs réseaux locaux (**au moins 2**).

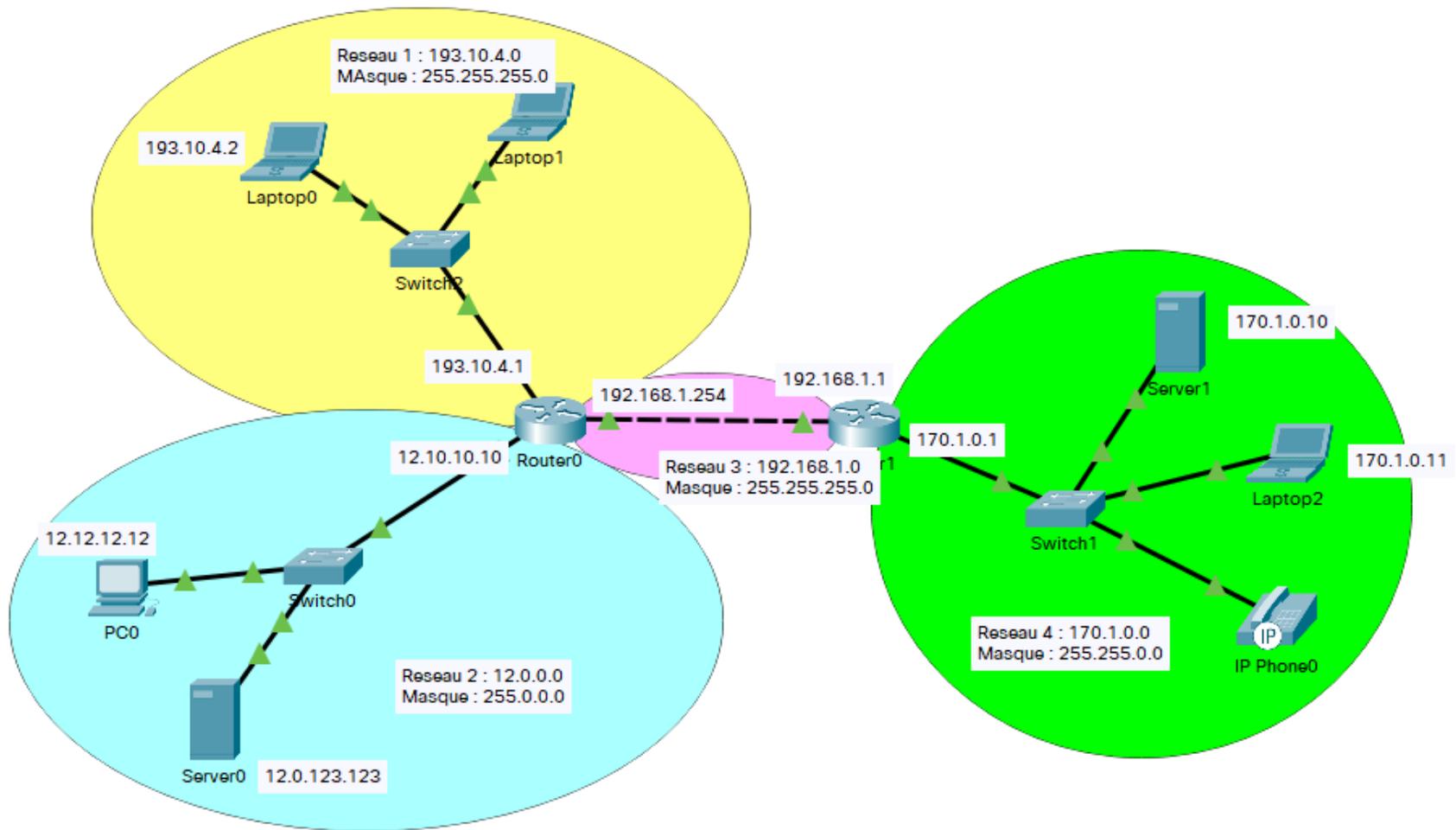
Dans ce cas, une carte réseau est appelée **une interface du routeur**



L'adresse de l'interface réseau doit appartenir au réseau

Un routeur sépare au moins 2 réseaux

Exemple



Mode non connecté

✦ *Tout équipement de niveau 3 à une table de routage*

◆ PC, routeur, mais ni HUB, ni switch

✦ Composition d'une table de routage

◆ 4 colonnes (ou lignes):

- @ réseau distant
- Masque du réseau distant
- @IP du saut suivant pour atteindre le réseau distant
- Interface de sortie

La table de routage ne donne que l'@IP du prochain système sur la route vers la destination

(hop by hop)

Fonctionnement table de routage (1)

- Exemple de table de routage du routeur 193.65.20.1

@IP réseau distant	Masque	@IP saut suivant	interface
193.65.20.0	255.255.255.0	193.65.20.1	eth0
136.30.0.0	255.255.0.0	193.65.20.254	eth0
0.0.0.0	0.0.0.0	193.65.20.100	eth0

- Utilisation de la table de routage

- Soit @Ipd adresse de destination finale
- Pour chaque ligne de la table de routage
 - faire un "et" binaire entre @Ipd et Masque
 - Comparer le résultat avec l'@IP réseau distant
 - Si différente, passer à la ligne suivante
 - Sinon @IP du prochain saut = @IP saut suivant pour le paquet
- Si plus de ligne, échec, abandon du paquet

Commande pour voir la table de routage : `netstat -r` ou `ip route`

Exemple table de routage (1)

IPv4 Table de routage

Itinéraires actifs :

Destination réseau	Masque réseau	Adr. passerelle	Adr. interface	Métrique
0.0.0.0	0.0.0.0	172.16.79.254	172.16.65.100	276
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306
172.16.64.0	255.255.240.0	On-link	172.16.65.100	276
172.16.65.100	255.255.255.255	On-link	172.16.65.100	276
172.16.79.255	255.255.255.255	On-link	172.16.65.100	276
192.168.56.0	255.255.255.0	On-link	192.168.56.1	276
192.168.56.1	255.255.255.255	On-link	192.168.56.1	276
192.168.56.255	255.255.255.255	On-link	192.168.56.1	276
224.0.0.0	240.0.0.0	On-link	127.0.0.1	306
224.0.0.0	240.0.0.0	On-link	192.168.56.1	276
224.0.0.0	240.0.0.0	On-link	172.16.65.100	276
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306
255.255.255.255	255.255.255.255	On-link	192.168.56.1	276
255.255.255.255	255.255.255.255	On-link	172.16.65.100	276

Itinéraires persistants :

Adresse réseau	Masque réseau	Adresse passerelle	Métrique
0.0.0.0	0.0.0.0	172.16.79.254	Par défaut

Sous windows, lecture de bas en haut....

Exemple table de routage (2)

Exemple sous linux :

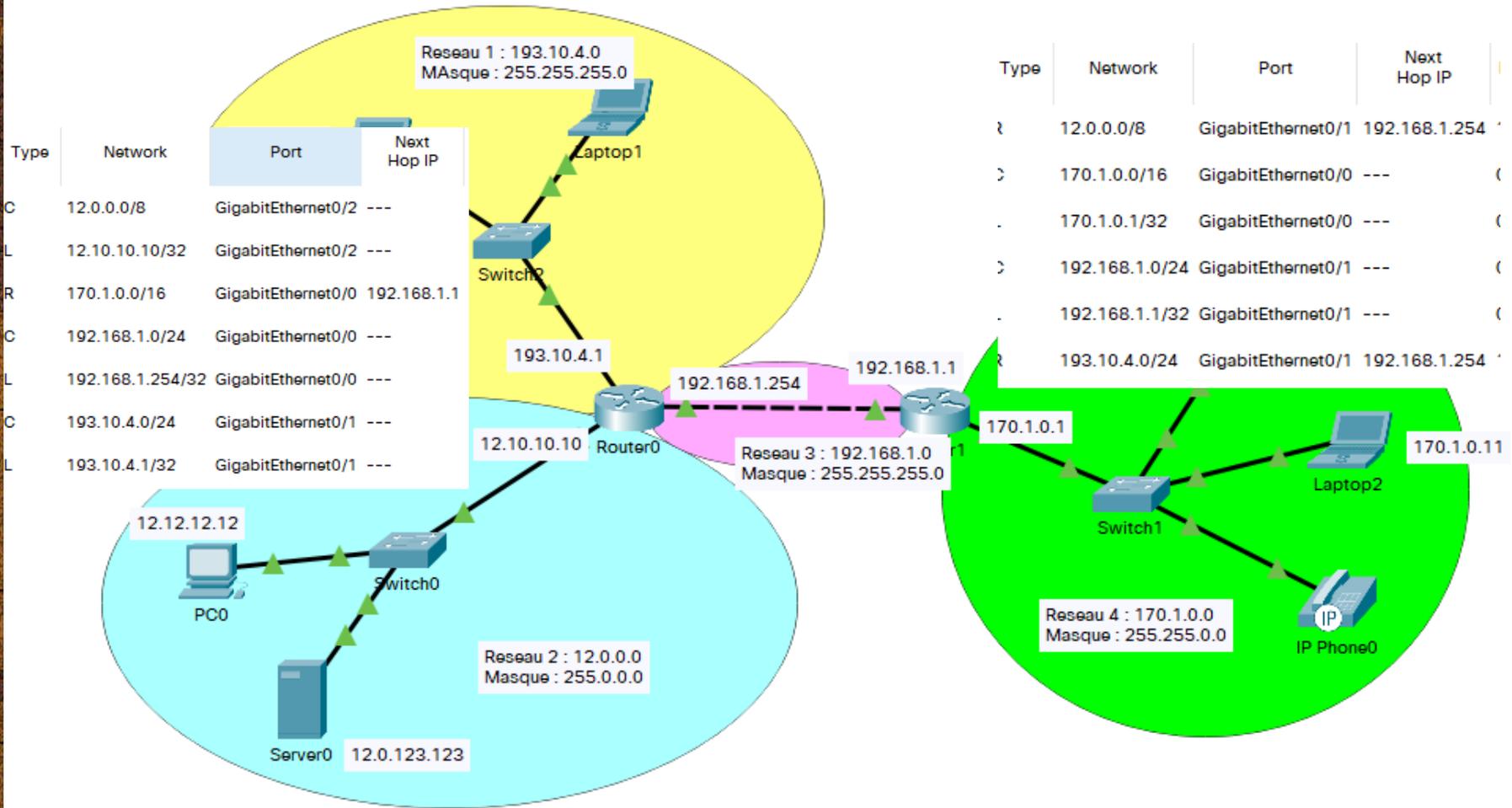
```
[palauren@turing ~]$ netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags         MSS Window  irtt  Iface
0.0.0.0          172.16.47.254   0.0.0.0         UG            0  0           0  enp109s0f1
172.16.32.0      0.0.0.0         255.255.240.0   U             0  0           0  enp109s0f1
192.168.122.0    0.0.0.0         255.255.255.0   U             0  0           0  virbr0
[palauren@turing ~]$
```

- Lecture de bas en haut
- 2 cartes réseaux : @IP : 172.16.32.21
@IP : 192.168.122;1
- Si gateway = 0.0.0.0 ou *, alors gateway inutile
même réseau que le destinataire

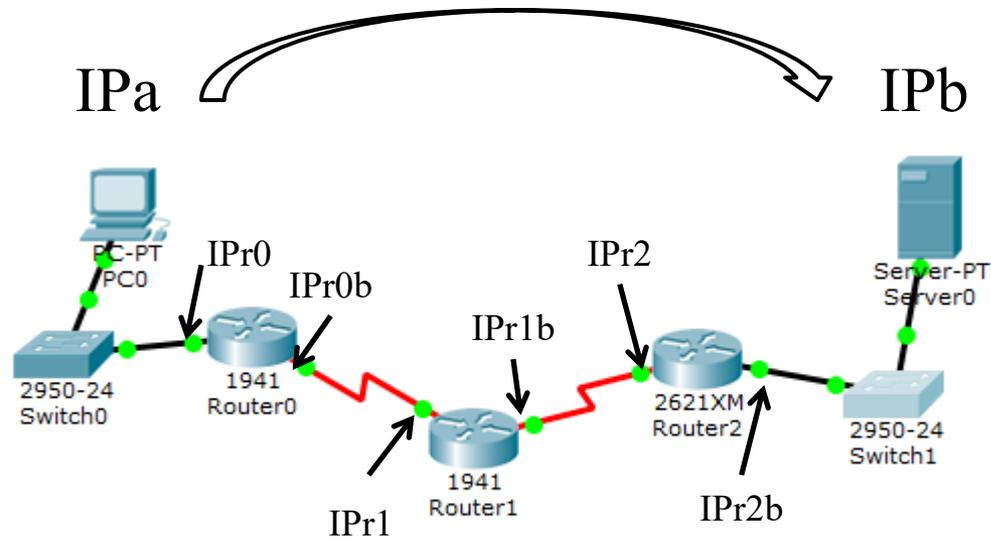
(paquet est presque arrivé à destination...)

```
[palauren@turing ~]$ ip route
default via 172.16.47.254 dev enp109s0f1 proto static metric 100
172.16.32.0/20 dev enp109s0f1 proto kernel scope link src 172.16.32.21 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```

Exemple suite

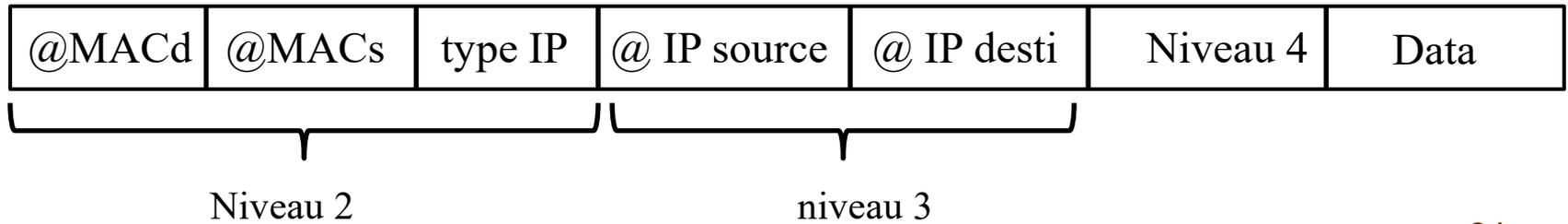


Utilisation routage (1)



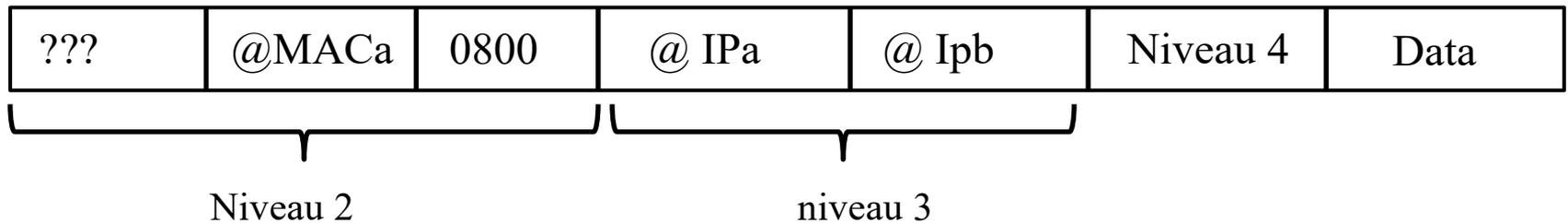
4 réseaux - (Ipa et IPr0) (IPr0b et IPr1) - (IPr1b et IPr2) - (IPr2b et Ipb)

Constitution de la trame



Utilisation routage (2)

Constitution de la trame



➤ utilisation de la table de routage

Mais récupération de l'@IP du saut suivant !!!
On cherche → @MAC

ARP : Address Resolution Protocol

correspondance @MAC ↔ @IP

ARP (1)

✦ ARP

- ✦ *But* : obtenir une correspondance entre @IP d'un PC et son adresse MAC
- ✦ *A* désire envoyer un message à une station *B*, et connaît son adresse IP. Mais, adresse MAC inconnue pour envoyer sa trame Ethernet.
- ✦ *A* envoie donc un **broadcast Ethernet ARP** qui contient l'adresse IP demandée (*B*) .
- ✦ Toutes les stations reçoivent ce message et examinent l'adresse IP demandée.
- ✦ Seule la station *B* répond à la requête ARP. Elle insère dans la réponse sa propre adresse MAC. Réponse en unicast.
- ✦ La station *A* récupère le message, **stocke dans sa table ARP la correspondance @IP ↔ @MAC** et peut maintenant envoyer des données à la station *B* en utilisant cette adresse MAC.

ARP (2)

✦ 2 étapes

- ✦ On regarde dans table ARP si correspondance (arp -a)
(stockage temporaire des informations)
- ✦ Une requête ARP est faite sur le réseau

No.	Time	Source	Destination	Protocol	Length	Info
111	13.464961	192.168.1.1	192.168.1.17	TCP	54	80 → 53795 [ACK] Seq=2473 Ack=4501 Win=13343 Len=0
112	13.502360	192.168.1.17	192.168.1.1	TCP	54	53796 → 80 [ACK] Seq=3746 Ack=2175 Win=254 Len=0
113	13.517112	SamsungE_be:e0:fe	Broadcast	ARP	42	Who has 192.168.1.17? Tell 192.168.1.18
114	13.517202	HonHaiPr_53:55:9b	SamsungE_be:e0:fe	ARP	42	192.168.1.17 is at 64:27:37:53:55:9b
115	13.518204	192.168.1.1	192.168.1.17	HTTP	633	HTTP/1.1 200 OK (text/html)

```
> Frame 113: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: SamsungE_be:e0:fe (bc:b1:f3:be:e0:fe), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: SamsungE_be:e0:fe (bc:b1:f3:be:e0:fe)
  Sender IP address: 192.168.1.18
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.17
```

```
C:\Users\Isima>arp -a
Interface : 172.16.65.100 --- 0xb
Adresse Internet      Adresse physique      Type
172.16.64.4           98-e7-f4-ec-32-33    dynamique
172.16.64.5           18-03-73-32-cb-11    dynamique
172.16.64.9           48-4d-7e-d0-29-b6    dynamique
172.16.64.38          b8-ca-3a-ba-2e-38    dynamique
172.16.64.70          18-03-73-d5-39-2b    dynamique
172.16.64.86          18-03-73-d6-6d-25    dynamique
172.16.64.99          00-1c-c0-52-ba-af    dynamique
172.16.64.102         18-03-73-d6-5a-75    dynamique
172.16.64.103         18-03-73-d6-61-61    dynamique
172.16.64.106         d4-be-d9-da-78-80    dynamique
172.16.64.112         00-24-e8-30-33-f3    dynamique
172.16.64.117         18-03-73-d6-5c-19    dynamique
172.16.64.130         c8-1f-66-d1-9a-f4    dynamique
172.16.64.133         b8-ac-6f-a2-18-44    dynamique
172.16.64.250         e0-cb-4e-12-78-0a    dynamique
172.16.64.252         90-e6-ba-60-50-2f    dynamique
172.16.65.13         d4-be-d9-63-78-f7    dynamique
172.16.65.65         d0-67-e5-3b-20-ab    dynamique
```

ARP (suite)

The screenshot displays a network simulation environment. The main workspace shows a topology with three nodes: a PC-PT PC0 on the left, a 2960-24TT Switch0 in the center, and a PC-PT PC1 on the right. All nodes are interconnected with bidirectional arrows. The interface includes a menu bar (File, Edit, Options, View, Tools, Extensions, Window, Help) and a toolbar with various icons. A status bar at the bottom indicates the time as 00:01:38.426 and provides play controls. On the right side, a Simulation Panel is open, featuring an Event List table with columns for Vis., Time(sec), Last Device, At Device, and Type. Below the table are controls for Reset Simulation, Constant Delay, and Play Controls. A list of Event List Filters is also visible, including ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDR, DHCP, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPsec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Mxshi, NDP, NETFLOW, NTP, OSPF, OSPFv6, PaqP, POP3, PPP, PPPoE, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STR, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP.

Vis.	Time(sec)	Last Device	At Device	Type
------	-----------	-------------	-----------	------

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDR, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPsec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Mxshi, NDP, NETFLOW, NTP, OSPF, OSPFv6, PaqP, POP3, PPP, PPPoE, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STR, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Utilisation routage (2 bis)

Attention

➤ La réponse à une requête ARP prend du temps

❖ Pas le temps d'attendre pour certains protocoles (ICMP)

⇒ Renvoi fail à la demande

Pour le ping (ICMP)

minimum 2 essais

Windows : 5 essais

Cisco : 4 essais

Linux : infini,...

Utilisation routage(4)

✦ Utilisation table de routage

- ◆ Toujours le même principe
- ◆ Si aucune route trouvée,
 - ⇒ abandon du paquet....
- ◆ Si paquet tourne en rond, mise en œuvre du TTL
 - ⇒ abandon du paquet ...

Création table de routage ??

- Statiquement
- Dynamiquement

Création statique

✦ Création de la table de routage statique

- ◆ Renseigne manuellement les 3 champs
 - @réseau destination
 - Masque du réseau destination
 - @IP du prochain routeur vers la destination
 - ◆ Remarque : l'adresse prochain routeur appartient au même réseau que le routeur

- ◆ Reste valide tout le temps

Même si la liaison tombe

Route prioritaire par rapport à du routage dynamique

Ne pas oublier d'indiquer la passerelle sur les PCs

Routage dynamique

✦ But : mettre à jour dynamiquement les tables de routage lorsque :

- La topologie interne de « l'AS » est modifiée (câble coupé, routeur ajouté, modification de l'adressage, ...)

✦ Deux classes d'algorithmes existent :

les algorithmes *à vecteurs de distance*

- ◆ chaque routeur ne connaît que ses voisins
- ◆ échange des tables de routage en utilisant un vecteur
- ◆ Utilisation et sélection des routes ayant le coût minimal
- ◆ Exemple : [RIP](#), IGRP, EIGRP

les algorithmes *à état de liens*

- ◆ Chaque routeur reconstruit une carte complète de l'AS
- ◆ Chaque routeur recherche la meilleure route vers les autres routeurs
- ◆ Exemple : [OSPF](#), IS-IS

Routage à vecteurs de distance (1)

✦ Basé sur l'algorithme de Bellman-Ford

(nommé DBF : Distributed Bellman-Ford)

- ◆ Les routeurs ne connaissent que leurs voisins (directement "connecté")

✦ Fonctionnement :

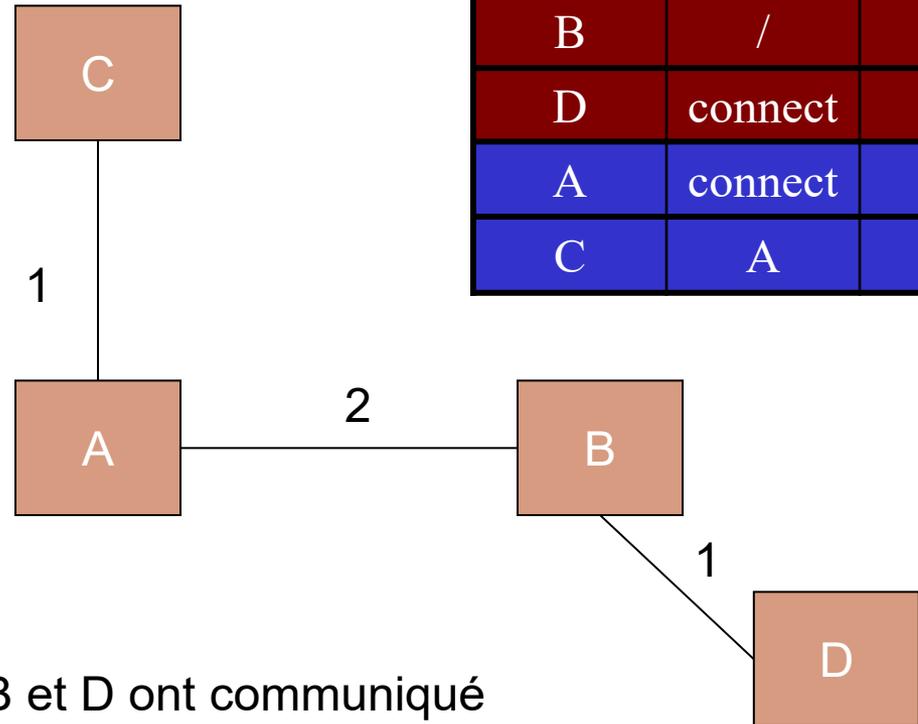
- ◆ On échange entre nœuds voisins **un vecteur donnant les coûts (vecteur de distance)** permettant d'atteindre toutes les destinations connues par le routeur et stockées dans la table de routage
- ◆ Un routeur qui reçoit ce vecteur, le compare avec ses propres coûts connus et **met à jour sa propre table de routage** :
 - si une route reçue comprend un plus court chemin
 - si une route reçue est inconnue
- ◆ Si la table d'un routeur est modifiée, le routeur enverra son vecteur de distance à tous ces voisins. Si plus aucune table n'est modifiée, l'algorithme se termine, on dit qu'il y a **convergence**.

Routage à vecteurs de distance (2)

C	Pass	coût
C	/	/
A	connect	1
B	A	3
D	A	4

A	Pass	coût
A	/	/
B	connect	2
D	B	3
C	connect	1

B	Pass	coût
B	/	/
D	connect	1
A	connect	2
C	A	3



B et D ont communiqué

B envoie sa table a A

Vecteur = (B=0)

C envoie sa table a A

Vecteur = (C=0)

A envoie sa table a B et C Vecteur = (A=0, B=3, C=1)

Routage à vecteurs de distance (3)

✦ Avantage :

Facile à mettre en œuvre

✦ Inconvénients :

- ✦ La taille des informations de routage est proportionnelle au nombre de routeurs du domaine
- ✦ Le nombre de réseaux s'est accru car adressage classless
 - peut donner de gros vecteurs de distance → lourd
- ✦ Coût de routage uniquement défini selon la distance entre 2 destinations
- ✦ *Maximum 15 sauts*

RIP / RIPv2

✦ Routing Information Protocol

- ✦ Protocole intra-domaine (IGP)
 - RFC 1058, 2453
 - Version 1.0 et 2.0
- ✦ Protocole à vecteur de distance
 - Utilisation algorithme de Bellman-Ford
 - Inondation de messages
 - Itératifs
 - Distribué

Algorithme à état de liens (1)

-
- ✦ Les routeurs maintiennent **une carte complète du réseau** et calculent les meilleurs chemins localement.
 - ◆ Utilisation d'une base de données (BD topologiques)
 - ◆ Utilisation d'un algorithme pour créer un arbre de plus court chemin (SPF)
 - ◆ Utilisation d'une table de routage

 - ✦ Un routeur teste périodiquement l'état des liens qui le relie à ses routeurs voisins (envoie d'une trame HELLO) pour savoir s'ils sont vivants.

 - ✦ Un routeur envoie périodiquement ces états (*Link-State Advertisement*) à tous les autres routeurs du domaine (*pas uniquement aux voisins*) ou lorsqu'il y a un changement de topologie.

 - ✦ Les routeurs ne communiquent pas la liste de toutes les destinations connues, mais seulement les informations ayant changées
 - Mise à jour partielle, évite de surcharger le réseau

Algorithme à état de liens (2)

- ✦ Lorsqu'un message parvient à un routeur, celui-ci met à jour sa BD topologique
- ✦ Il recalcule localement pour chaque lien modifié, la nouvelle route selon l'algorithme de Dijkstra (*Shortest Path First algorithm*) qui détermine le plus court chemin pour toutes les destinations à partir d'une même source.
- ✦ Tous les routeurs ayant la même BD topologique, chacun obtient le même résultat → convergence

<u>Avantages</u>	<u>Inconvénients</u>
Convergence rapide sans boucle	Capacité de calcul important
Métrique précise et couvrant plusieurs besoins	Plus de mémoire pour stocker les informations
Taille des messages échangés entre routeurs petite	Fort débit lors de la convergence de l'algorithme
Calcule des routes sur chaque routeur	

OSPF

✦ Open Shortest Path First (rfc 1247, 2328)

- ✦ Protocole intra-domaine (IGP)
- ✦ Protocole à état de liens

✦ Link-State Database (LSDB)

- ✦ Structure de données fondamentales
- ✦ Une copie est maintenue par tous les routeurs
- ✦ Contient une description du graphe des routeurs de l'area
- ✦ Chaque lien vers un réseau est associé à une entrée de la LSDB avec une métrique associée.
- ✦ Utilisation d'area pour séparer les gros réseaux