

Cours Réseaux

Chapitre 4 Couche réseau

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

Université de Perpignan

Chapitre 4: Couche réseau

Objetif du chapitre:

- Comprendre les principes derrière les services de la couche réseau:
 - Modèle des services de la couche réseau
 - Fonctionnement d'un routeur
 - routage (sélection du chemin)
 - Gérer l'envergure du réseau
 - Sujet avancés: IPv6, mobilité
- Instanciation, implémentation dans l'internet

Chapitre 4: Introduction

Notre objectif:

- Avoir une “idée” et la terminologie
- Plus de profondeur, détail seront plus tard dans le cours.
- Approche: utiliser Internet comme exemple

Grand ligne du cours:

- Qu'est ce que l'internet
- Quesqu'un protocole?
- Coeur de réseau, réseaux d'accès
- Accès au réseau et aspect matériel
- Internet/ISP structure
- Performance: perte, délai
- Les couches de protocoles
- (Historique des réseaux)
- Modèle de réseau

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- IPv4 adressage
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

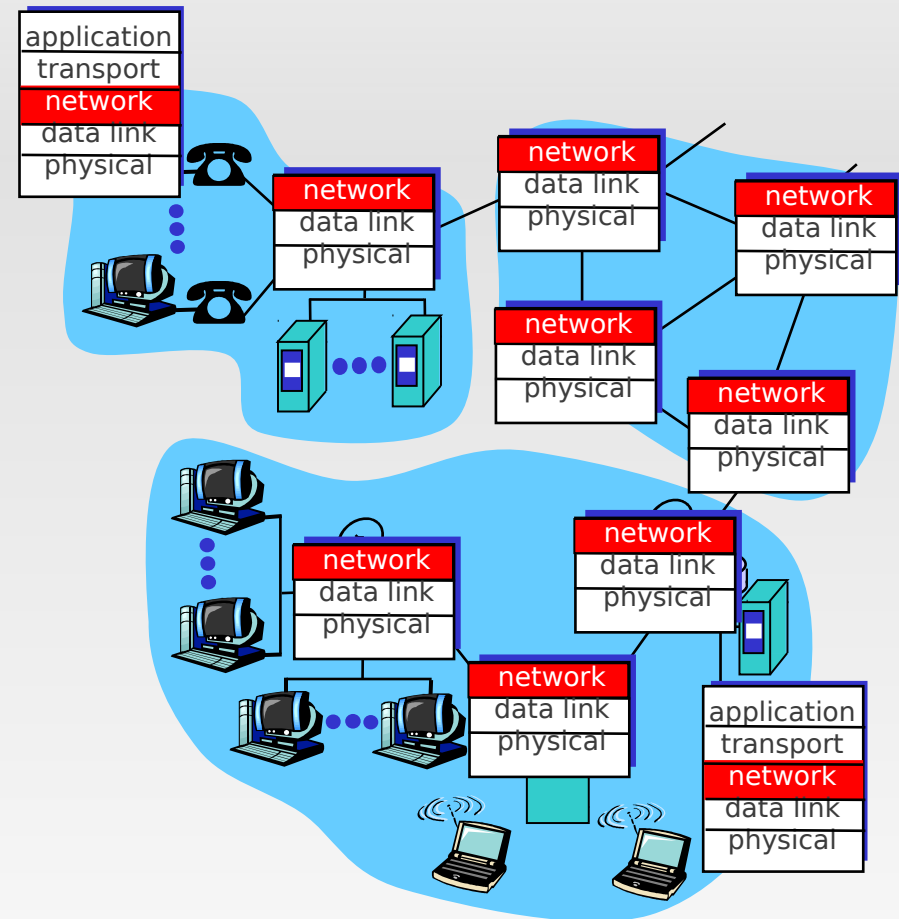
4.7 routage *broadcast* et *multicast*

Fonctions de la couche réseau

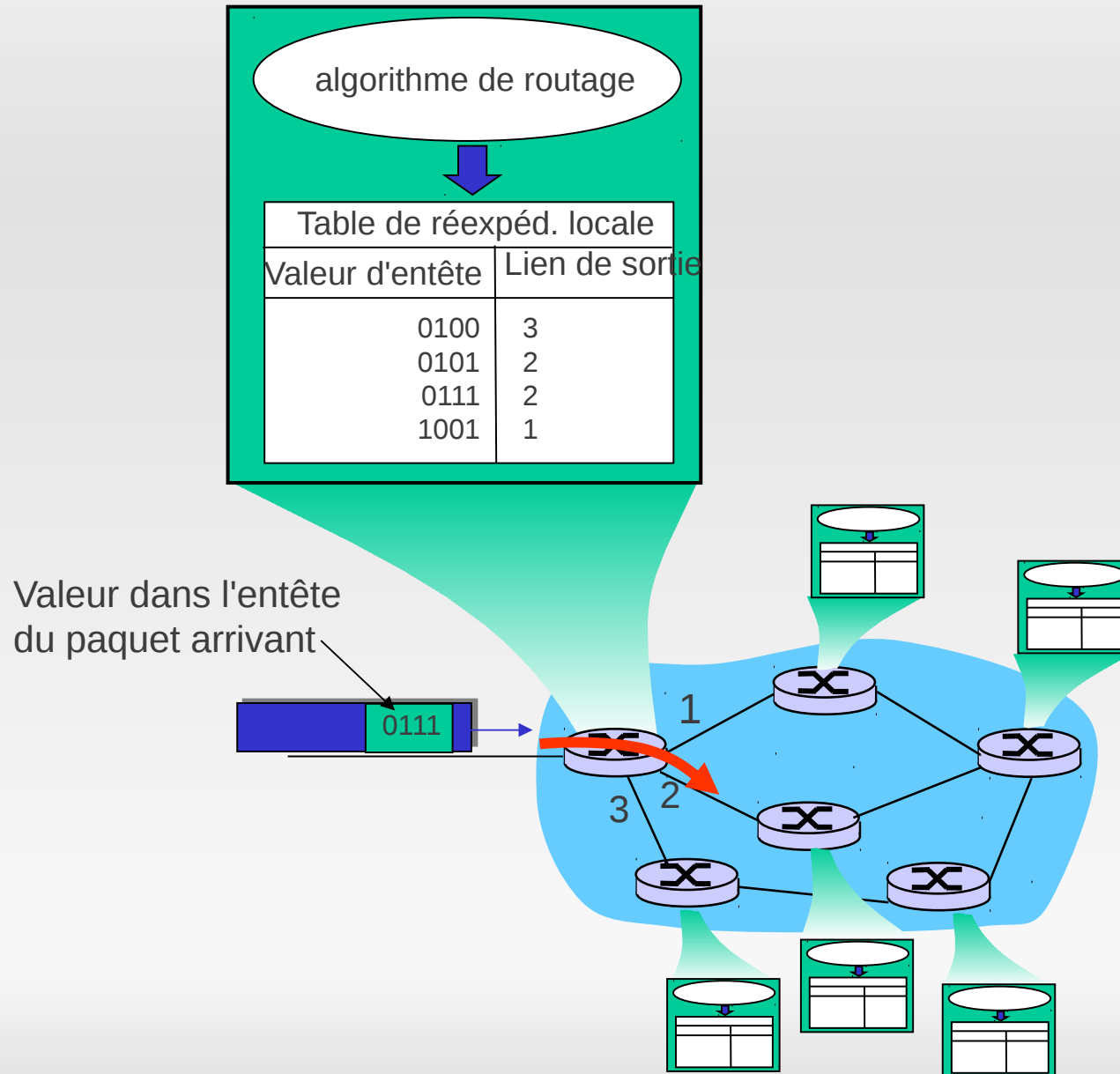
- Transport des paquets de l'expéditeur vers le destinataire
- Protocole de la couche réseau présent dans les hôtes mais aussi dans les routeurs

Deux fonctions importantes :

- *Choix de l'itinéraire à emprunter:*
Chemin pris par les paquets algorithme de routage
- *Réexpédiation:* déplacement des paquets d'un routeur à un autre sur le bon port



Interaction entre routage et réexpédition



Établissement de la connexion

- 3^{eme} fonction importante dans *certaines* architectures de réseau:
 - ATM, frame relay, X.25
- Avant l'écoulement des datagrammes, les deux hôtes *et* les routeurs concernés établissent une connexion virtuelle
 - Les routeurs sont alors partie prenante
- Le service de connexion réseau vs transport :
 - **Réseau:** entre deux hôtes (peut aussi faire intervenir des routeurs dans le cas des circuits virtuels)
 - **Transport:** entre deux processus

Modèle de service de la couche réseau

Q: Quel *modèle de service* pour un “canal” transportant des datagrammes de l'expéditeur au destinataire?

Exemple de service pour des datagrammes individuels:

- Livraison garantie
- Livraison garantie avec moins de 40 sec de délai

Exemple de service pour un flux de datagrammes:

- Datagrammes livrés dans l'ordre
- Une bande passante minimum garantie
- Préservation du temps entre les paquets

Service de la couche réseau:

Architecture Réseau	Modèle de Services	Garanties ?				Contrôle de Congestion
		Bandwidth	Perte	Ordre	Temps	
Internet	best effort	aucun	non	non	non	non (via perte)
ATM	CBR	constant	oui	oui	oui	non congestion
ATM	VBR	garantie	oui	oui	oui	non congestion
ATM	ABR	garantie minimum	non	oui	non	yes
ATM	UBR	aucune	non	oui	non	non

- Le modèle internet peut être étendu: Intserv, Diffserv
 - On le verra peut-être plus tard...

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagrammes

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagramme
- IPv4 adressage
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

Connexion de la couche réseau and service sans connexion

- Les réseaux à datagrammes sont des réseaux sans connexion
- Les réseaux à circuit virtuel (VC) sont des réseaux avec connexion
- Analogue aux services de la couche transport, mais:
 - **service:** hôte-à-hôte
 - **Pas de choix:** le réseau offre l'un ou l'autre des services
 - **Implémentation:** dans le coeur du réseau

Circuits virtuels (CV)

“chemin de la source à la dest. ressemble plus à un circuit téléphonique”

- Recherche de performance
- Le réseau agit tout au long du chemin allant de la source au destinataire

- Établissement de l'appel, et sa terminaison, lors de chaque appel avant que l'envoi de données puissent débuter
- Chaque paquet transporte un identifiant de circuit virtuel (pas d'adresse de destination)
- *Chaque* routeur sur le chemin source-destination maintient des “variables d'état” pour chaque connexion le traversant
- lien, ressource des routeurs (bande passante, tampons mémoires) peut être allouer à un CV (ressources dédiées = service préétabli)

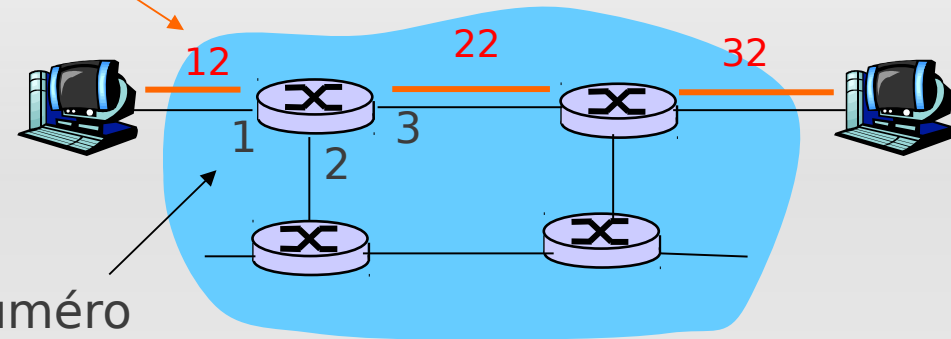
Implémentation de CV

Un CV consiste en:

1. Chemin de la source à la destination
 2. Numero de CV, un numéro pour chaque lien se trouvant sur le chemin
 3. Entrées dans les tables de réexpéditions des routeurs se trouvant sur le chemin
- Paquet appartenant à un CV transporte le numéro du CV (plutôt qu'une adresse de destination)
 - Numéro CV peut être change sur chaque lien.
 - Nouveau nombre CV vient d'une table de réexpédition

Table de réexpédiation

Numéro du CV



Numéro d'interface

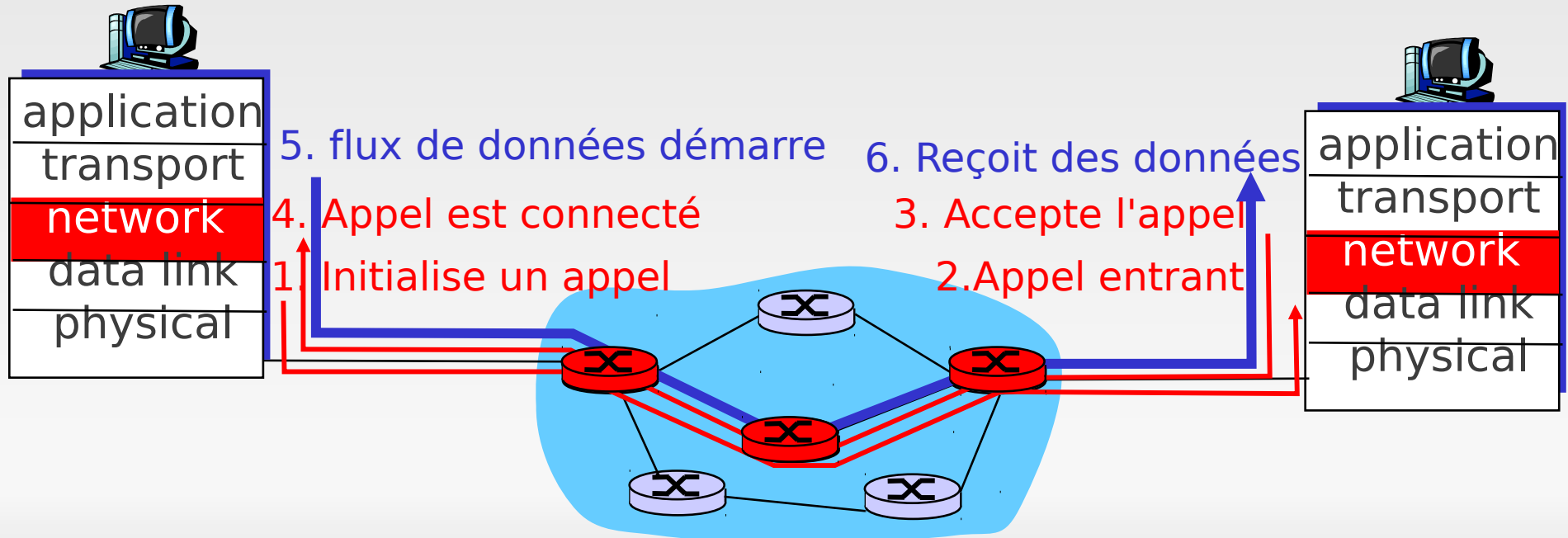
Table de réexpédiation dans le routeur du nord-ouest:

Interface entrante	VC # entrant	Interface sortante	VC # sortant
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Les routeurs maintiennent l'information sur l'état de la connexion!

Circuits virtuels: protocoles de signalement

- Utilisé pour établir, maintenir et fermer un VC
- Utilisé dans ATM, frame-relay, X.25
- N'est pas utilisé dans l'internet d'aujourd'hui



Réseau à datagramme

- Pas d'établissement de connexion au niveau réseau
- Routeurs: pas d'état concernant les connexions bout-en-bout
 - Pas de concept de connexion au niveau réseau
- Réexpédition de paquet utilisant l'adresse du destinataire
 - Des paquets peuvent prendre des chemins différents entre la source et la destination

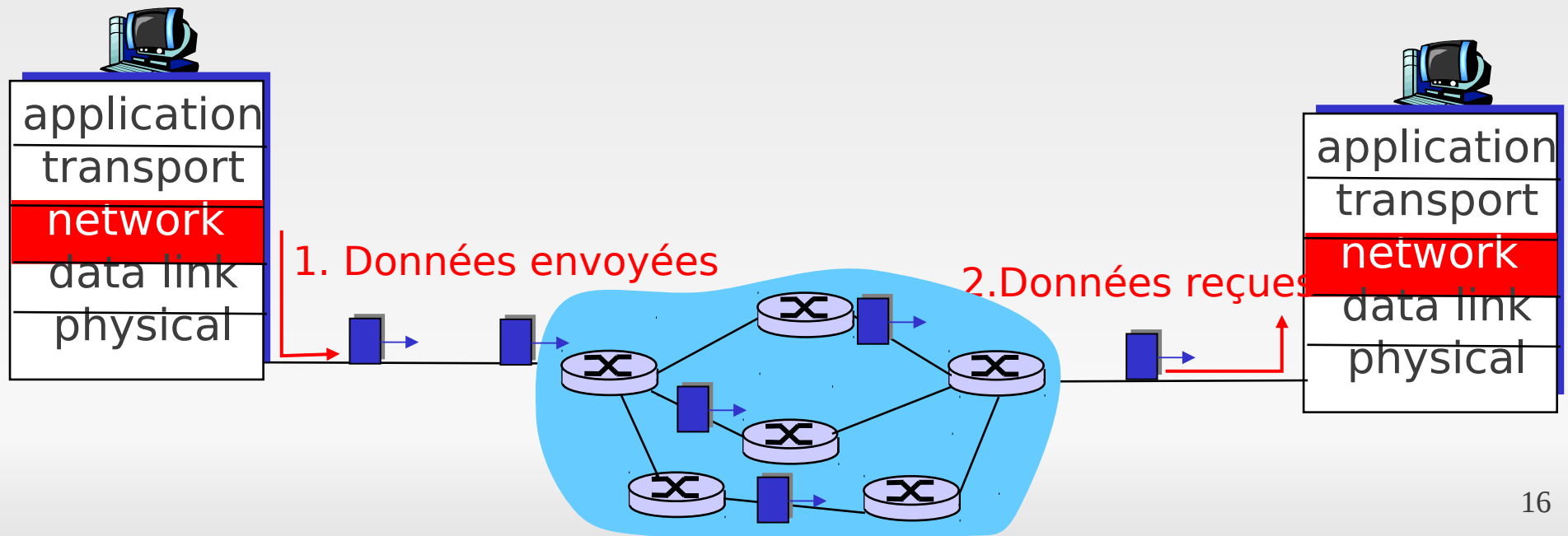


Table de routage/réexpédition

4 milliard d'
entrées possibles

<u>Eventail d'adresse de dest.</u>	<u>Interface sortie</u>
11001000 00010111 00010000 00000000 à travers	0
11001000 00010111 00010111 11111111	
11001000 00010111 00011000 00000000 à travers	1
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 à travers	2
11001000 00010111 00011111 11111111	
sinon	3

Résolution par plus long préfixe

<u>Préfixe</u>	<u>Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
sinon	3

Exemples

DA: 11001000 00010111 00010**110** 10100001

Quelle interface?

DA: 11001000 00010111 00011**000** 10101010

Quelle interface?

Datagramme ou CV : pourquoi ?

Internet (datagramme)

- Donnée échangée entre ordinateurs
 - Service “élastic”, pas de gestion strict du temps req.
- Ordinateurs “intelligent”
 - Peuvent s’adapter, faire du contrôle, de la gestion d’erreur
 - Réseau simple, complexité au extrémitées
- Lien de plusieurs sortes
 - Caractéristiques différentes
 - Difficultés d’avoir un service uniforme

ATM (CV)

- Héritier du téléphone
- Conversation humaine :
 - Temps strict, confiance nécessaire
 - Besoin de services garantis
- Extrémité du réseau très simple
 - Téléphones
 - Complexité dans le réseau

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- IPv4 adressage
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

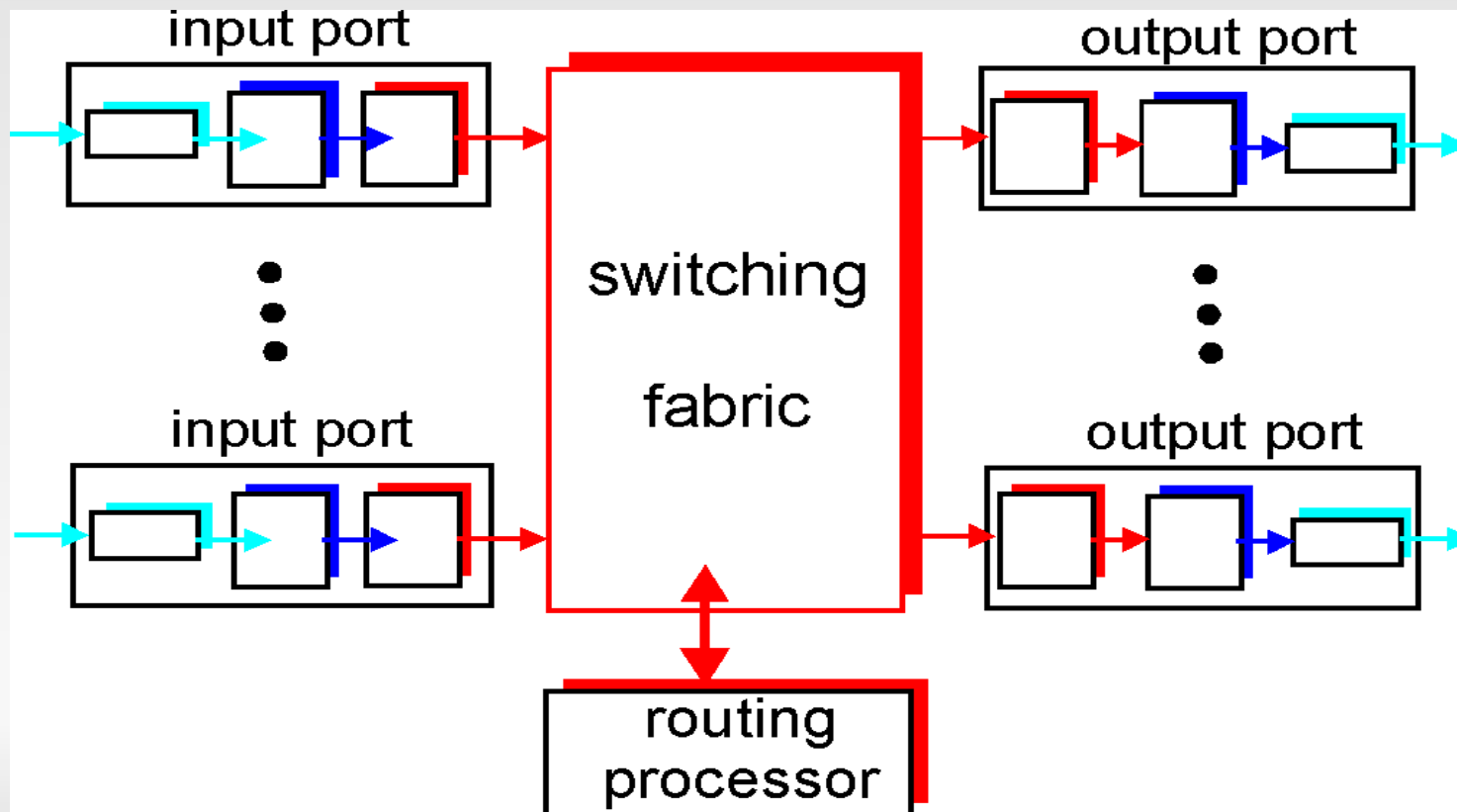
- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

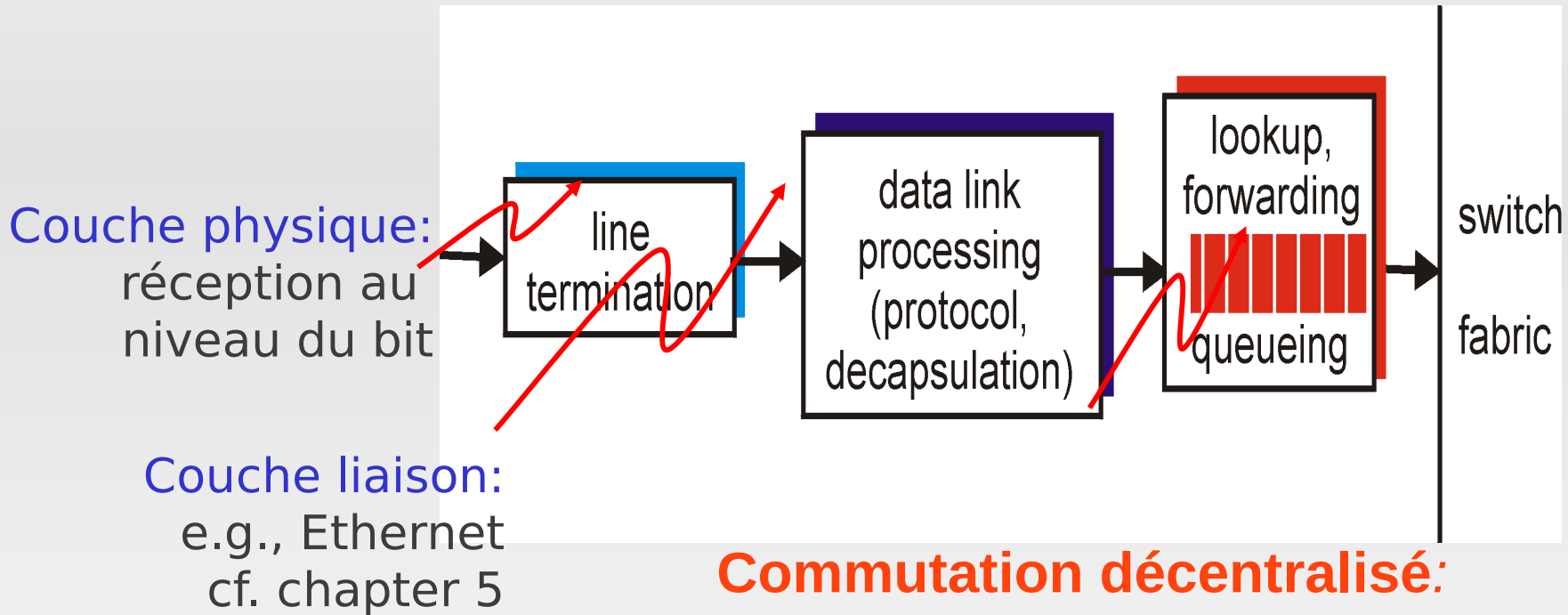
Vue d'ensemble de l'architecture d'un routeur

Deux fonctions clefs d'un routeur:

- Exécution d'algorithmes/protocoles de routage (RIP, OSPF, BGP)
- *Réexpédition* des datagrammes entrant vers les ports de sortant



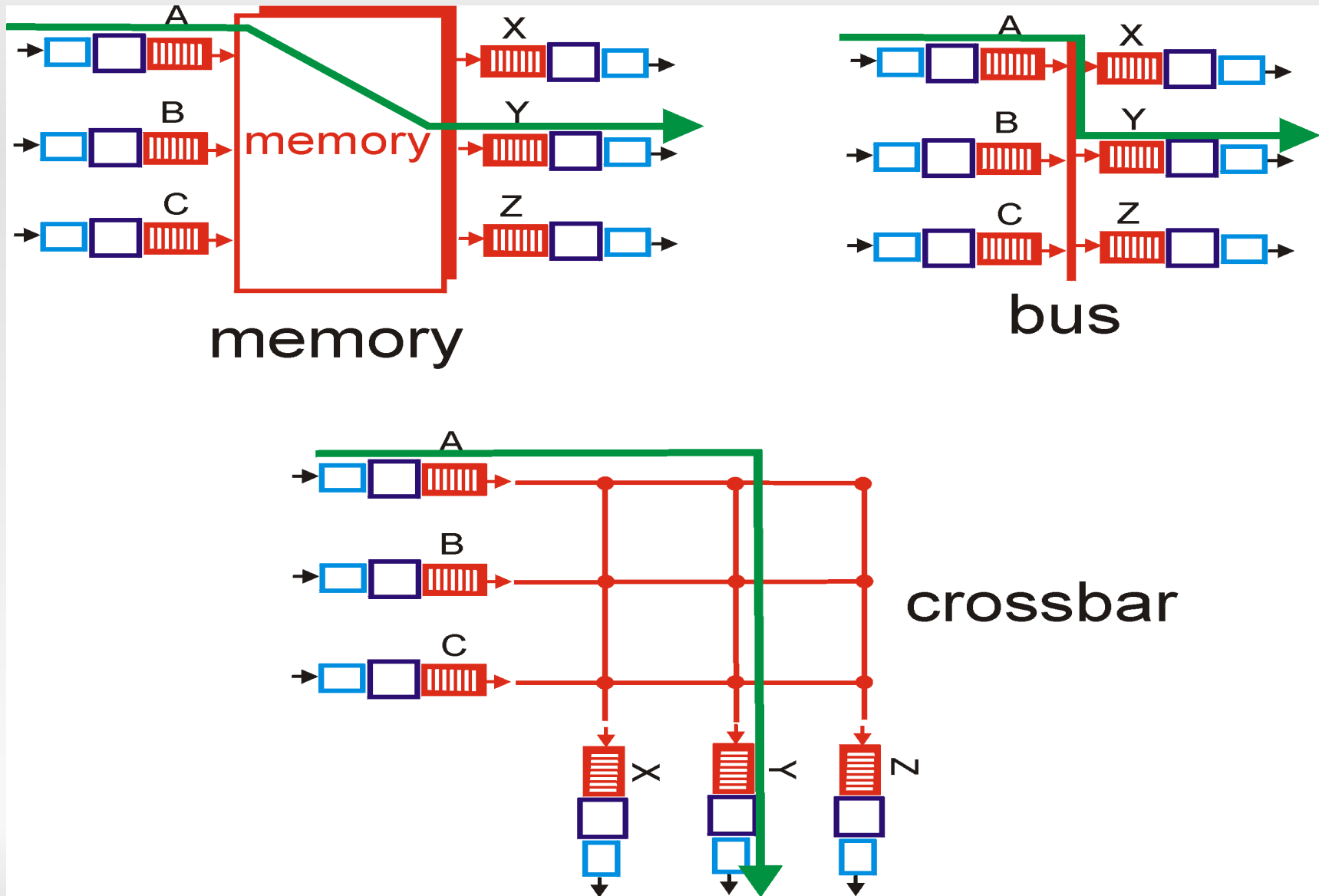
Fonction au niveau des ports



Commutation décentralisé:

- Pour une destination de datagramme donnée, un accès à une table de routage donne le port de sortie
- But: traiter les données entrantes à la vitesse de la liaison "line speed"
- Mise en file d'attente: si des datagrammes arrivent plus vite que n'est opérée la commutation dans la fabrique de commutation (switch fabric)

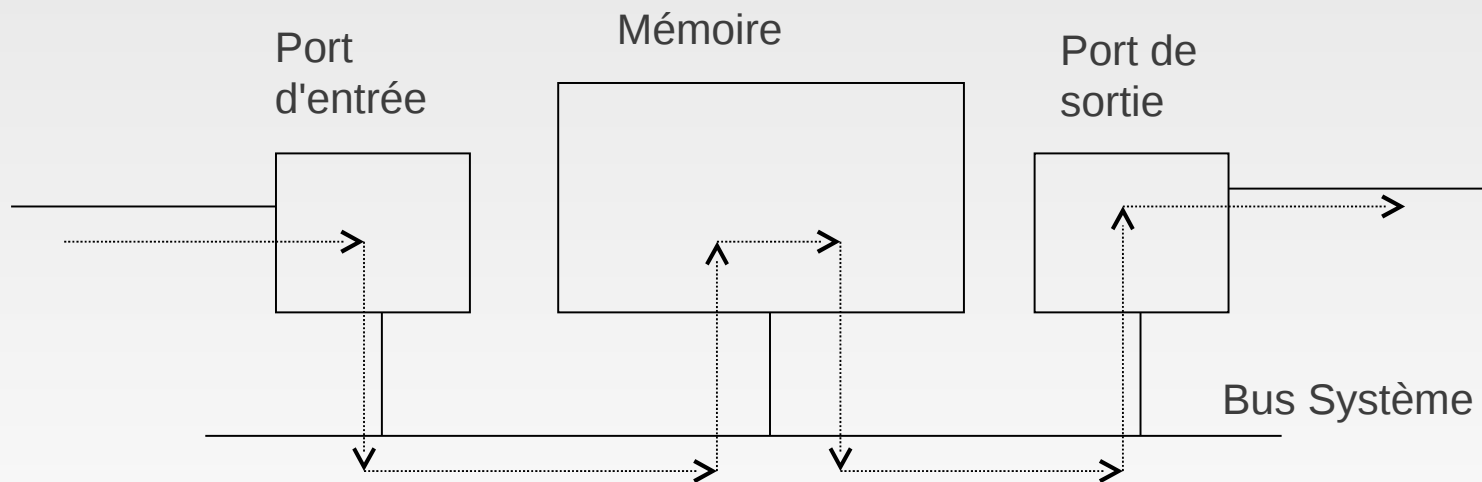
Trois types de "switching fabric"



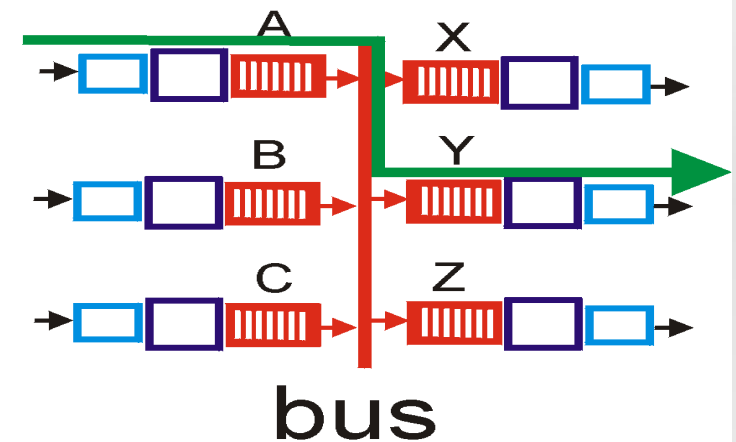
Commutation avec de la mémoire

Routeurs de première génération:

- Ordinateur classique avec une commutation sous contrôle du CPU
- Paquet copié sur la mémoire système
- Vitesse limitée par la bande passante de la mémoire (2 croisements de bus par datagramme)



Commutation avec un bus

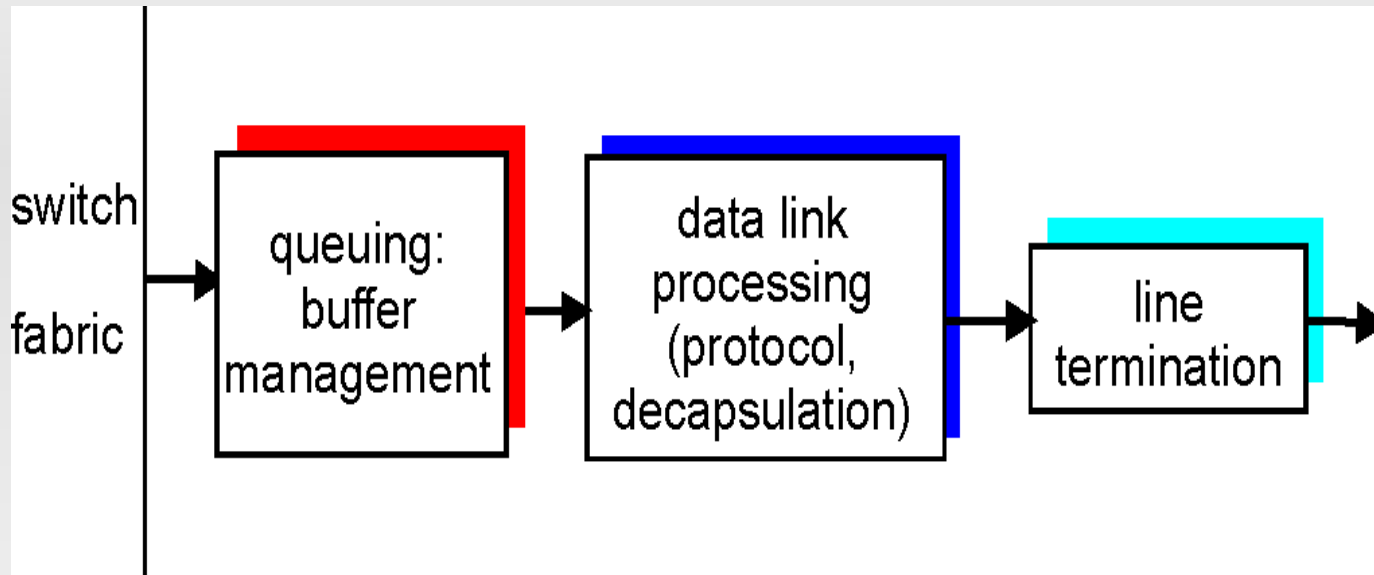


- Datagramme du port d'entrée de la mémoire au port de sortie la mémoire est transférée via un bus partagé.
- La vitesse de commutation est limitée par la bande passante du bus
- Bus d'1 Gbps, Cisco 1900: vitesse suffisante pour des routeurs d'accès d'entreprise (pas réseau régional ou pour les routeurs dits "colonne vertébrale" d'internet)

Commutation via un réseau d'interconnexion

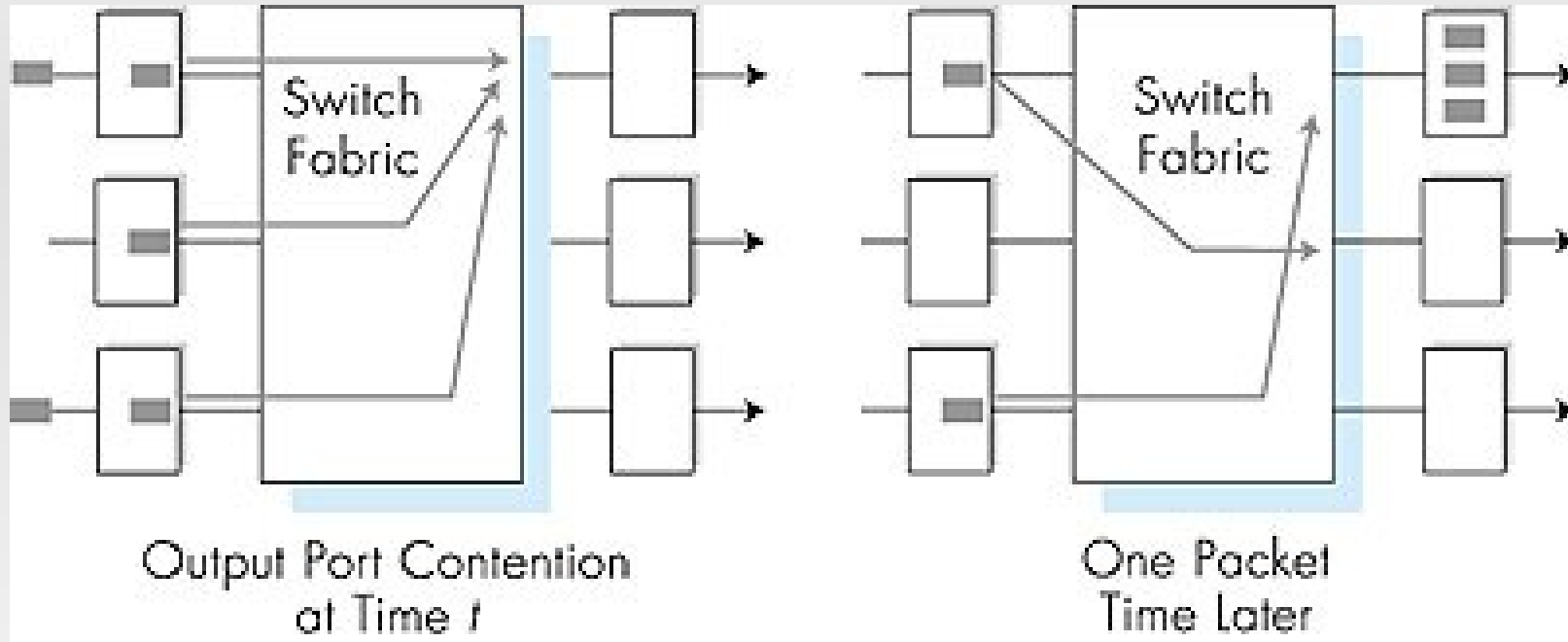
- Dépasser la limitation liée à la bande passante du bus
- Réseaux Banyan, autre réseau d'interconnexion initialement développé pour connecter des processeurs dans les multi-processeurs
- Design avancé: fragmentation des datagrammes en des cellules de taille fixe, commute les cellules à travers la switch fabric.
- Cisco 12000: commute Gbps à travers le réseau d'interconnexion

Port de sortie



- *Mise en tampon mémoire* nécessaire quand les datagrammes arrivent d'une fabrique plus vite que le taux de transmission
- *Discipline d'ordonnement* choisit les datagrammes à transmettre parmi les datagrammes dans la file d'attente

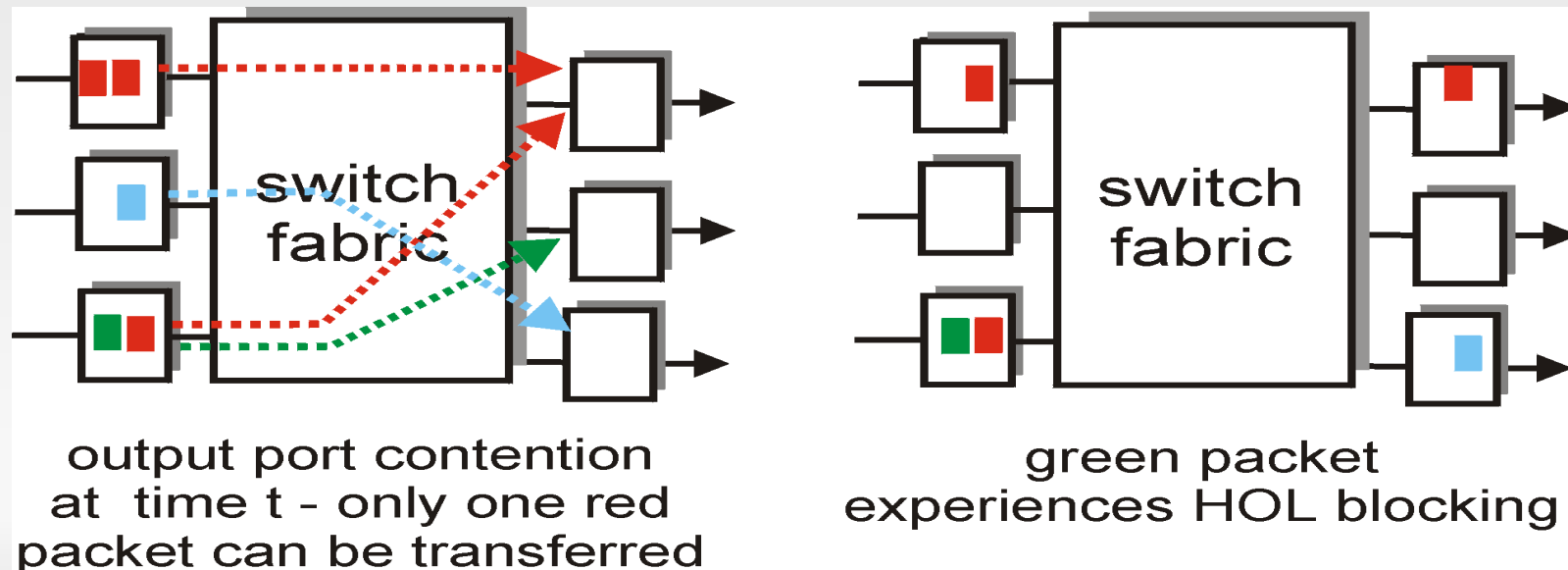
Mise en attente sur les ports de sortie



- Stockage sur un tampon mémoire lorsque le taux arrivées via le la switch fabric dépasse le débit des liaisons sortantes
- *Mise en tampon mémoire* quand les datagrammes plus vite que la vitesse de la liaison de sortie
- *Attente dans la file (délai) et perte dû à un débordement au niveau des port de sorti*

Mise en attente sur les ports de sortie

- La switch Fabric plus lente que les ports d'entrée combinés
-> la file d'attente peut se produire sur les ports d'entree
- **Blocage Head-of-the-Line (HOL):** les datagrammes en début de file d'attente empêche les autres datagrammes d'avancer
- *Délai d'attente et perte dû au débordement des ports d'entrée!*



Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- IPv4 adressage
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

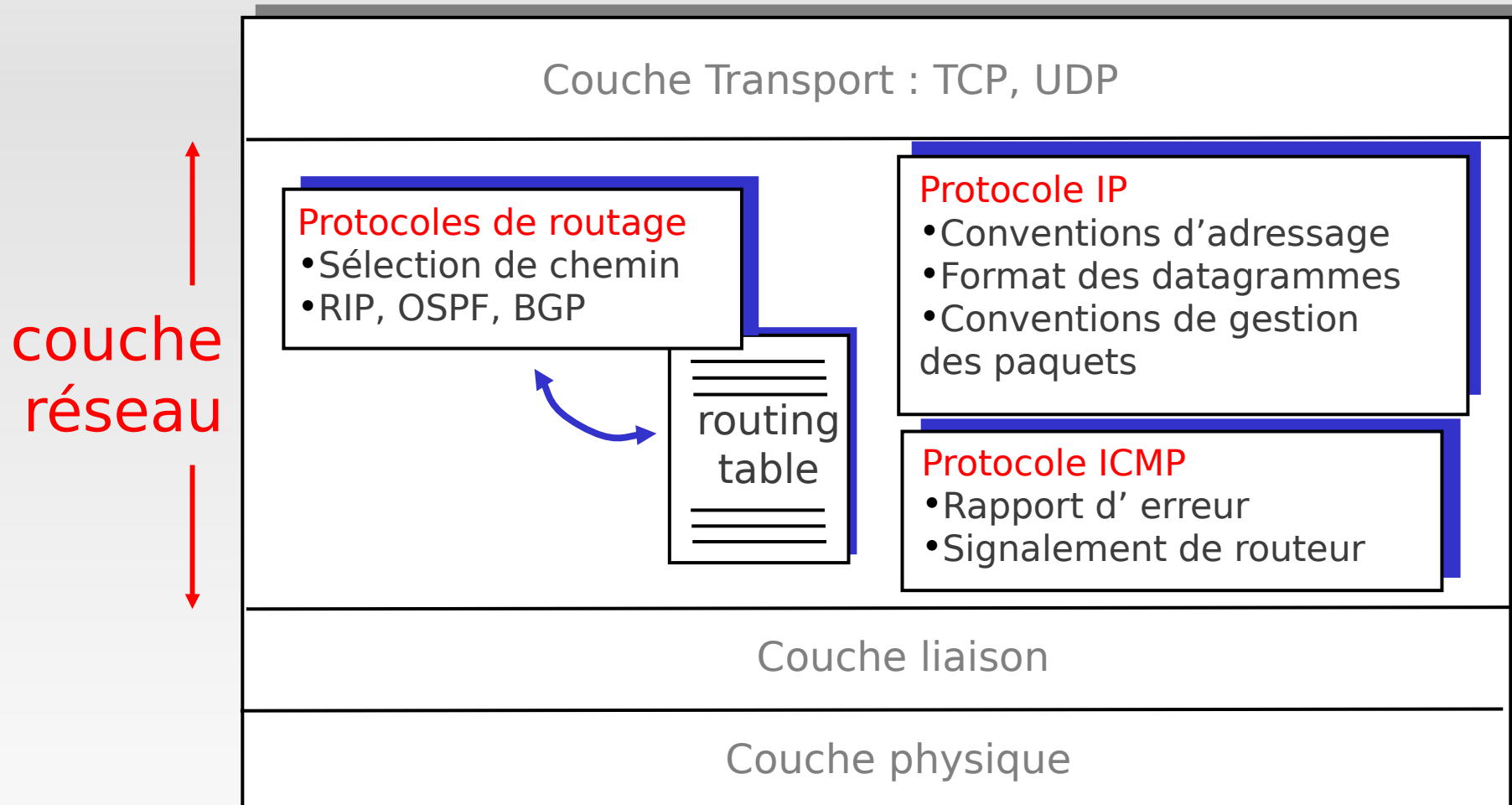
4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

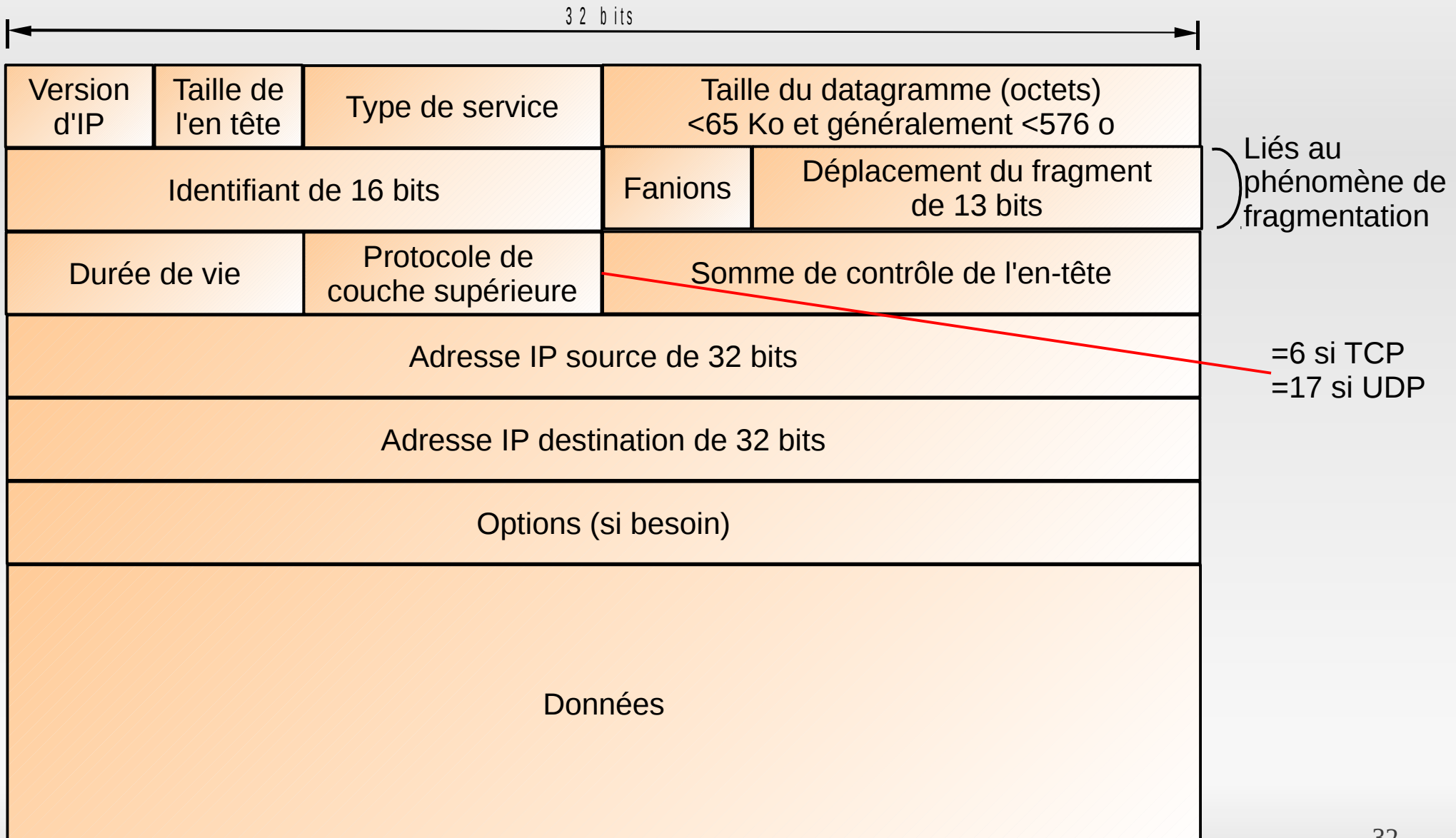
4.7 routage *broadcast* et *multicast*

Protocole Internet (IP)

fonctions de la couche réseau des routeurs :

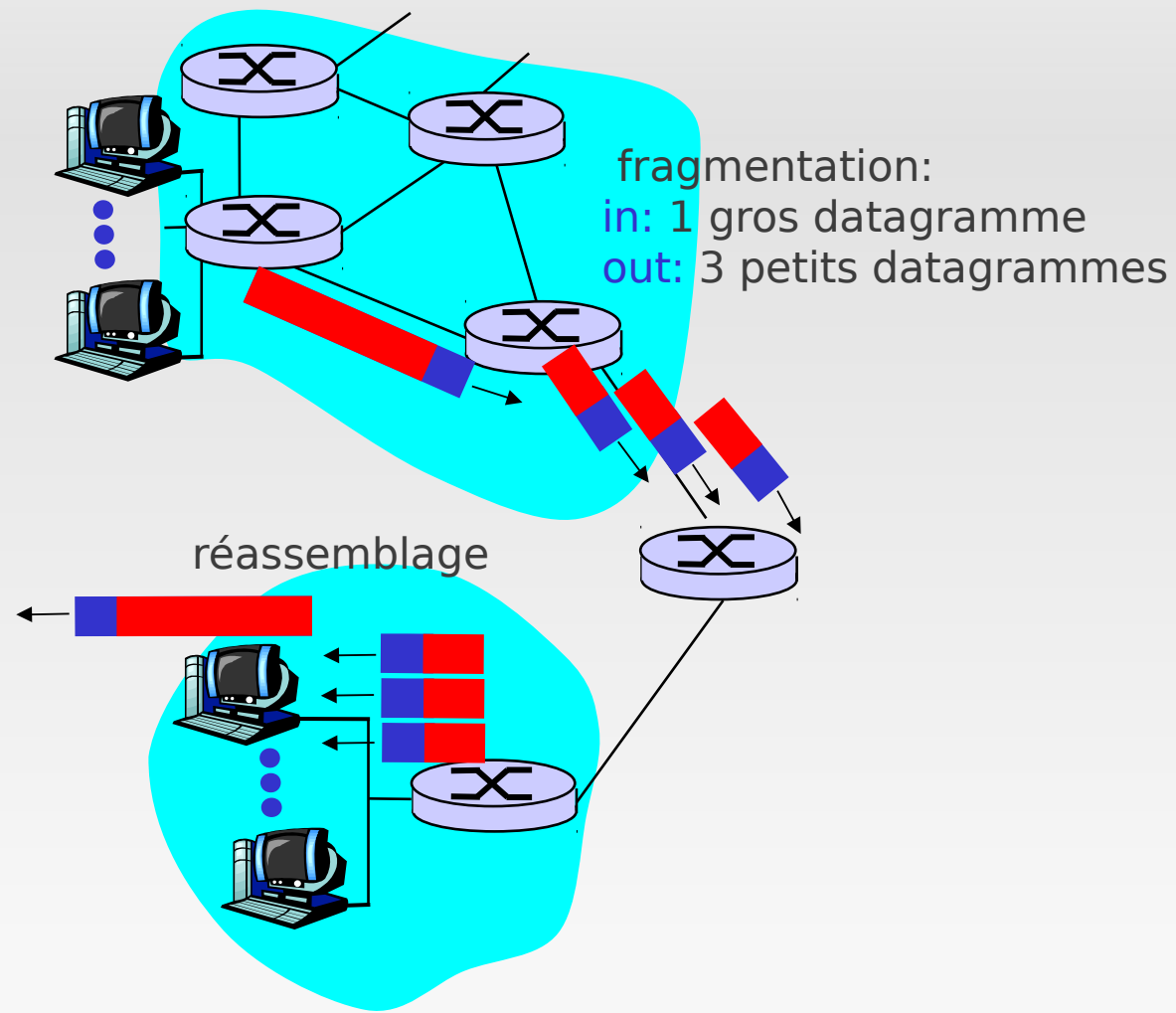


Format du datagramme IP



Fragmentation des datagrammes IP

- Les liaisons sur le réseau ont des MTU (max.transfer unit) – taille du plus gros paquet admis sur la liaison
 - Différents types de liens, différents MTUs (Ethernet 1.5Ko, Internet 576 o)
- Gros datagramme IP fragmenté sur le réseau
 - Un datagramme devient plusieurs datagrammes
 - Le réassemblage se fait à la destination finale
 - Les bits de l'en-tête IP sont utilisés pour identifier et ordonner les fragments



Fragmentation IP et Ré-assemblage

	length =4000	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

1 gros datagramme devient
plusieurs petits datagrammes

	length =1500	ID =x	fragflag =1	offset =0	
--	-----------------	----------	----------------	--------------	--

	length =1500	ID =x	fragflag =1	offset =1480	
--	-----------------	----------	----------------	-----------------	--

	length =1040	ID =x	fragflag =0	offset =2960	
--	-----------------	----------	----------------	-----------------	--

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et reseaux a datagram

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagramme
- **adressage IPv4**
- IPv6

4.5 Algorithmes de routage

- Etat de lien
- Vecteur de distance
- Routage hiérarchique

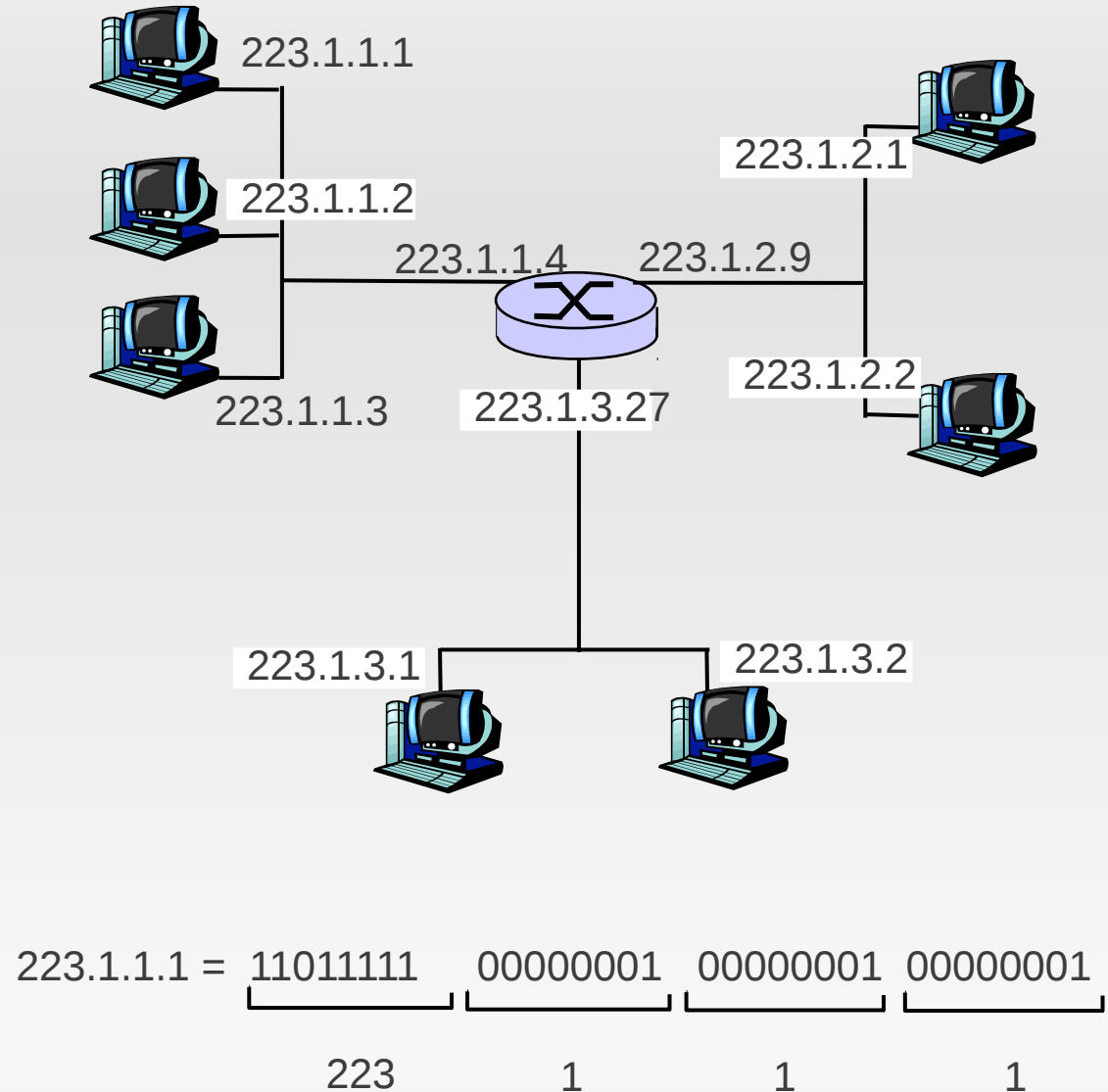
4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

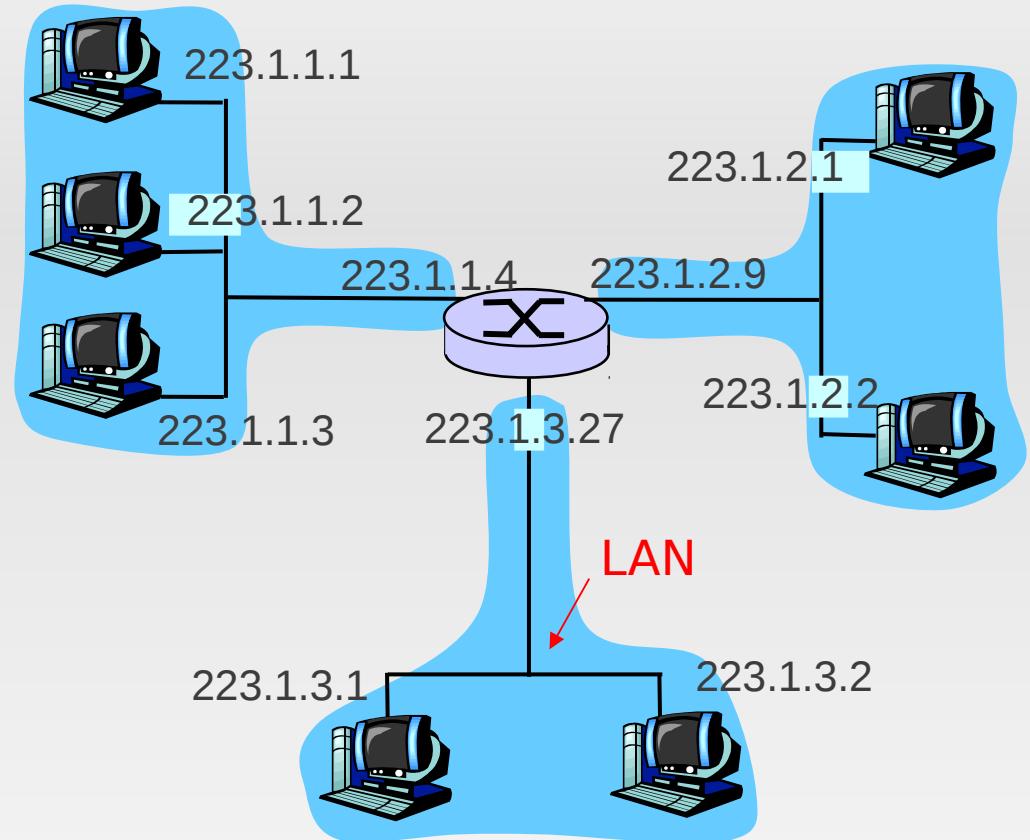
Adressage IP : introduction

- **Adresse IP:**
Identifiant sur 32 bit pour les hôtes, les interfaces des routeurs
- **Interface :**
Connexion entre les hôtes , les routeurs et les liens physiques
 - Un routeur à plusieurs interfaces
 - Un hôte peut avoir plusieurs interfaces
 - L'adressage IP est associé avec les interfaces et non avec les hôtes où les routeurs



Adressage IP

- Adresse IP:
 - Bit de poids fort : partie réseau
 - Bit de poids faible : partie hôte
- *Qu'est ce qu'un réseau ?* (d'un point de vue IP)
 - interfaces avec des adresse IP avec la même partie réseau
 - L'un et l'autre peuvent s'atteindre physiquement sans modification dans le routeur



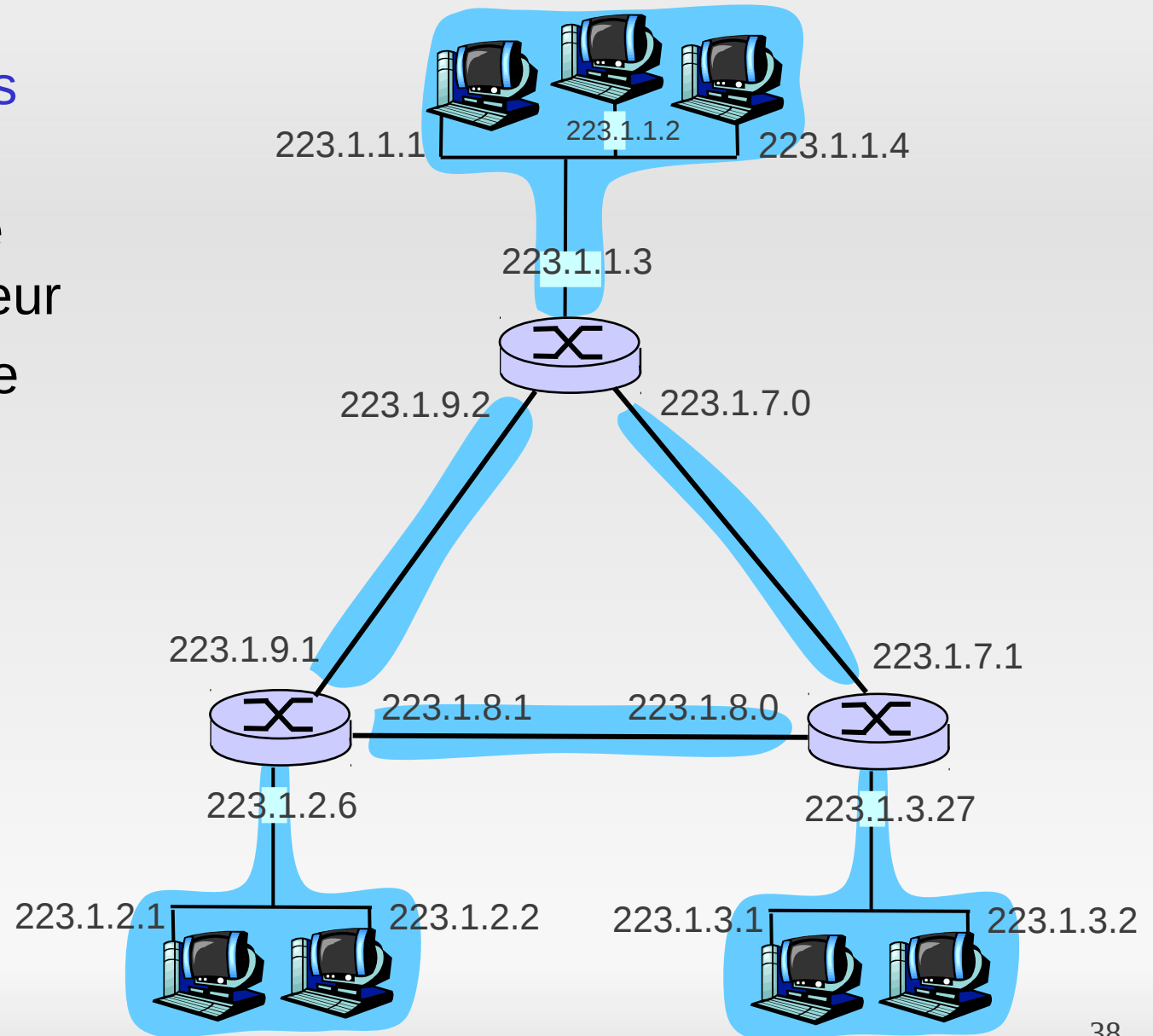
Réseau constitué de 3 réseaux IP

Adressage IP

Comment trouver les réseaux?

- Détacher chaque interface du routeur
- Créer des îlots de réseaux isolés

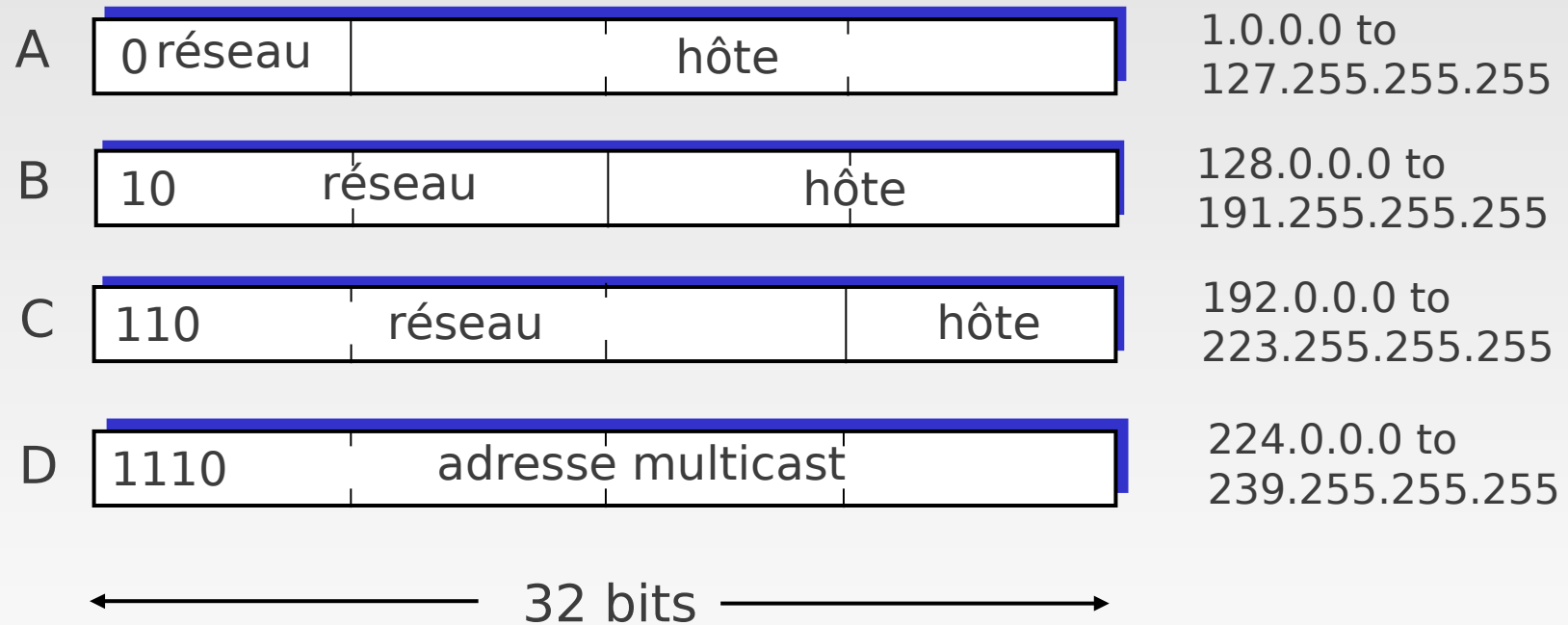
Système interconnecté constitué de 6 réseaux



Classes d'adresse IP

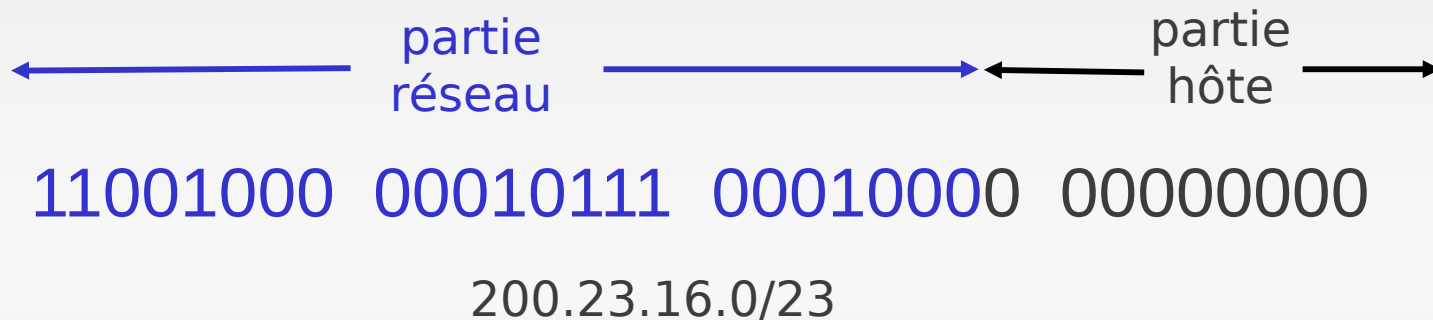
Après la notion de “réseau”, examinons les adresses IP :

Classes



Adressage IP : CIDR

- Utilisation strict des classes :
 - Utilisation inefficace de l'espace d'adressage
 - Exp. : classe B peut allouer 65 000 hôtes, même si il n'y en a que 2000
- **CIDR: Classless InterDomain Routing**
 - Portion de l'adresse dédiée au réseau
 - Format : **a.b.c.d/x**, où x est le numéro de bits de l'adresse dédiée au réseau

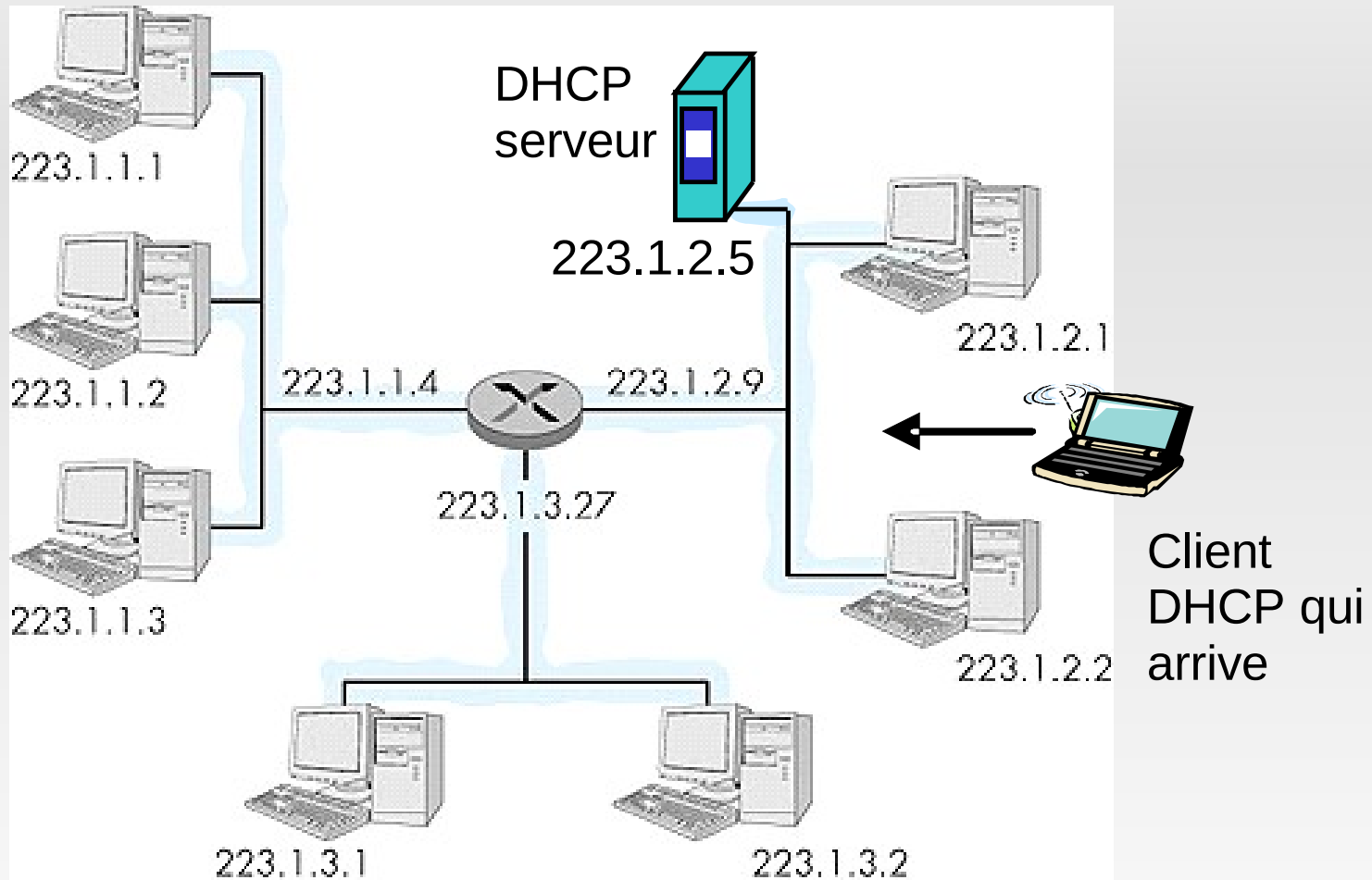


Adresses IP : comment en obtenir une ?

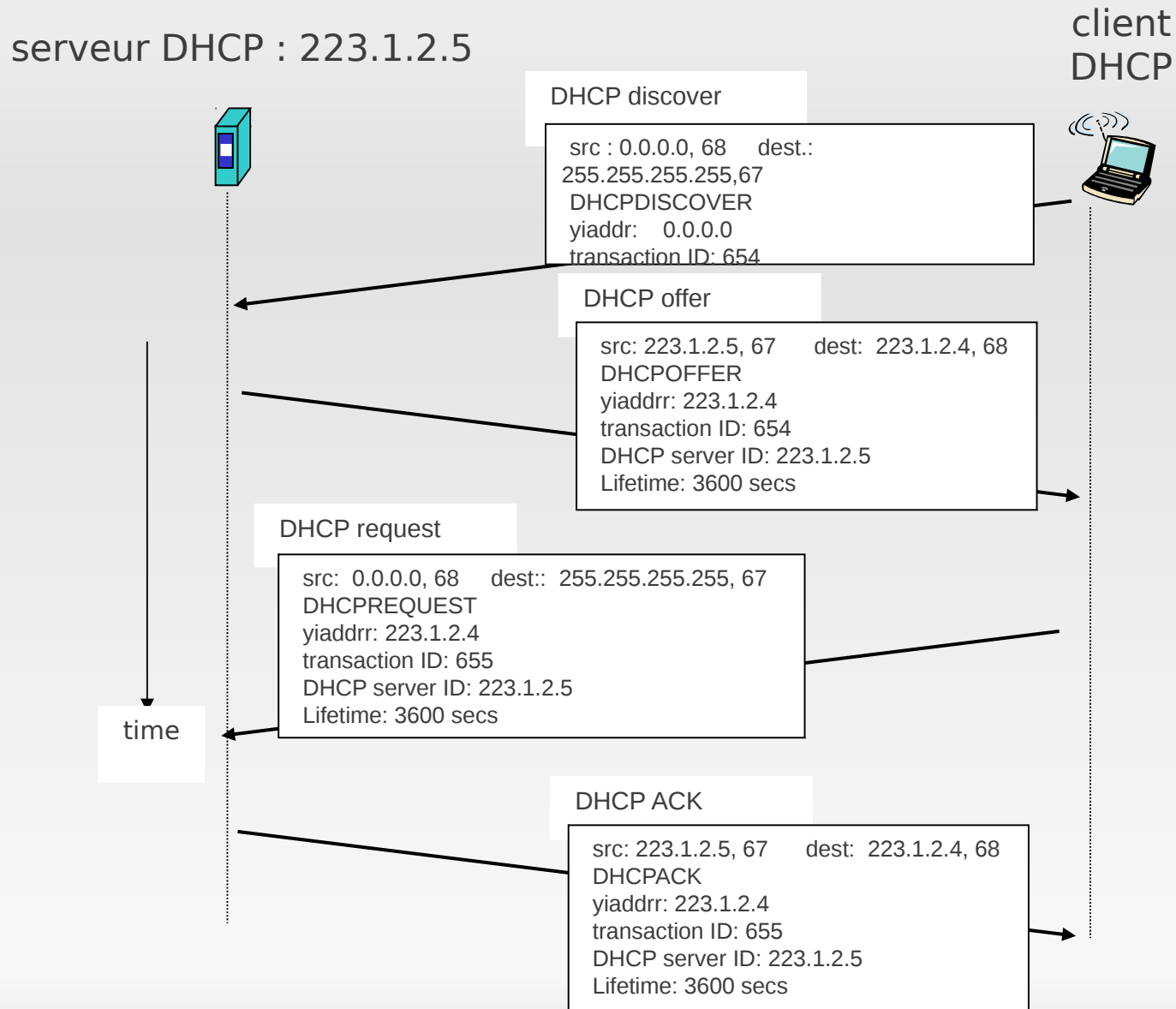
Q: comment les hôtes obtiennent une adresse IP ?

- Codé en dur par l'admin système dans un fichier
 - Windows: control-panel->network->configuration->tcp/ip->properties
 - UNIX: /etc/rc.config
- **DHCP**: **D**ynamic **H**ost **C**onfiguration **P**rotocol: obtention dynamiquement d'une adresse : "plug-and-play"
 - Apprécié des admin réseau (évite la configuration poste par poste)
 - Gestion d'un pool d'adresse d'une quantité inférieure au nombre de client potentiel(FAI)
 - Succès accru due à la mobilité des appareils se connectant à un réseau.
 - L'hôte broadcasts avec des msg "DHCP discover"
 - Le serveur DHCP répond avec un msg "DHCP offer"
 - L'hôte demande une @ IP avec un msg "DHCP request"
 - Le serveur DHCP renvoie une adresse avec un msg "DHCP ACK"

Scénario DHCP client-serveur



Scénario DHCP client-serveur



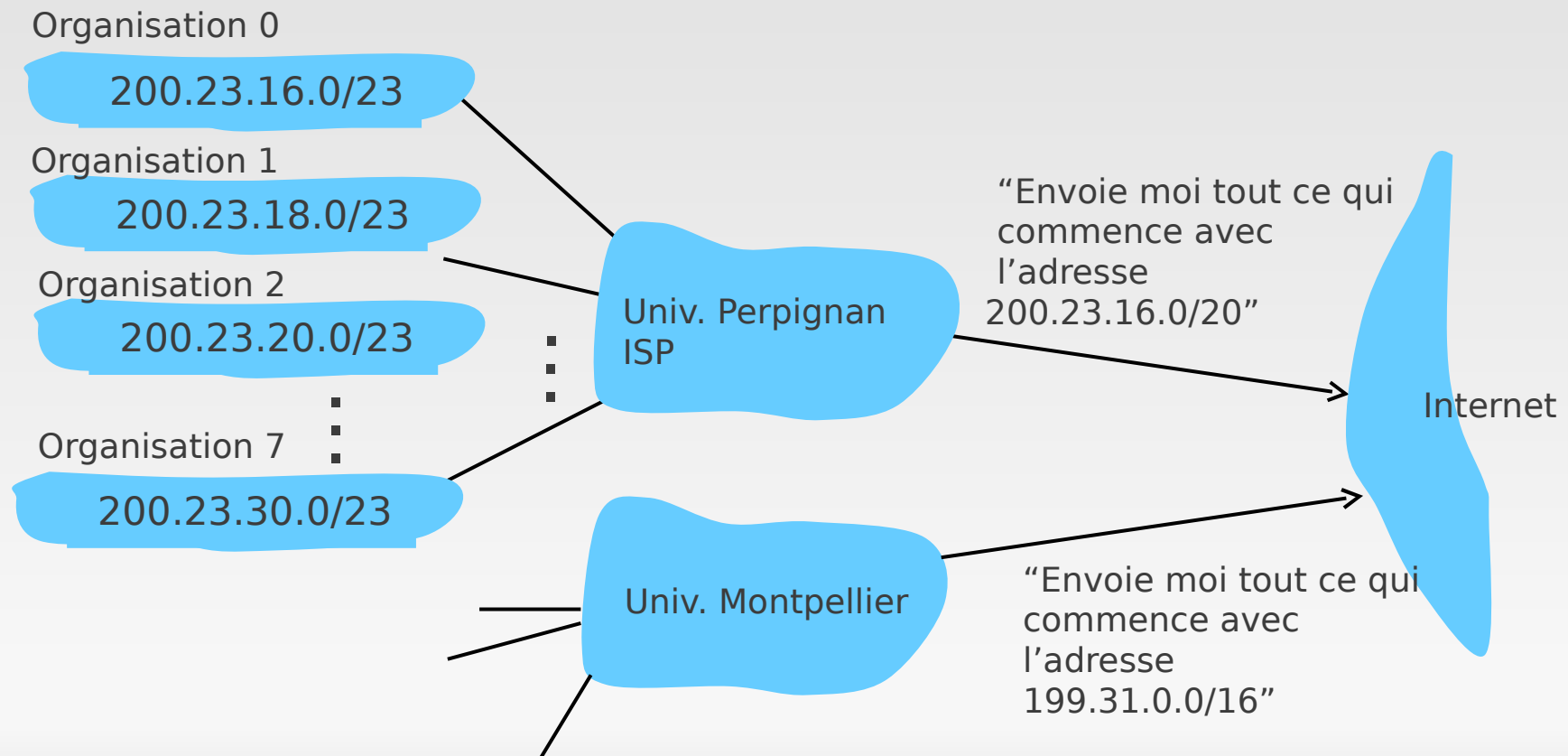
Adresses IP : comment en obtenir une ?

Réseau (portion de réseau):

Bloc du FAI	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organisation 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organisation 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organisation 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...
Organisation 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

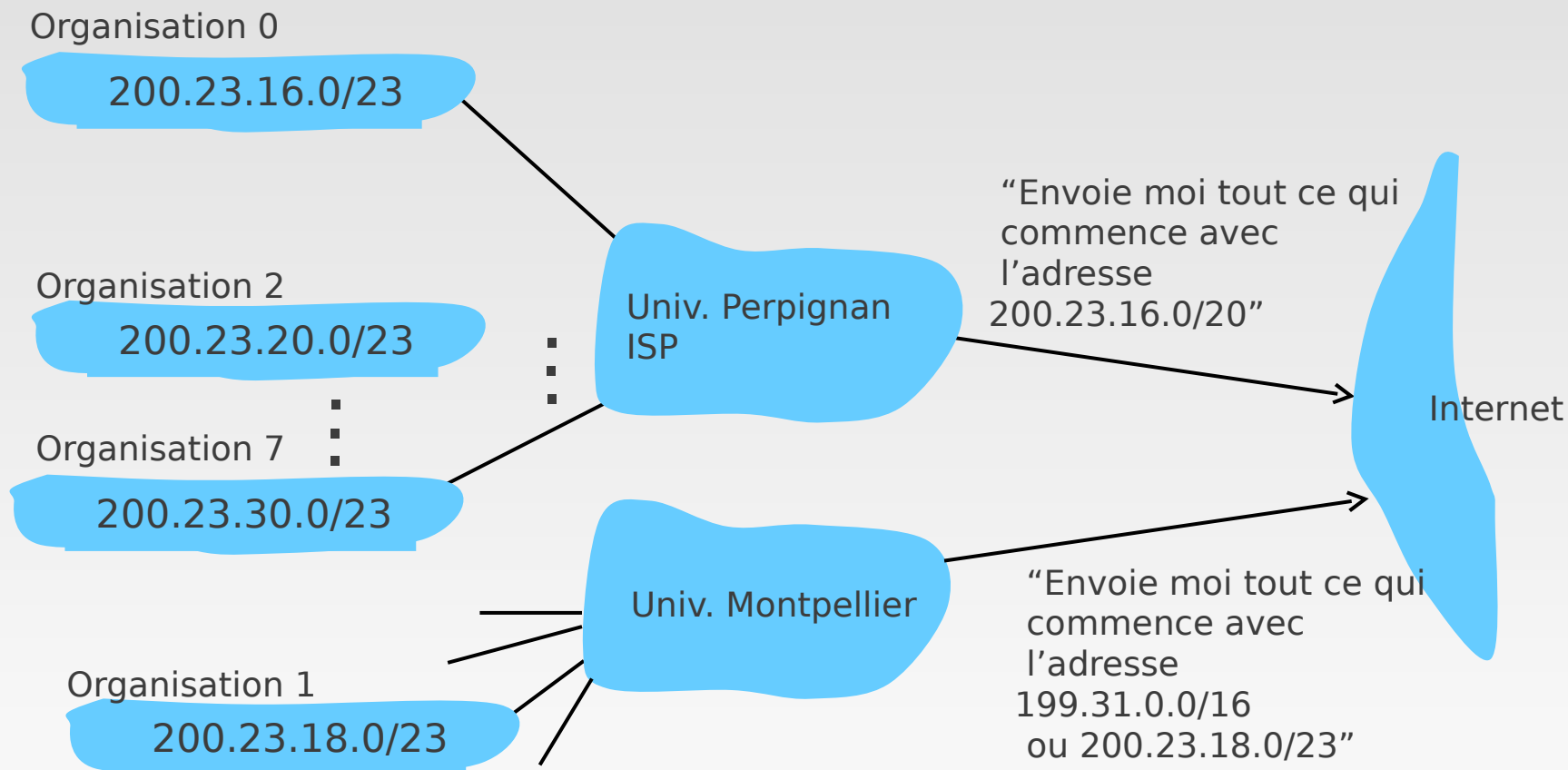
Adressage hiérarchique : association de route

L'adressage hiérarchique permet un routage de l'information efficace



Adressage hiérarchique : routage plus spécifique

Univ. Montpellier a une route spécifique vers l'organisation 1



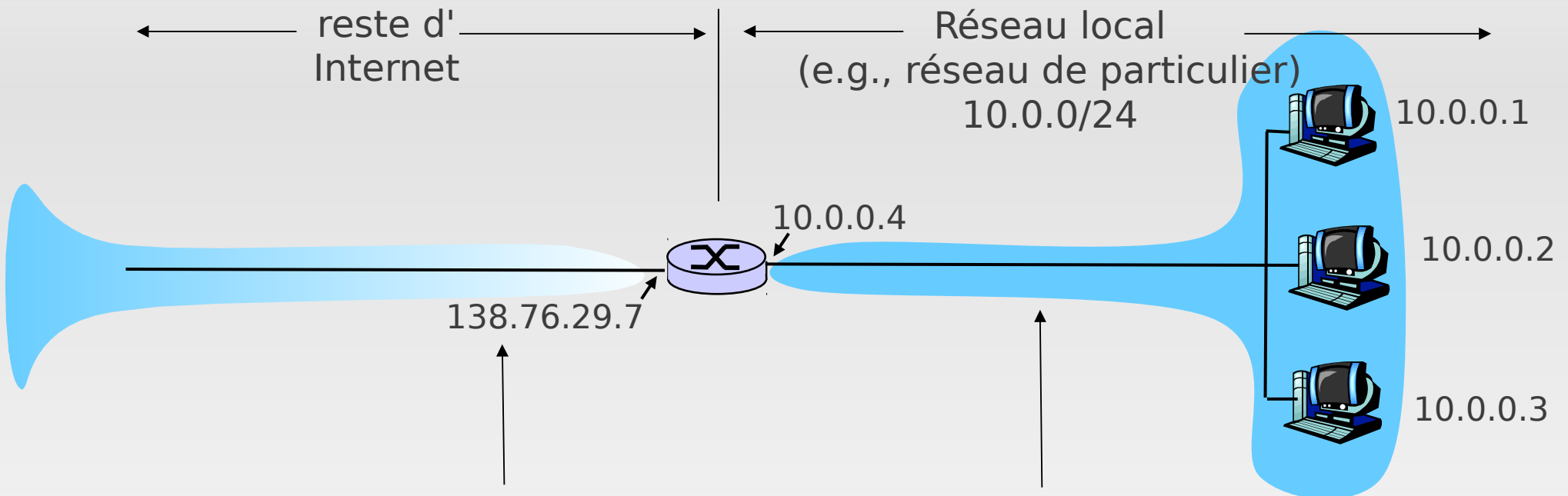
Adressage IP : le dernier mot...

Q: Comment un FAI obtient un block d'adresse ?

R: **ICANN**: Internet **C**orporation for **A**ssigned
Names and **N**umbers

- Allocation d'adresse
- Gestion des DNS
- Assignation des noms de domaines, résolution des disputes

NAT: Network Address Translation



Tous les datagrammes *quittant* le réseau local ont la *même* adresse NAT IP source: 138.76.29.7, mais différents numéros de port source

Datagrammes avec une source ou une destination dans ce réseau ont 10.0.0/24 pour adresse pour la source ou la destination (comme d'hab.)

NAT: Network Address Translation

- **Motivation:** un réseau local utilise juste une adresse IP pour autant que le monde extérieur est concerné:
 - De multiples adresses IP provenant de l'ISP ne sont plus nécessaires: juste une adresse IP pour tous les appareils
 - On peut changer les adresses des appareils dans le réseau local sans en référer au monde extérieur
 - On peut changer d'ISP sans changer l'adresse des appareils dans le réseau local
 - Les appareils à l'intérieur du réseau local ne sont pas explicitement adressables ou visibles du monde extérieur (une sécurité supplémentaire).

NAT: Network Address Translation

Implémentation: le routeur NAT doit:

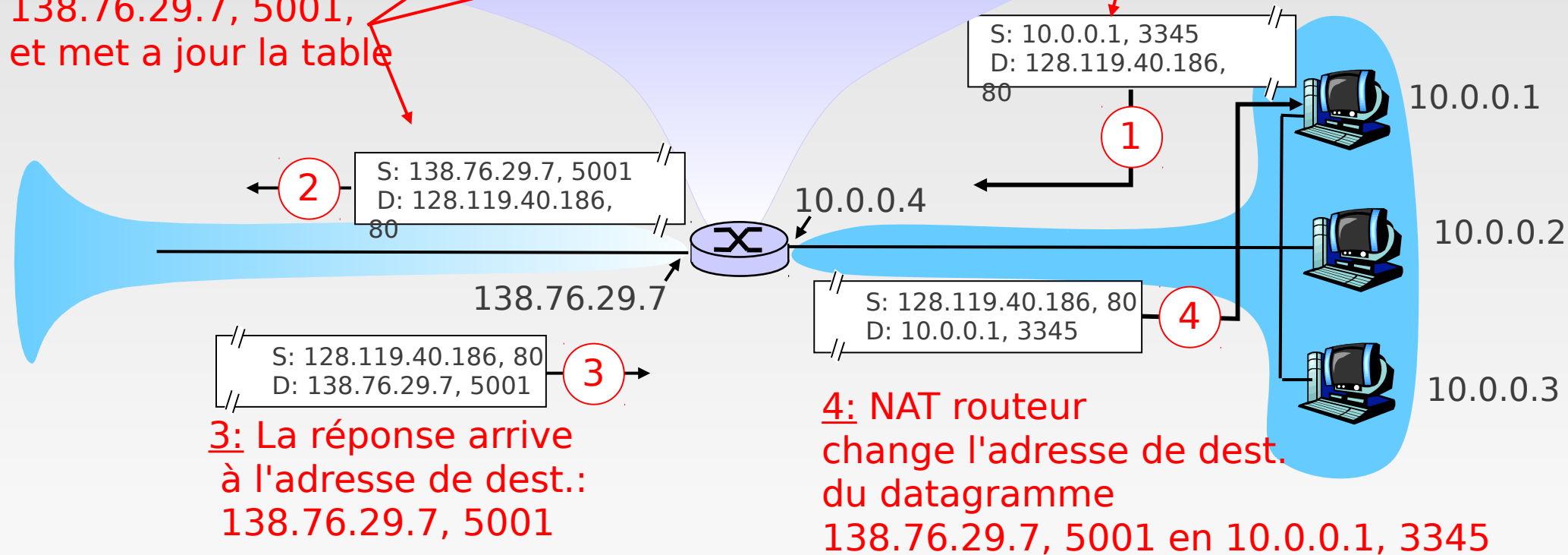
- *Pour les datagrammes sortant:* remplacer (adresse IP de la source et le port #) de chaque datagramme sortant par (adresse IP du NAT et un nouveau port #)
... clients/serveurs lointains répondront avec (adresse IP du NAT IP le nouveau port #) comme adresse de destination.
- *Rappelez vous que (dans la table de translation du NAT)* on associe à chaque (adresse IP de la source et le port #) à (adresse IP du NAT et nouveau port #)
- *Les datagrammes entrant:* remplacer (adresse IP du NAT et le nouveau port #) dans le champs de destination de chaque datagramme par (l'adresse IP de la source adresse IP et le port #) correspondant stockés dans la table de translation du NAT

NAT: Network Address Translation

2: routeur NAT change l'adr. de la source dans le datagramme 10.0.0.1, 3345 en 138.76.29.7, 5001, et met a jour la table

Table de translation du NAT	
addr. côté WAN	addr côté LAN
138.76.29.7, 5001	10.0.0.1, 3345
.....

1: hôte 10.0.0.1 envoie un datagramme à 128.119.40.186, 80



NAT: Network Address Translation

- Champs de numéro de port sur 16-bit
 - 60,000 connexions simultanées avec un simple a adressage côté LAN!
- NAT est controversé:
 - Les routeurs devraient fonctionner seulement jusqu'à la couche 3 (réseau)
 - Viole le principe de bout-en-bout
 - Un NAT doit probablement être pris en compte par les développeurs d'applications, e.g., applications P2P
 - Le manque d'adresse devrait être résolu par IPv6

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

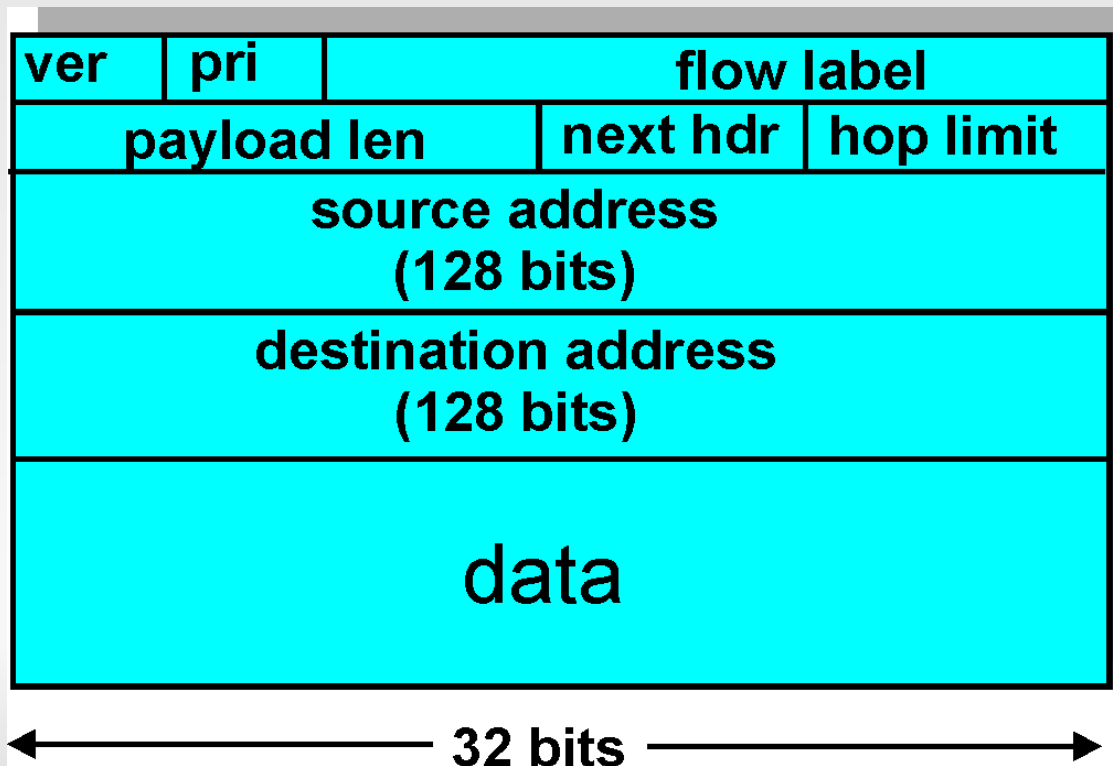
- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

- **Motivation initiale:** l'espace des adresses 32 sera bientôt totalement alloué.
- Autre motivation:
 - Le format des entêtes des paquets facilite le traitement et la réexpédition des paquets dans un routeur.
 - Les changements dans l'entête pour faciliter la qualité des services
- **Format des datagrammes IPv6 :**
 - Un entête de longueur fixe sur 40 octets
 - Pas de fragmentation autorisé

Entête IPv6

- Priorité:** identifie la priorité parmi les datagrammes.
- Label de flux:** identifie les datagrammes dans le même “flux.” (le concept de “flux” pas très bien défini).
- Prochaine entête:** identifie le protocole supérieur



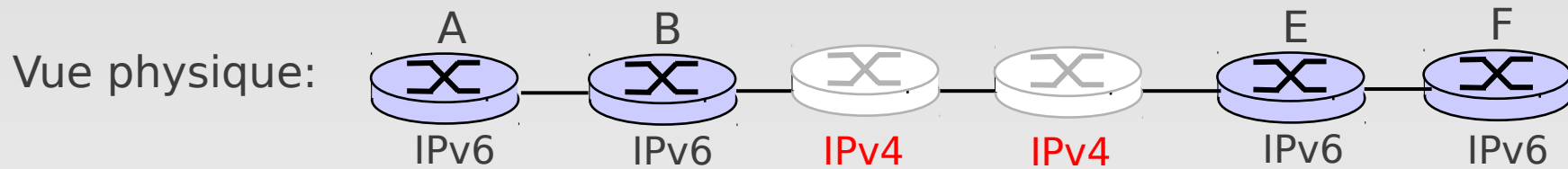
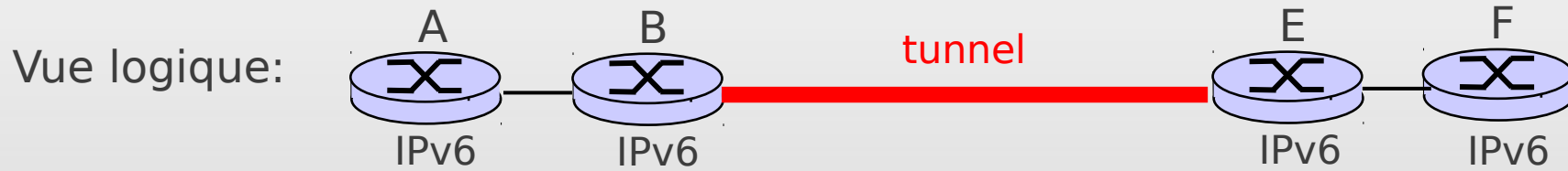
D'autres changements

- *Somme de contrôle (checksum)*: a été supprimée pour réduire le temps de traitement dans chaque routeur
- *Options*: permises, mais en dehors de l'entête, leurs présences est indiquées par le champs "Next Header"
- *ICMPv6*: nouvelle version d'ICMP
 - De nouveaux types de message, e.g. "Paquet trop gros"
 - Fonctions de gestion des groupes multicast

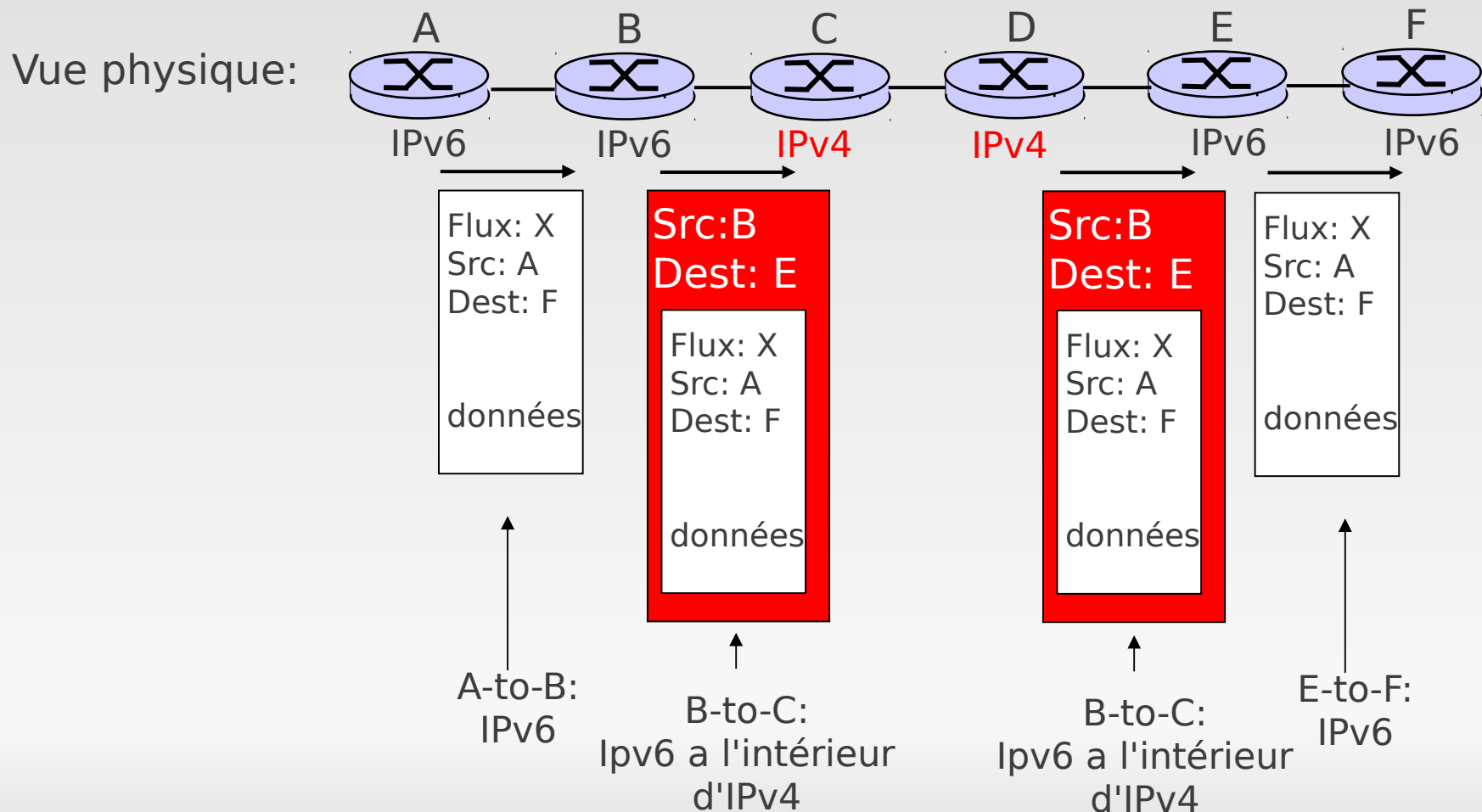
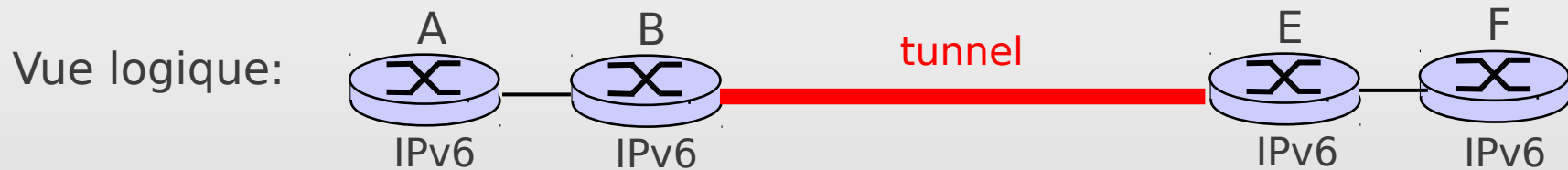
Transition de IPv4 a IPv6

- Tous les routeurs pourraient être mis à niveau simultanément
 - Pas de “flag days”
 - Comment le réseau va fonctionner avec un mélange d' IPv4 et d'IPv6?
- *Utilisation de tunnel:* IPv6 transporté comme des données quelconques dans un datagramme Ipv4 parmi les routeurs IPv4

Utilisation de tunnel



Utilisation de tunnel



Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

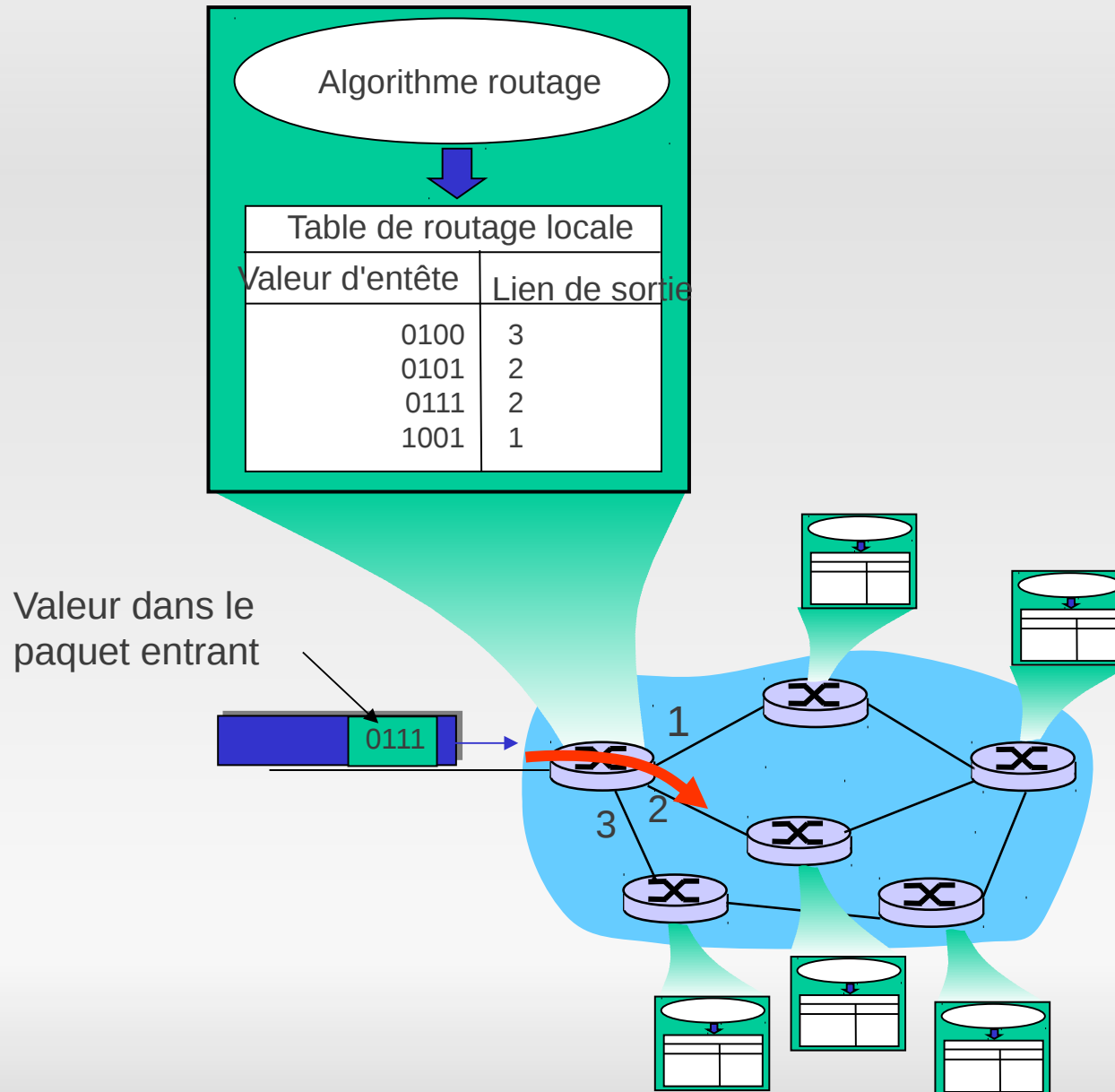
- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

Interaction routage et réexpédition



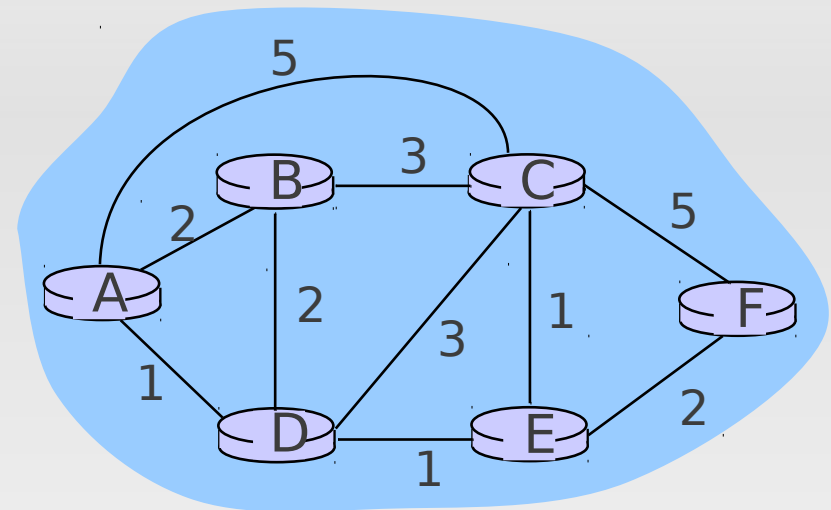
Routage

Protocole de Routage

But: déterminer le bon chemin (séquence de routeurs) de la source à la dest.

Basés sur l'algorithmique de graphe :

- Les noeuds sont les routeurs
- Les arêtes sont les liens physiques
 - Pondération par le temps, coût €, niveau de congestion



□ “bon” chemin :

- Classiquement celui de coût minimum
- Mais d'autres définitions sont possibles

Classification des Algorithmes de Routage

Information décentralisée ou centralisée ?

Centralisé :

- Tout les routeurs ont une vision complète de la topologie et du coût des liens
- => Algorithme de routage par “état de lien”

Décentralisé :

- Les routeurs ont une connaissance de leurs voisins et du coût des liens vers ses voisins
- Processus de calcul itératif avec échange d'informations entre voisin
- => Algorithme à “vecteur de distance”

Statique ou dynamique ?

Statique :

- Les chemins change doucement au cours du temps

Dynamique :

- Les chemins changent plus souvent
 - Mise à jour périodique
 - Réponse au changement de coût

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

Un algorithme de routage par état de lien

Algorithme de Dijkstra's

- Connaissance de la topologie du réseau, et du coût de chaque lien
 - Accomplis par un broadcast de l'état des liens
 - Tous les noeuds ont la même information
- Calcul du meilleur chemin d'un noeud à une autre
 - Donne une **table de routage** pour ce noeud
- itératif: après k itérations, connaissance du meilleur chemin jusqu'à la k -ième destination

Notation:

- $c(i,j)$: coût du lien du noeud i à j . coût infini si i et j ne sont pas voisin direct
- $D(v)$: valeur courante du cout du chemin de la source à la destination v
- $p(v)$: noeud prédécesseur le long du chemin de la source à v
- N : ensemble de noeuds dont le coût minimum est définitivement connu

Algorithme de Dijkstra's

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

1 **Initialisation:**

2 $N = \{A\}$

3 Pour chaque sommet v

4 si v adjacent à A

5 Alors $D(v) = c(A, v)$

6 Sinon $D(v) = \textit{infini}$

7

8 **Faire**

9 trouver w qui n'est pas dans N tel que $D(w)$ est minimum

10 ajouter w à N

11 Mettre à jour $D(v)$ pour tout les v adjacents à w et qui ne sont pas dans N :

12 $D(v) = \min(D(v), D(w) + c(w, v))$

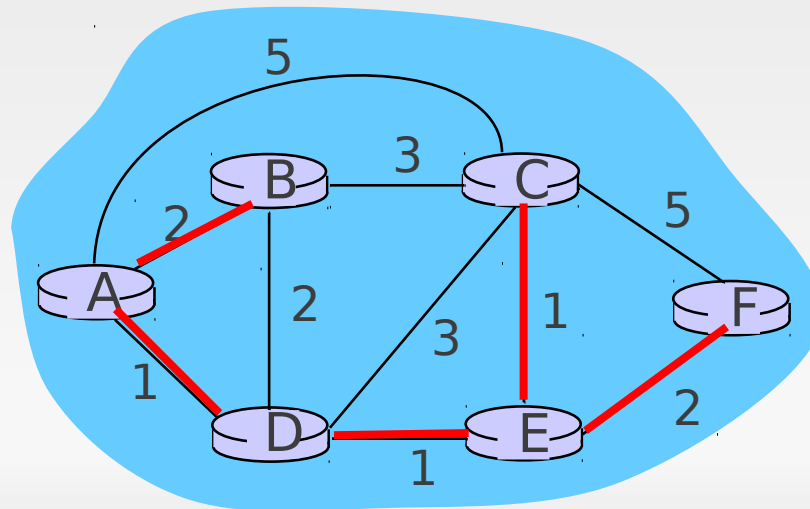
13 /* nouveau coût vers v est soit l'ancien coût vers v ou le plus

14 court chemin vers w plus le coût de w à v */

15 **Pour tout les noeuds de N**

Algorithme de Dijkstra : exemple

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infini	infini
1	AD	2,A	4,D		2,D	infini
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



Algorithme de Dijkstra: exemple (2)

L'arbre couvrant par plus court chemin à partir de u:

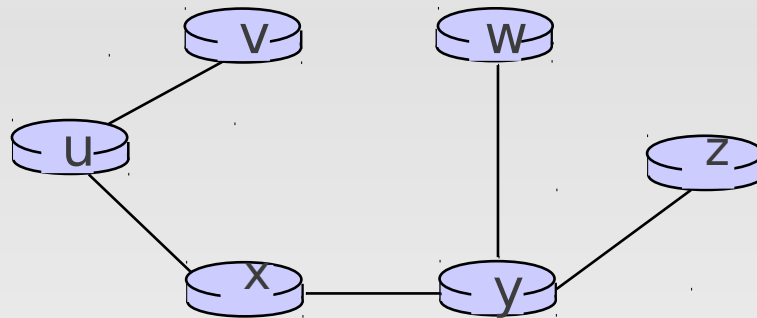


Table de routage dans u:

destination	lien
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

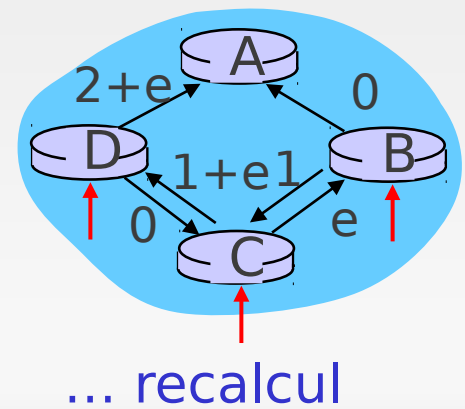
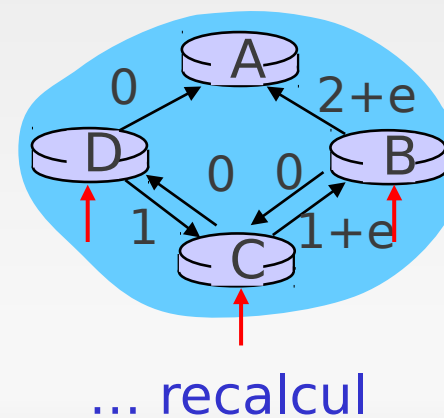
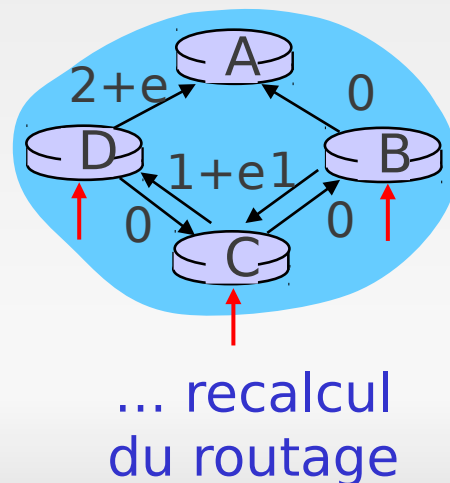
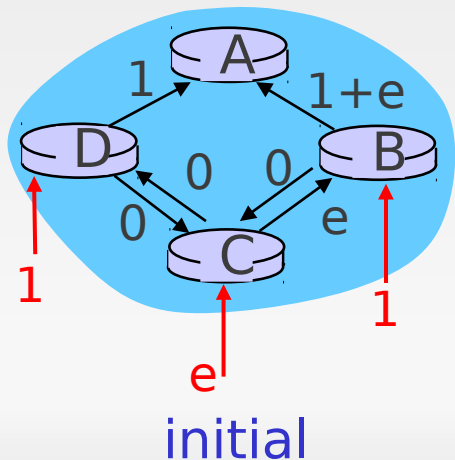
Discussion autour de l'algorithme de Dijkstra's

Complexité de l'algorithme : n noeuds

- Chaque itération : test de tous les noeuds, w , pas dans N
- $n*(n+1)/2$ tests : $O(n^2)$
- Implémentation plus efficace : $O(n \log(n))$

Variantes :

- ex., fonction de coût = quantité de trafic



Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- ICMP
- IPv6

4.5 Algorithmes de routage

- Etat de lien
- **Vecteur de distance**
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

Algorithme à vecteur de distance

Iteratif :

- On continue jusqu'à ce qu'il n'y ait plus d'échange entre noeuds
- *Arrêt- automatique* : pas de "signal" stop

Asynchrone :

- Les noeuds n'ont besoin d'échange d'information que dans un état de blocage

Distribué :

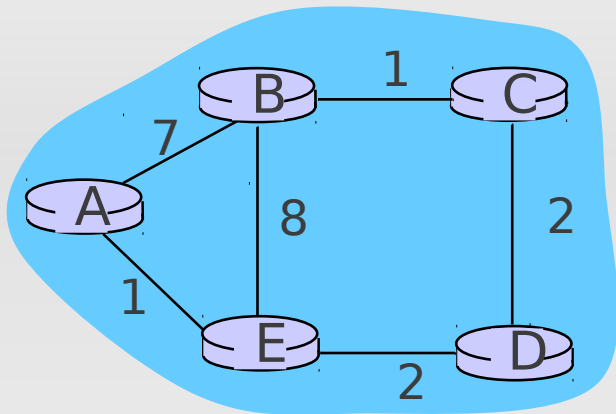
- Chaque noeud communique seulement avec ses voisins directs

Structure de la table de distance

- Chaque noeud en a une copie
- Ligne pour chaque destination possible
- Colonne pour chaque voisin direct
- Exemple : dans le noeud X , pour la destination Y via le voisin Z :

$$\begin{aligned} D^X(Y,Z) &= \text{distance de } X \text{ à } Y, \\ &\quad \text{via } Z \\ &= c(X,Z) + \min_w \{D^Z(Y,w)\} \end{aligned}$$

Table de distance : exemple



$$\begin{aligned}
 D^E_D(C,D) &= c(E,D) + \min_w \{D^D(C,w)\} \\
 &= 2+2 = 4
 \end{aligned}$$

$$\begin{aligned}
 D^E_D(A,D) &= c(E,D) + \min_w \{D^D(A,w)\} \\
 &= 2+3 = 5 \quad \text{loop!}
 \end{aligned}$$

$$\begin{aligned}
 D^E_D(A,B) &= c(E,B) + \min_w \{D^B(A,w)\} \\
 &= 8+6 = 14 \quad \text{loop!}
 \end{aligned}$$

Coût de la destination via

$D^E_D()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

La table de distance donne la table de routage

Coût de la destination via

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

Lien sortant, coût

A	A,1
B	D,5
C	D,4
D	D,4

destination

Table de distance → Table de routage

Routage à vecteur de distance : aperçu

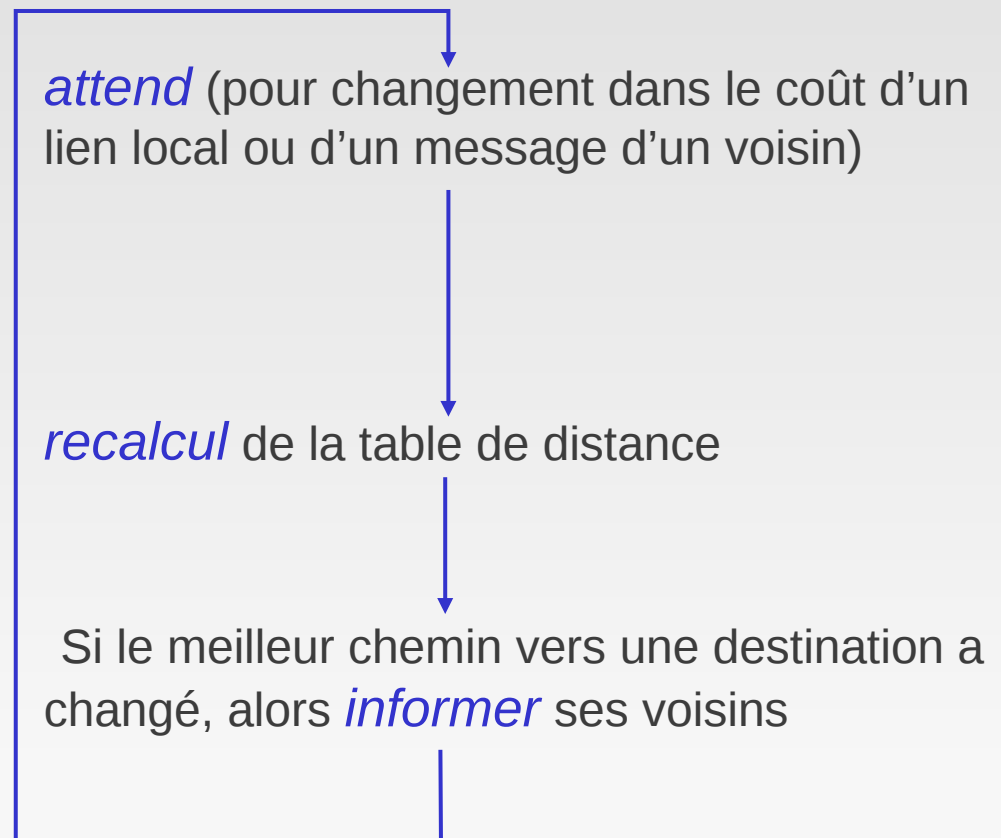
Itératif, asynchrone : chaque itération local causée par :

- Changement de coût du lien local
- Message d'un voisin : son meilleur chemin vers un voisin vient de changer

Distribué :

- Chaque noeud informe ses voisins seulement lors d'une modification dans le coût des chemins
 - Les voisins informent leurs autres voisins si nécessaire

Chaque noeud :



Algorithme à vecteur de distance : (Init.)

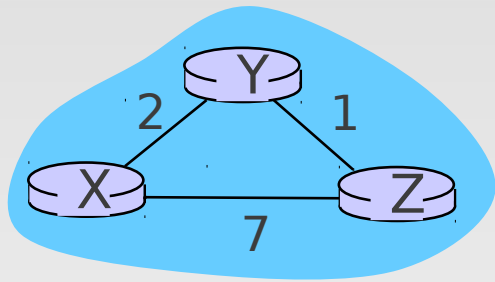
A chaque noeud, X :

- 1 **Initialisation:**
- 2 Pour tout les noeuds v adjacents :
- 3 $D^X(*,v) = \text{infini}$ /* l'opérateur $*$ signifie "pour toute les lignes" */
- 4 $D^X(v,v) = c(X,v)$
- 5 Pour toutes les destinations, y
- 6 Envoyer $\min_w D^X(y,w)$ à chacun des voisins

Algorithme à vecteur de distance (boucle)

```
8  boucle
9  attend (jusqu'à un changement du coût vers mon voisin  $V$ 
10     où lorsque je reçoie une mise à jour de  $V$ )
11
12  si ( $c(X,V)$  change par  $d$ )
13     /* change les coûts de toutes les destinations via le voisin  $V$  par  $d$  */
14     /* note:  $d$  peut être  $>0$  ou  $<0$  */
15     Pour toutes les destinations  $y$ :  $D^X(Y,V) = \bar{D}^X(Y,V) + d$ 
16
17  sinon si (mise à jour reçu de  $V$  pour la destination  $Y$ )
18     /* plus court chemin de  $V$  vers  $Y$  a changé */
19     /*  $V$  a renvoyé une nouvelle valeur pour son min  $D^V(Y,w)$  */
20     /* appelons cette nouvelle valeur "newval" */
21     Pour la destination  $Y$ :  $D^X(Y,V) = c(X,V) + newval$ 
22
23  si un nouveau minimum est apparu  $D^X(Y,w)$  pour une destination  $Y$ 
24     Renvoyer sa nouvelle valeur min  $D^X(Y,w)$  à chaque voisin
25
26  toujours
```

Algorithme à vecteur de distance : exemple.



		cost via X	
		Y	Z
d e s t	Y	2	∞
	Z	∞	7

		cost via Y	
		X	Z
d e s t	X	2	∞
	Z	∞	1

		cost via Z	
		X	Y
d e s t	X	7	∞
	Y	∞	1

		cost via Z	
		Y	Z
d e s t	Y	2	8
	Z	3	7

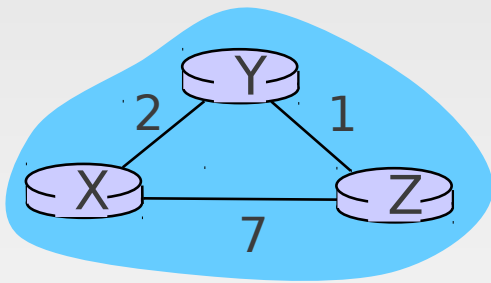
$$D^X(Y,Z) = c(X,Z) + \min_w \{D^Z(Y,w)\}$$

$$= 7 + 1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w \{D^Y(Z,w)\}$$

$$= 2 + 1 = 3$$

Algorithme à vecteur de distance : exemple



D ^X		cost via	
D		Y	Z
d	Y	2	∞
e	Z	∞	7
s			
t			

D ^Y		cost via	
D		X	Z
d	X	2	∞
e	Z	∞	1
s			
t			

D ^Z		cost via	
D		X	Y
d	X	7	∞
e	Y	∞	1
s			
t			

D ^X		cost via	
D		Y	Z
d	Y	2	8
e	Z	3	7
s			
t			

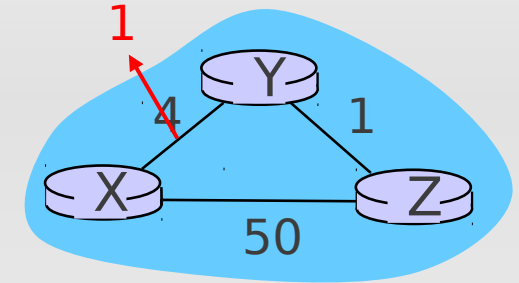
D ^Y		cost via	
D		X	Z
d	X	2	8
e	Z	9	1
s			
t			

D ^Z		cost via	
D		X	Y
d	X	7	3
e	Y	9	1
s			
t			

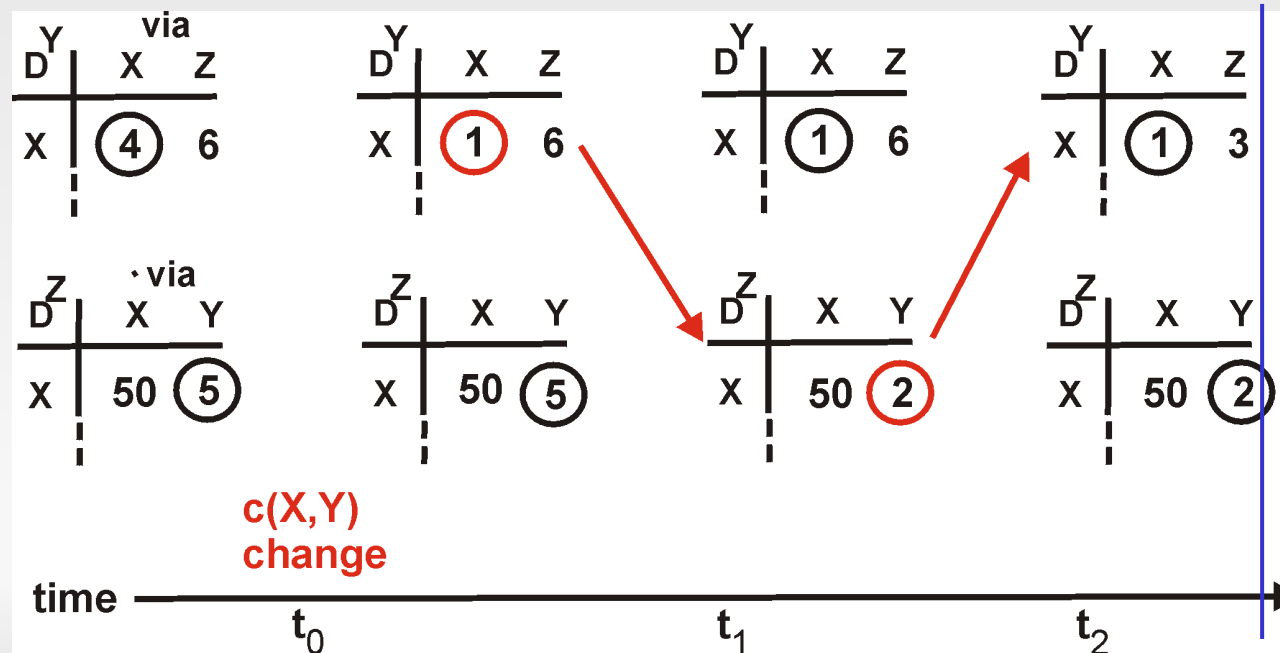
Vecteur de distance : changement du coût d'un lien

Le coût change lorsque :

- Le noeud détecte une modification locale
- Mise à jour de la table de distance (ligne 15 de l'algorithme)
- Si le coût change dans le meilleur chemin et que l'on informe ses voisins (lignes 23, 24 de l'algorithme)



“les bonnes nouvelles se propagent vite”

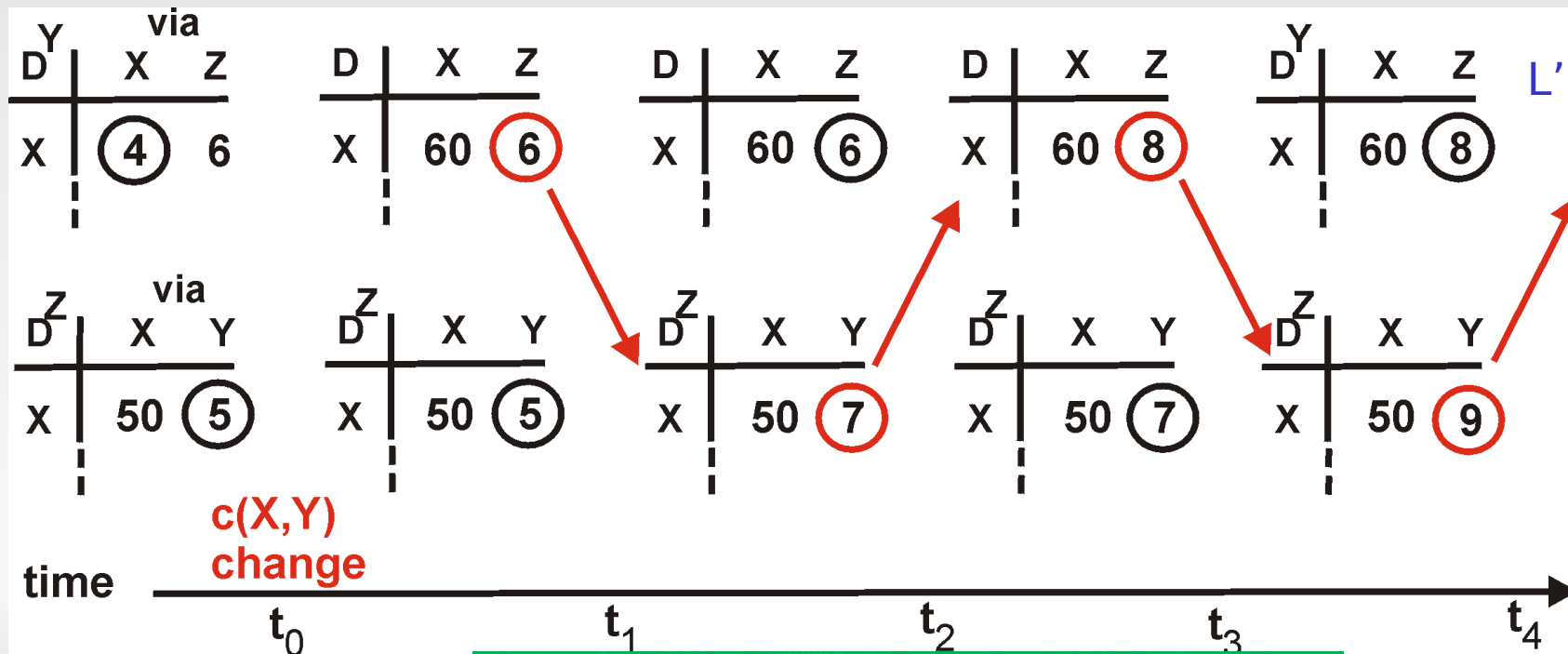
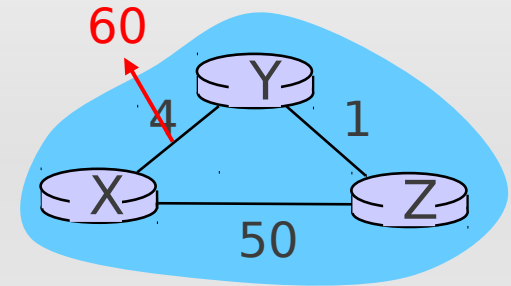


fin de l'algorithme

Vecteur de distance : changement du coût d'un lien

Changement du coût d'un lien :

- Les bonnes nouvelles se propagent vite
- Les mauvaises nouvelles voyagent lentement - problème de l'infini

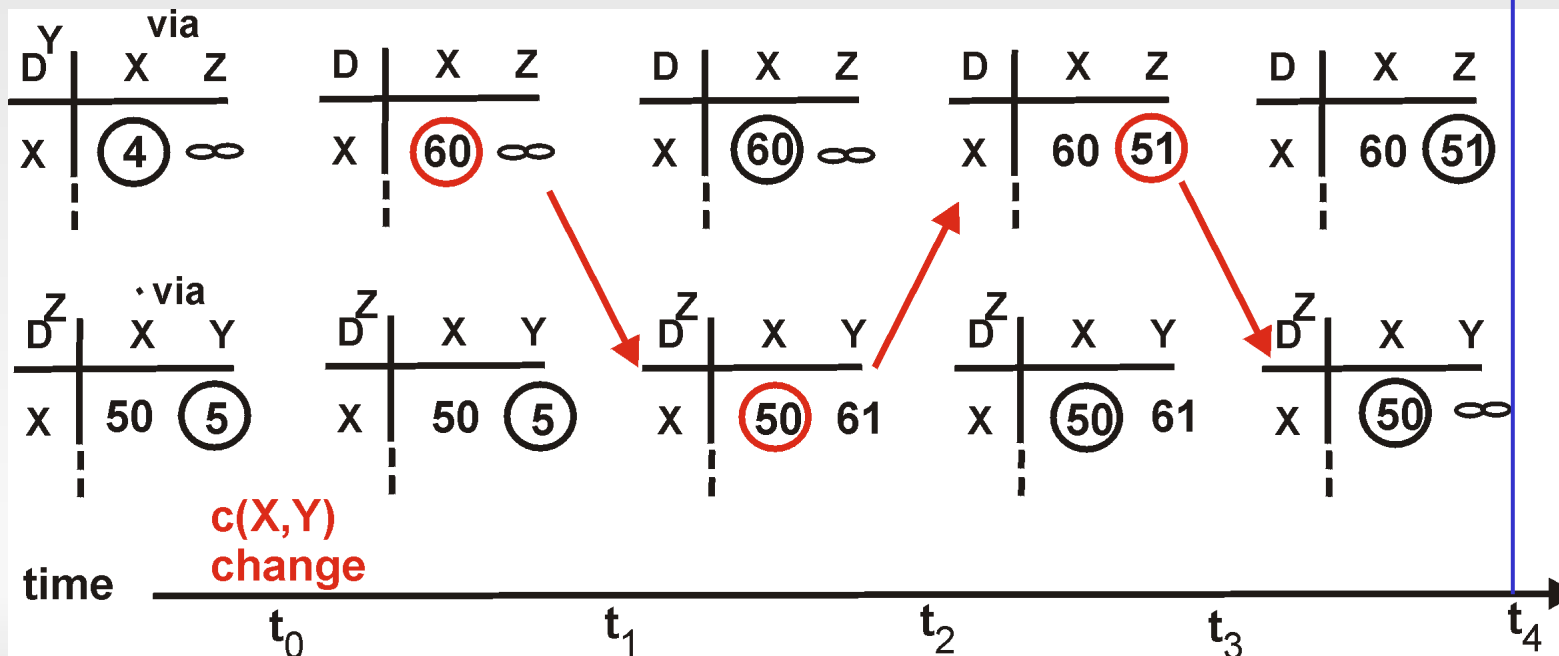
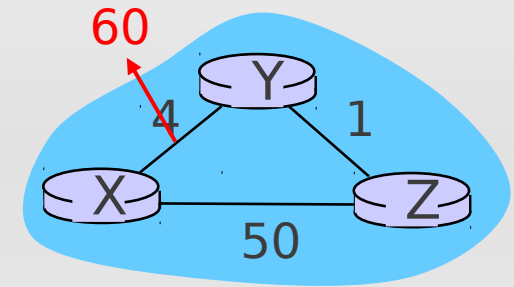


L'algorithme continue

Vecteur de distance : problème

Si la route jusqu'à X via Z de la table Y change pour l'infini :

- Z dit à Y que sa distance de Z à X est infinie (donc Y ne routera pas vers X via Z)
- Est ce que cela résoudra le problème de l'infini ?



L'algorithme se termine

Comparaison des algorithmes LS et VD

Complexité des messages

- LS (état de lien) :
avec n noeuds, E liens, $O(nE)$ msgs envoyés à chaque fois
- VD (vecteur de distance) : échange entre voisin seulement
 - Le temps de convergence varie

Vitesse de convergence

- LS: algorithme en $O(n^2)$ avec $O(nE)$ msgs
 - Peut avoir des oscillations
- VD: le tps de convergence varie
 - Boucle de routage
 - Problème de l'infini

Robustesse: que se passe-t-il si une malfonction apparaît ?

LS:

- Le noeud peut propager une information de coût de noeud incorrecte
- Chaque noeud ne calcule que sa propre table

VD:

- Le noeud peut propager une information de coût de chemin incorrecte.
- Chaque table de noeud utilisée par les autres
 - L'erreur se propage à travers le réseau

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

- État de lien
- Vecteur de distance
- **Routage hiérarchique**

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

4.7 routage *broadcast* et *multicast*

Routage hiérarchique

Notre étude du routage est idéalisée

- Tous les routeurs sont identiques
- réseau “plat”

... *pas* vrai en pratique

Étendue: avec 200 millions de destinations possibles:

- Il est **impossible** de stocker toutes les destinations dans les tables de routage!
- L'échange de table de routage inonderait alors toutes les liaisons!

Autonomie administrative

- internet = réseau de réseaux
- Chaque administrateur réseau peut vouloir contrôler le routage dans son propre réseau

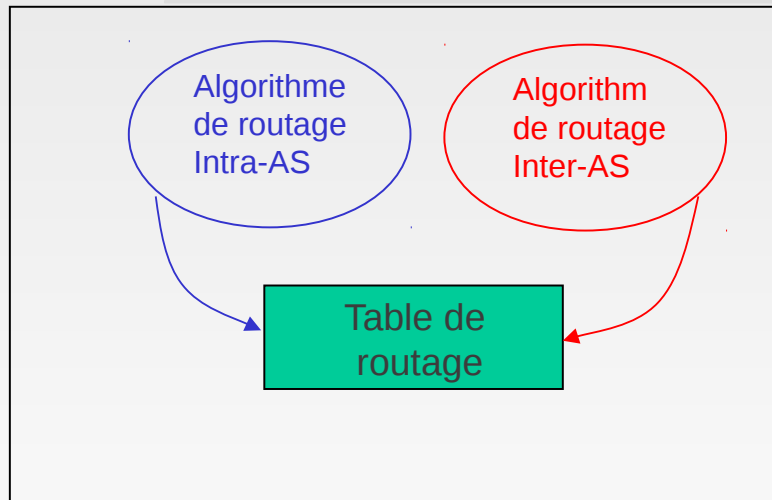
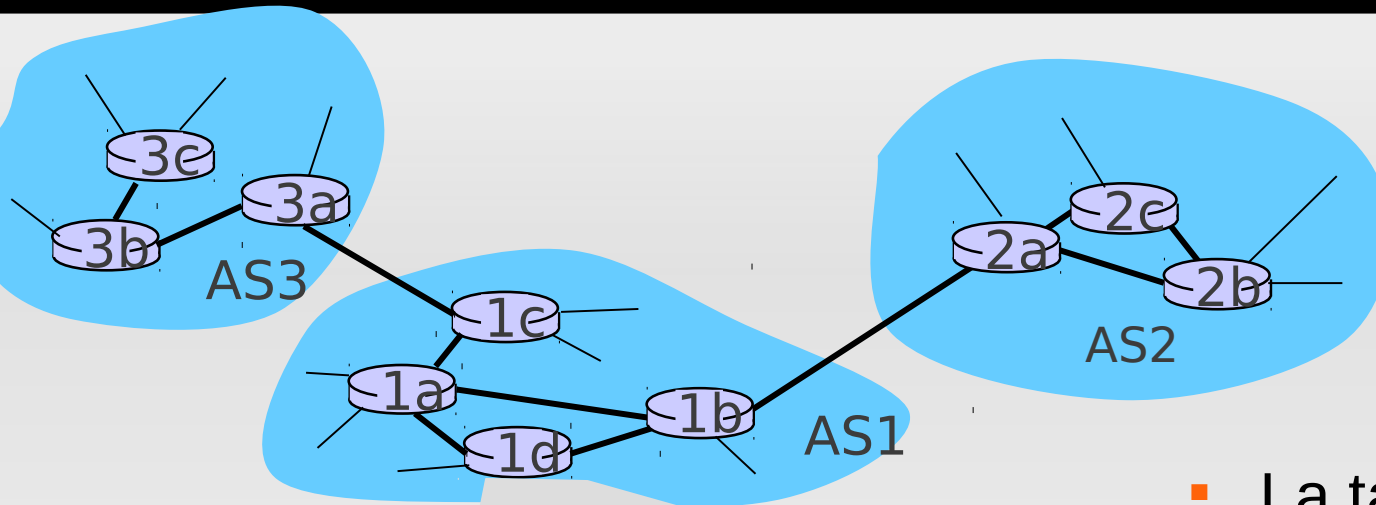
Routage hiérarchique

- Partitionner les routeurs en région, « système autonome » ou “autonomous systems” (AS)
- Des routeurs dans le même système autonome utilise le même protocole de routage
 - “intra-AS” protocole de routage
 - Les routeurs dans des AS différents peuvent utiliser des protocoles de routages différents.

Routeur “passerelle” ou “Gateway”

- Est relié directement à un routeur d'un autre AS.

AS interconnectés



- La table de routage est configurée à la fois avec un algorithme de routage intra-AS et un inter-AS
 - Intra-AS donne les entrées pour les destinations internes
 - Inter-AS & Intra-AS donne les entrées pour des destinations externes

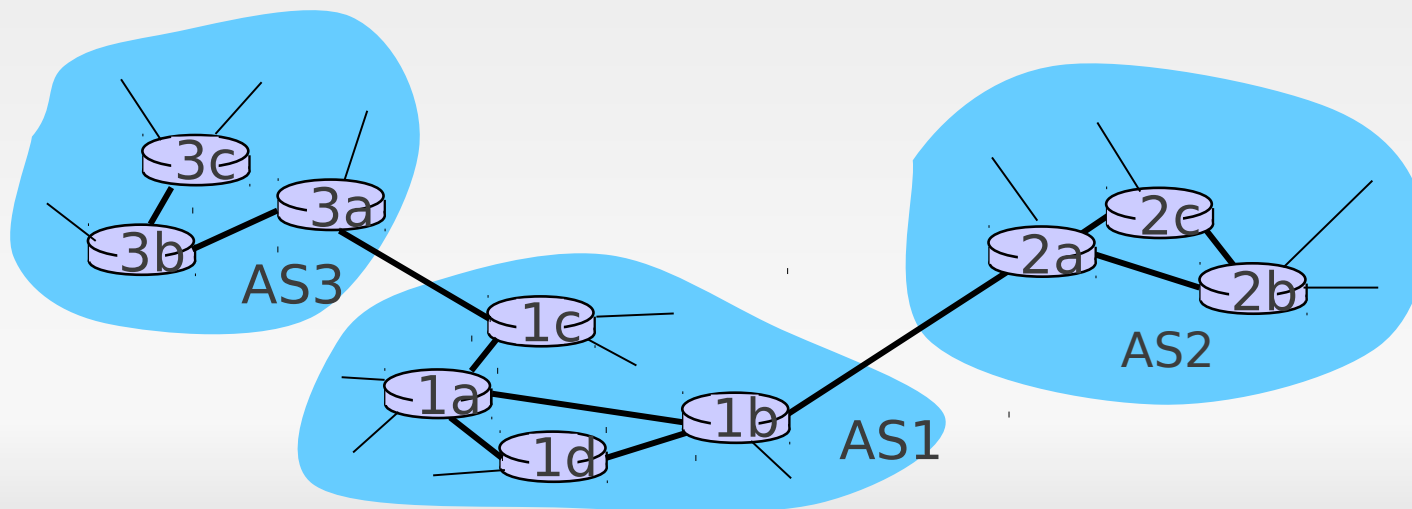
Tâches Inter-AS

- Supposons qu'un routeur dans AS1 reçoivent un datagramme destiné à un hôte externe au AS1
 - Le routeur devrait expédier le paquet vers l'un des routeurs passerelles, mais lequel?

AS1 a besoin:

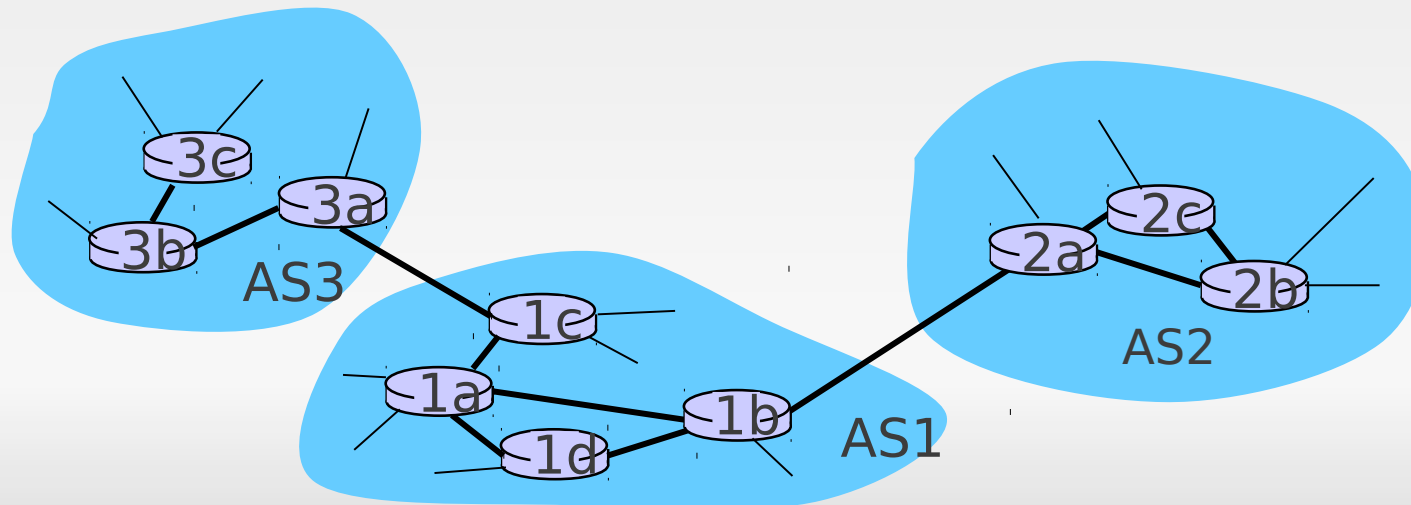
1. De connaître quelles destinations sont accessibles via AS2 et via AS3
2. De propager cette information sur l'accessibilité à tous les routeurs de AS1

C'est le travail du routage inter-AS!



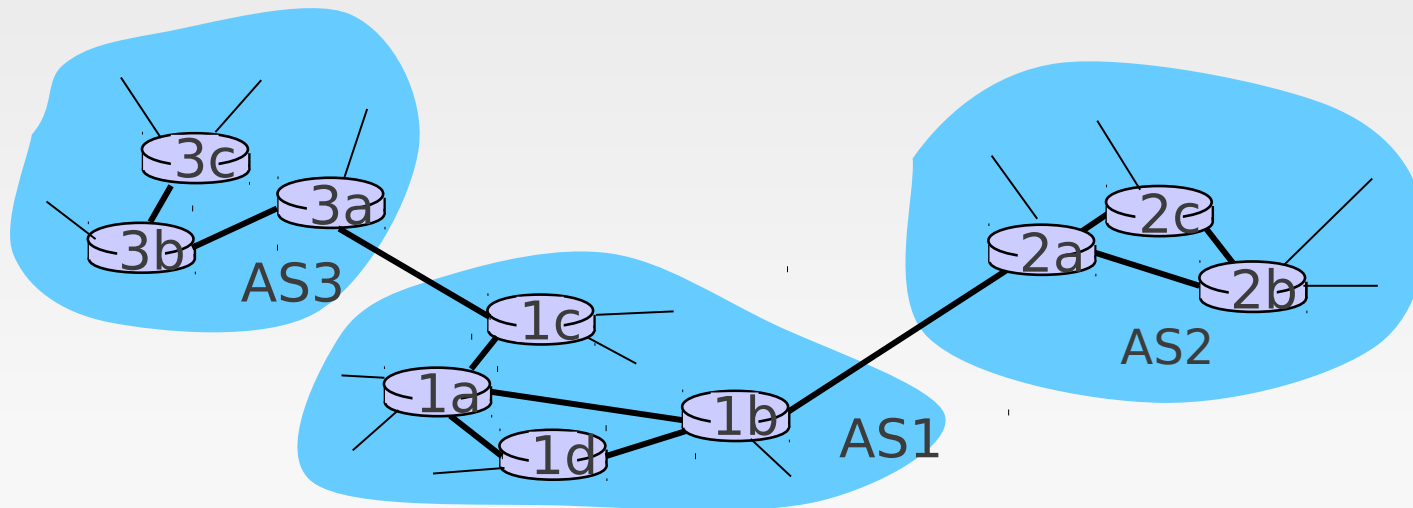
Exemple: établissement d'une table dans le routeur 1d

- Supposons que AS1 sache (via le protocole inter-AS) que le sous-réseau x est accessible via AS3 (passerelle 1c) mais pas via AS2.
- Le protocole Inter-AS propage cette information à tous les routeurs internes de AS1.
- Le routeur 1d détermine avec le protocole intra-AS que son interface l est sur le chemin le moins coûteux vers 1c.
- Mettre dans la table de routage l'entrée (x,l) .



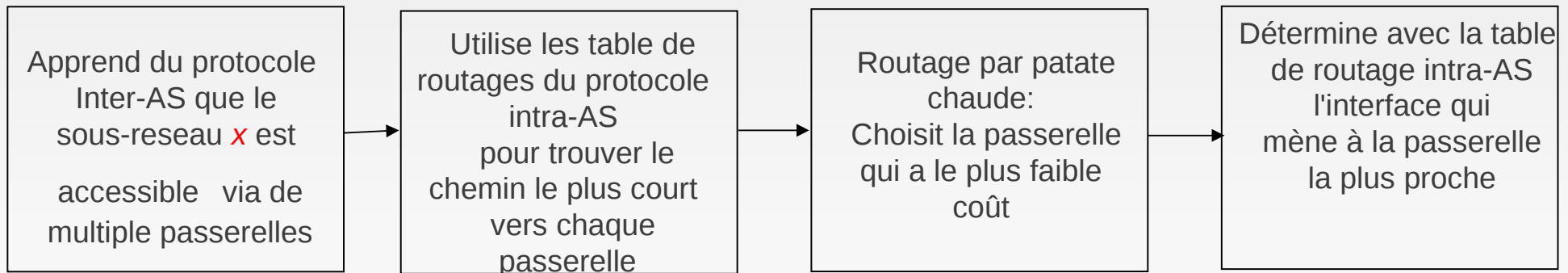
Exemple: Choisir parmi de multiple AS

- Maintenant supposons que l'AS1 apprenne du protocole inter-AS que le sous-réseau **x** est accessible à partir de AS3 **et** de AS2.
- Pour configurer une table de routage, le routeur 1d doit déterminer vers quelle passerelle il doit expédier son paquet pour la dest. **x**.
- C'est aussi le travail du protocole de routage inter-AS!



Exemple: Choisir parmi de multiple AS

- Maintenant supposons que l'AS1 apprenne du protocole inter-AS que le sous-réseau **x** est accessible à partir de AS3 **et** de AS2.
- Pour configurer sa table de routage, le routeur 1d doit déterminer vers quelle passerelle il doit envoyer le paquet pour la destination **x**.
- C'est aussi le travail du protocole de routage inter-AS!
- **Routage avec la patate chaude:** envoie le paquet au routeur le plus proche des deux routeurs.



Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- IPv4 adressage
- IPv6

4.5 Algorithmes de routage

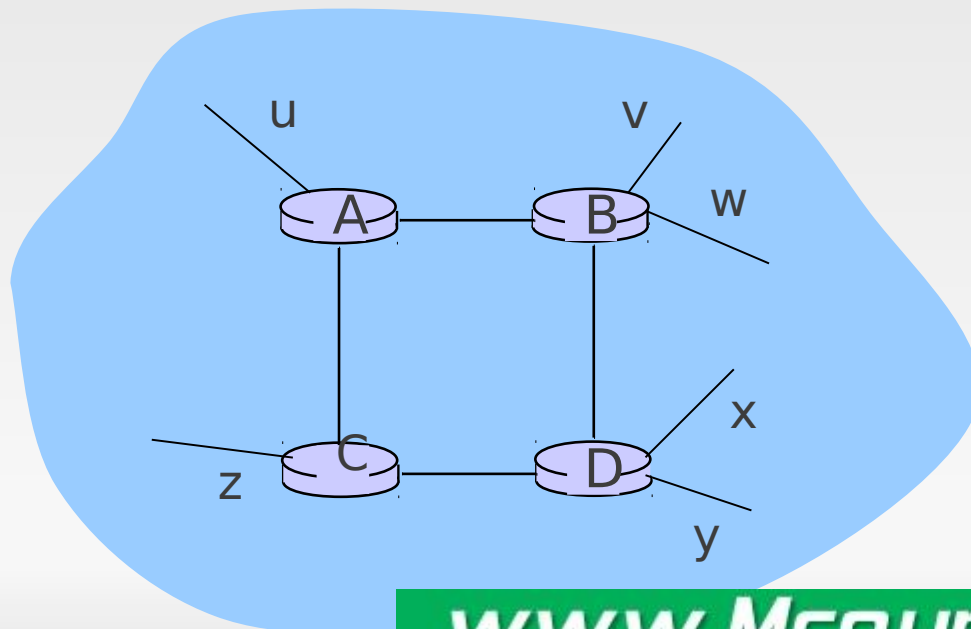
- État de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

RIP (Routing Information Protocol)

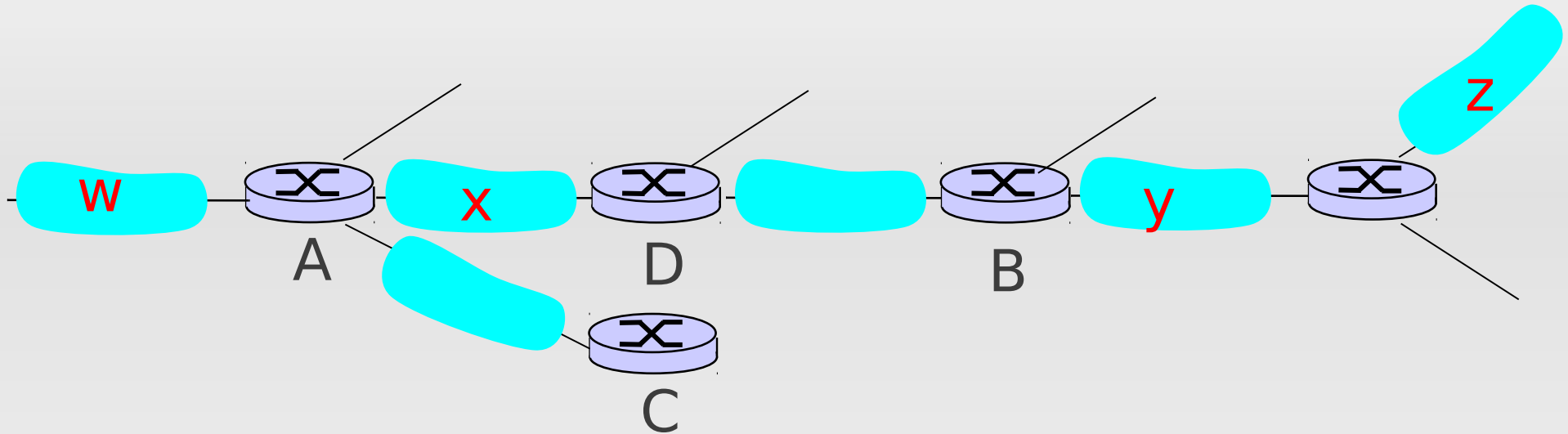
- Algorithme à vecteur de distance
- Inclus dans la distribution BSD-UNIX en 1982
- La métrique de distance = nombre de bond/saut (max = 15 bonds)



Du routeur A au sous-réseaux:

<u>destination</u>	<u>bond</u>
u	1
v	2
w	2
x	3
y	3
z	2

RIP (Routing Information Protocol)



Réseau de det.	Prochain routeur	# de bond
w	A	2
y	B	2
z	B	7
x	--	1
....

Table de routage de D

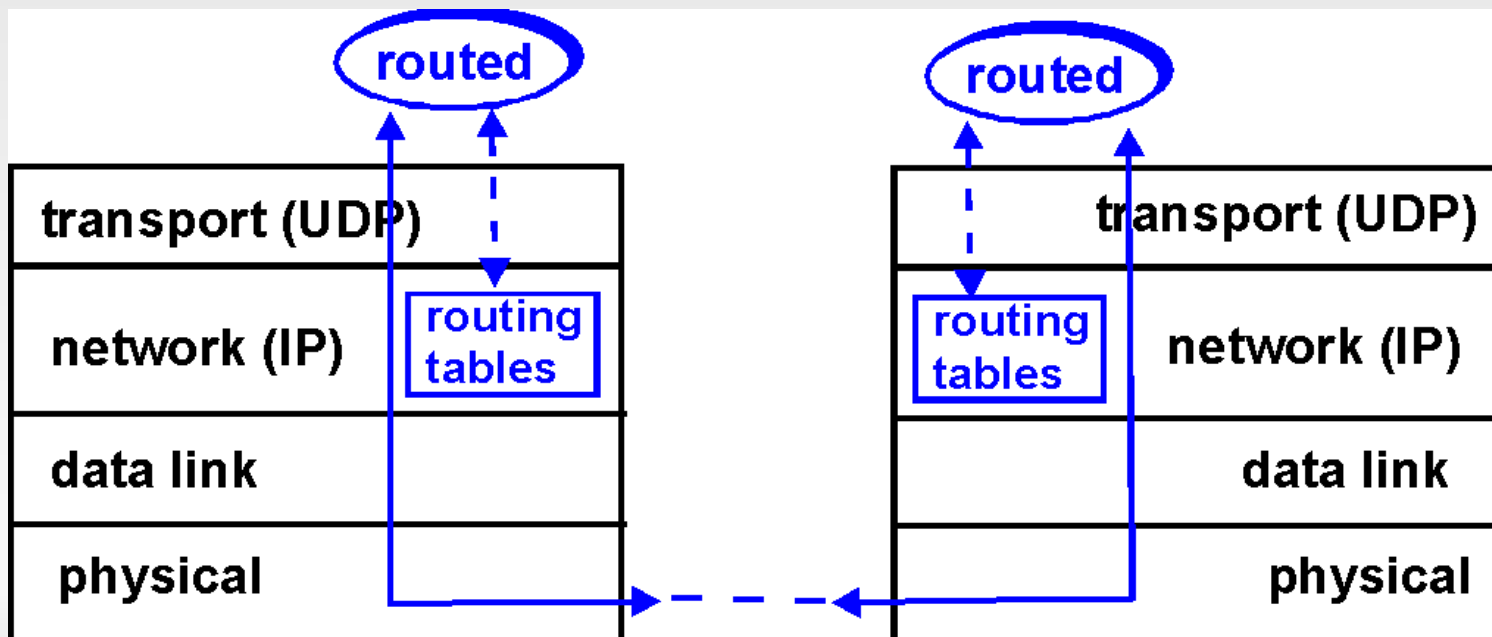
RIP: problèmes de liaison

Si aucun message après 180 sec. --> le voisin/lien est déclaré mort !

- Les routes via ce voisin sont invalidées
- Nouveaux messages envoyés aux voisins
- Les voisins répondent par un nouveau message (si les tables ont changées)
- L'information se propage très rapidement sur le réseau tout entier

Les tables RIP

- Les table de routage RIP sont administrées au niveau application appelée “daemon route-d”
- Message envoyé dans un paquet UDP, répété périodiquement



Les tables RIP : exemple

Routeur: *giroflée.eurocom.fr*

Destination	Gateway	Flags	Ref	Use	Interface
127.0.0.1	127.0.0.1	UH	0	26492	lo0
192.168.2.	192.168.2.5	U	2	13	fa0
193.55.114.	193.55.114.6	U	3	58503	le0
192.168.3.	192.168.3.5	U	2	25	qaa0
224.0.0.0	193.55.114.6	U	3	0	le0
default	193.55.114.129	UG	0	143454	

- 3 réseaux de classe C attachés
- Les routeurs ne connaissent que les chemins du réseau
- Le routeur envoie par default vers l'extérieur
- L'adresse de multicast est : 224.0.0.0
- Interface qui boucle sur elle-même (pour debug)

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

- Etat de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

- RIP
- OSPF
- BGP

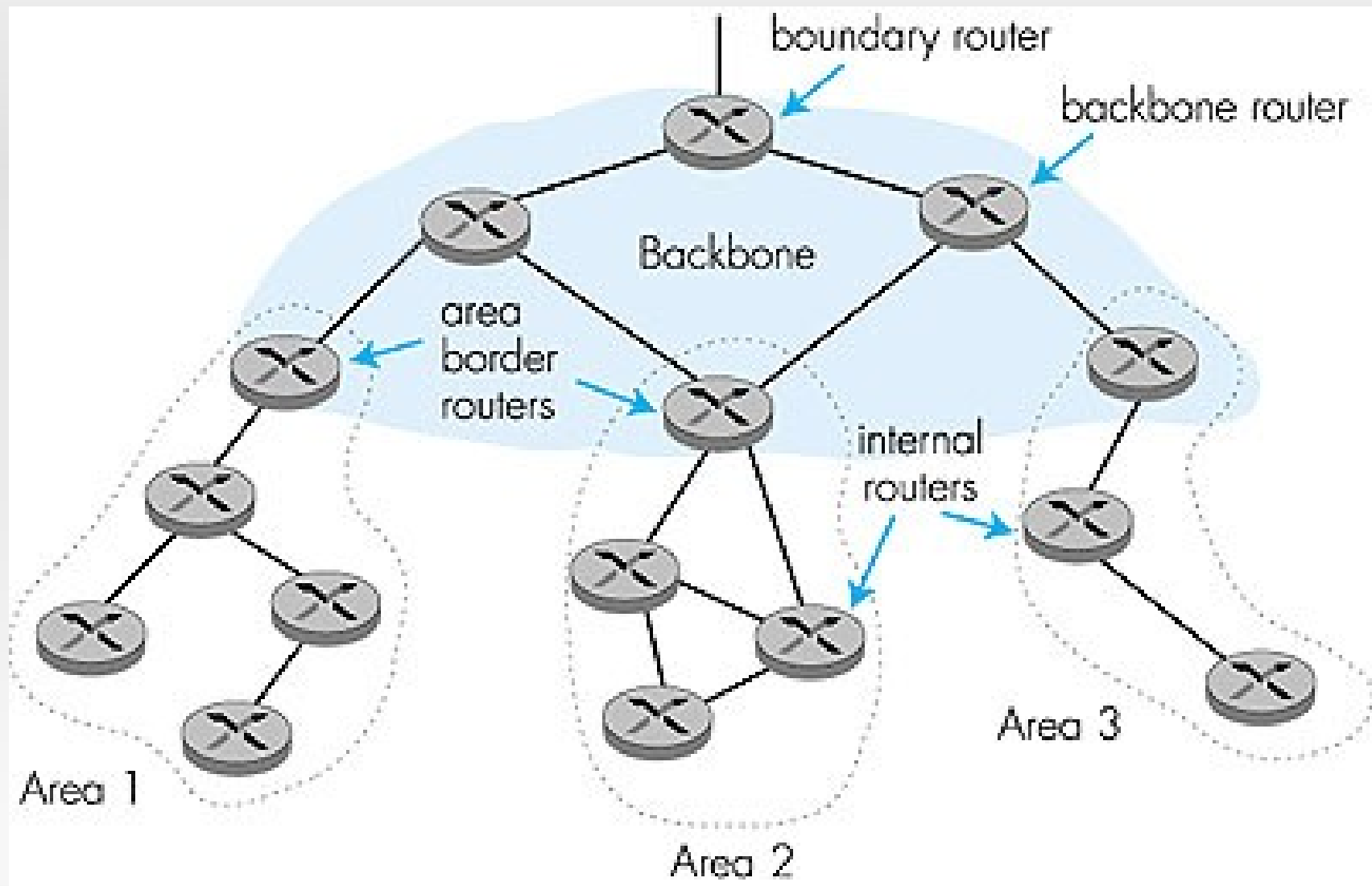
OSPF (Open Shortest Path First)

- “open”: public
- Basé sur un algorithm de routage par état de lien (LS)
 - Dissémination des paquets LS
 - Carte du réseau dans chaque noeud
 - Calcul du routage basé sur Dijkstra
- Les messages OSPF ont une entrée par routeur voisin
- Les messages sont disséminés sur tout l’AS (innondation)
 - Transport dans des messages OSPF directement sur IP (au lieu d’utiliser TCP ou UDP)

OSPF fonctions avancées (!= RIP)

- **Sécurité** : tout les msg OSPF sont authentifiés (pour résoudre les intrusions malicieuses)
- **Multi-path** : autorisé, par le biais de chemin de même coût (seulement 1 dans RIP)
- Pour chaque lien, utilisation de métriques différentes en fonction de l'utilisation (ex: liaison satellite ne coûte pas cher pour le meilleur effort; et très cher pour le temps réel)
- Support **uni- et multi-cast** intégré :
 - Multicast OSPF (MOSPF) basé sur les données topologiques d'OSPF
- **OSPF Hiérarchique** dans les gros domaines

OSPF hiérarchique



OSPF hiérarchique

- **Hiérarchie à 2 niveaux** : zone locale, backbone/dorsale.
 - Message sur l'état des liens seulement en zone locale
 - Chaque noeud a une vision détaillée de la topologie de la zone locale et de la direction du plus court chemin vers les autres
- **Routeur intra-zone** : connaissance du réseau dans son domaine et informe les autres routeurs intra-zone
- **Routeur backbone** : utilise le routage OSPF uniquement sur le backbone.
- **Routeur inter-zone** : connecté aux autres AS.

Chapitre 4: Plan

4.1 Introduction

4.2 Circuit virtuel et réseaux à datagramme

4.3 Fonctionnement d'un routeur

4.4 IP: Internet Protocol

- Format des datagrammes
- Adressage IPv4
- IPv6

4.5 Algorithmes de routage

- Etat de lien
- Vecteur de distance
- Routage hiérarchique

4.6 Routage dans l'Internet

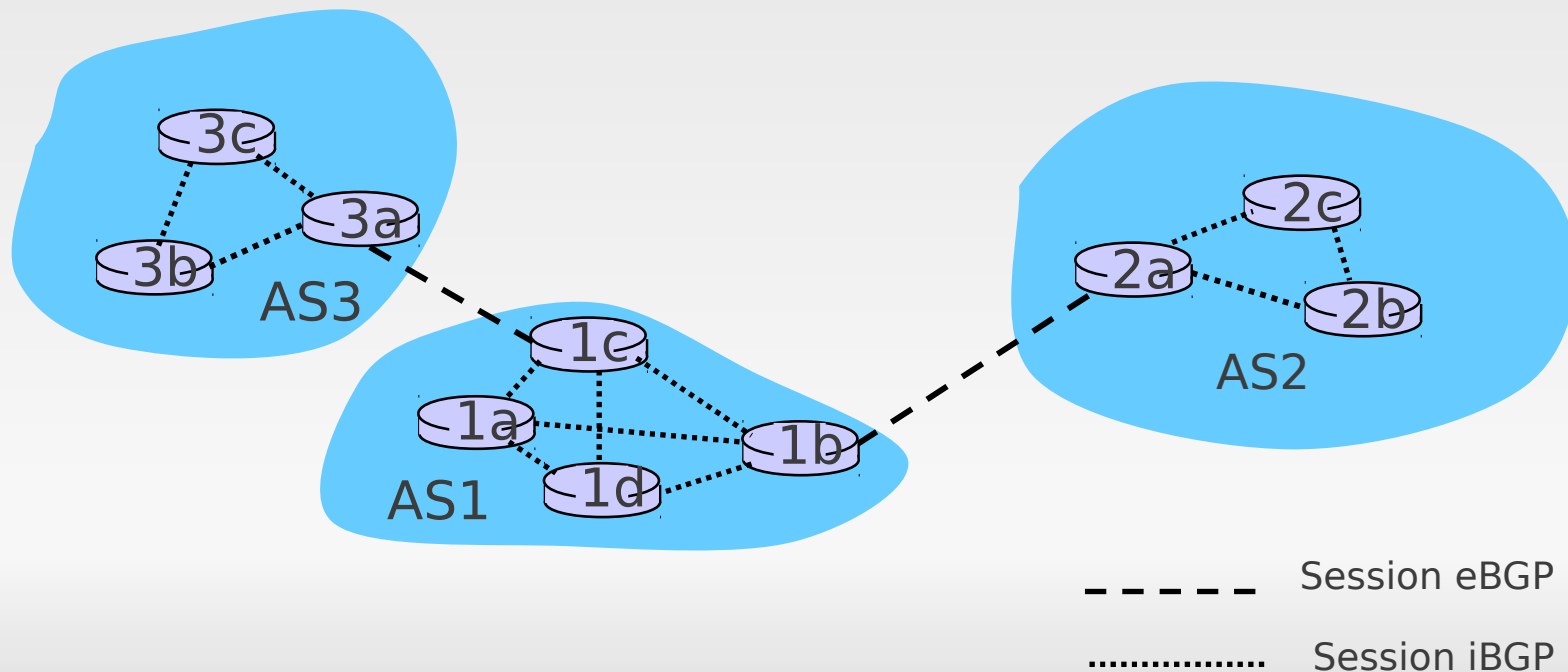
- RIP
- OSPF
- **BGP**

Routage Internet inter-AS: BGP

- **BGP (Border Gateway Protocol):** le standard, de facto
- BGP permet à chaque AS de:
 1. Obtenir d'AS voisin une information sur l'accessibilité d'un sous-réseau.
 2. Propager l'information sur l'accessibilité à tous les routeurs internes d'un AS.
 3. Déterminer les “bons” chemins vers des sous-réseaux basés sur l'accessibilité et des aspects politiques.
- Permet à un sous-réseau d'avertir son existence au reste d'internet: *“Je suis là!”*

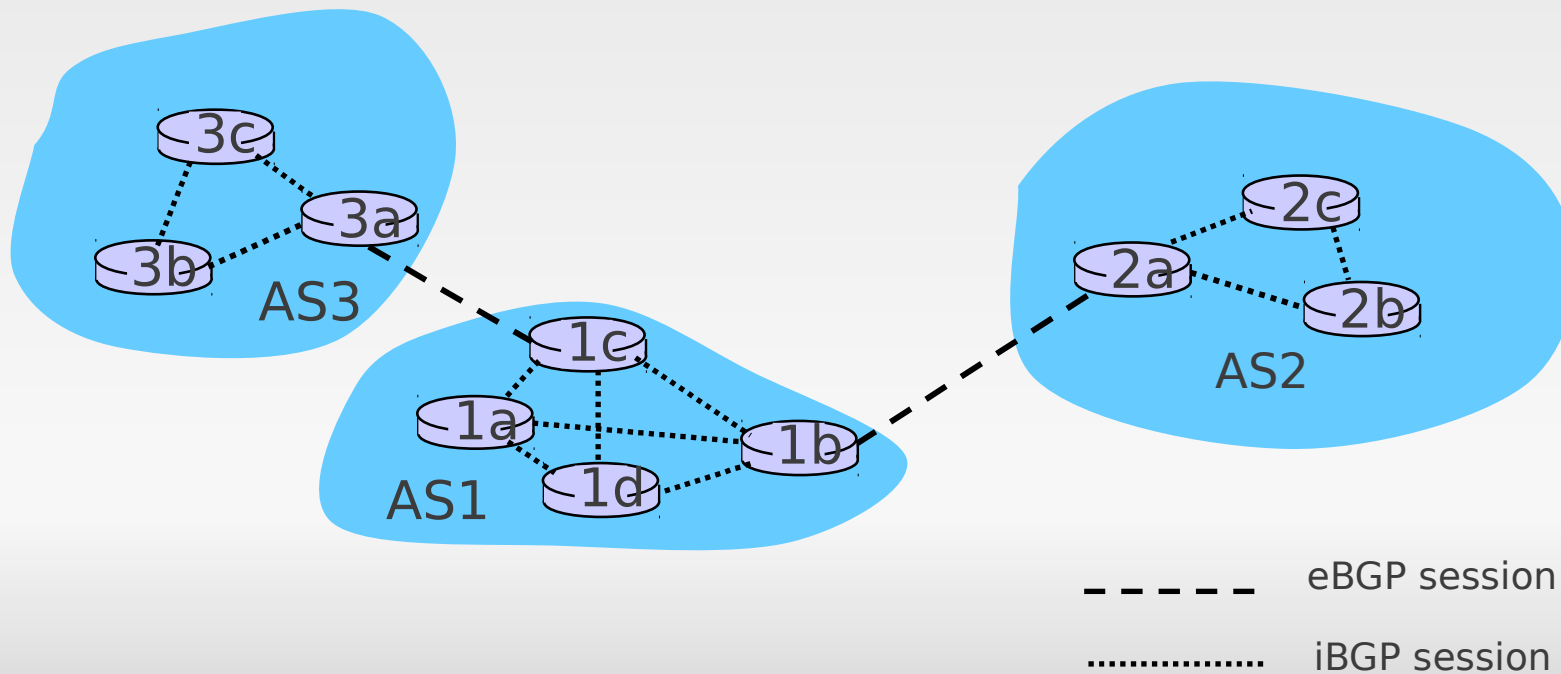
Les bases de BGP

- Une paire de routeurs (BGP pairs) échangent des informations de routage sur des connections TCP semi-permanente: des **sessions BGP**
 - Les sessions BGP ne correspondent pas forcément à des liaisons physiques.
- Lorsque AS2 avertit d'un préfixe à AS1, AS2 **promet** qu'il amènera chaque datagramme destiné à ce préfixe à bon port.
 - AS2 peut rassembler des préfixes dans son avertissement.



Information distribuée concernant l'accessibilité

- Avec une session eBGP entre 3a et 1c, AS3 envoie l'information concernant l'accessibilité des préfixes à AS1.
- 1c peut alors utiliser iBGP pour distribuer ces nouvelles informations sur les préfixes accessibles à tous les routeurs dans AS1
- 1b peut alors re-avertir AS2 de nouvelles informations d'accessibilité sur la session eBGP 1b-to-2a
- Lorsqu'un routeur apprend un nouveau préfixe, il crée une entrée pour le préfixe dans sa table de routage.



Attributs des chemins & routes

BGP

- Un message d'information sur préfixe inclus des attributs BGP.
 - préfixe + attributs = “route”
- Deux attributs importants:
 - **AS-PATH**: contient le nom des AS par lesquelles le message d'information est passé, e.g., AS 67, AS 17
 - **NEXT-HOP**: Indique précisément le routeur interne vers le prochain AS. (il peut y avoir plusieurs lien de l'AS actuel vers le prochain-AS.)
- Lorsqu'un routeur passerelle reçoit un nouveau message (et donc une nouvelle route) pour un préfixe, il utilise une **politique d'importation** pour accepter/décliner l'information.

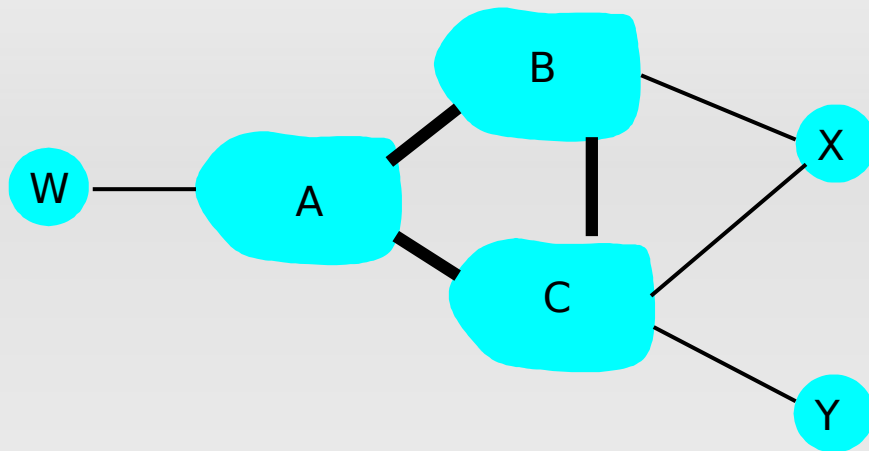
Sélection de route dans BGP

- Un routeur peut connaître plus d'une route vers un préfixe. Le routeur doit sélectionner une des routes à emprunter.
- Règles d'élimination:
 1. Préférence locale, décision politique
 2. AS-PATH le plus court
 3. (NEXT-HOP) le routeur vers le prochain saut est le plus proche: patate chaude!
 4. D'autres critères

Messages BGP

- Messages d'échange BGP avec TCP.
- Messages BGP :
 - **OPEN**: ouvrir une connexion TCP à un pair et authentifier l'expéditeur
 - **UPDATE**: informe d'un nouveau chemin (ou rejette un ancien chemin)
 - **KEEPALIVE** garder la connexion ouverte en l'absence d' UPDATES; aussi ACKs des requêtes OPEN
 - **NOTIFICATION**: notification d'erreurs dans les msg précédent; utilisée aussi pour fermer une connexion

BGP: contrôle de qui route qui



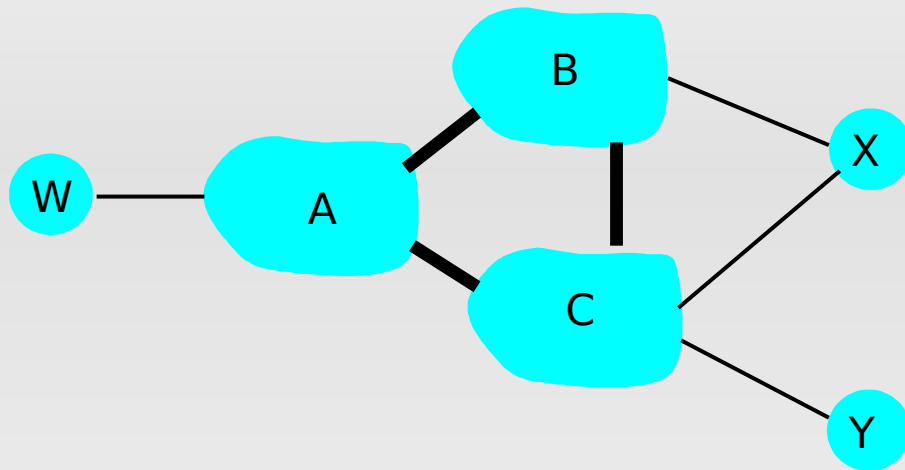
légende

 Fournisseur de réseau

 Réseau client

- A,B,C sont des **fournisseurs de réseau**
- X,W,Y sont des clients
- X est attaché à 2 réseaux
 - X ne veut pas de route au départ de B vers C via X
 - Donc X n'informera pas B d'une route vers C

BGP: contrôle de qui route qui



légende

Fournisseur de réseau

Réseau client

- A informe B du chemin AW
- B informe W du chemin BAW
- Est ce que B doit informer C du chemin BAW ?
 - Non ! B ne retire rien de router CBAW vue que ni W ni C sont des clients de B
 - B veut forcer C à router vers W via A
 - B veut router uniquement de/vers ses clients !

Pourquoi existe-t-il différents types de routage Intra- et Inter-AS ?

Politique:

- Inter-AS: nécessité de cohésion dans le routage sur le net
- Intra-AS: Un seul admin., donc pas de standard nécessaire

Echelle :

- Le routage hiérarchique économise des tailles de table et réduit le trafic utilisé pour les mises à jour.

Performance:

- Intra-AS: peut se concentrer essentiellement sur les performances
- Inter-AS: la politique peut dominer les performances