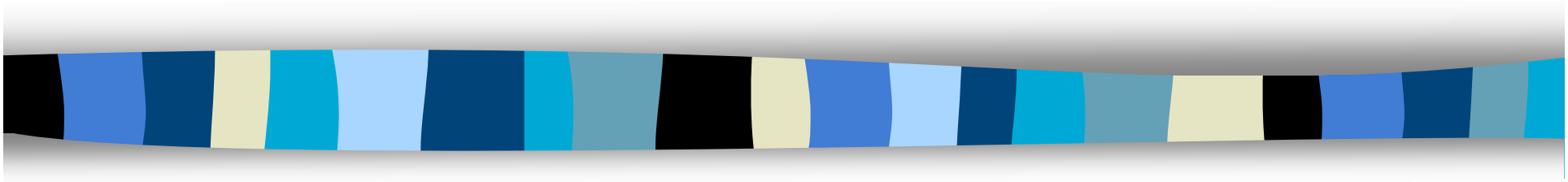
Routage et réseaux IP



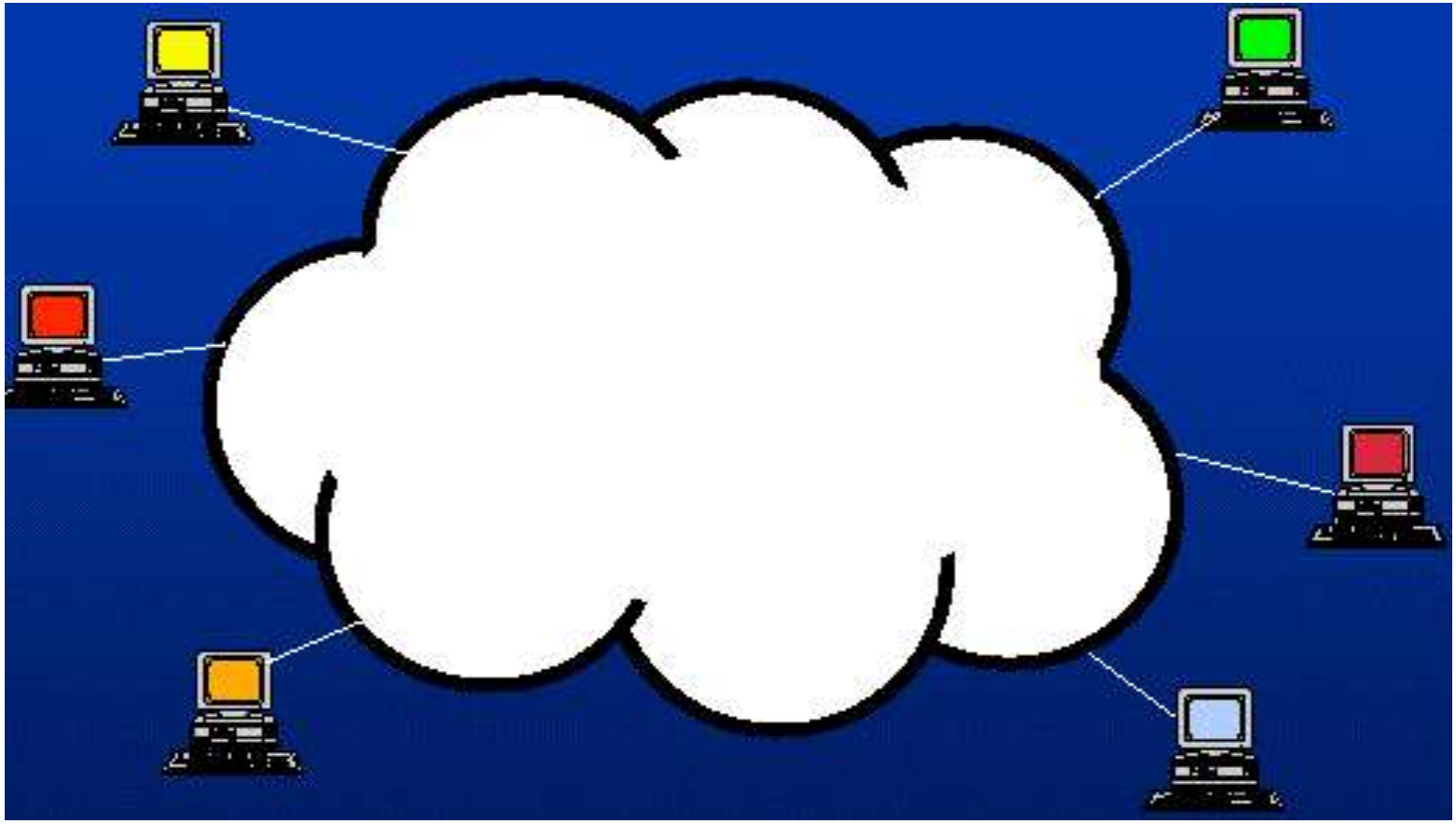
C. Pham
Université de Pau et des Pays de l'Adour
Département Informatique
http://www.univ-pau.fr/~cpham
Congduc.Pham@univ-pau.fr

Copyright

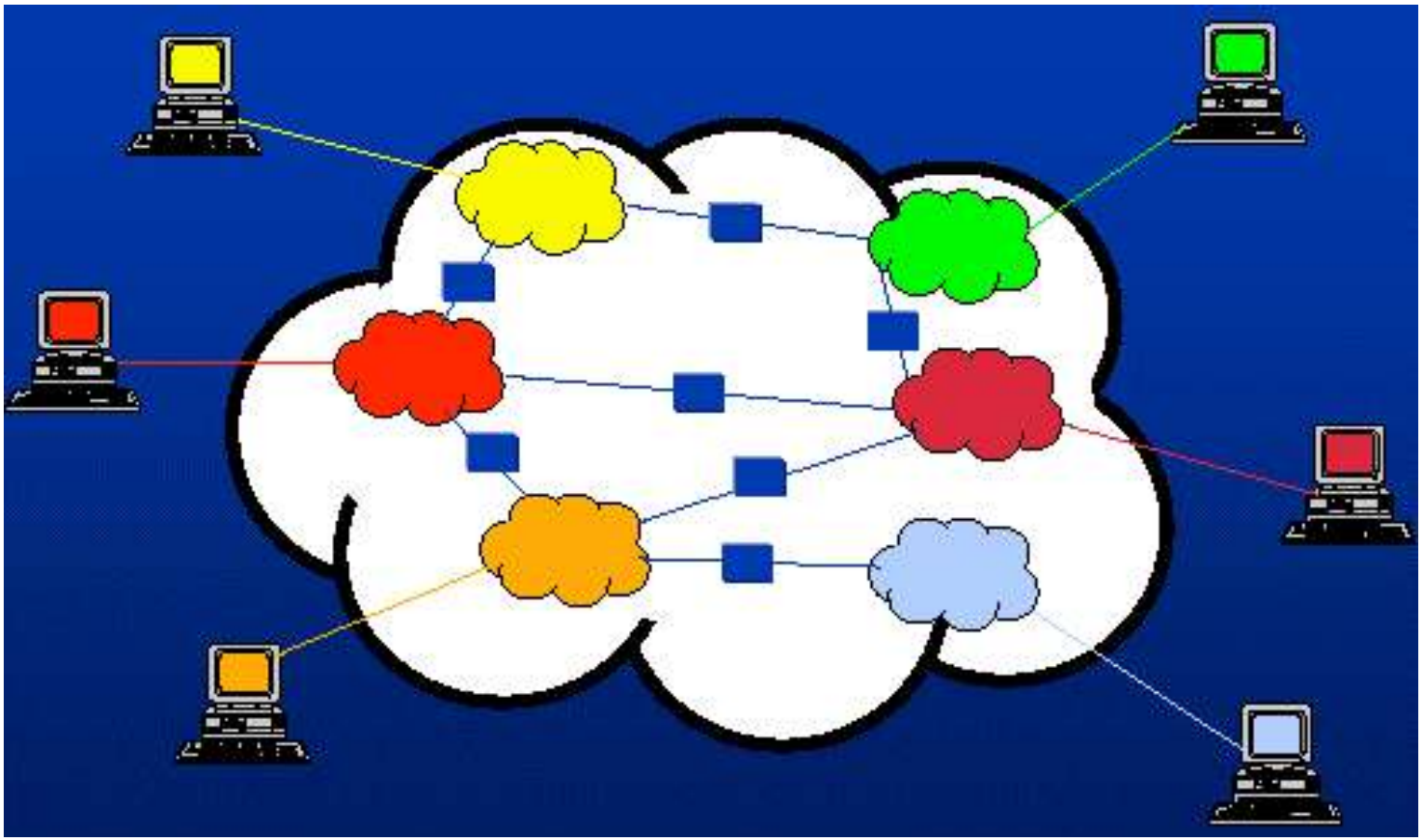
Références

- **Ces supports ont empruntés des image et des informations d'un grand nombre de sources:**
  - A. Tanenbaum, "Computer Network"
  - Keshav, "An Engineering Approach to Computer Networking"
  - L. Toutain, "Réseaux Locaux et Internet"
  - Supports de cours de Shivkumar Kalyanaraman (et de manière indirecte J. Kurose, I. Stoica…)
  - Supports de cours de l'UREC
  - Supports de Cisco
  - Sources diverses sur l'Internet

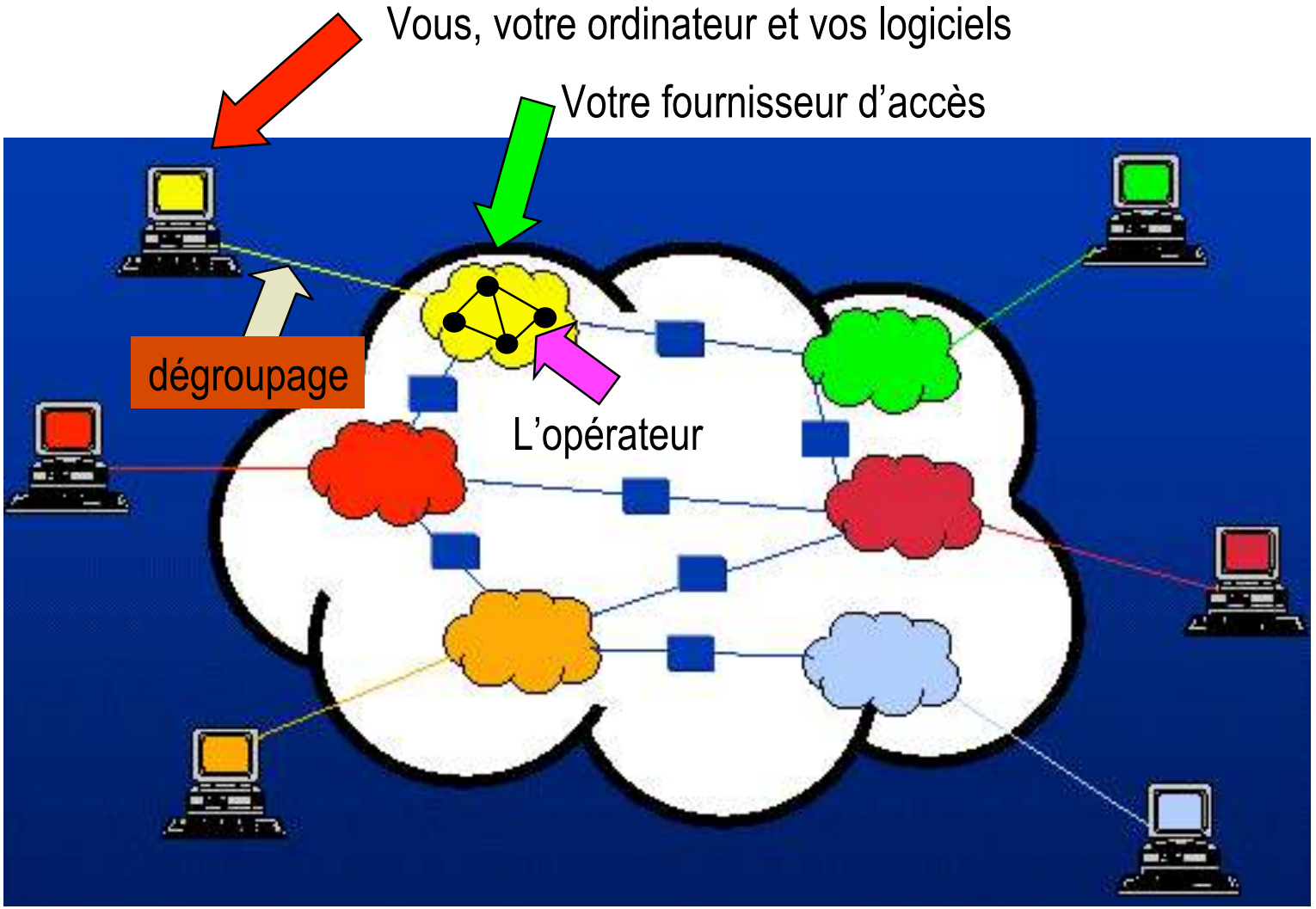
L’Internet du point de vue de l ’utilisateur



L'Internet, en vrai...

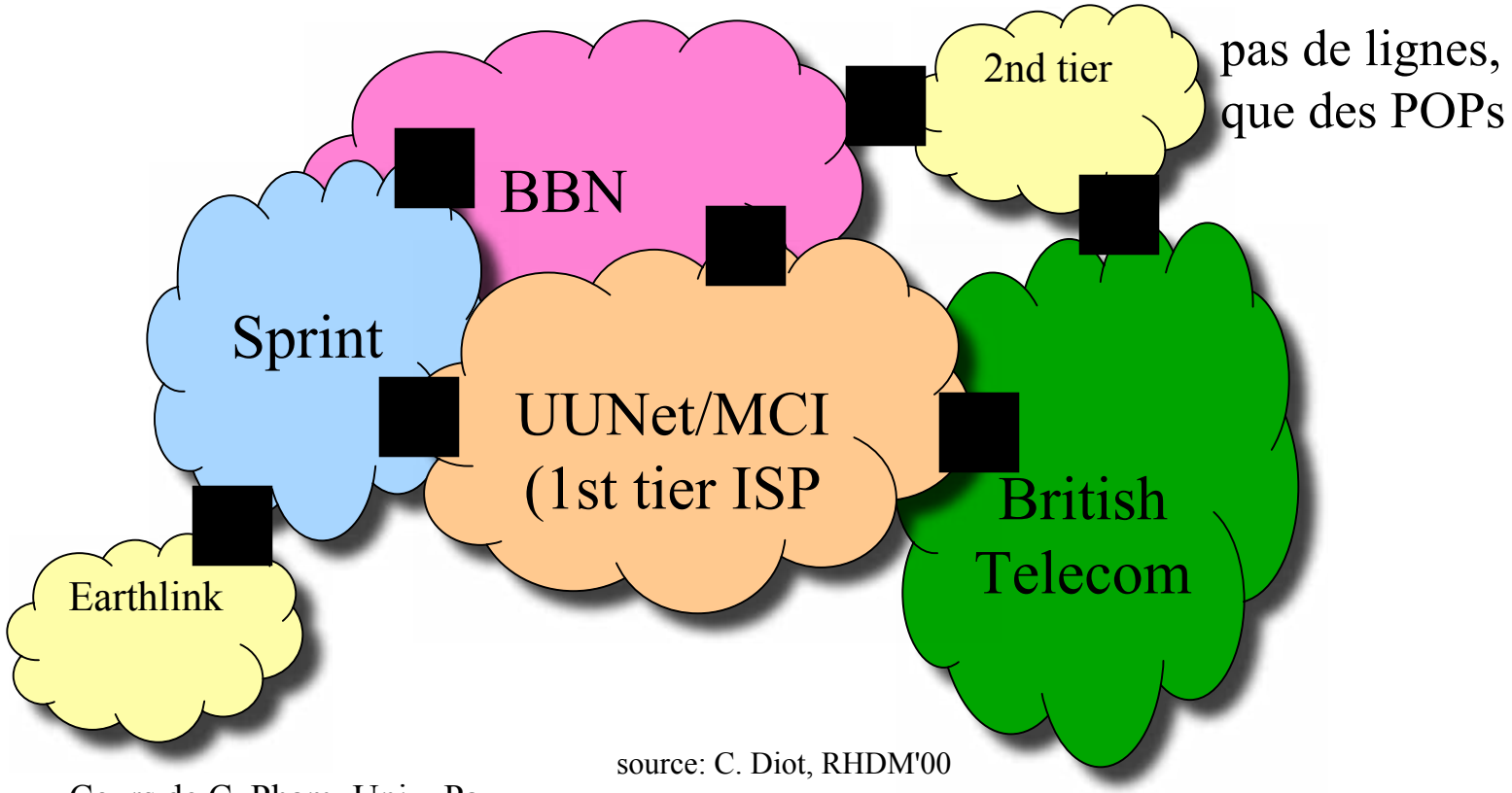


L'Internet, en vrai...



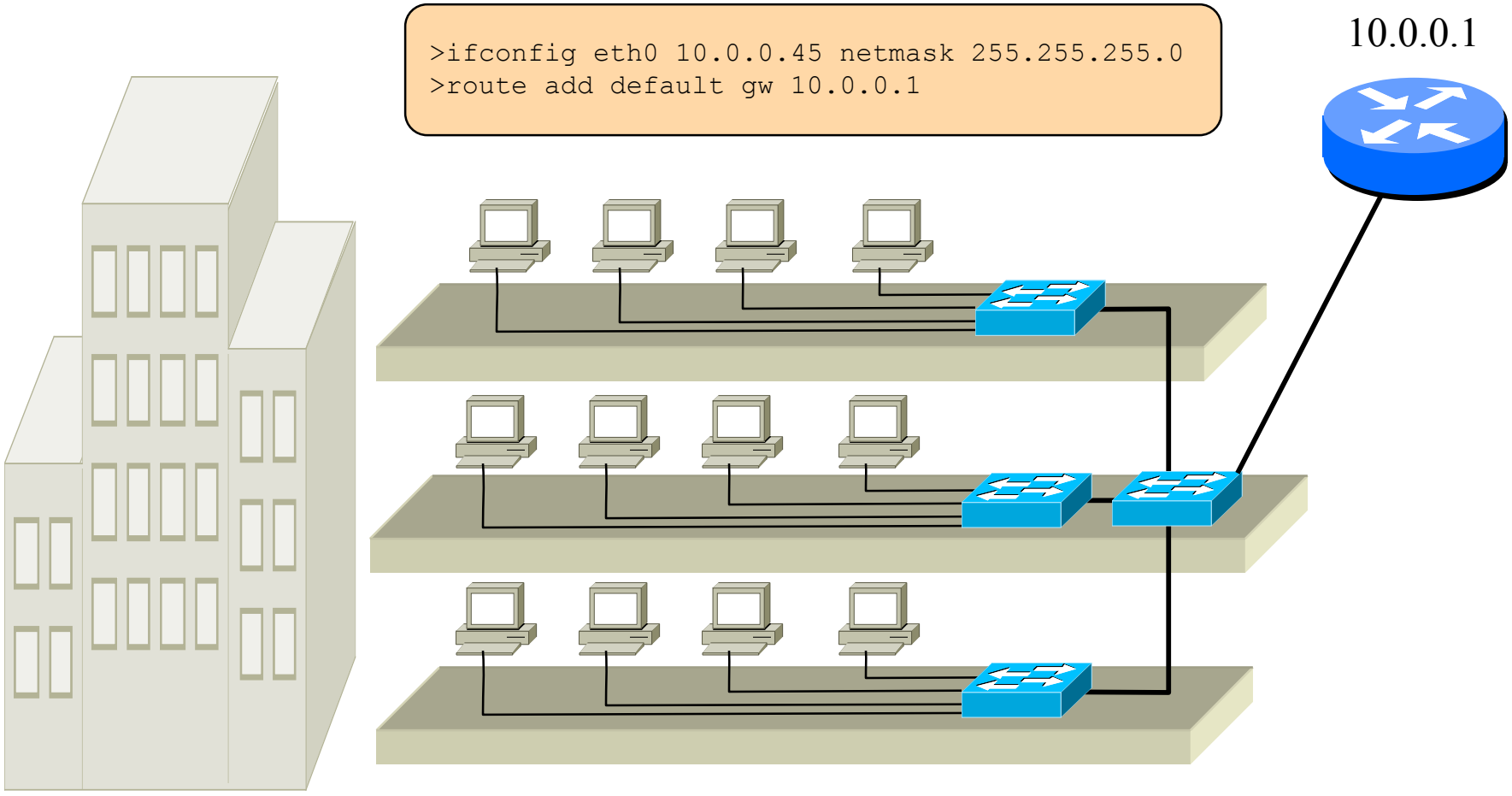
Le visage de l'Internet aujourd'hui

- **Les « 1st tier ISP » possèdent des lignes.**
- **L'interconnexion se produit essentiellement à des points de peering privé.**



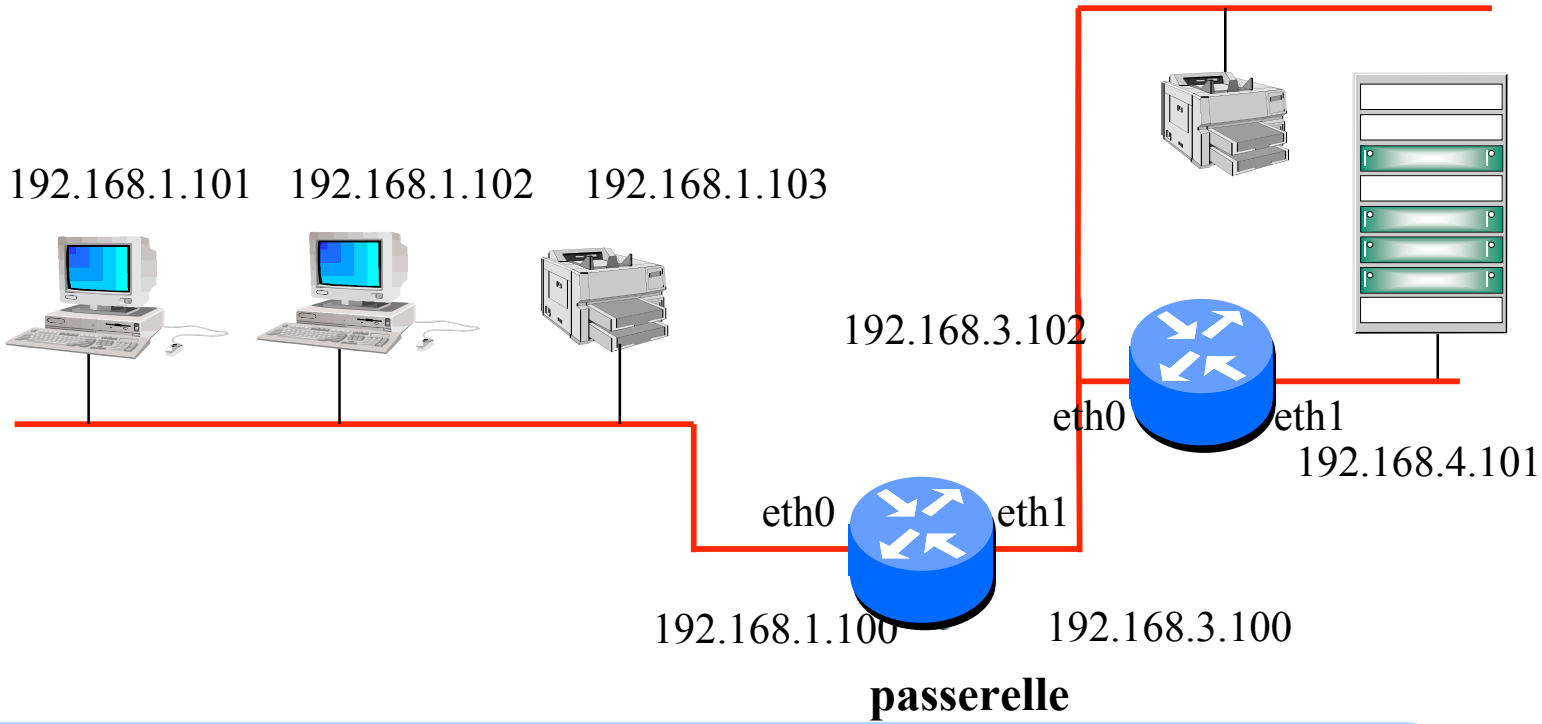
source: C. Diot, RHDM'00

Qu'y a t-il après le réseau local?



Routeurs

- = passerelle avec matériel & logiciel dédiés



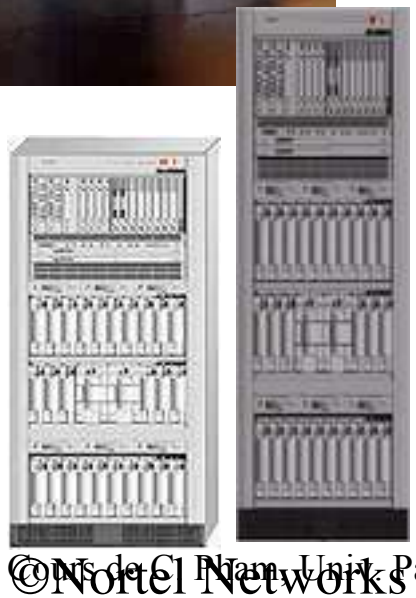
Un routeur est un matériel réseau spécifique, conçu spécialement pour le routage. Faire un routeur avec un PC à plusieurs cartes est possible, mais peu efficace!

High Performance Routers

©Juniper

©Procket Networks

©Alcatel



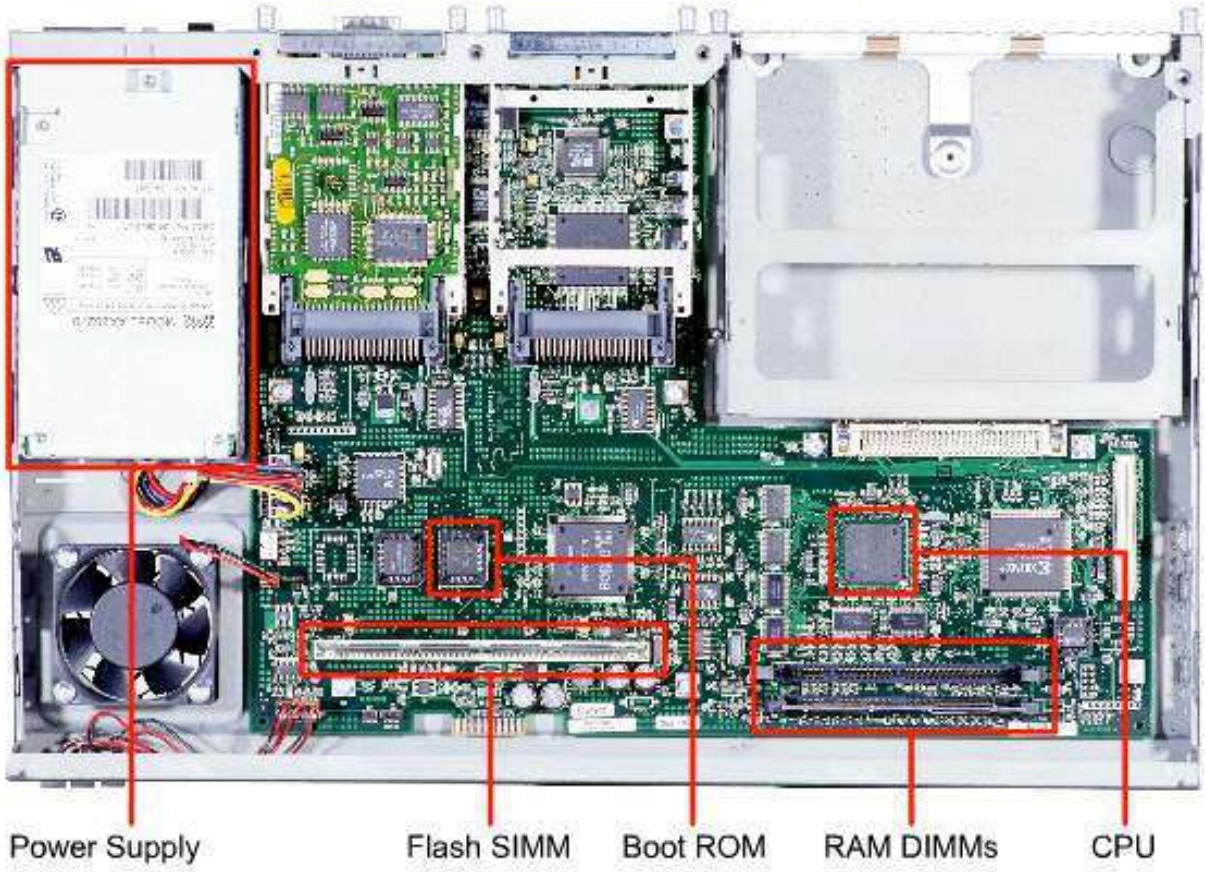
©Nortel Networks

©Lucent

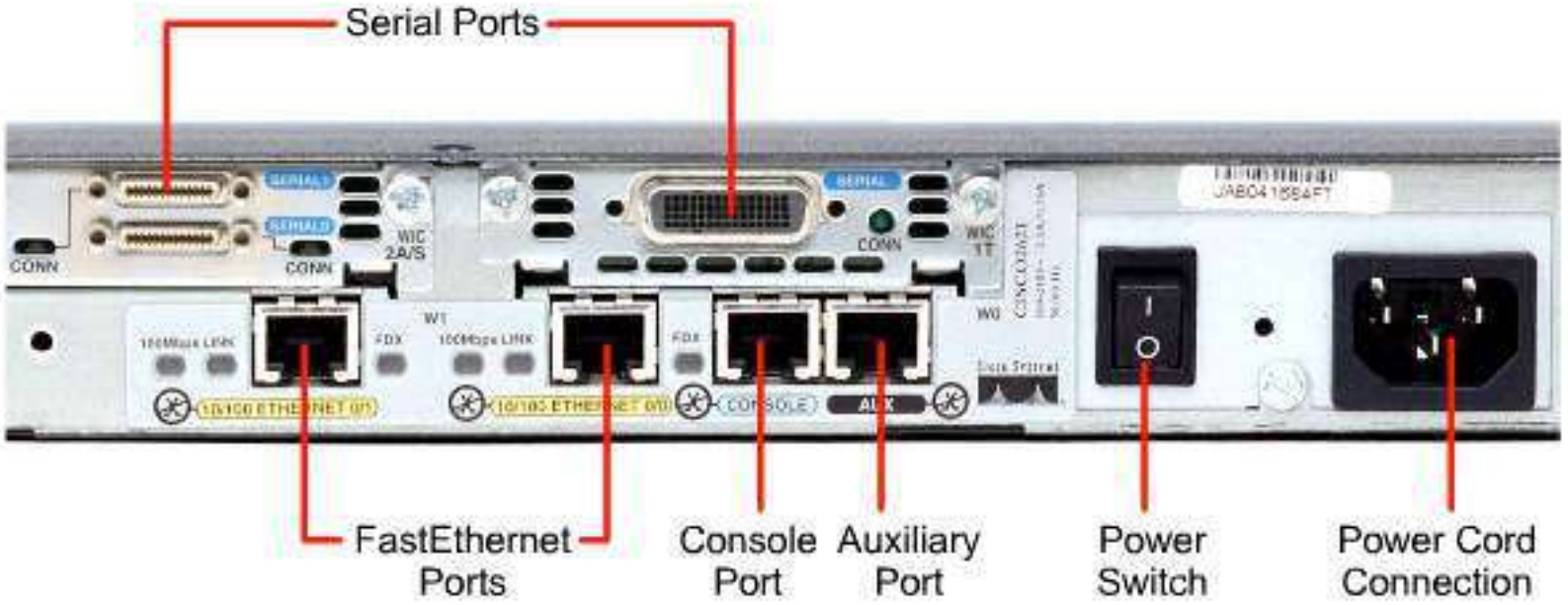
and more…

Added by C. Pham

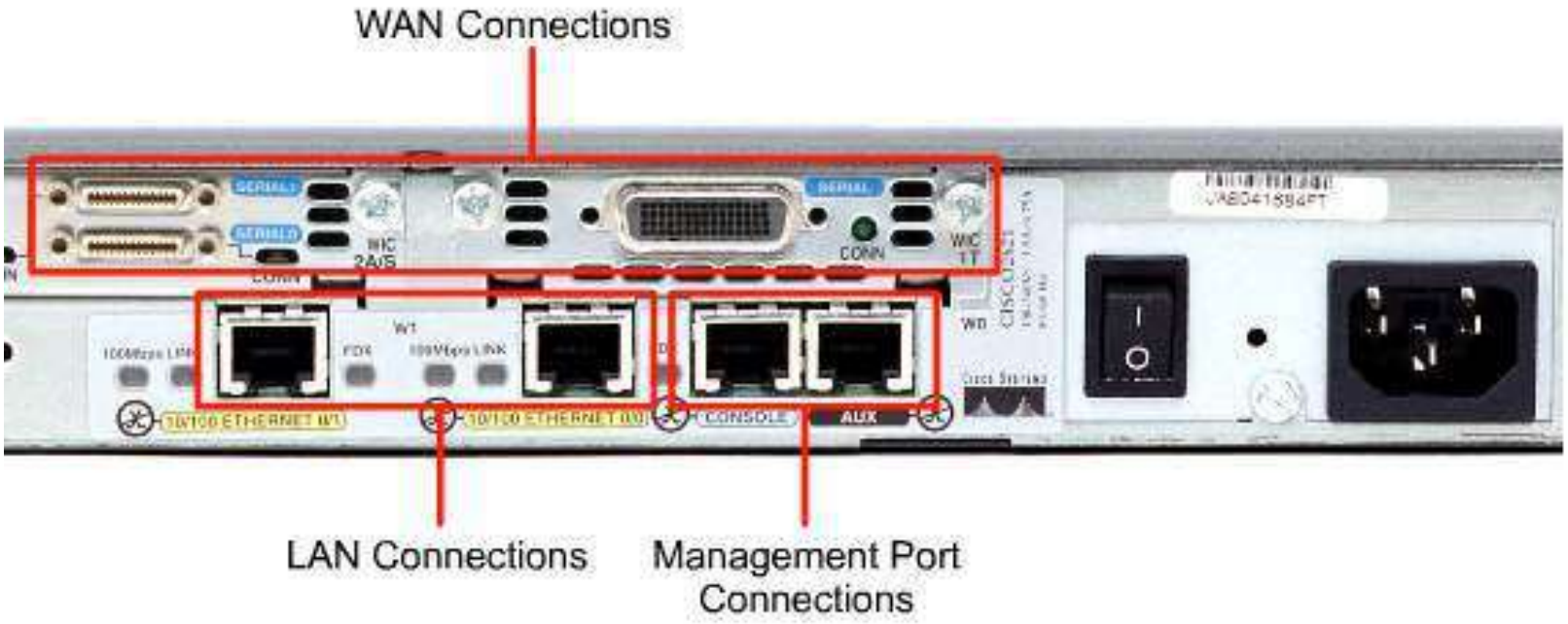
Internal Components of a 2600 Router



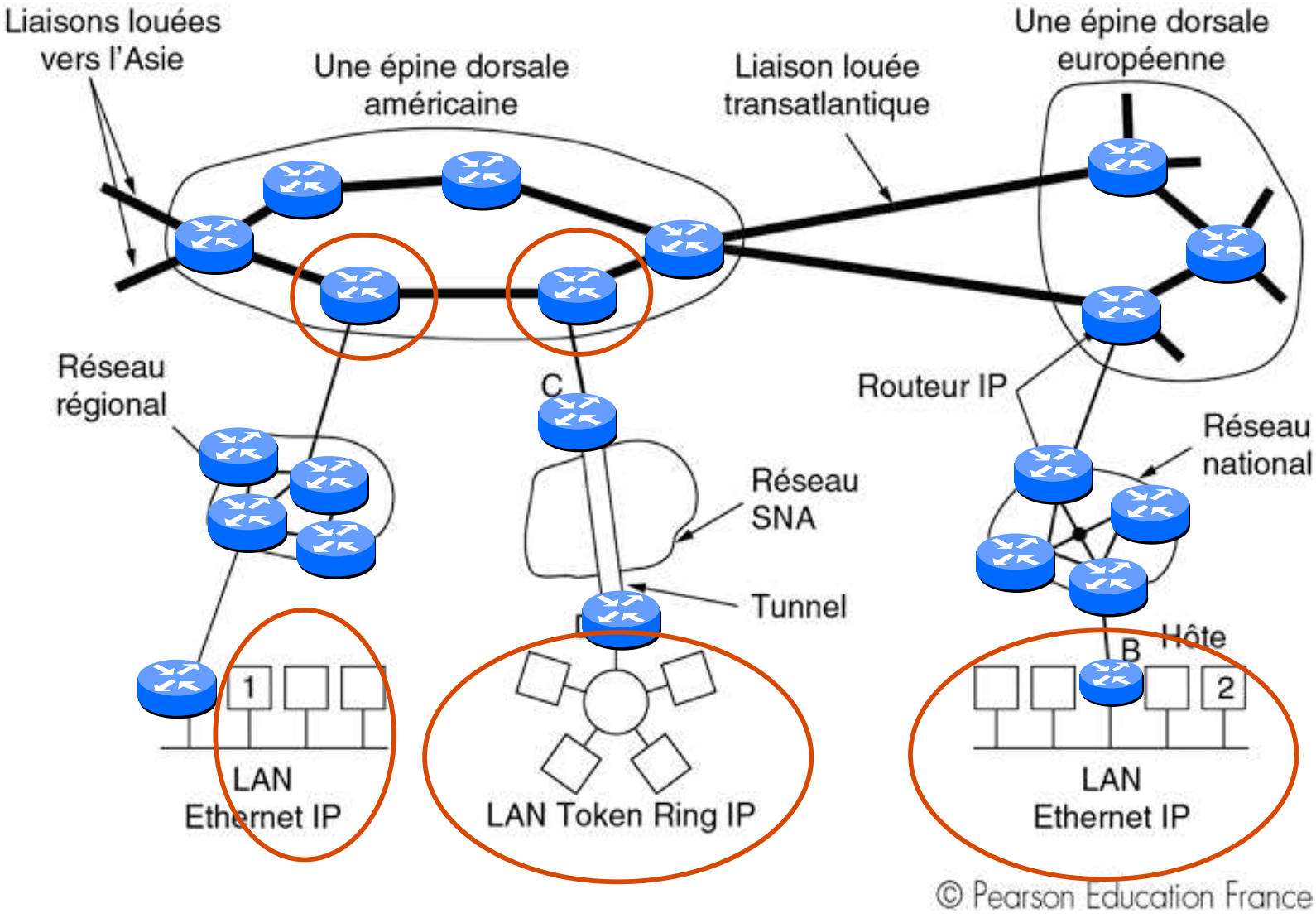
External Connections on a 2600 Router



Router External Connections



La diversité des réseaux



Qu'est-ce qu'un réseau longue distance?

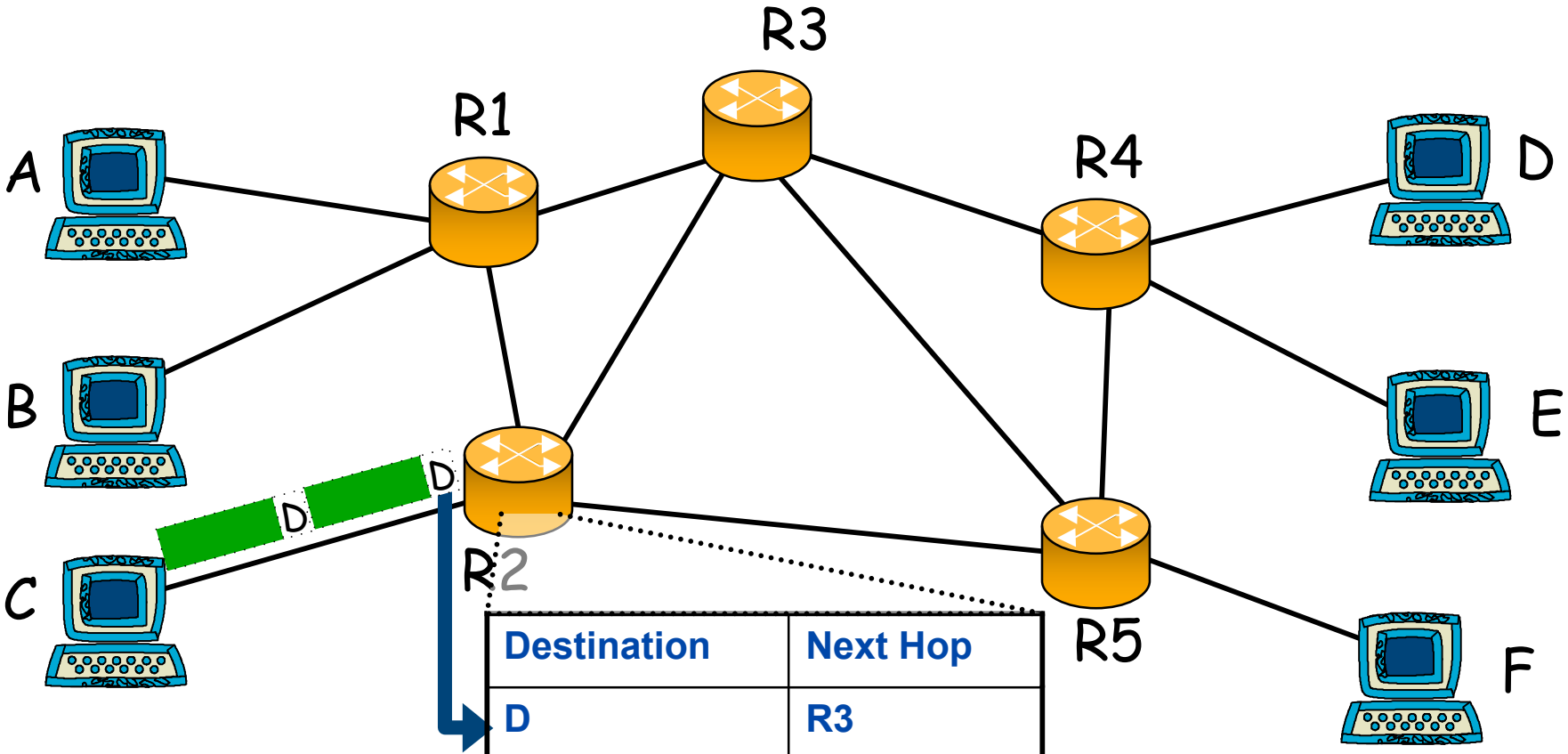
- **Réseaux longue distance**
  - grande couverture géographique,
  - hétérogénéité des modes de transmission,
  - mélange de réseaux publics et privés,
  - agrégation du trafic,
  - tarification par des opérateurs.

- **Comment aller plus loin?**
  - technique de transmission WAN
  - noeuds de "commutation » appelé routeurs,
  - interconnexion de réseaux,
  - introduction du routage.

Choix pour les protocoles de routage

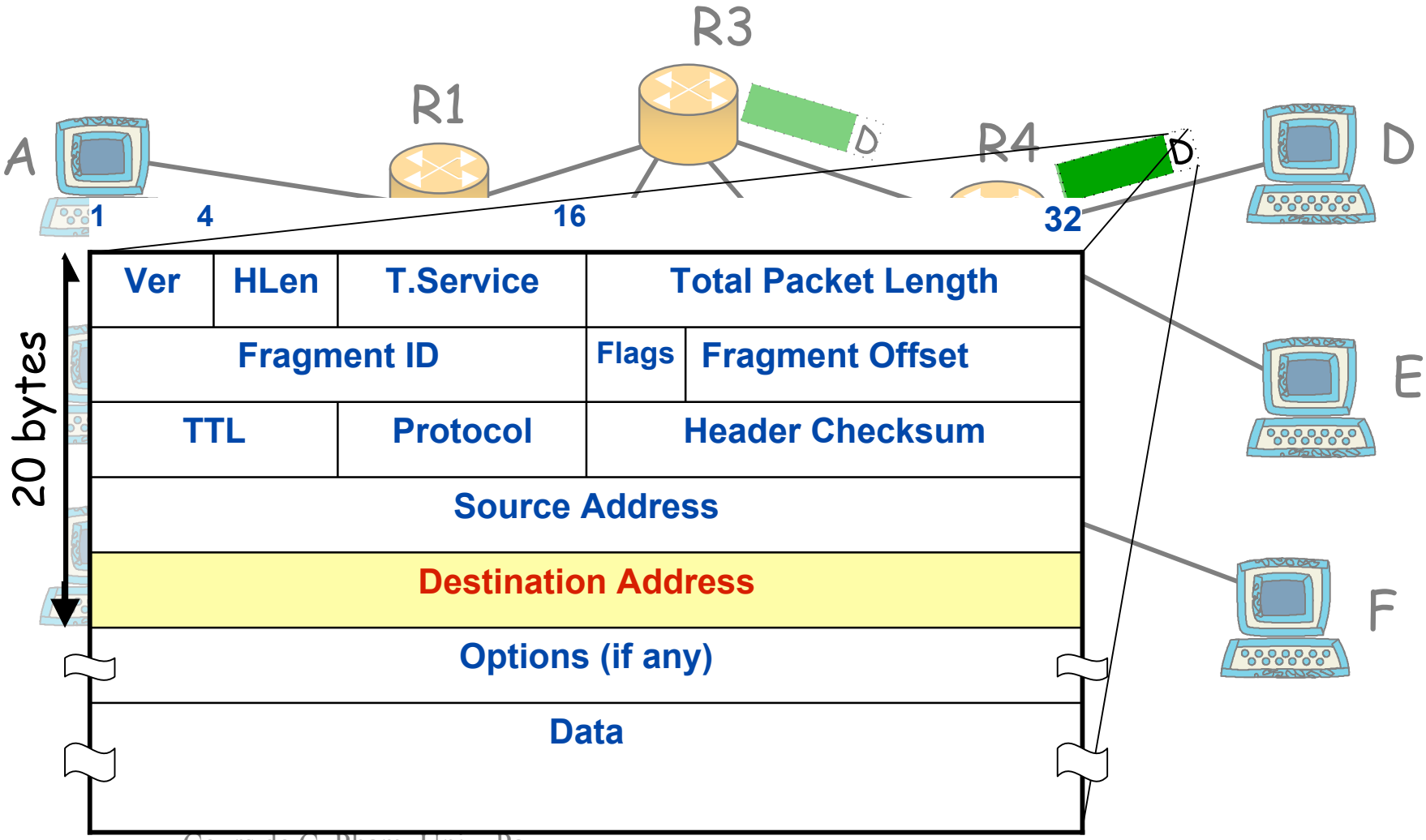
- **Centralisé vs distribué**
  - centralisé est simple mais sujet aux pannes et à la congestion.
- **Routage par la source vs hop-by-hop**
  - taille de l'entête importante pour le routage par la source.
- **Stoquastique vs déterministe**
  - stoquastique répartie la charge mais dé-séquence.
- **Dépendant ou indépendant de l'état?**
  - dépendant de l'état plus efficace mais plus complexe.

Le routage de proche en proche illustré



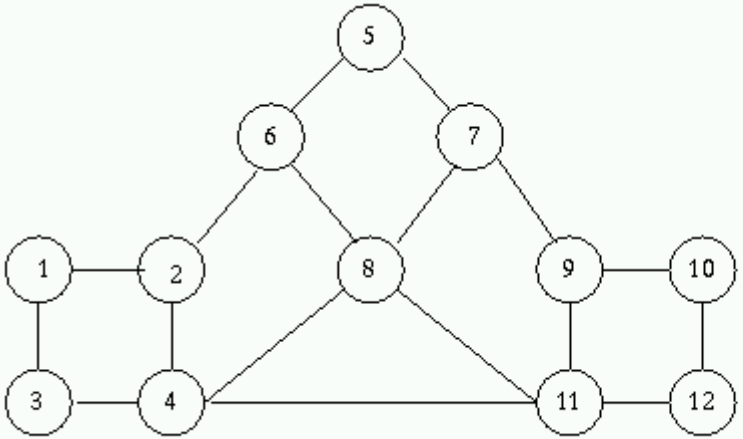
Destination	Next Hop
D	R3
E	R3
F	R5

Le routage IP



Routage: principes de base

- Un algorithme de routage rempli une table de routage dans les routeurs



Dest	Hop	Dest	Hop
1	-	7	3
2	2	8	2
3	3	9	2
4	3	10	2
5	2	11	3
6	2	12	3

- PB: Choix local sur un système global

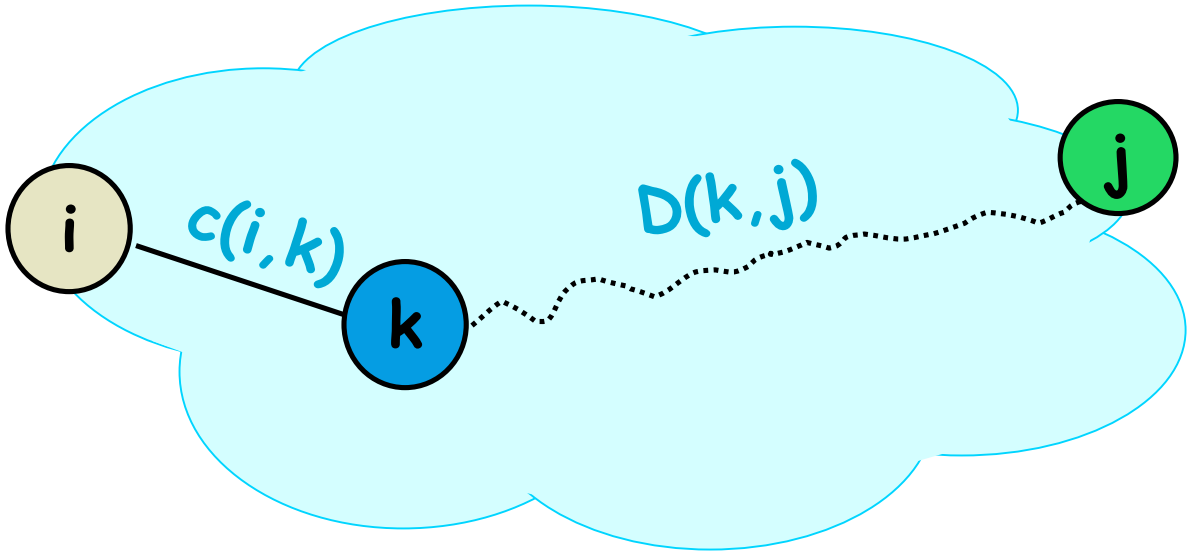
Les protocoles de routages pour réseaux paquets

- **Vecteur de distance (Distance-Vector, DV)**
  - chaque routeur ne connaît initialement que le coût de ses propres liaisons, les routeurs échangent entre-eux des informations de coûts,
  - chaque routeur n'a qu'une vision partielle du réseau: coût vers chaque destination,
  - fonctionne bien sur des systèmes de petite taille.

- **Etat des liens (Link-State, LS)**
  - chaque routeur construit une vision complète de la topologie du réseau à partir d'informations distribuées,
  - ne pas confondre connaître la topologie et connaître tous les noeuds terminaux,
  - fonctionne sur des grands réseaux.

Critère de “consistence”

- **Un sous-ensemble d’un plus court chemin est aussi le plus court chemin entre les 2 deux nœuds intermédiaires**
- **Corollaire:**
  - Si le plus court chemin d’un nœud i à un nœud j, avec une distance de D(i,j) passe par un nœud voisin k avec un lien de coût c(i,k), alors: D(i,j) = c(i,k) + D(k,j)



Connaître les voisins de ses voisins…

- **Un routeur connaît ses voisins directement connectés…**
- **…ainsi que le "coût" pour y aller**
- **Coût = métrique pour comparer les routes entre elles**
  - La route la plus "courte" est celle dont le coût est minimal
  - Voisin direct = 1 saut, donc coût de 1 si la métrique est le nbr de saut
  - Voisin direct donc on connaît le débit du lien physique avec lequel on est connecté: débit faible=grand coût
- **Par propagation des informations de voisinage et de coût, chaque routeur va construire une table de routage**

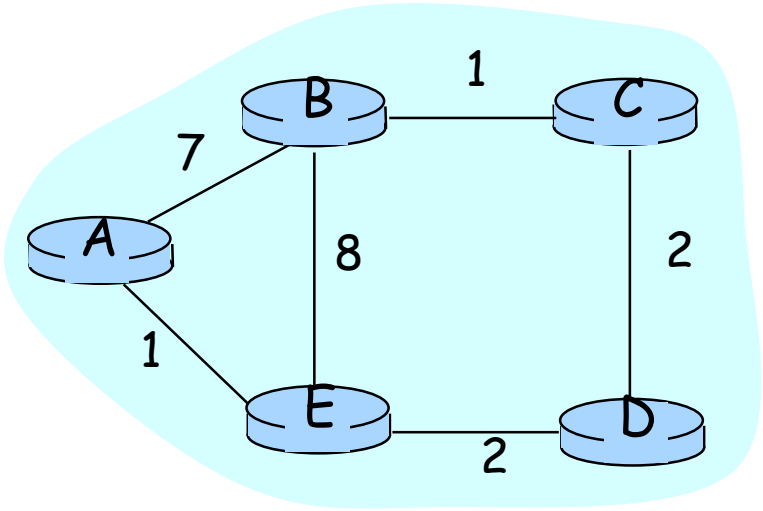
Notion de convergence

- **On parle de convergence lorsque tous les routeurs ont la même information de routage**
- **En cas de non convergence, les ressources du réseau peuvent être inaccessibles**
  - Les paquets sont acheminés vers d'autres destinations.
    - On parle de trou noir "Black holes" (les paquets disparaissent)
    - Bouclage du processus de routage (routing loops)
- **Le processus de convergence est déclenché après changement d'état d'un routeur ou d'un lien.**

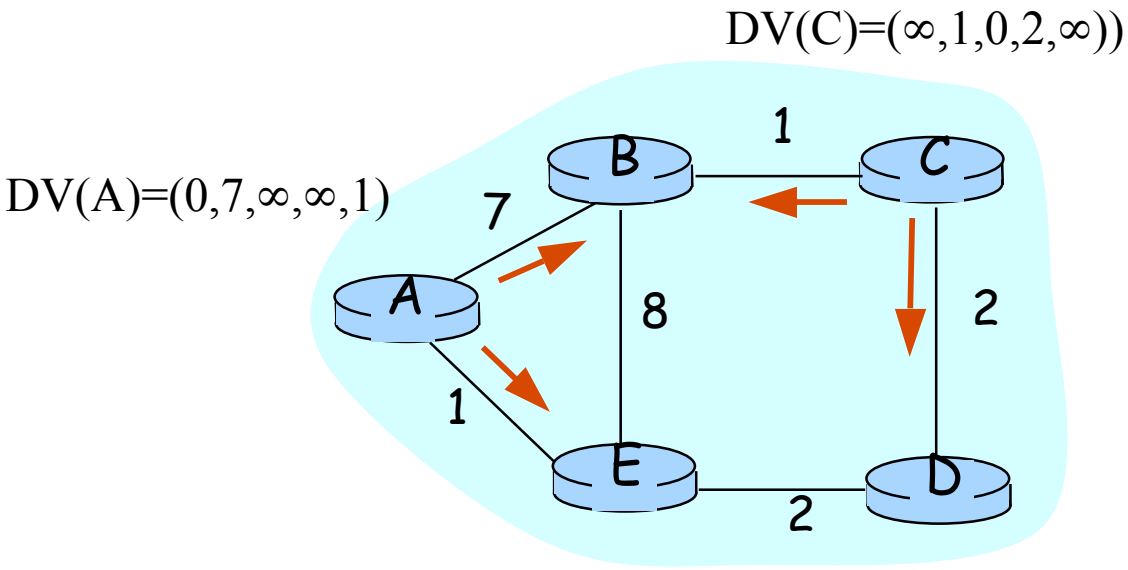
L'approche vecteur de distance (1)

- **Vecteur de distance (Distance-Vector, DV)**
  - chaque routeur ne connaît initialement que le coût de ses propres liaisons vers ses voisins direct. C'est le vecteur initial
  - chaque routeur va échanger son vecteur initial avec tous ses voisins
  - après un certain nombre d'itérations, chaque routeur va connaître le coût vers chaque destination,
  - fonctionne bien sur des systèmes de petite taille.

$$DV(A)=(0,7,\infty,\infty,1)$$



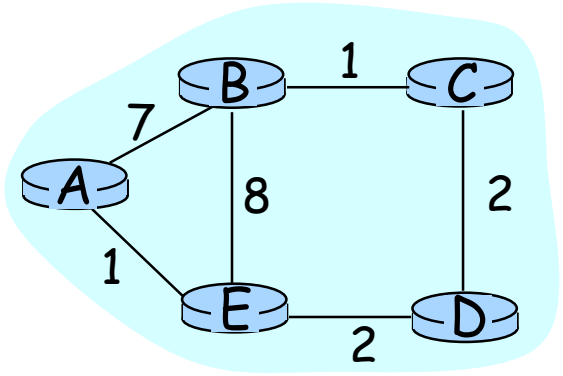
L'approche vecteur de distance (2)



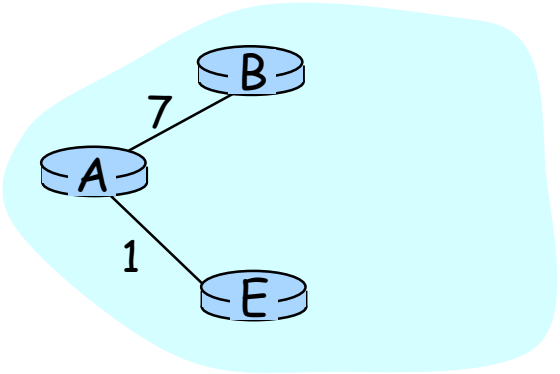
Pas obligatoirement de synchronisation dans les envois de messages

L'approche vecteur de distance (2)

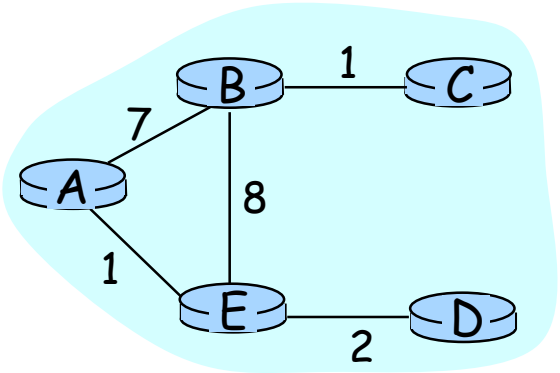
- **Condition de consistence: D(i,j) = c(i,k) + D(k,j)**
- **L'algorithme DV (Bellman-Ford) évalue cette condition de manière récursive**
  - À la m-iéme itération, le critère de consistence est vérifié, en supposant que chaque nœud N "voit" les noeuds et les liens à m-sauts (ou moins) de lui (i.e. on a une vision à m-sauts)



Réseau d'étude



Vision de A à 1-saut (après la 1ère itération)



Vision de A à 2-sauts (après la 2nd itération)

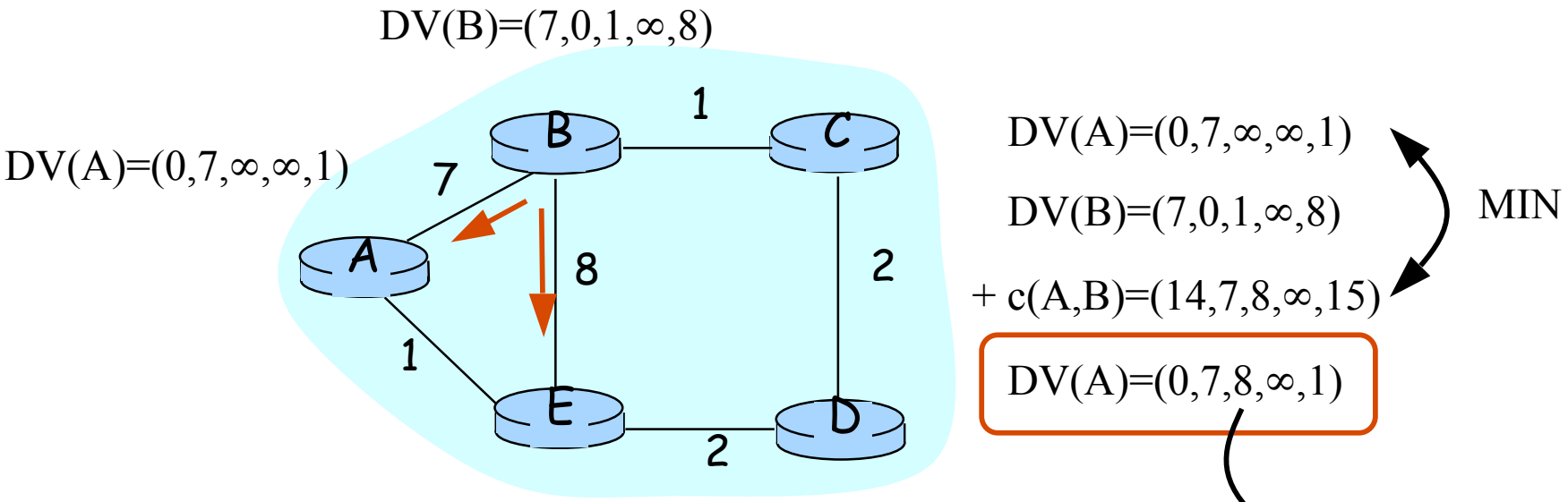
Algorithme DV (1)

- **Valeurs initiales (itération 1):**
  - D(i,i) = 0 ;
  - D(i,k) = c(i,k) si k est un voisin (i.e. k est à 1 saut); et
  - D(i,j) = INFINITY pour tous non-voisin j.
- **D(i,*) est le vecteur de distance du noeud i.**
- **L'algorithme maintient une table de "forwarding" pour toute destination j, initialisée comme suit:**
  - Prochain-saut(i) = i;
  - Prochain-saut(k) = k si k est un voisin, et
  - Prochain-saut(j) = INCONNU si j n'est pas un voisin.

Algorithme DV (2)

- **Après chaque itération chaque noeud i échange sont vecteur de distance D(i,*) avec ses voisins directs.**
- **Pour tout voisin k, si c(i,k) + D(k,j) < D(i,j), alors:**
  - D(i,j) = c(i,k) + D(k,j)
  - prochain-saut(j) = k
- **Après chaque itération, le critère de consistence est vérifié**
  - Après m itérations, chaque nœud connaît le plus court chemin vers tous les autres nœuds situés à m sauts ou moins.
  - i.e. chaque nœud à une vision à m sauts du réseau.
  - L'algorithme converge en O(d) iterations: d est le diamètre maximum du réseau.

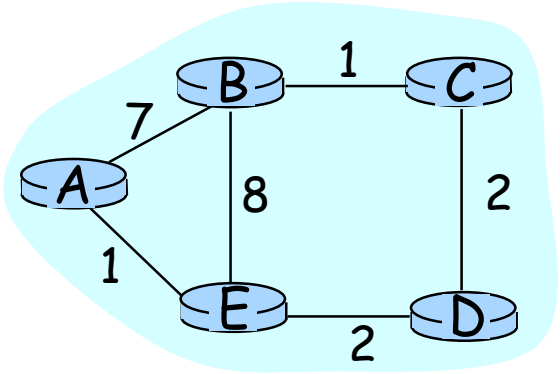
Algorithme DV (3)



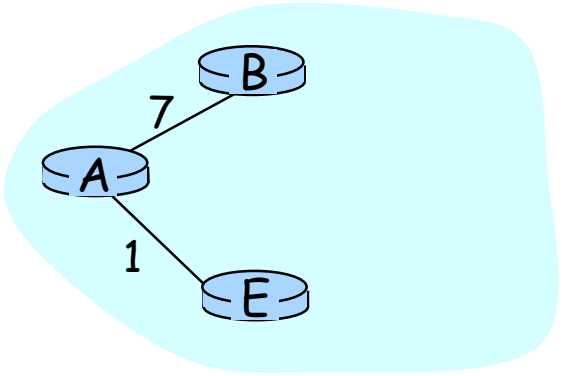
- **A reçoit de B: DV(B,*)=(7,0,1, ∞,8)**
- **Pour tout voisin k, si c(i,k) + D(k,j) < D(i,j), alors:**
  - D(i,j) = c(i,k) + D(k,j)
  - prochain-saut(j) = k
- **Pour voisin B, si c(A,B)+D(B,C) < D(A,C), alors:**
  - D(A,C) = c(A,B) + D(B,C)
  - prochain-saut(C) = B

Algorithme DV (4)

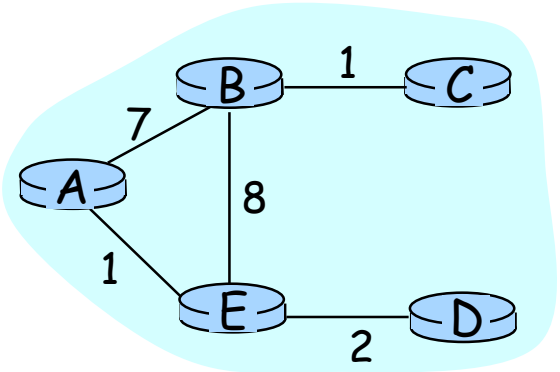
- **Vecteur de distance de A: D(A,*):**
  - Après Itération 1 est: [0, 7, INFINITY, INFINITY, 1]
  - Après Itération 2 est: [0, 7, 8, 3, 1]
  - Après Itération 3 est: [0, 7, 5, 3, 1]
  - Après Itération 4 est: [0, 6, 5, 3, 1]



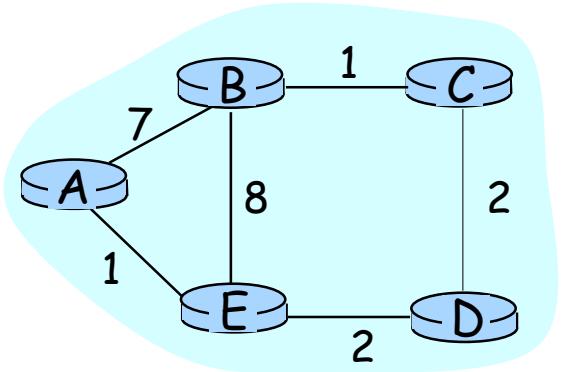
Réseau d'étude



Vision de A à 1-saut (après la 1ère itération)

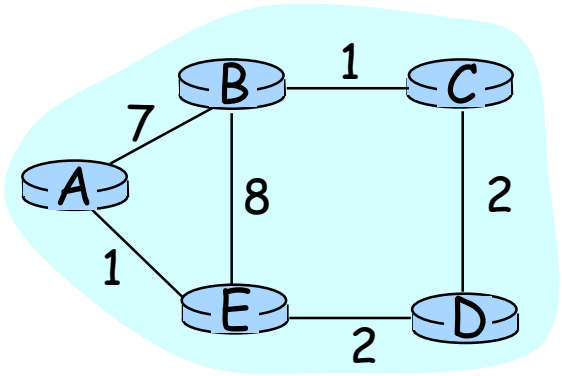


Vision de A à 2-sauts (après la 2nd itération)



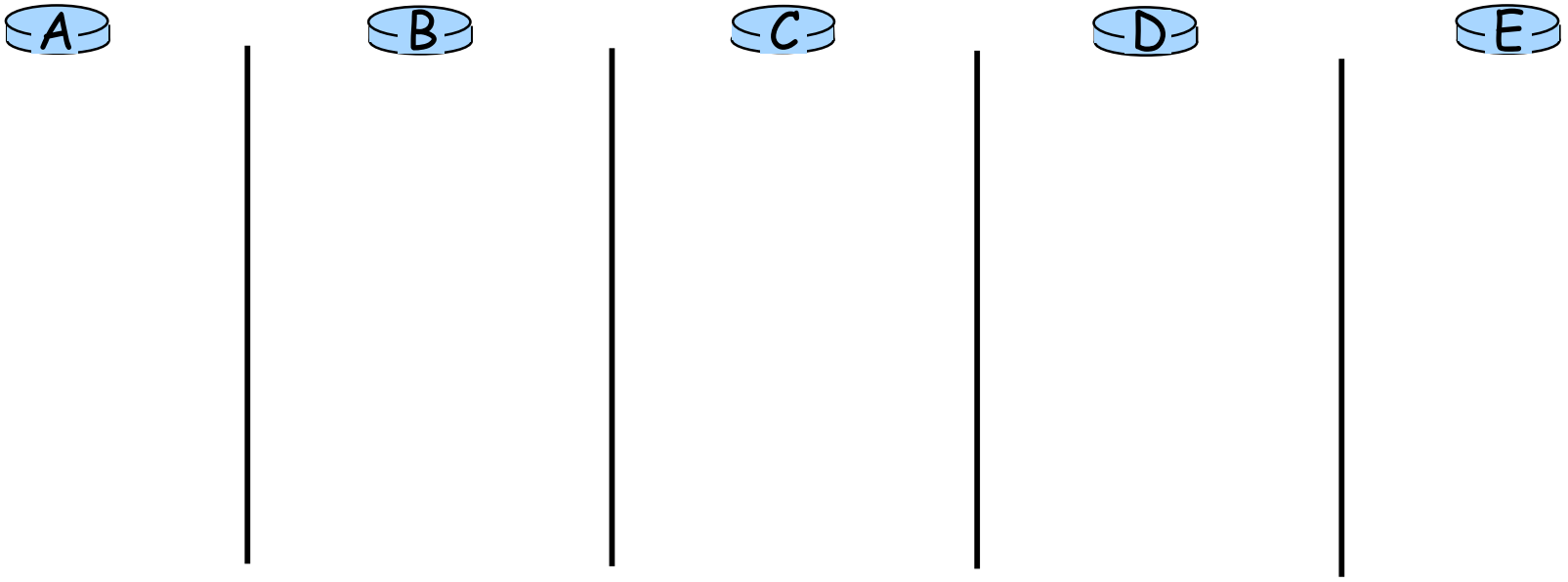
Vision de A à 3-sauts (après la 3e itération)

Pas à pas…au tableau

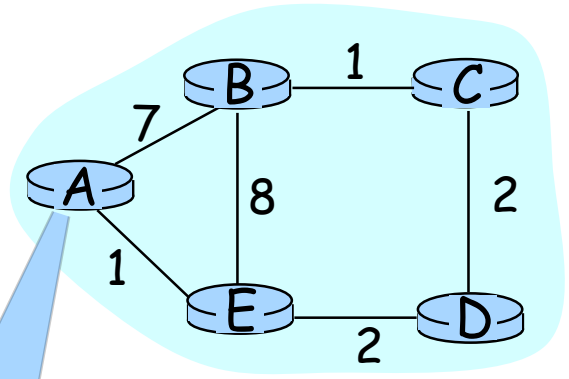


Réseau d'étude

Pour pouvoir le faire, il faut fixer l'ordre des routeurs pour l'envoi des messages: B, C, D, E



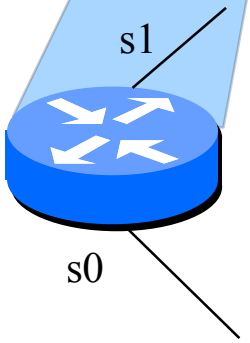
Vecteur de distance en pratique



Réseau d'étude

B	s1
E	s0

B	s1
C	s1
E	s0



DV(A)=(0,7,∞,∞,1)

DV(B)=(7,0,1,∞,8)

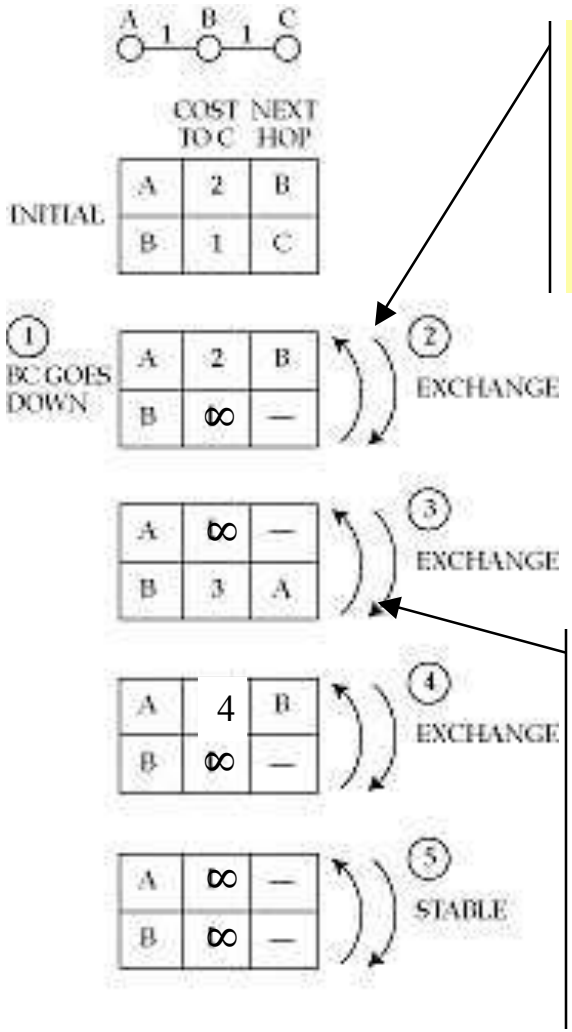
MIN

+ c(A,B)=(14,7,8,∞,15)

DV(A)=(0,7,1,∞,1)

Problèmes avec le Distance Vector

- **Comptage à l'infini**
  - Les packets peuvent osciller entre A et B



B envoie à A son nouveau DV. En même temps, A envoie à B son DV indiquant une 2-hop chemin pour C.

A découvre que B n'a plus de chemin vers C et lui indique donc le changement. B par contre croit avoir un moyen de joindre C avec un 3-hop chemin.

Split horizon

– ne jamais dire à un voisin Y le coût vers X si Y est le prochain routeur pour aller vers X

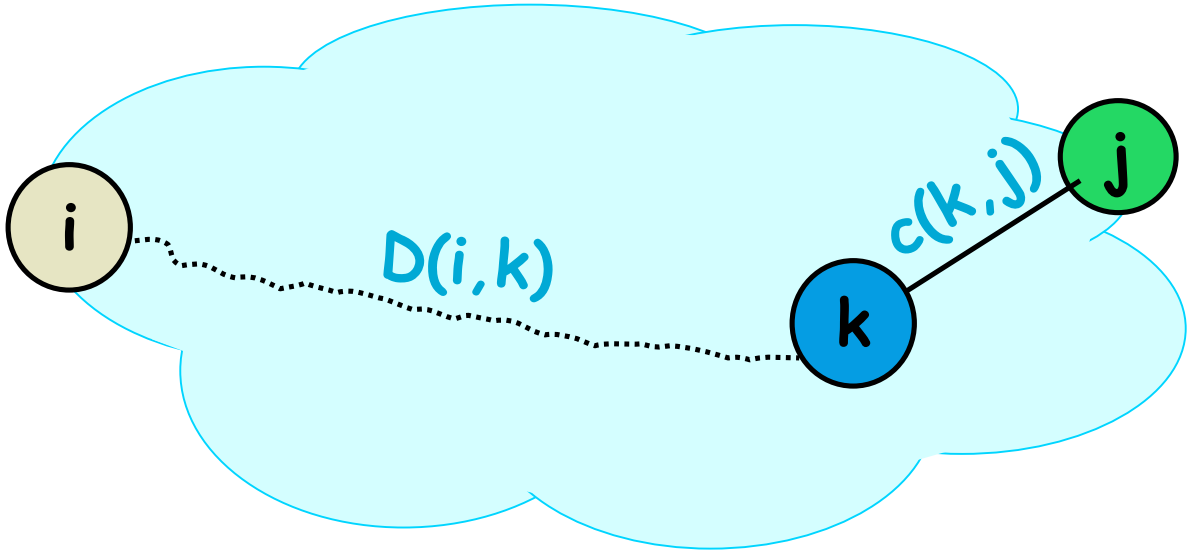
– ne fonctionne pas sur des configuration à 3 chemins!

Approche “état des liens” (1)

- **Dans l’approche distance vector, les routeur ne connaissent que le coût pour chaque destination**
- **Dans l’approche “etat des liens” (link state) les routeurs connaissent la topologie entière, et calculent chacun les meilleures routes**
  - Les calculs sont indépendants
  - Convergence plus rapide
  - Gestion de plus grands réseaux
  - Potentiellement plus robuste
- **2 phases principale**
  - Échanger des informations de topologie
  - Calcul des plus court chemin

Approche “état des liens” (2)

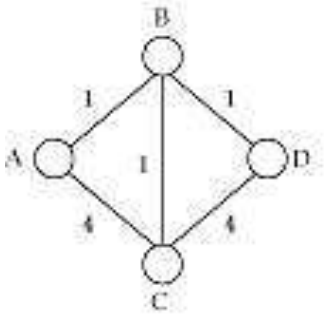
- **L’approche état des liens est itérative, et pivote autour des destinations j, and leur prédécesseurs k = p(j)**
  - Une autre vue de du critère de consistence est utilisée:
  - D(i,j) = D(i,k) + c(k,j)



- **Chaque noeud i collecte tous les états c(*,*) d’abord puis exécute localement l’algorithme de plus court chemin (Dijkstra).**

Diffusion de la topologie

- **Un routeur décrit son voisinage avec un *link state packet (LSP)***



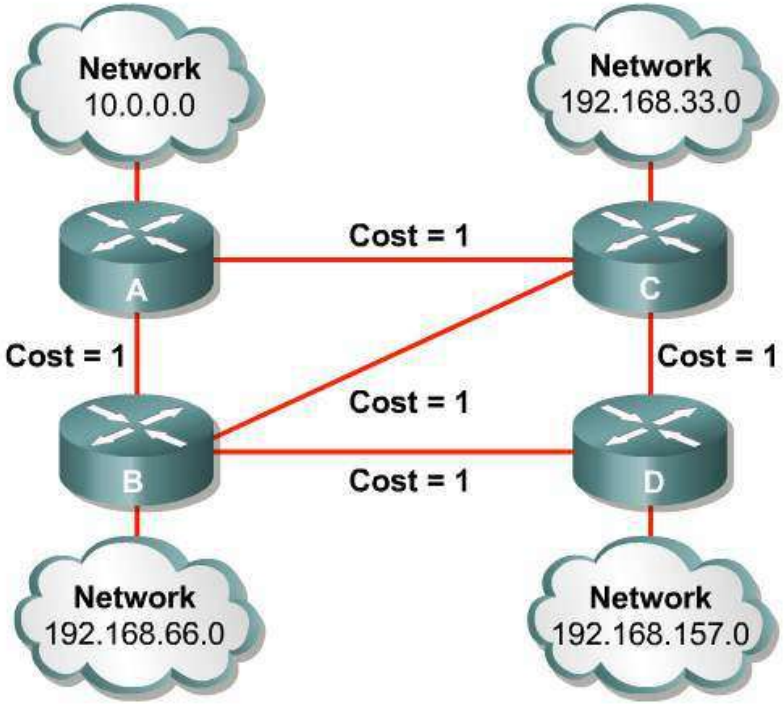
LSPs CREATED BY A

A	B	1
A'S ID	B'S ID	COST

A	C	4

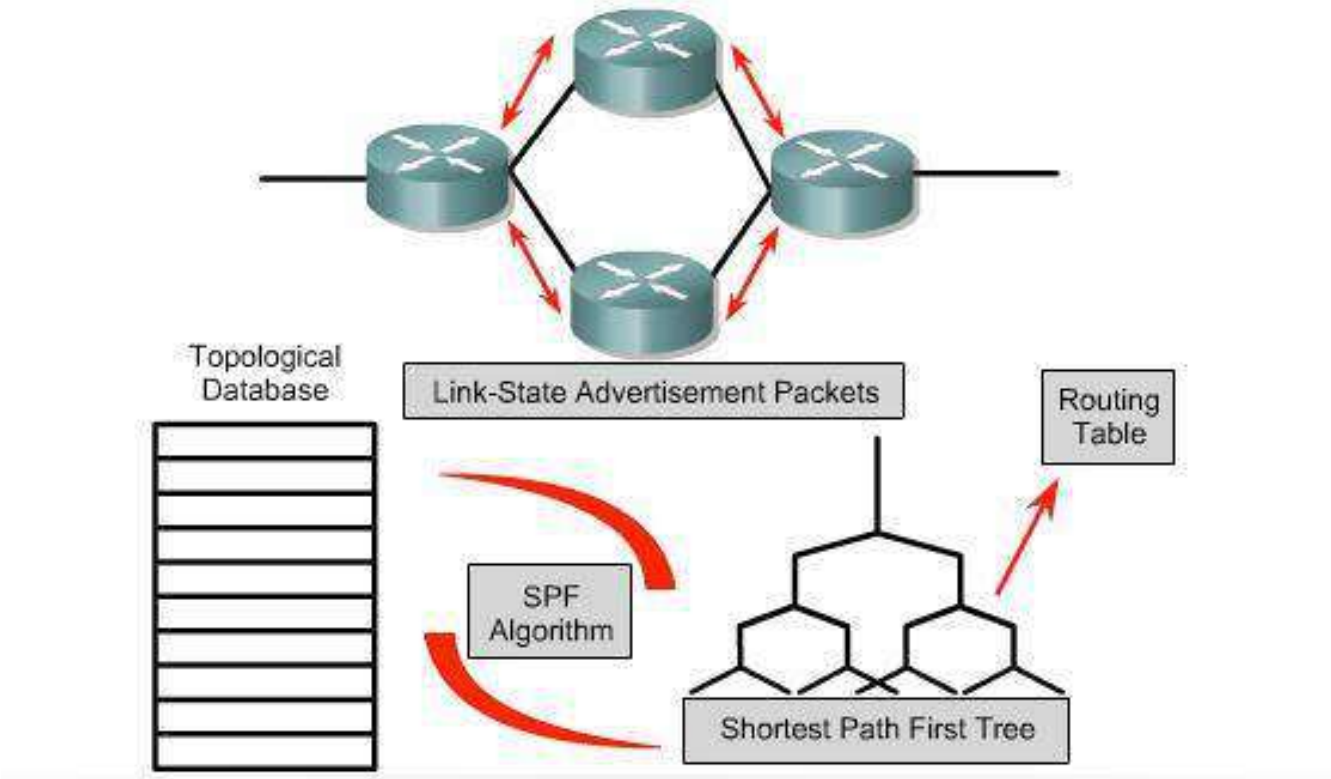
- **Utilise une diffusion contrôlée pour distribuer l'information dans le réseau**
  - Garde le LSP dans une base de données de LSP
  - Si nouvelle, transmet sur chaque interface, sauf l'interface entrante
  - Un réseau avec E sommets transmettra au plus 2E fois

Topologie entière du réseau



Router	Destination	Next Hop	Cost
A	192.168.66.0	B	1
A	192.168.33.0	C	1
A	192.168.157.0	B	2
A	192.168.157.0	C	2
B	10.0.0.0	A	1
B	192.168.33.0	C	1
B	192.168.157.0	D	1
C	10.0.0.0	A	1
C	185.134.0.0	B	1
C	192.168.157.0	D	1
D	10.0.0.0	B	2
D	10.0.0.0	C	2
D	192.168.66.0	B	1
D	192.168.33.0	C	1

Etat des liens: vue d'ensemble



Routers send LSAs to their neighbors. The LSAs are used to build a topological database. The SPF algorithm is used to calculate the shortest path first tree in which the root is the individual router. A routing table is then created.

Link State (LS) Approach…

- **After each iteration, the algorithm finds a new destination node j and a shortest path to it.**
- **After m iterations the algorithm has explored paths, which are m hops or smaller from node i.**
  - It has an m-hop view of the network just like the distance-vector approach
- **The Dijkstra algorithm at node i maintains two sets:**
  - set N that contains nodes to which the shortest paths have been found so far, and
  - set M that contains all other nodes.
  - For all nodes k, two values are maintained:
    - D(i,k): current value of distance from i to k.
    - p(k): the predecessor node to k on the shortest known path from i

Dijkstra: Initialization

- **Initialization:**
  - D(i,i) = 0 and p(i) = i;
  - D(i,k) = c(i,k) and p(k) = i if k is a neighbor of i
  - D(i,k) = INFINITY and p(k) = UNKNOWN if k is not a neighbor of i
  - Set N = { i }, and next-hop (i) = i
  - Set M = { j | j is not i}

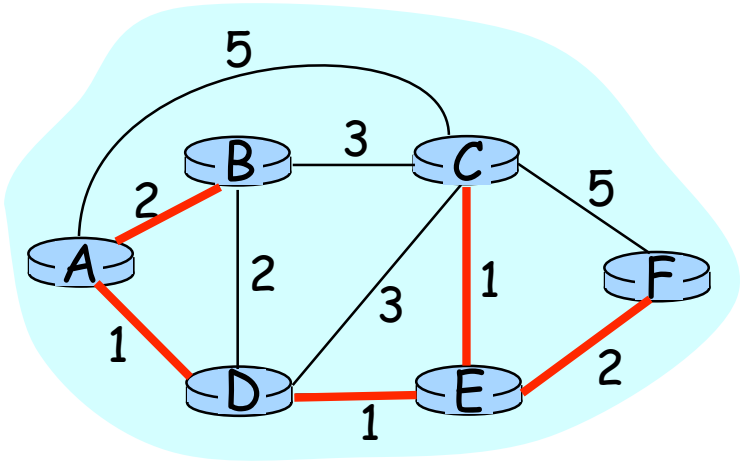
- **Initially set N has only the node i and set M has the rest of the nodes.**
- **At the end of the algorithm, the set N contains all the nodes, and set M is empty**

Dijkstra: Iteration

- **In each iteration, a new node j is moved from set M into the set N.**
  - Node j has the minimum distance among all current nodes in M, i.e. D(i,j) = min {l ε M} D(i,l).
  - If multiple nodes have the same minimum distance, any one of them is chosen as j.
  - Next-hop(j) = the neighbor of i on the shortest path
    - Next-hop(j) = next-hop(p(j)), if p(j) is not i
    - Next-hop(j) = j, if p(j) = i
- **Now, in addition, the distance values of any neighbor k of j in set M is reset as:**
    - If D(i,k) < D(i,j) + c(j,k), then
    - D(i,k) = D(i,j) + c(j,k), and p(k) = j.
- **This operation is called "relaxing" the edges of node j.**

Dijkstra's algorithm: *example*

Step	set 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	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					

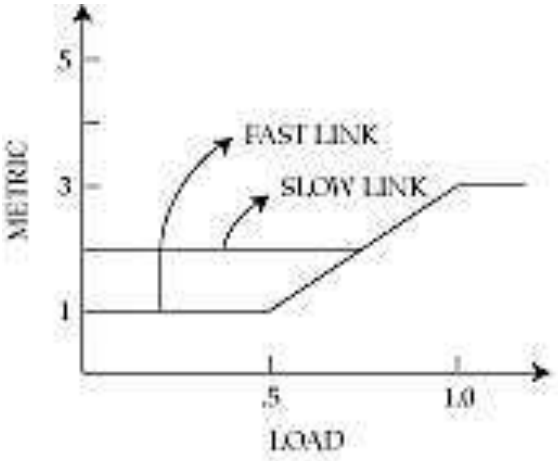


The shortest-paths spanning tree rooted at A is called an SPF-tree

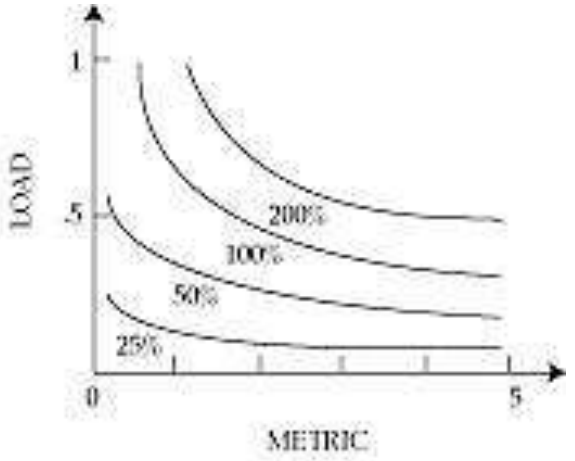
Misc: How to assign the Cost Metric?

- **Choice of link cost defines traffic load**
  - Low cost = high probability link belongs to SPT and will attract traffic
- **Tradeoff: convergence vs load distribution**
  - Avoid oscillations
  - Achieve good network utilization
- **Static metrics (weighted hop count)**
  - Does not take traffic load (demand) into account.
- **Dynamic metrics (cost based upon queue or delay etc)**
  - Highly oscillatory, very hard to dampen (DARPAnet experience)
- **Quasi-static metric:**
  - Reassign static metrics based upon overall network load (demand matrix), assumed to be quasi-stationary

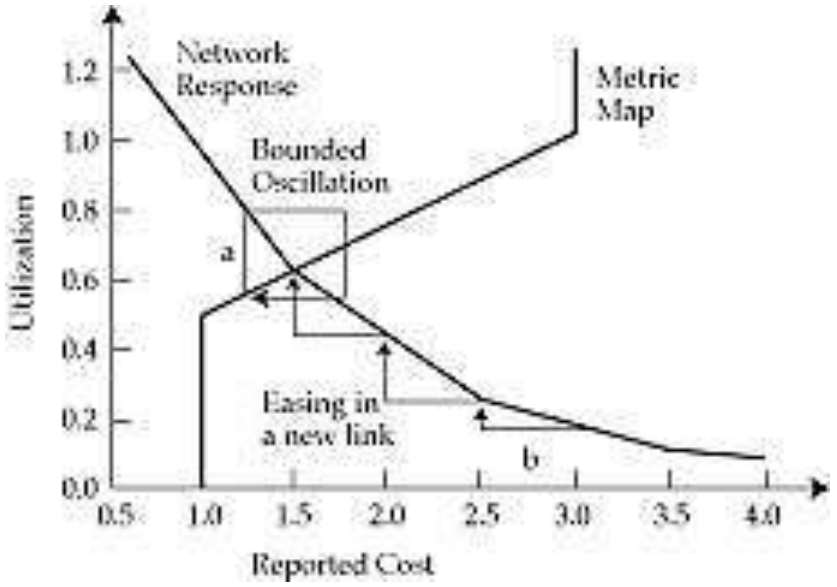
Routing dynamics



(a) METRIC MAP

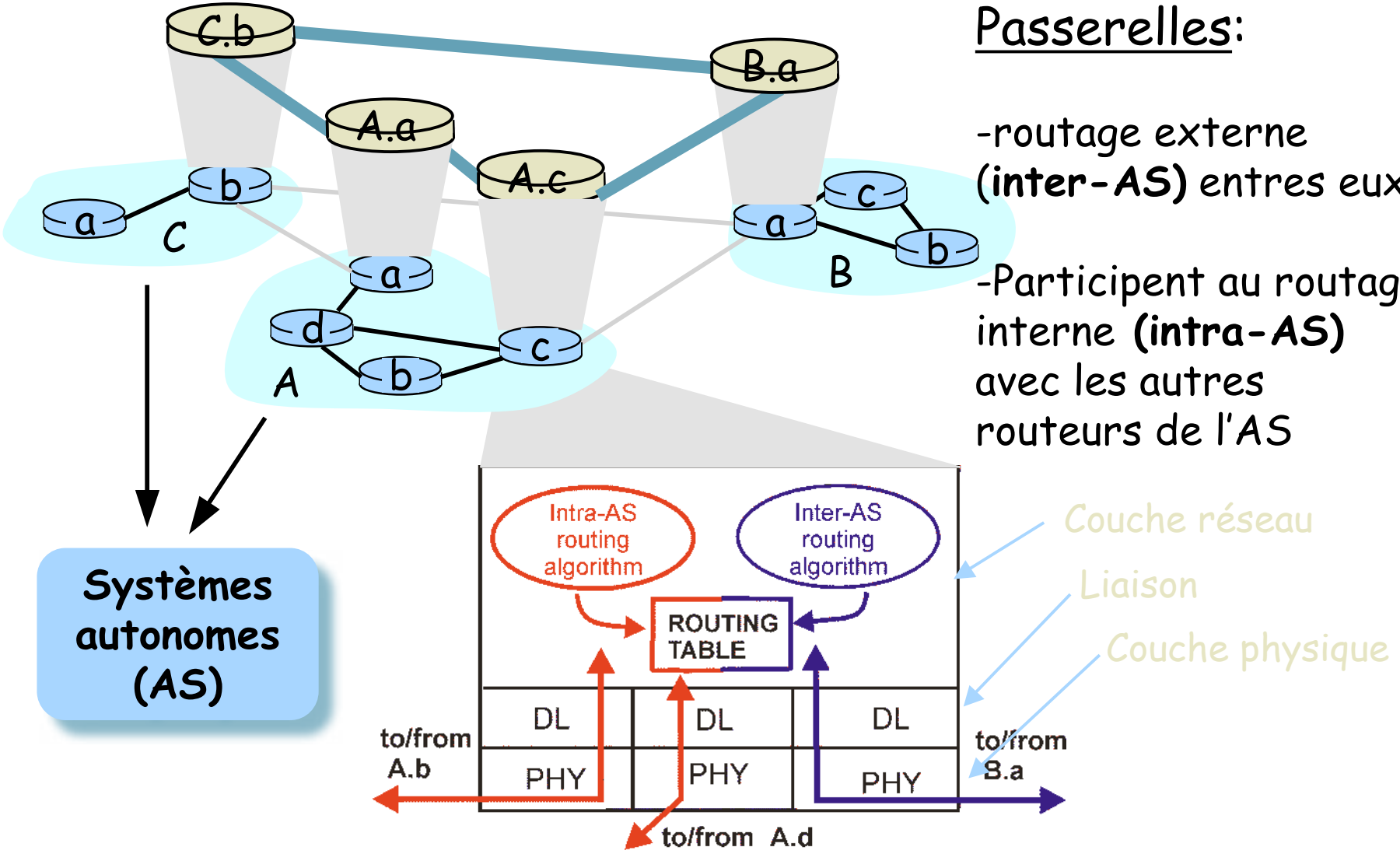


(b) NETWORK RESPONSE MAP



Added by C. Pham

Le vrai routage dans l'Internet

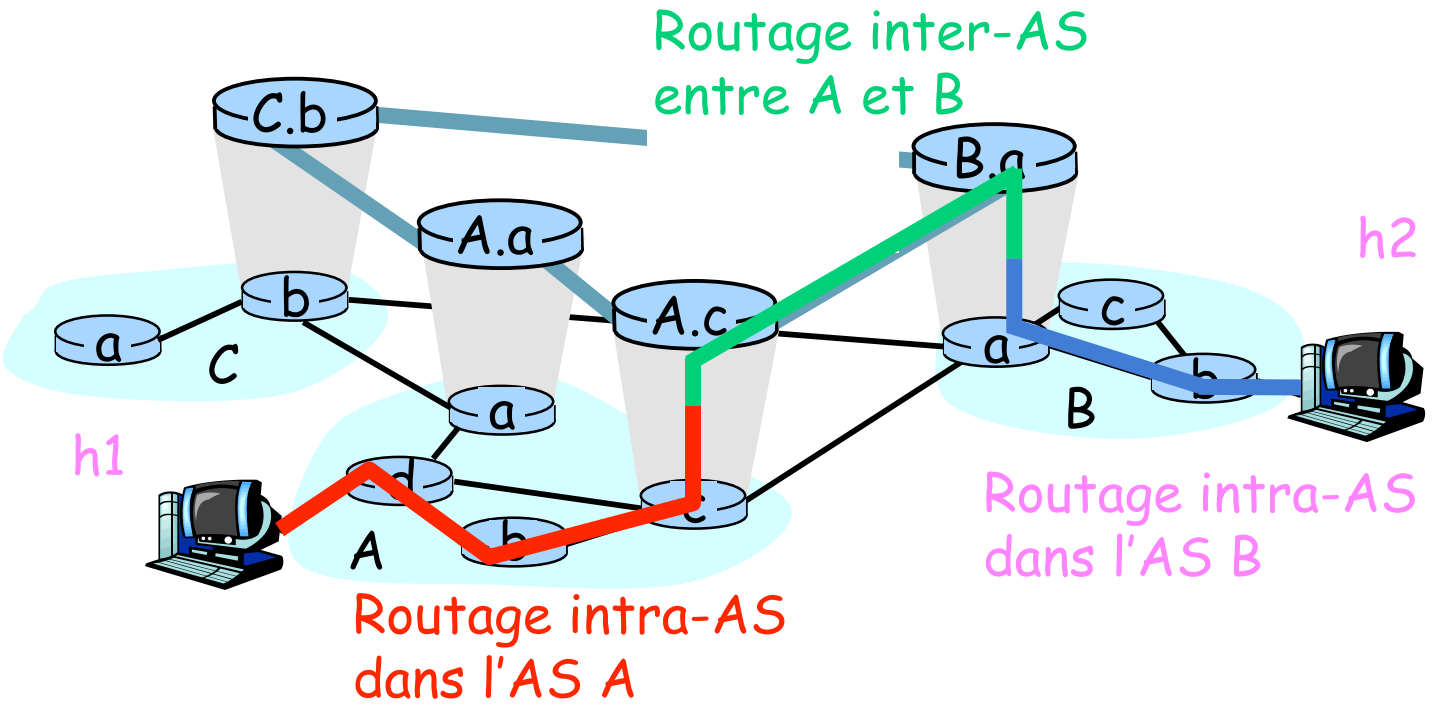


Passerelles:

-routage externe **(inter-AS)** entres eux

-Participent au routage interne **(intra-AS)** avec les autres routeurs de l'AS

Exemple de routage interne et externe

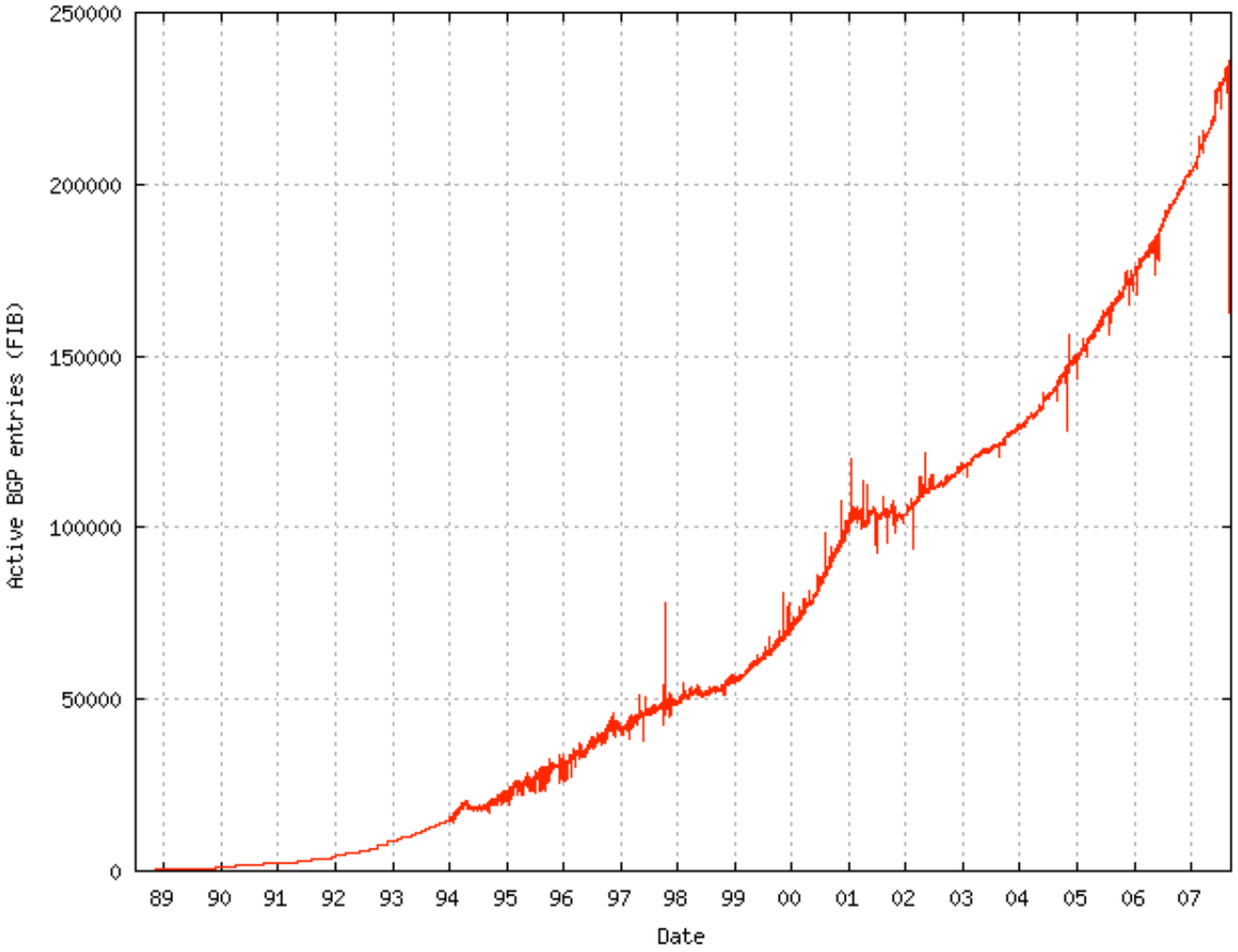


Seulement quelques routeurs (de 0 à 50) dans chaque AS

Les protocoles de routage (interne)

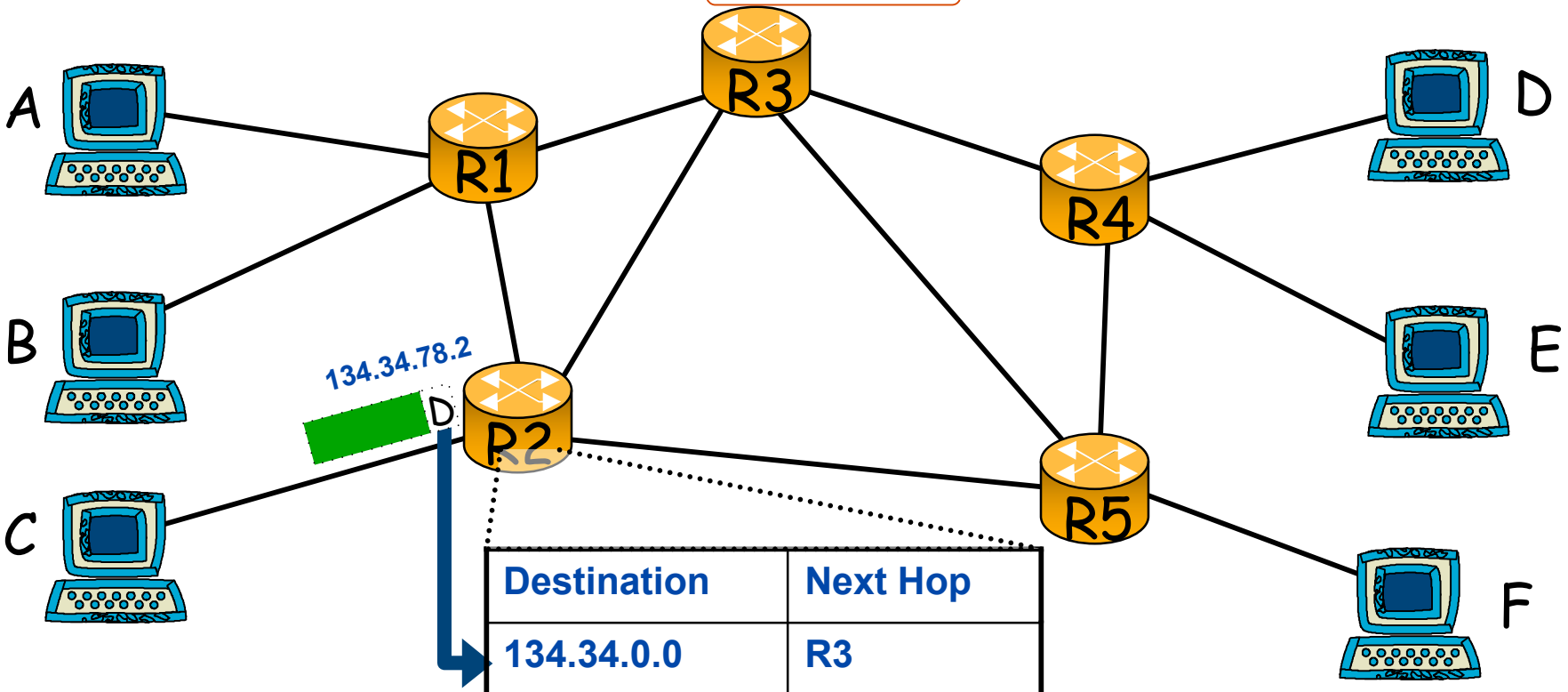
- **RIP (v1 et v2)**
  - Routing Information Protocol, v2 supporte le VLSM
  - Nombre de saut comme métrique
  - Nombre de saut maximum = 15
  - Mise à jour des tables de routage toutes les 30s
- **IGRP**
  - Interior Gateway Routing Protocol (Cisco)
  - Bande passante et délai comme métrique
  - Mise à jour des tables de routage toutes les 30s
- **OSPF**
  - Open Shortest Path First, supporte le VLSM
  - Notion de zones administratives
  - Utilise SPF (Dijkstra) pour calculer le plus court chemin
  - Le coût d'un lien dépend de la capacité (10^8/capacité)
  - Paquet HELLO toutes les 10s ou 30s
- **EIGRP**
  - Enhanced IGRP (Cisco), supporte le VLSM
  - Utilise l'équilibrage
  - Utilise DUAL (Diffused Update Algorithm) pour calculer le + court chemin

Evolution du nombre d'entrée dans un routeur inter-domaine BGP



Ce qu'il y a vraiment dans les tables

134.34.78.2 : 10000010.00010010.01001110.00000000
134.34.0.0 : 10000010.00100010.00000000.00000000
67.0.3.0 : 01000011.00000000.00000011.00000000
134.12.0.0 : 10000010.00001100.00000000.00000000



Destination	Next Hop
134.34.0.0	R3
67.0.3.0	R3
134.12.0.0	R5

Exemple du lookup

Prefix	lien de sortie
01*	K
10011*	M
110*	C
1*	D
...	

- @: 11001010000, lien de sortie: C
- @: 10101010000, lien de sortie: D

Contraintes de performance

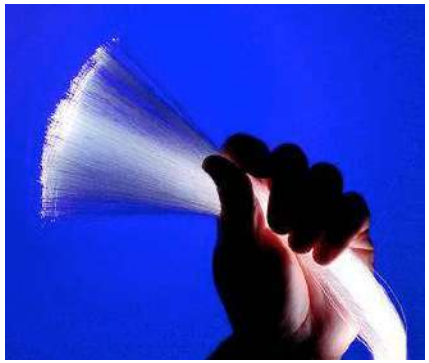
- **Avec les débits actuellement rencontrés, un routeur doit effectuer des millions d'opérations à la seconde**

Année	Débit ligne	40B (Mpps)	84B (Mpps)	354B (Mpps)
1997-98	0.155	0.48	0.23	0.054
1998-99	0.622	1.94	0.92	0.22
1999-00	2.5	7.81	3.72	0.88
2000-01	10.0	31.25	14.88	3.53
2002-03	40.0	125	59.52	14.12
GEthernet	1.0	3.13	1.49	0.35

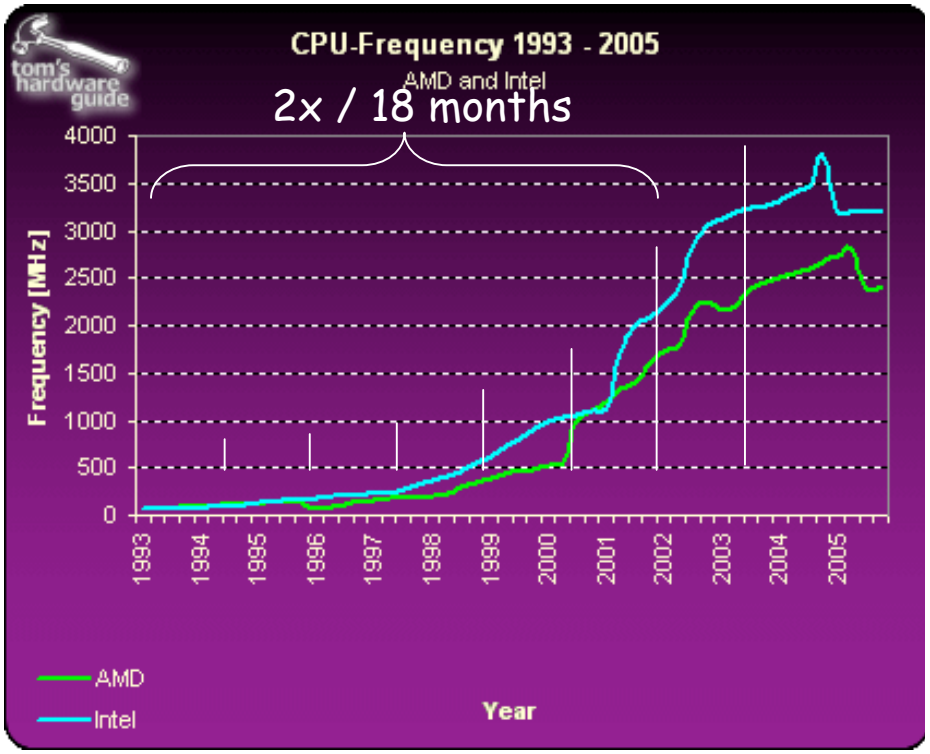
Distribution de la taille des paquets et difficultés du lookup

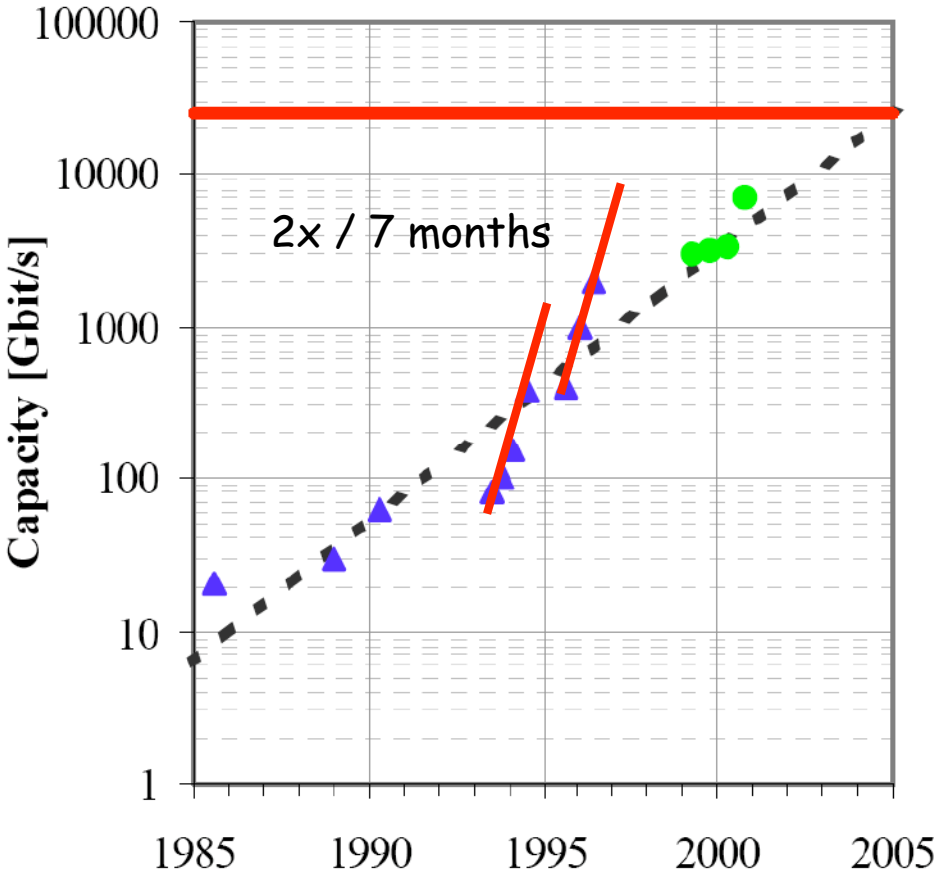
- **Sur un lien de réseau dorsale (backbone)**
  - 75% des paquets sont plus petits que 552 octets,
  - environ 50% des paquets sont plus petits que 44 octets (paquets d'acquittements),
  - 10% des paquets sont plus grands que 1500 octets.

- **Difficultés du lookup**
  - les tables de routage peuvent avoir des milliers d'entrées,
  - le prefix des adresses de destinations sont de longueurs variables, par exemple: 100101* ou 1* ou 10101100001000100010101O,
  - l'adresse de destination peut correspondre à plusieurs prefix, il faut prendre la plus longue.



La révolution optique!

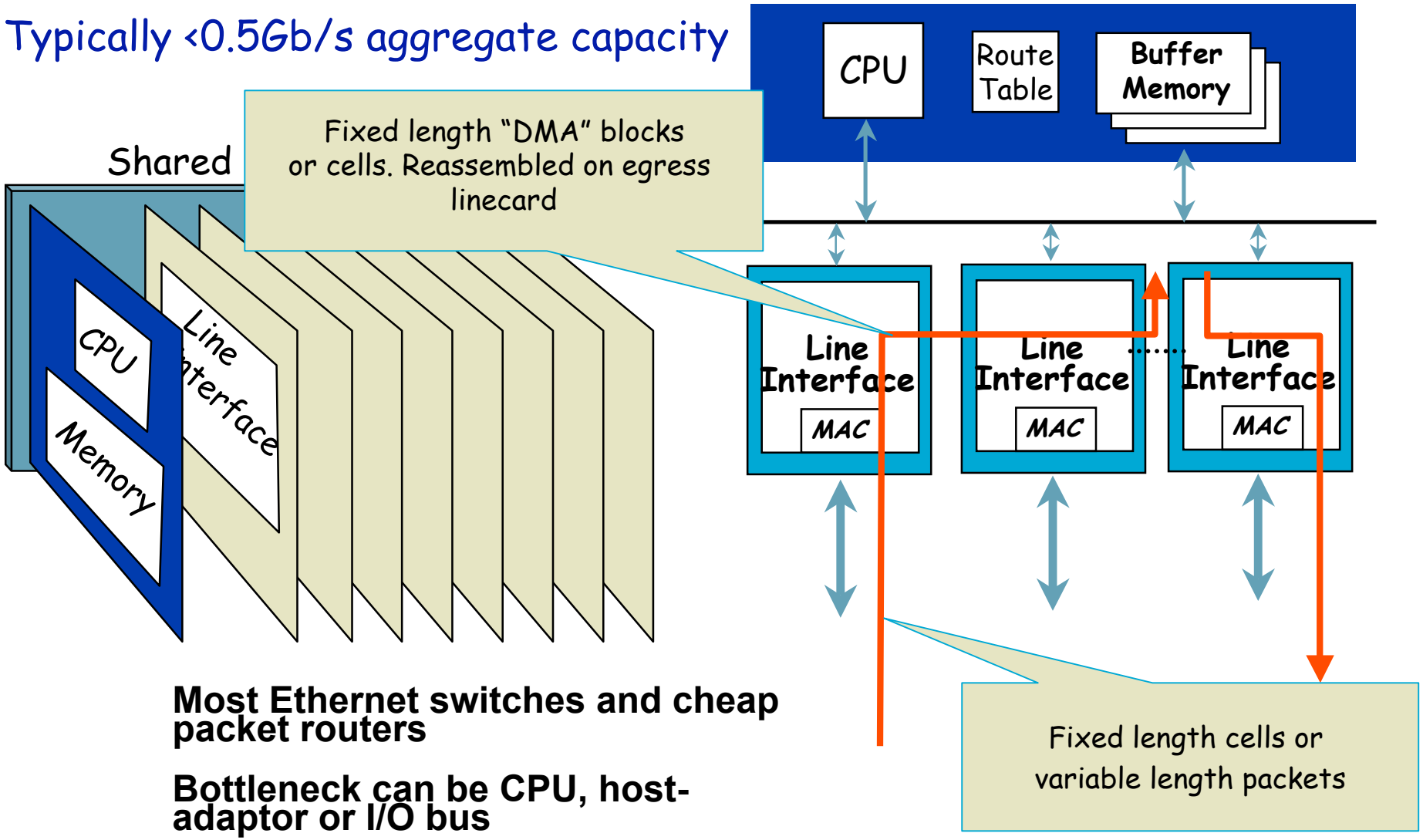




Source « Optical fibers for Ultra-Large Capacity Transmission » by J. Grochocinski

Routeurs de 1ère génération

Typically <0.5Gb/s aggregate capacity



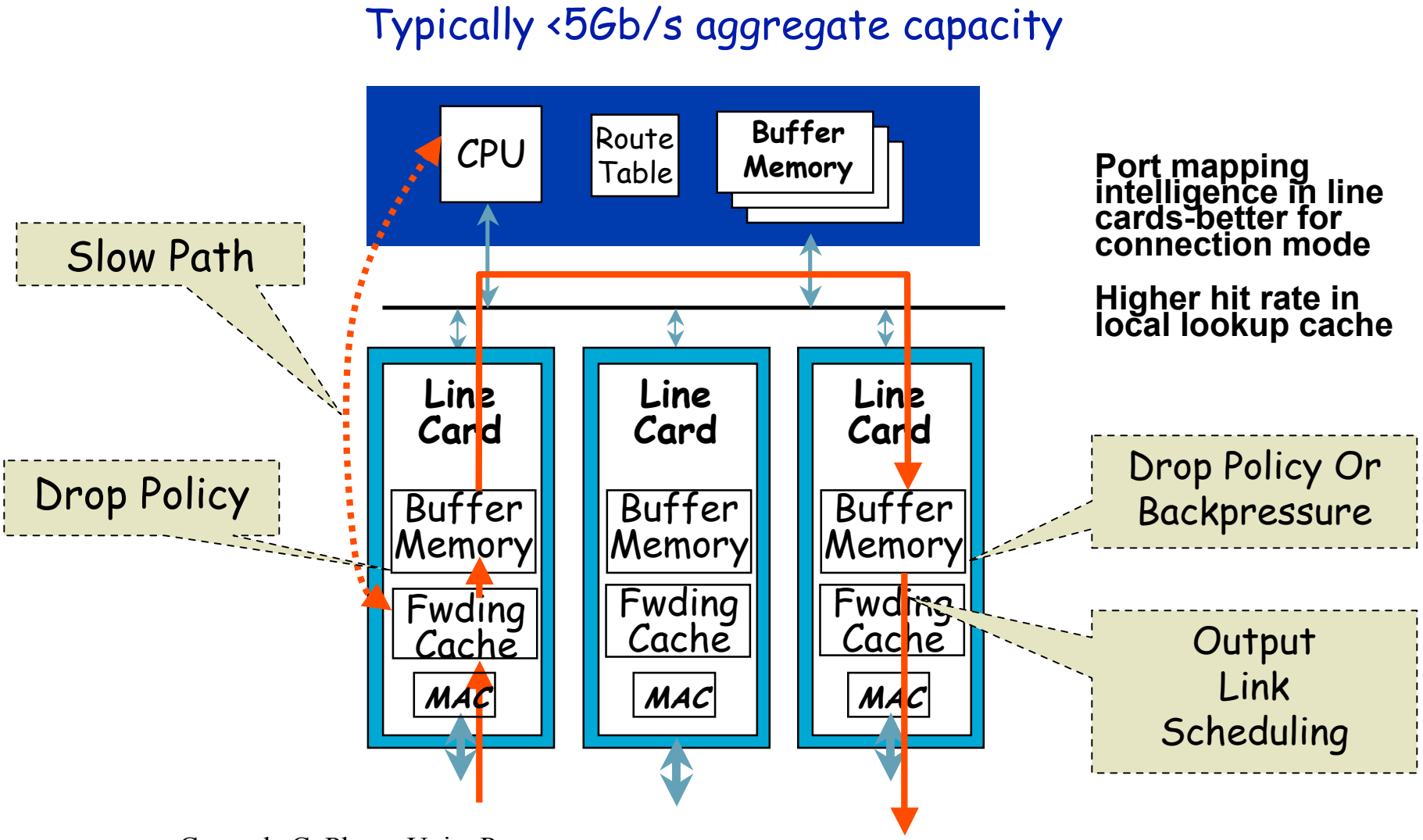
Most Ethernet switches and cheap packet routers

Bottleneck can be CPU, host-adaptor or I/O bus

Limitations

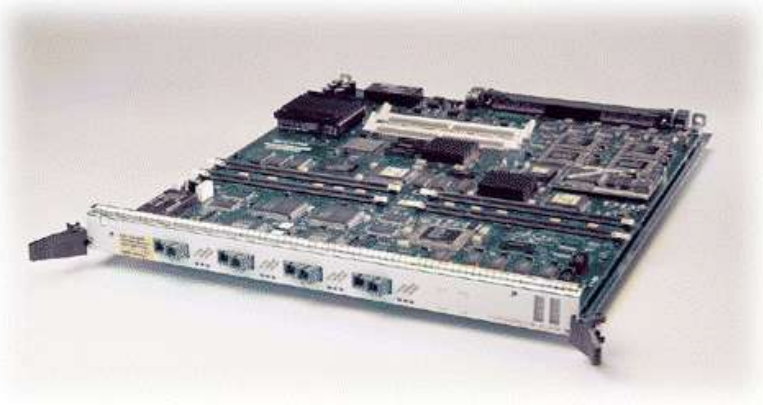
- **First generation router built with 133 MHz Pentium**
  - Mean packet size 500 bytes
  - Interrupt takes 10 microseconds, word access take 50 ns
  - Per-packet processing time is 200 instructions = 1.504 µs
- **Copy loop**
  - register <- memory[read_ptr]
  - memory [write_ptr] <- register
  - read_ptr <- read_ptr + 4
  - write_ptr <- write_ptr + 4
  - counter <- counter -1
  - if (counter not 0) branch to top of loop
- **4 instructions + 2 memory accesses = 130.08 ns**
- **Copying packet takes 500/4 *130.08=16.26µs; interrupt 10 µs**
- **Total time = 27.764 µs => speed is 144.1 Mbps**
- **Amortized interrupt cost balanced by routing protocol cost**

Routeurs de 2ème génération



Les équipements

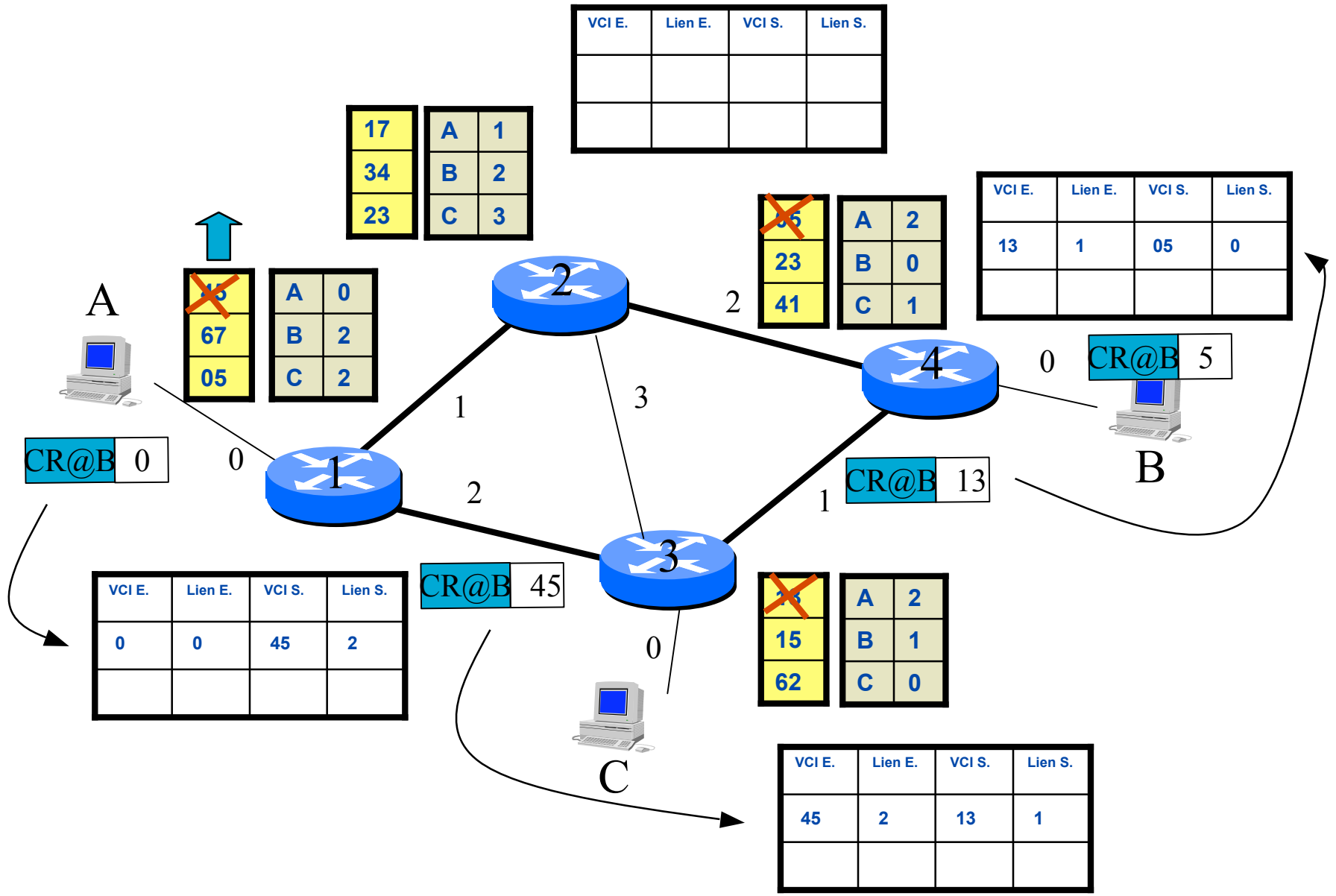
- gamme Cisco 12000, routeurs d'Internet
- Cisco OC-3/STM-1 Carte d'interface Packet-over-SONET/SDH
- Nortel Networks: gamme OPTera Metro 3000



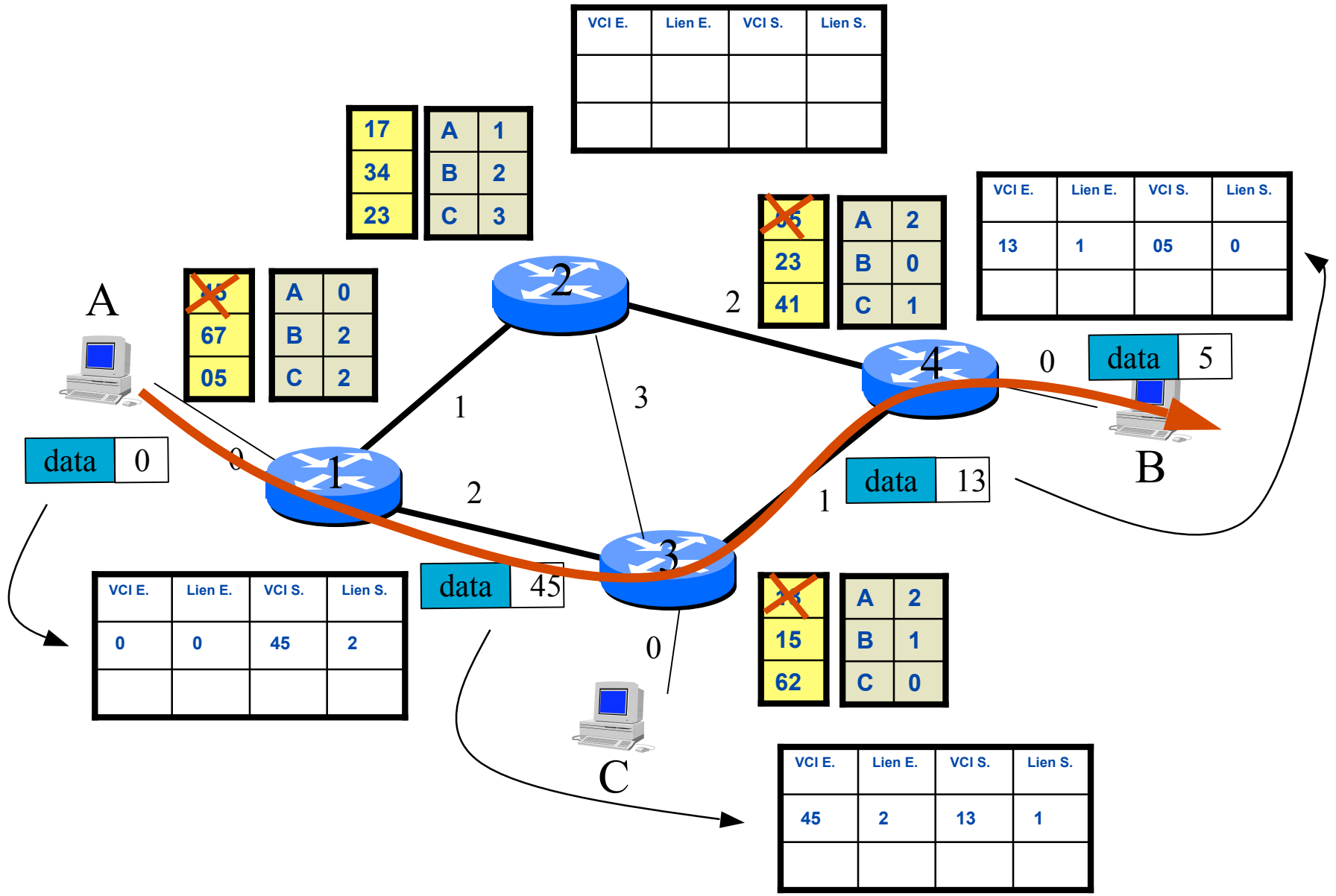
Le routage à circuit virtuel

- **Le routage IP est dit non-connecté ou mode datagramme**
  - Les paquets sont indépendants les uns des autres
  - Selon la table de routage, le chemin peut changer d'un paquet à un autre
- **Le routage à circuit-virtuel crée un chemin valable pour la durée de la connexion: routage semi-dynamique**
- **Initialement utilisé dans X.25 pour réduire le coût du routage:**
  - Création d'un chemin à l'ouverture de connexion
  - tous les paquets d'une même connexion emprunte le même chemin (circuit virtuel)
- **Principe**
  - Fonctionne toujours en mode connecté
  - Une demande d'ouverture de connexion crée un chemin qui sera identifié avec des labels (étiquettes)
  - Dans chaque routeur sur le chemin, des tables vont conserver les différents chemins (circuits virtuels) ouverts

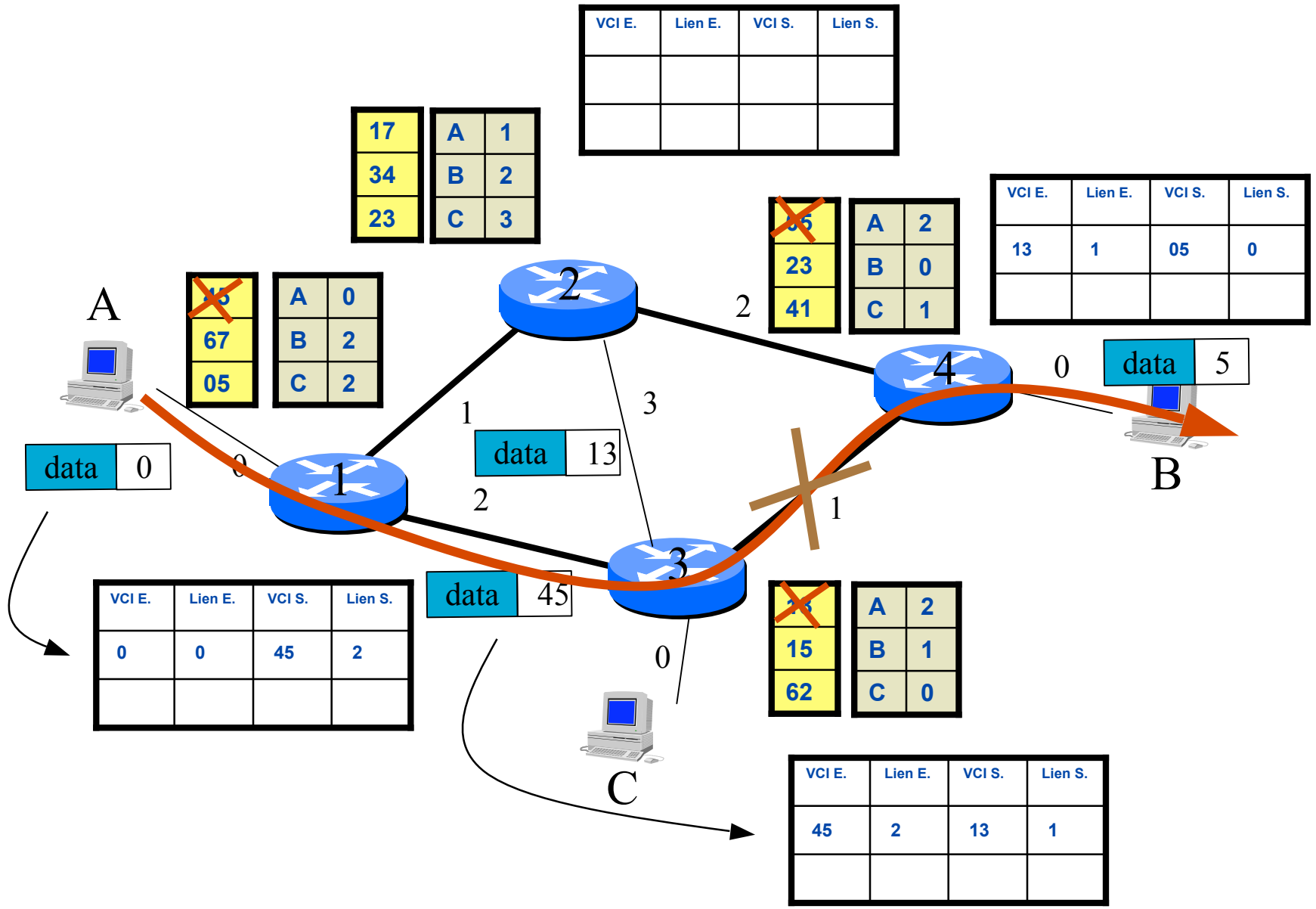
Etablissement d'un circuit virtuel (1)



Etablissement d'un circuit virtuel (2)



Panne de liens



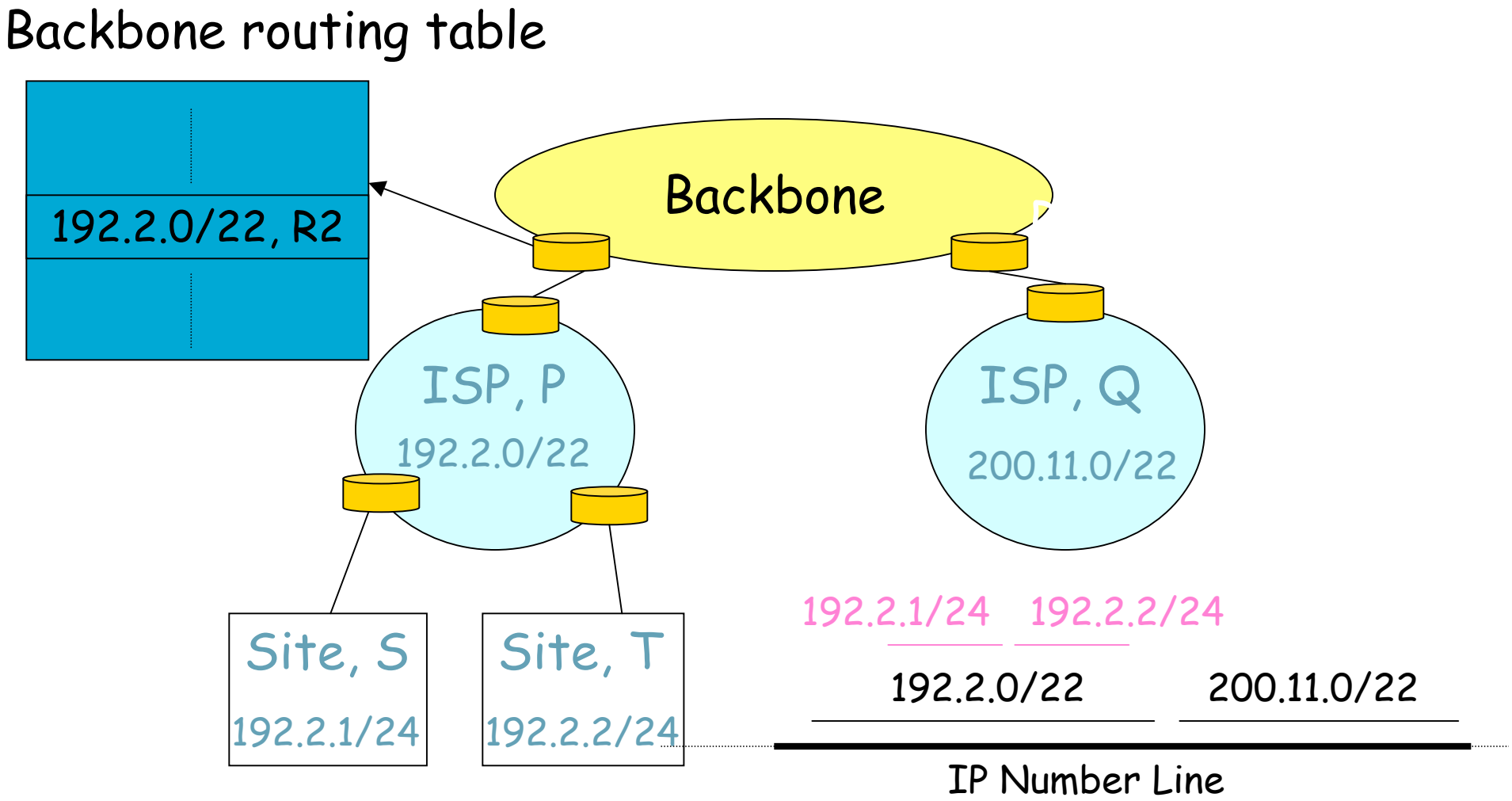
CIDR/VLSM

- **Classless Inter Domain Routing (rfc 1466)**
  - adressage sans notion de classe
  - attribution d'adresses de classes C consécutives
  - il suffit de connaître l'adresse de début et l'adresse de fin

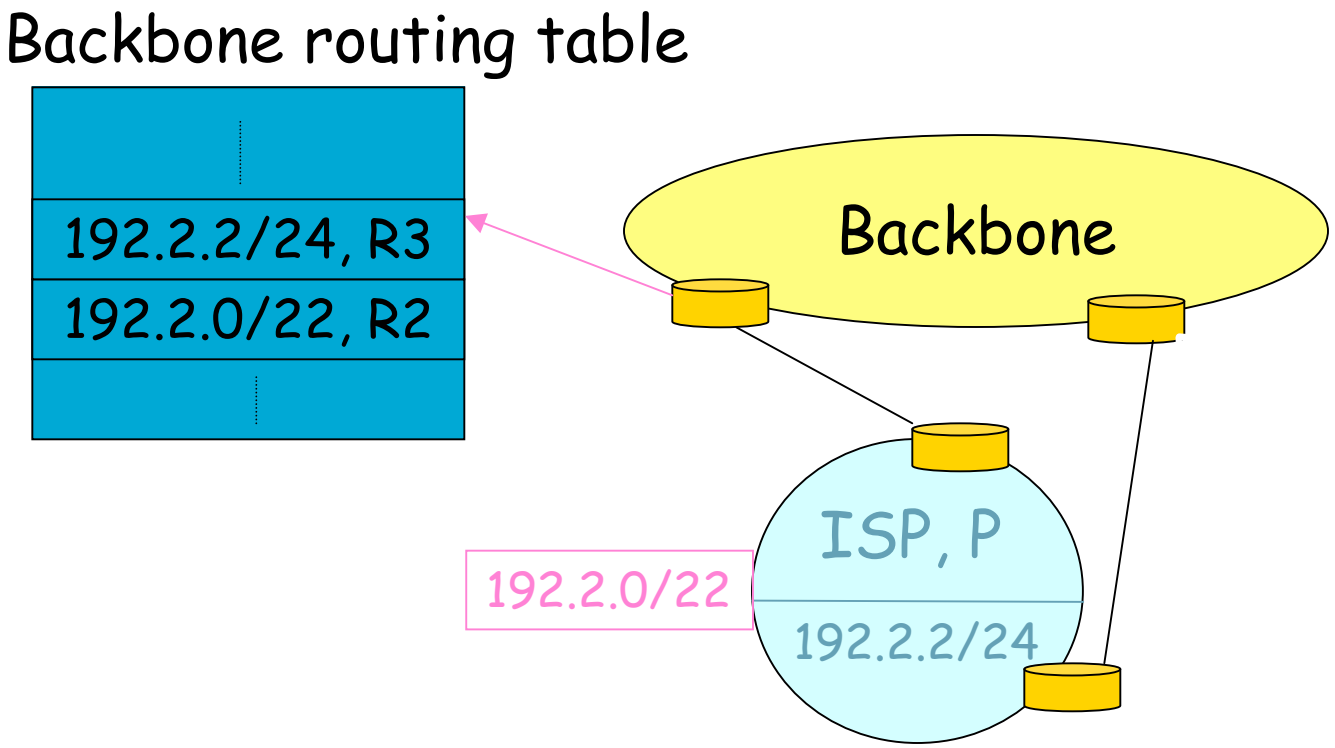
société	nb d'adresses	nb de classe C	adresse de début	adresse de fin	adresse de début	netmask	netmask en binaire
A	< 2048	8	192.24.0	192.24.7	192.24.0	255.255.248.0	255.255. 1111 1000 .0
B	< 4096	16	192.24.16	192.24.31	192.24.16	255.255.240.0	255.255. 1111 0000 .0
C	< 1024	4	192.24.8	192.24.11	192.24.8	255.255.252.0	255.255. 1111 1100 .0
D	< 1024	4	192.24.12	192.24.15	192.24.12	255.255.252.0	255.255. 1111 1100 .0
E	< 512	2	192.24.32	192.24.33	192.24.32	255.255.254.0	255.255. 1111 1110 .0
F	< 512	2	192.24.34	192.24.35	192.24.34	255.255.254.0	255.255. 1111 1110 .0

source L. Toutain

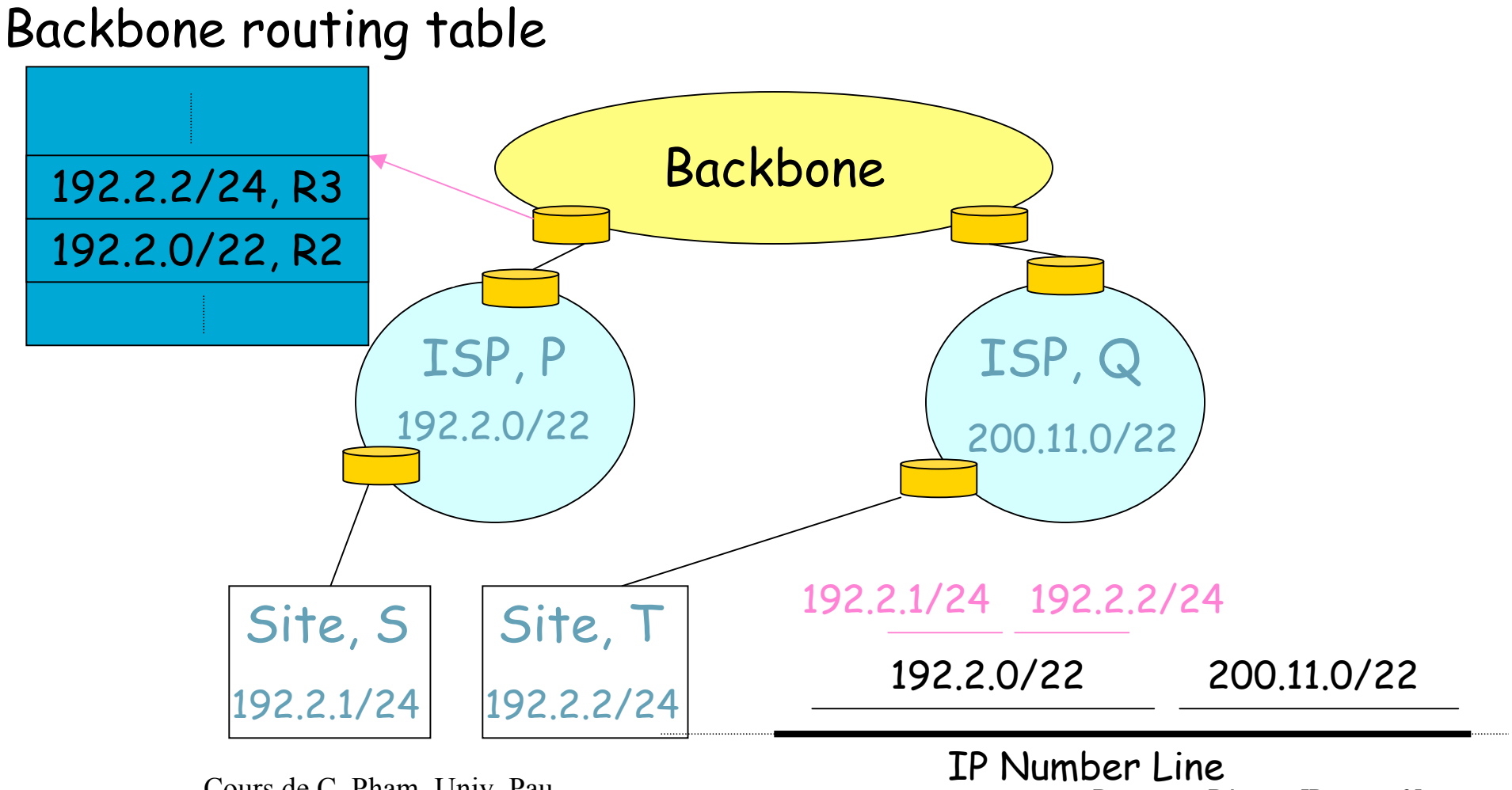
Ex: Agrégation avec CIDR/VLSM



Prefixes non-agrégeable
(1) Multi-homed Networks



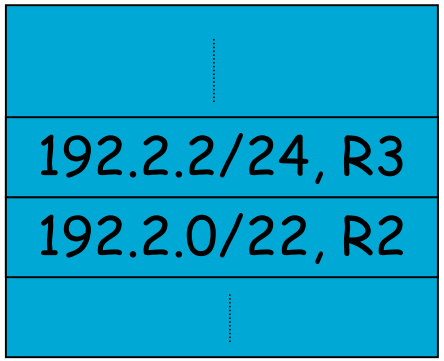
Prefixes non-agrégeable
(2) Changement de provider



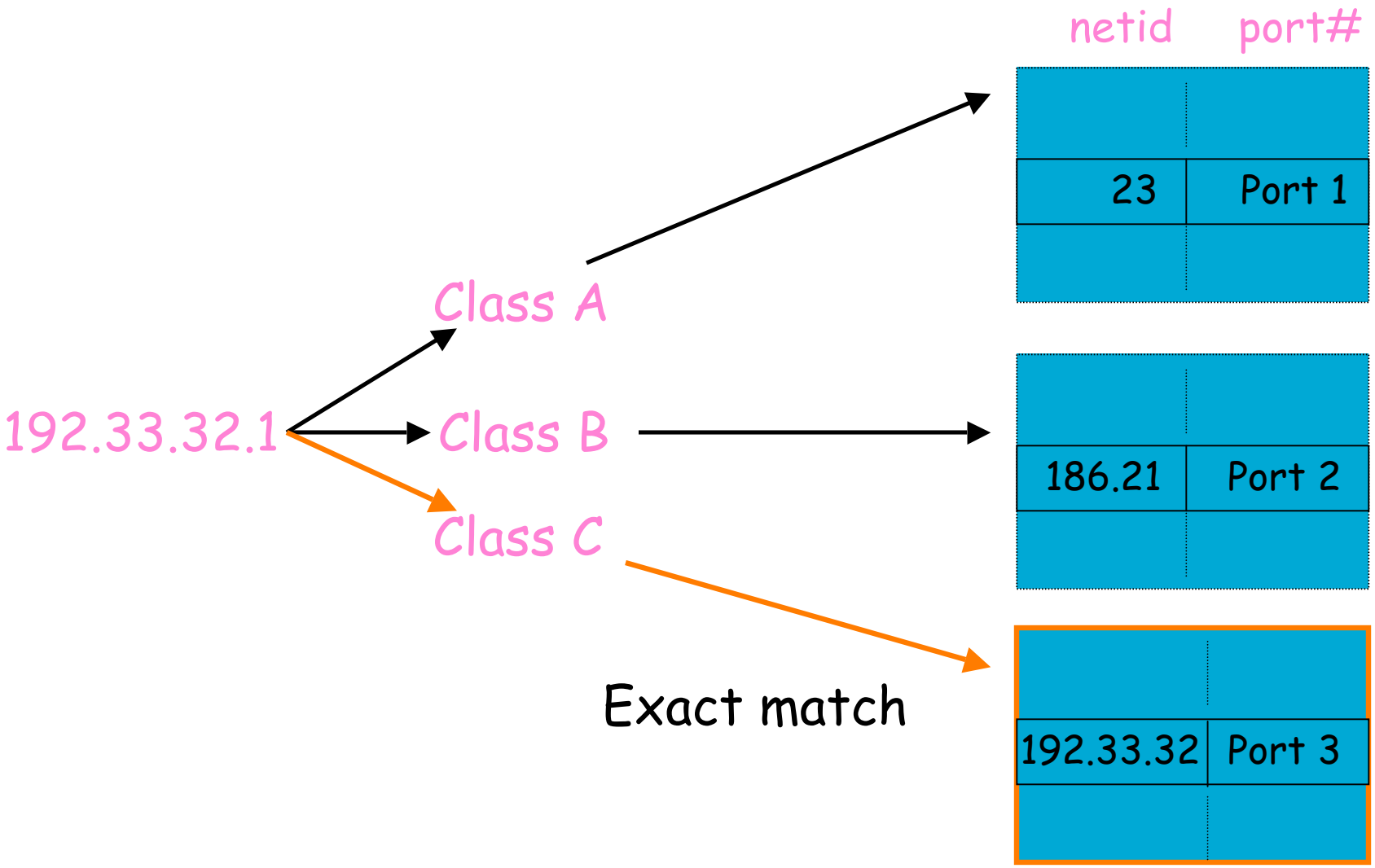
La phase de lookup

- **Lookup**
  - le lien de sortie dépend de l'adresse de destination et du contenude la table de routage,
  - la recherche se fait avec Best Matching Prefix (BMP) dans les tables de routage,
  - la rapidité est primordiale.

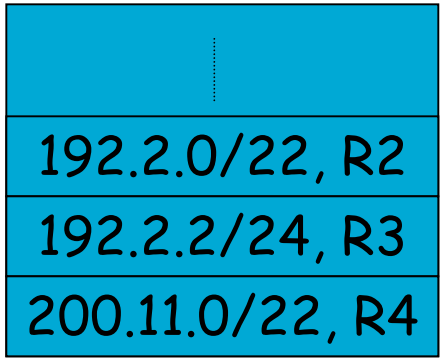
- **Best Matching Prefix**
  - table de routage = paires d'entrées (Prefix, Lien),
  - pour une \@IP donné, le lien de sortie avec le plus long prefix est choisi.

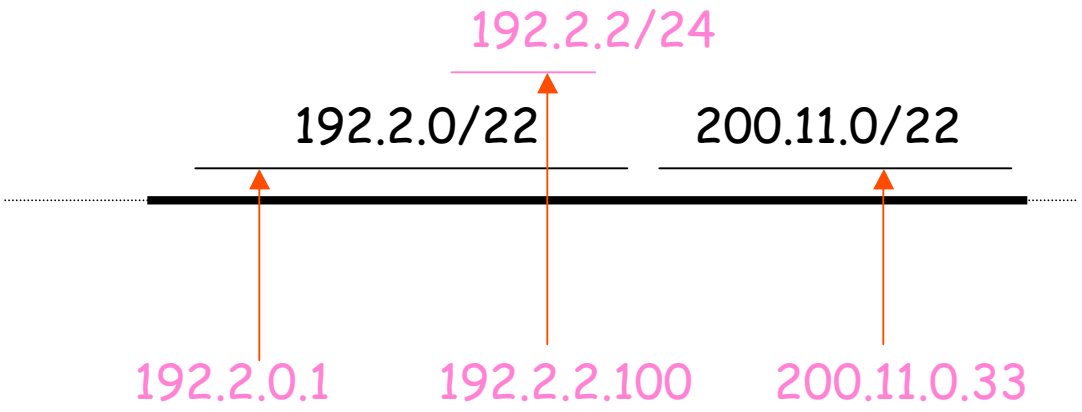


Lookups avec distinction de classes



Lookups avec CIDR/VLSM





Find the most specific route, or the longest matching prefix among all the prefixes matching the destination address of an incoming packet