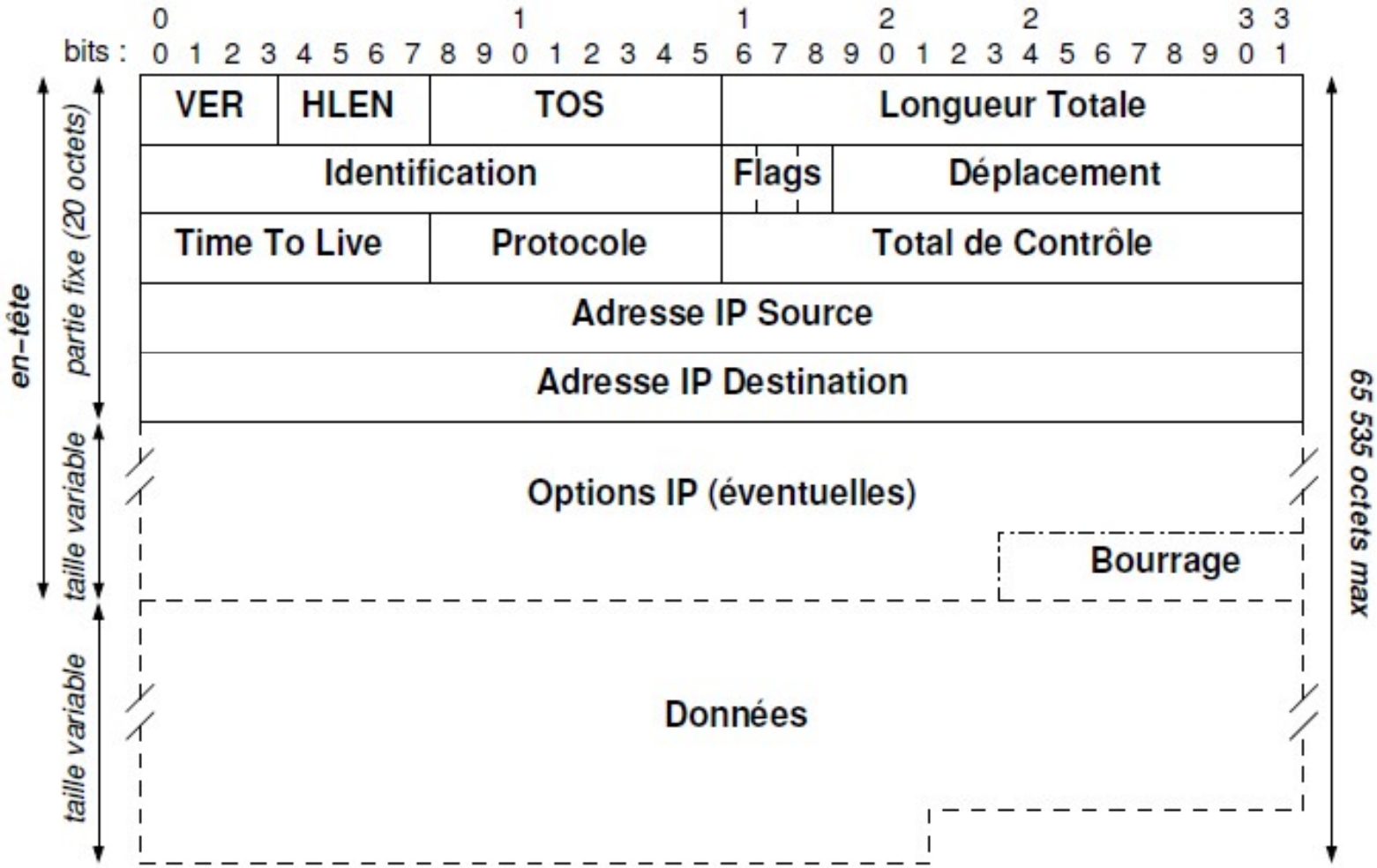
Chapitre 2 :

Routage & Adressage IP

Le protocole IPv4

- Le protocole IP (*Internet Protocol, RFC 791)* est un protocole de niveau réseau qui a pour objet l'interconnexion de réseaux hétérogènes
  - masquer aux applications les réseaux physiques traversés
- Il assure sans connexion un service non fiable de délivrance de datagrammes IP
  - perte, duplication, desequencement possible des paquets
- IP n'entretient aucune variable d'état, il n'effectue que les tâches élémentaires
  - l'adaptation de la taille des unités de données aux capacités d'emport du réseau physique traversé (MTU, *Maximum Transfert Unit)* : fragmentation
  - la désignation des machines d'extrémité : adressage IP
  - l'acheminement dans le réseau logique : routage

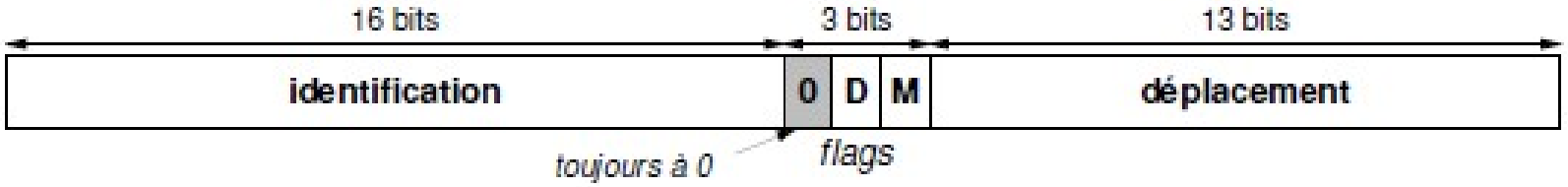
Format du datagramme IP



Champs du datagramme IP

- VER : Version du protocole IP qui doit interpréter ce datagramme(4 bits).
  - La version actuelle la plus “déployée” est 4
  - dans le datagramme IPv6, ce champ est maintenu et vaut 6
- HLEN : (ou IHL *: Internet Header Length*)(4 bits). Cette valeur est à multiplier par 4 pour connaître le nombre d’octets constituant l’en-tête
  - nécessaire car le champ option est de longueur variable
  - Peut varier de 5 (soit 20 octets d’entête) à 15 (60 octets d’entête).
- TOS : *Type Of Service* (8 bits). spécifie le type d’acheminement attendu
- Longueur Totale : (16 bits) Donne la taille totale en octets du datagramme (ou du fragment). Ainsi, un datagramme ne peut pas excéder 65535 octets

Champs du datagramme IP (fragmentation)



- Identification : valeur identifiant le datagramme d'origine
  - Les fragments d'un datagramme ont les mêmes identification, IP Source, IP Destination et Protocole
- bit D(on't Fragment) : le datagramme ne doit pas être fragmenté (détruit et message ICMP si impossible)
- bit M(ore) : a 0 si datagramme non fragmenté ou si c'est le dernier fragment ; sinon vaut 1
- Déplacement (Offset) : déplacement X 8 est la position absolue (ou numéro) du premier octet de données de ce datagramme dans le datagramme d'origine.
  - Vaut 0 si pas de fragmentation ou dans le premier fragment
  - Tous les fragments, sauf le dernier, doivent avoir une longueur multiple de huit

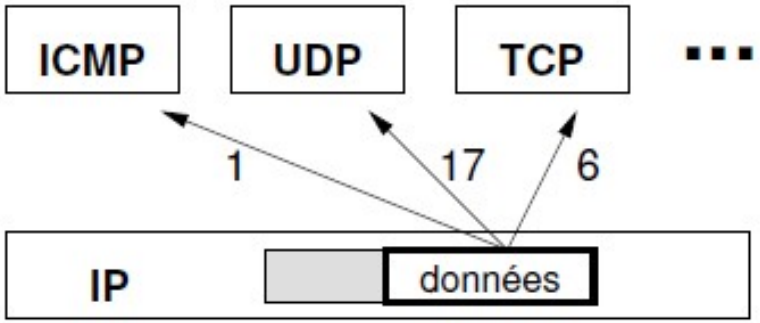
Datagramme IP : Champ TTL

- Time To Live (TTL): indiqué par l'émetteur pour limiter :
  - la "durée de vie" du datagramme en secondes (plus vraiment d'actualité)
  - le nombre de routeurs traverses par le datagramme (nombre de sauts)
- Valeur maximum fixé par l'émetteur (255, 128, 64...)
- décrémenté par routeurs et stations traitant le datagramme :
  - de 1 à chaque traversée d'un routeur (saut)
  - du temps passé en file d'attente
- Si atteint 0, le datagramme est détruit, et l'émetteur est informé par un message ICMP
- Empêcher un datagramme de circuler indéfiniment

Datagramme IP : Champ Protocole

- Protocole : indique à IP l'origine du champ données (protocole transporté)
  - démultiplexage vers les protocoles de la couche supérieure
- Quelques exemples de valeurs du champ protocole :

Valeur	Acronyme	Protocole
1	ICMP	*Internet Control Message Protocol*
2	IGMP	*Internet Group Message Protocol*
6	TCP	*Transmission Control Protocol*
8	EGP	*External Gateway Protocol*
17	UDP	*User Datagram Protocol*
89	OSPF	*Open Short Path First*



Champs du datagramme IP

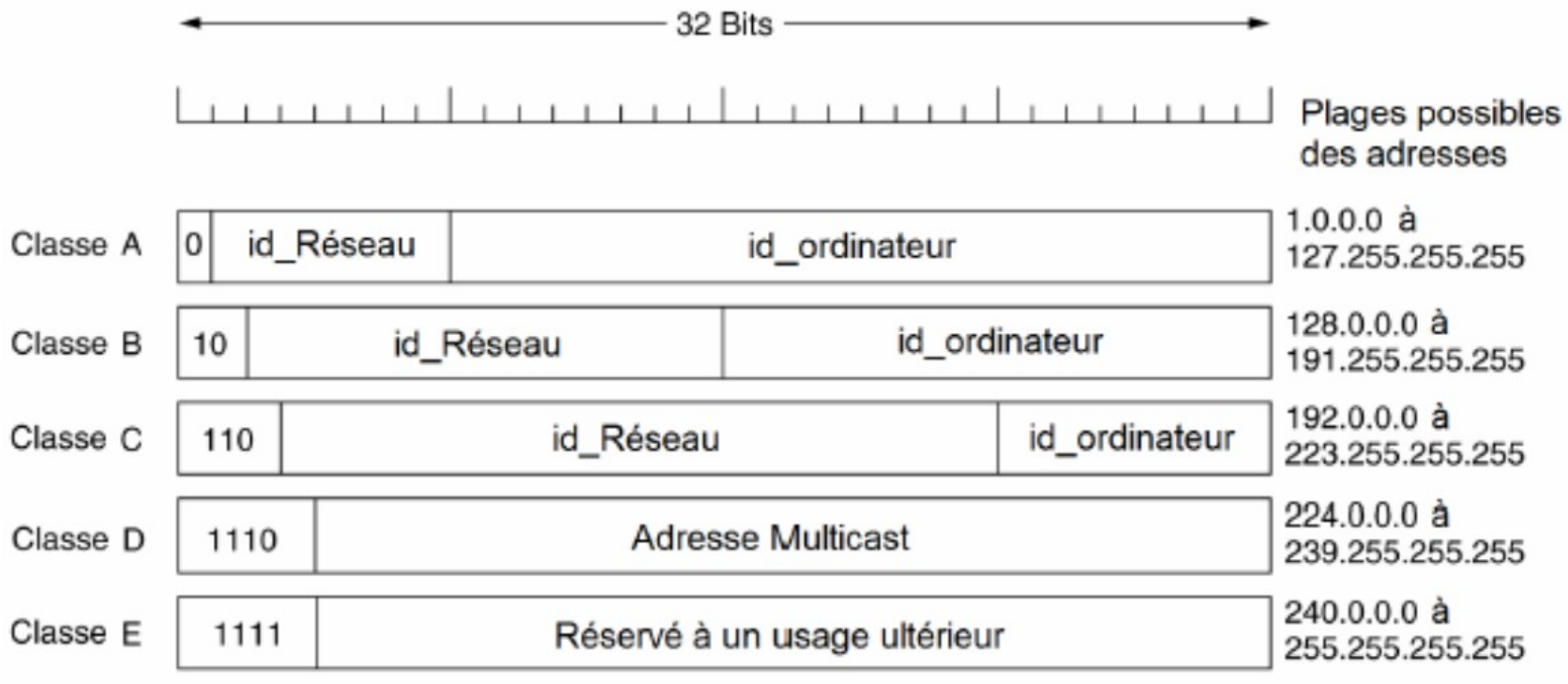
- Total de Contrôle (checksum) : est le complément à 1 de la somme en complément à 1 des données de l'en-tête découpées en mots de 16 bits
  - permet de contrôler l'intégrité de l'entête
  - recalculé par chaque système intermédiaire (modification du champ TTL et fragmentation éventuelle)
- Adresse IP Source : identifie l'hôte à l'origine du datagramme
  - permet de retourner un message à l'émetteur (ICMP, UDP…)
- Adresse IP Destination : identifie le destinataire final du datagramme

Datagramme IP : Champ Option

- Le champ option, de longueur variable, est codé : code option (type), longueur, valeur
  - analysé dans chaque routeur
    - il est peu voire pas utilisé car peu de machines sont capables de le gérer
  - peut-être suivi de bits de bourrage pour assurer l'alignement de l'en-tête IP sur des mots de 32 bits
  - en cas de fragmentation, certaines options sont copiées dans tous les datagrammes
  - parmi ces options, on trouve des options d'enregistrement de la route, d'estampille horaire, routage strict, etc...

Adresses IPv4

- Adresses universelles, codées sur 32 bits, indépendantes des adresses MAC
- L'espace d'adressage IP est découpé en 5 classes pour des besoins différents



Adresses IPv4

- Certaines combinaisons de bits pour l'id_ordinateur et/ou l'id_réseau ont un rôle particulier
  - les 32 bits à 0 (adresse 0.0.0.0) indique "cet" ordinateur : utilisée temporairement lorsqu'un hôte ne connaît pas encore son adresse
  - id_ordinateur tout à 0 est l'adresse du réseau
  - les 32 bits à 1 (adresse 255.255.255.255) est l'adresse de diffusion limitée (limited broadcast) représentant tous les hôtes du réseau présent. Un hôte peut l'utiliser (comme destination) pour envoyer un message a tous les hôtes (actifs) de son réseau
  - id_ordinateur tout à 1 est l'adresse de diffusion dirigée (directed broadcast) dans le réseau id_réseau
  - Id_réseau tout à 1 pour la classe A (adresses de la forme 127.x.y.z) : cela correspond aux adresses de rebouclage (loopback)

Adresses IPv4

- Les adresses des réseaux/hôtes appartiennent aux classes A, B ou C
- Adresses potentielles de réseaux par classe :

classe	adr. min	adr. max
A	1.0.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.0.0	223.255.255.0

- Pour un réseau donné, comme 139.124.0.0, les adresses des hôtes vont de 139.124.0.1 à 139.124.255.254. Elles sont attribuées librement par l'administrateur du réseau

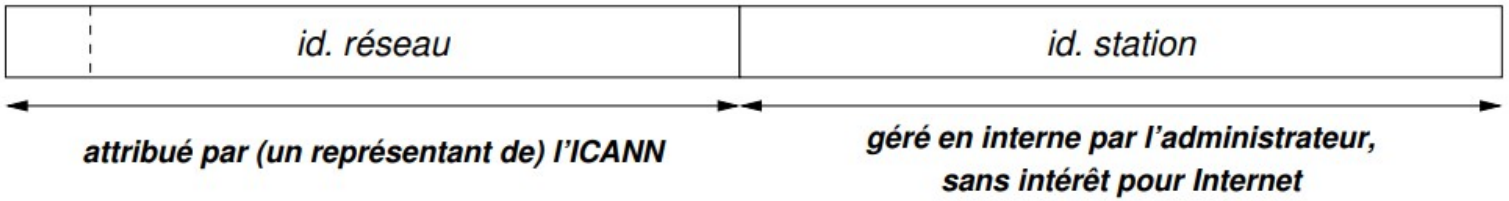
Sous-réseaux IP : motivation

- Dans la version d'origine d'IP, une adresse IP est constituée d'un id. réseau et d'un id. station dans ce réseau
- Le routage classique n'utilisait que l'id. réseau
  - à chaque réseau physique doit être affectée une adresse de réseau unique
- Or, la multiplication des réseaux pose plusieurs problèmes :
  - la gestion de toutes les adresses de réseau devient très lourde
  - les tables de routage deviennent gigantesques
  - le schéma d'adressage peut être saturé

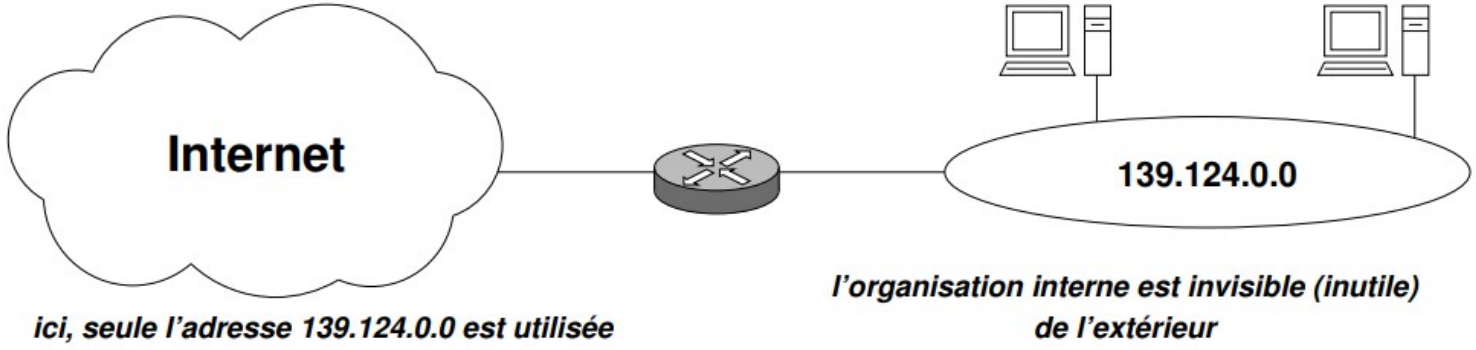
- Il est devenu nécessaire de réduire le nombre de réseaux à gérer, notamment en permettant à plusieurs réseaux physiques de partager la même adresse de réseau

Sous-réseaux IP : principe

- Pour connecter un réseau à Internet, un administrateur demande une adresse de réseau (classe A, B ou C)
- L'organisation interne du réseau est à la charge de l'administrateur :
  - plan d'adressage (affectation des adresses IP). En particulier, l'administrateur utilise comme il le souhaite la partie *id. station* :

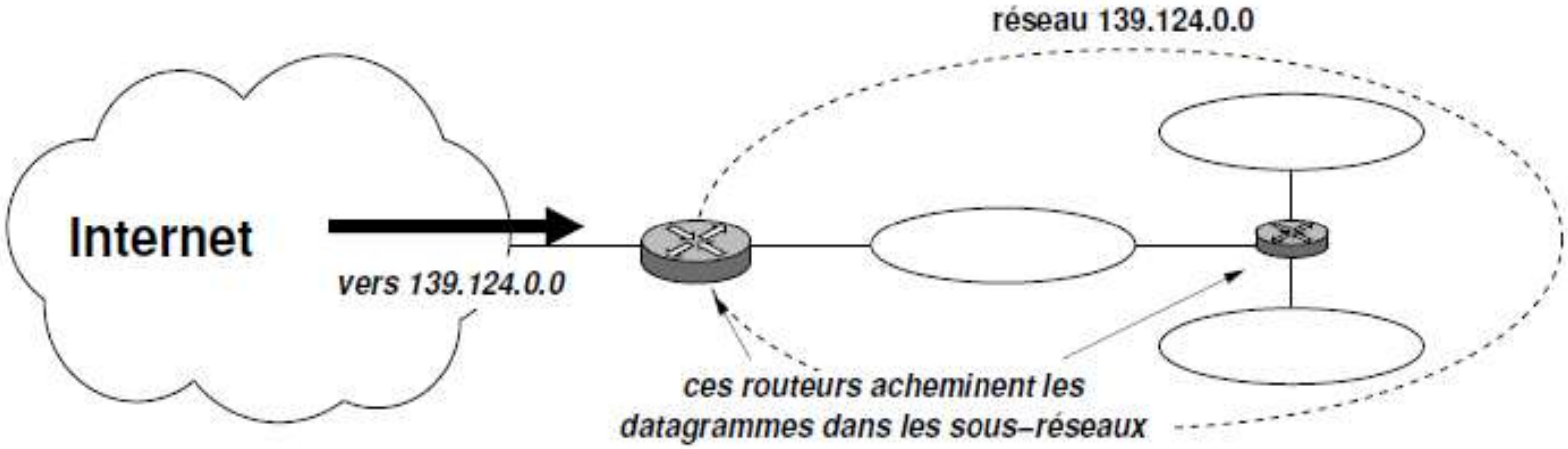


  - définition des routes sur les ordinateurs et routeurs de ce réseau
- Du point de vue d'Internet (des autres réseaux et routeurs), seule l'adresse du réseau est prise en compte



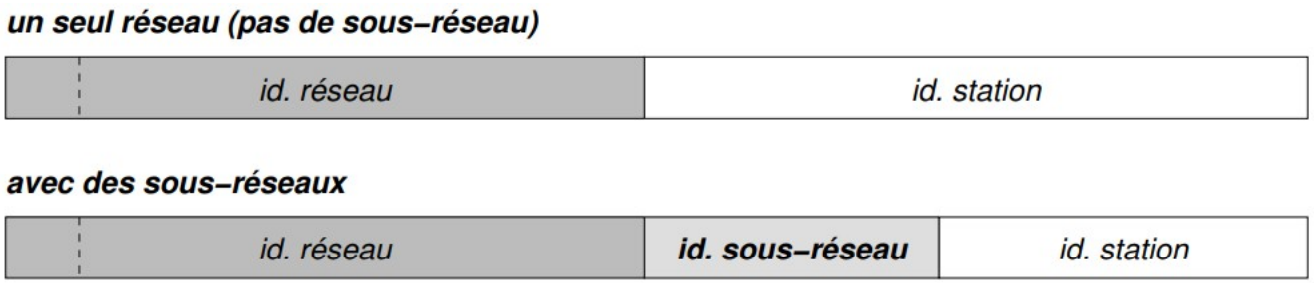
Sous-réseaux IP : principe

- L'administrateur dispose d'une adresse de réseau (telle que 139.124.0.0) mais de plusieurs réseaux physiques, appelés **sous-réseaux**
- La présence de plusieurs réseaux physiques est une question interne
- Les routeurs d'Internet se contentent d'acheminer les datagrammes vers "le réseau" 139.124.0.0
  - l'organisation interne est invisible (inutile) de l'extérieur
- À charge des routeurs internes d'acheminer les datagrammes à travers les sous-réseaux



Sous-réseaux IP : identifiant de sous-réseau

- Pour distinguer les sous-réseaux, l'administrateur réserve une partie de l'*id. station*, appelée l'**identifiant de sous-réseau** :

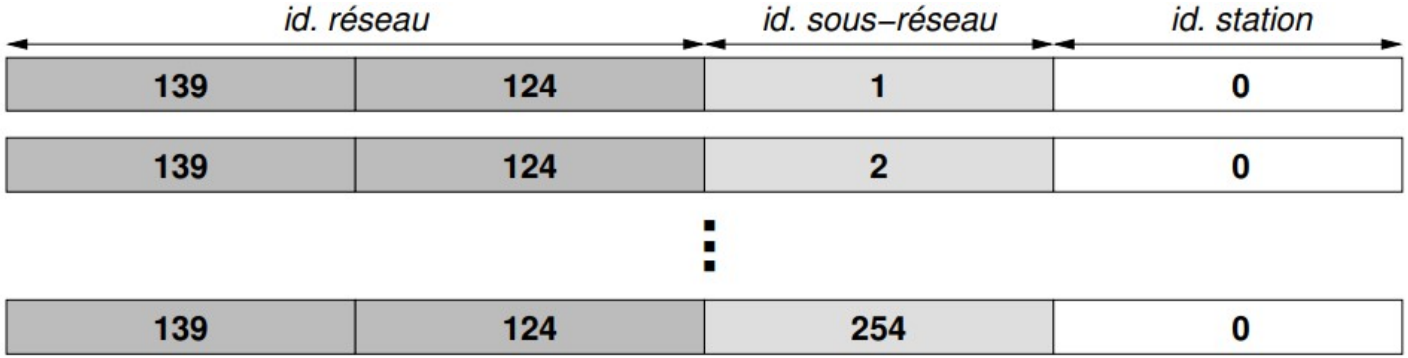


- Les stations et routeurs internes doivent en tenir compte pour leurs décisions de routage (accessibilité directe ou indirecte de la destination)
- La taille de l'*id. sous-réseau* dépend du nombre de sous-réseaux, en tenant compte des contraintes (recommandations) suivantes :
  - l'*id. sous-réseau* tout à 0 est réservé (conflit avec l'adresse du réseau)
  - l'*id. sous-réseau* tout à 1 est réservé (conflit avec l'adresse de diffusion dirigée)

  d'où : 2^n - 2 sous-réseaux avec *n* bits pour l'*id. sous-réseau*
- En pratique, ces recommandations ne sont pas toujours suivies car considérées obsolètes, mais nous les suivrons le plus souvent. . .

Sous-réseaux IP : identifiant de sous-réseau

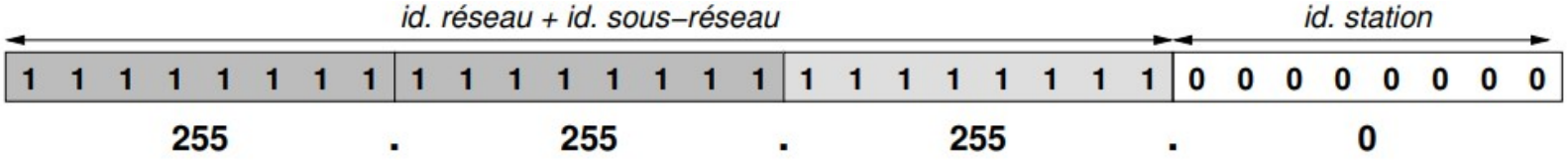
- Dans le réseau 139.124.0.0 :
  - si on utilise un octet pour l'*id. sous-réseau*, on peut avoir jusqu'à 254 sous-réseaux d'au plus 254 stations.
    - les sous-réseaux auront pour adresses : **139.124.1**.0, **139.124.2**.0, . . . jusqu'à **139.124.254**.0



  - si on n'utilise que 3 bits, on peut avoir jusqu'à 6 sous-réseaux d'au plus 8 190 stations.
    - les sous-réseaux auront pour adresses : **139.124.32**.0, **139.124.64**.0, **139.124.96**.0, **139.124.128**.0, **139.124.160**.0 et **139.124.192**.0

Sous-réseaux IP : masques de sous-réseaux

- En interne, pour identifier le sous-réseau, il faut prendre en compte la partie *id. sous-réseau*
- Or, le schéma d'adressage en classe ne permet que d'extraire la partie *id. réseau*
- D'où l'usage généralisé des **masques de sous-réseaux** :
  - ses bits à 1 indiquent où se trouvent les parties **id. réseau et id. sous-réseau**
  - ses bits à 0 indiquent où se trouve la partie *id. station*
- Exemple : pour le réseau 139.124.0.0 et l'*id. sous-réseau* sur un octet, le masque est 255.255.255.0 :



Sous-réseaux IP : masques de sous-réseaux

- Une adresse IP est toujours associée à un masque de sous-réseau codé aussi sur 32 bits
- Pour obtenir l'adresse du réseau, tous les bits de l'id_ordinateur sont mises à 0 :
  - adresse du réseau = (Adresse IP) **ET logique** (Masque)
- Pour obtenir l'adresse de diffusion (broadcast), tous les bits de l'id_ordinateur sont mises à 1 :
  - adresse de diffusion = (Adresse IP) **OU logique** (le complément à 1 du masque)
- L'identifiant hôte = (Adresse IP) **ET logique** (le complément à 1 du masque)

Exemple

- Soit l'adresse IP 192.168.1.72/24

Adresse IP	1100 0000 . 1010 1000 . 0000 0001	0100 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111	0000 0000	255.255.255.0
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001	0000 0000	192.168.1.0
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0000	0100 1000	0.0.0.72
Adresse broadcast	1100 0000 . 1010 1000 . 0000 0001	1111 1111	192.168.1.255

Généralisation des masques

- Dans la version d'origine d'IP, une adresse IP est constituée d'un id_réseau et d'un id_ordinateur dans ce réseau selon la classe
- L'usage des masques fait maintenant partie du standard IP
- Toutes les stations/routeurs les utilisent
- Masques par défaut selon les classes :
  - 255.0.0.0 (ou /8) pour les réseaux de classe A
  - 255.255.0.0 (ou /16) pour les réseaux de classe B
  - 255.255.255.0 (ou /24) pour les réseaux de classe C
- Où /n est la notation CIDR (*Classless Inter-Domain Routing)* qui signifie que la partie identifiant le réseau se trouve sur les n premiers bits

Sous-réseaux IP et diffusion

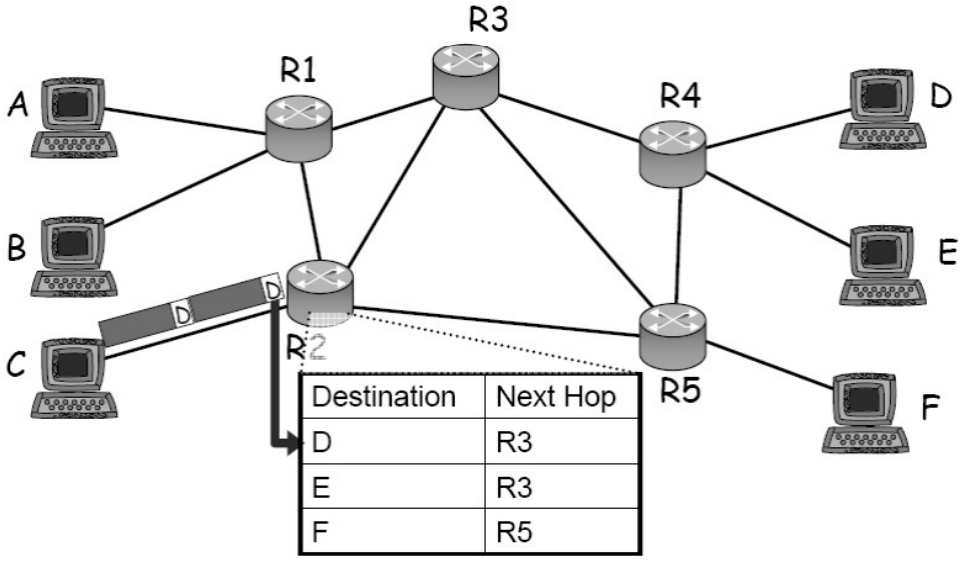
- Adresse de diffusion 139.124.255.255 :
  - dans tout le réseau 139.124.0.0 si non subnetté
  - dans tous les sous-réseaux de 139.124.0.0 si subnetté
- Adresse 139.124.1.255 pour sous-réseau 139.124.1.0/24 :
  - diffusion dans le seul sous-réseau 139.124.1.0

Sous-réseaux IP : flexibilité

- Bien que ce ne soit pas du tout conseillé, l'identifiant sous-réseau n'est pas forcement un groupe de bits contigus
- Il est possible de regrouper plusieurs entrées de la table de routage en une seule, en jouant sur le masque associe (agrégation d'adresse)
- On n'est pas limité au subnetting d'un réseau : le subnetting d'un sous-réseau est aussi possible
- Les exemples précédents n'ont montré que le subnetting avec des sous-réseau de taille égale (la taille de la partie id.sous-réseau était la même pour tous les sous-réseaux). Mais on peut aussi subnetter avec des sous-réseaux de taille variable par la technique du sous-adressage variable (VLSM)

Routage

- Le routage, dans un réseau, consiste à assurer le transit des blocs d'un point d'entrée à un point de sortie désigné par son adresse, c'est-à-dire la détermination d'un chemin à travers le réseau entre une machine émettrice et une machine réceptrice, toutes les deux identifiées par leurs adresses.
  - Chaque station ou routeur dispose de sa propre table de routage



Destination	Next Hop
D	R3
E	R3
F	R5

R2 atteint D par l'intermédiaire de R3 :

⟹R3 est son «next-hop» sur le chemin vers D.

Tables de routage

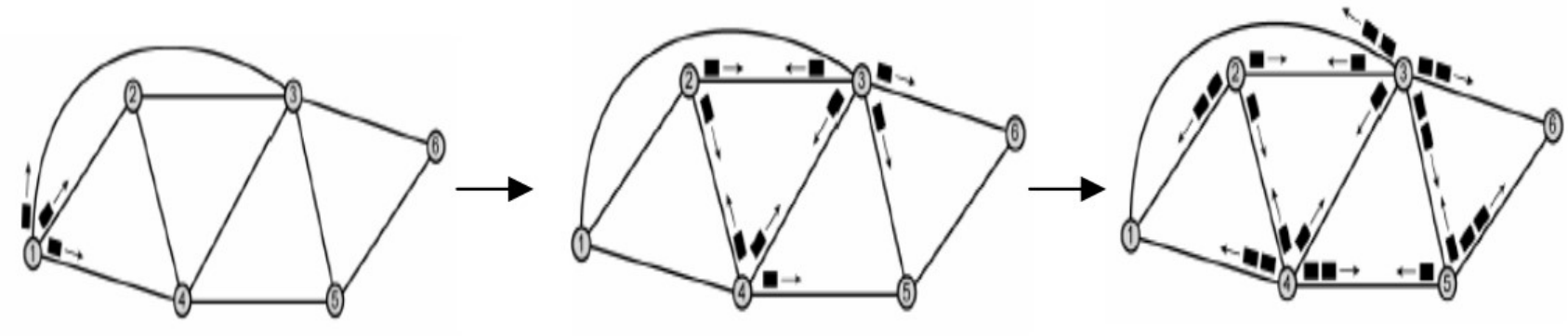
- Une table contient autant d'entrées que de destinations (réseaux) connues de l'hôte
- Une entrée est un couple
  (réseau de destination, adresse du saut suivant)
- Pour une destination donnée, la table n'indique pas le chemin à suivre, seulement le routeur à qui confier le datagramme
  - le chemin est une information repartie
  - ce routeur doit être directement accessible
- Une destination dont l'adresse de réseau ne figure pas dans la table est inaccessible
- Une table de routage peut contenir une **route par défaut** (pour les destinations inconnues).

Protocole de routage

- Décrit comment sont construites les tables de routage
- Composantes d'un protocole de routage :
  - Algorithmes : dans le cas de protocole de routage les algorithmes sont utilisés pour faciliter la détermination du meilleur chemin
  - Messages du protocole de routage : utilisés pour découvrir les voisins et échanger l'information de routage

Routage par inondation

- Chaque paquet entrant est envoyé sur toutes les lignes de sorties, sauf celle d'où provient le paquet
  - Ceci pourrait conduire à la congestion du réseau
- Placer un compteur de sauts dans l'entête de chaque paquet
  - ce compteur est initialisé à l'émission par le nombre de sauts autorisés et décrémenté à chaque saut
  - les paquets sont éliminés lorsque le compteur atteint zéro
- Pour éviter les bouclages, les paquets sont numérotés
  - chaque nœud mémorise cet identifiant et détruit les paquets déjà vus

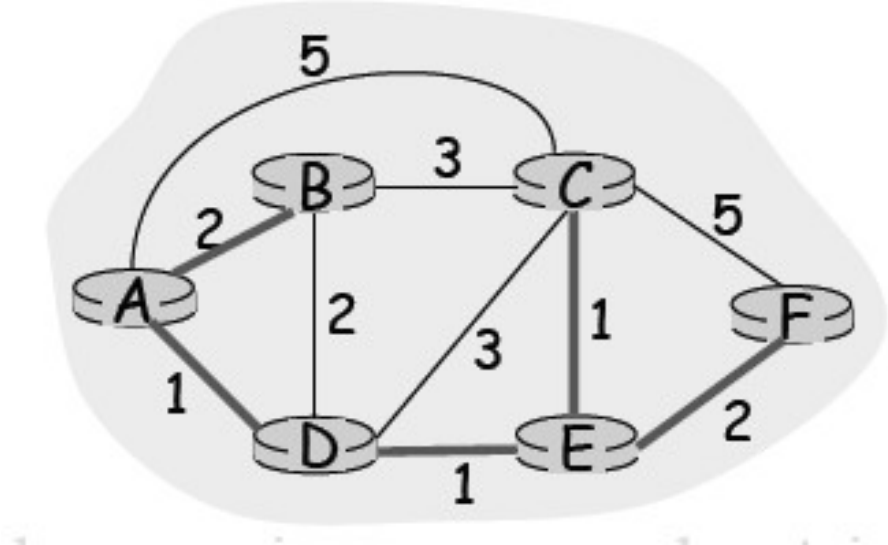


Routage statique

- Les algorithmes ne fondent pas leurs décisions en fonction des mesures du trafic et de la topologie
- Une table est construite par l'administrateur du réseau lors de configuration du réseau et à chaque changement de topologie
  - Les routes sont calculées, puis implémentées dans le routeur
  - Il n'existe pas de solution de secours en cas de rupture d'un lien
- Le routage statique n'est pas optimal, il convient parfaitement aux petits réseaux et aux réseaux dans lesquels il n'existe pas de redondance dans les routes
- Et si des routeurs sont ajoutés ou supprimés ?
  - Configurer les tables de routages des routeurs supplémentaires
  - Modifier les tables de routage de chacun des routeurs déjà présents

Calcul des routes (Algorithme de Dijkstra)

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					



⇒

TR de A		
Dest.	Next	Coût
A	-	0
B	B	2
C	D	3
D	D	1
E	D	2
F	D	4

Routage dynamique

- Motivation du routage dynamique :
  - Dans un environnement complexe, la mise en œuvre du routage statique est souvent difficile à maintenir
  - La mise en place d'un mécanisme de routage dynamique permet de faciliter les mises à jour
- Principe de fonctionnement général du routage dynamique :
  - Les routeurs communiquent des informations entre eux
  - Chaque routeur met à jour sa table de routage à partir des informations reçues depuis les autres
- Le but est de maintenir des informations associées aux routes en fonction de l'état du réseau

Routage au moindre coût

- Dans ce mode de routage, chaque nœud maintient à jour des tables indiquant quel est le plus court chemin pour atteindre le nœud destination
- Chaque lien a un coût affecté ou calculé. Ce coût ou métrique peut être exprimé en :
  - nombre de sauts
  - en km, distance réelle
  - en temps de latence dans les files d'attente
  - en délai de transmission
  - fiabilité...
- Dans certains protocoles de routage, un nœud peut maintenir plusieurs tables de routage et ainsi acheminer les données en fonction d'une qualité de service requise
- Les algorithmes de routage au moindre coût diffèrent selon la manière dont ils prennent en compte ces coûts pour construire les tables de routage
- Deux classes d'algorithmes :
  - Algorithmes à vecteur de distances
  - Algorithmes dits à état des liens

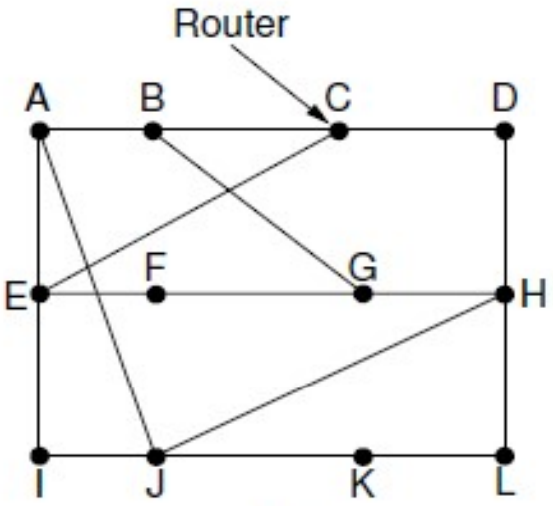
Routage à vecteur de distances

- Algorithme de Bellman-Ford distribué (ou Ford-Fulkerson 1962)
- Algorithme simple basé sur :
  - l'échange d'informations entre routeurs adjacents (liaison directe)
    - vecteur de distance (≠ table de routage)
  - propagation de proche en proche de l'accessibilité du réseau
- Fonctionne bien sur des systèmes de petite taille
  - utilisé sur des sites avec quelques routeurs pour éviter les configurations manuelles
- Chaque routeur n'a qu'une vision partielle du réseau

Principe

- Les routeurs ne connaissent initialement que leurs propres liaisons
  - La table de routage de chacun ne comporte qu'une entrée, elle indique que le coût pour se joindre est nul (locale)
- Périodiquement, ils diffusent leurs vecteurs de distance (table de routage sans les interface) à leur voisins
- A la réception d'un vecteur, un routeur intègre l'information dans sa table :
  - rajout des entrées nouvelles en indiquant l'interface d'arrivée
  - modifier le coût des entrées
    - si un plus court chemin est proposé
    - si un plus long chemin est proposé par la même interface que celle de la table
- Les échanges successifs doivent amener à la convergence
  - après un certain nombre d'itérations, chaque routeur va connaître le coût vers chaque destination
    - les échanges ultérieurs n'apportent aucune connaissance nouvelle

Exemple (1)



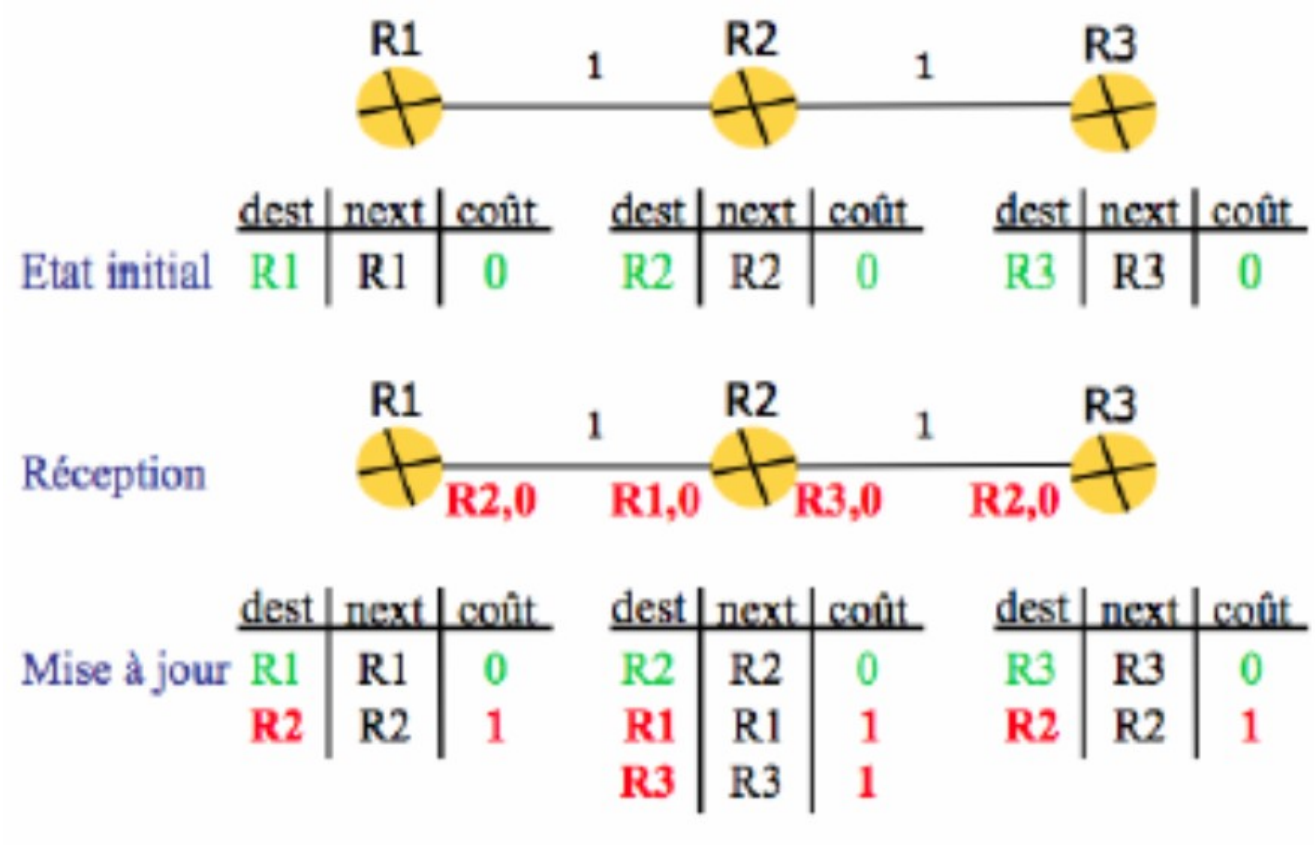
(a)

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

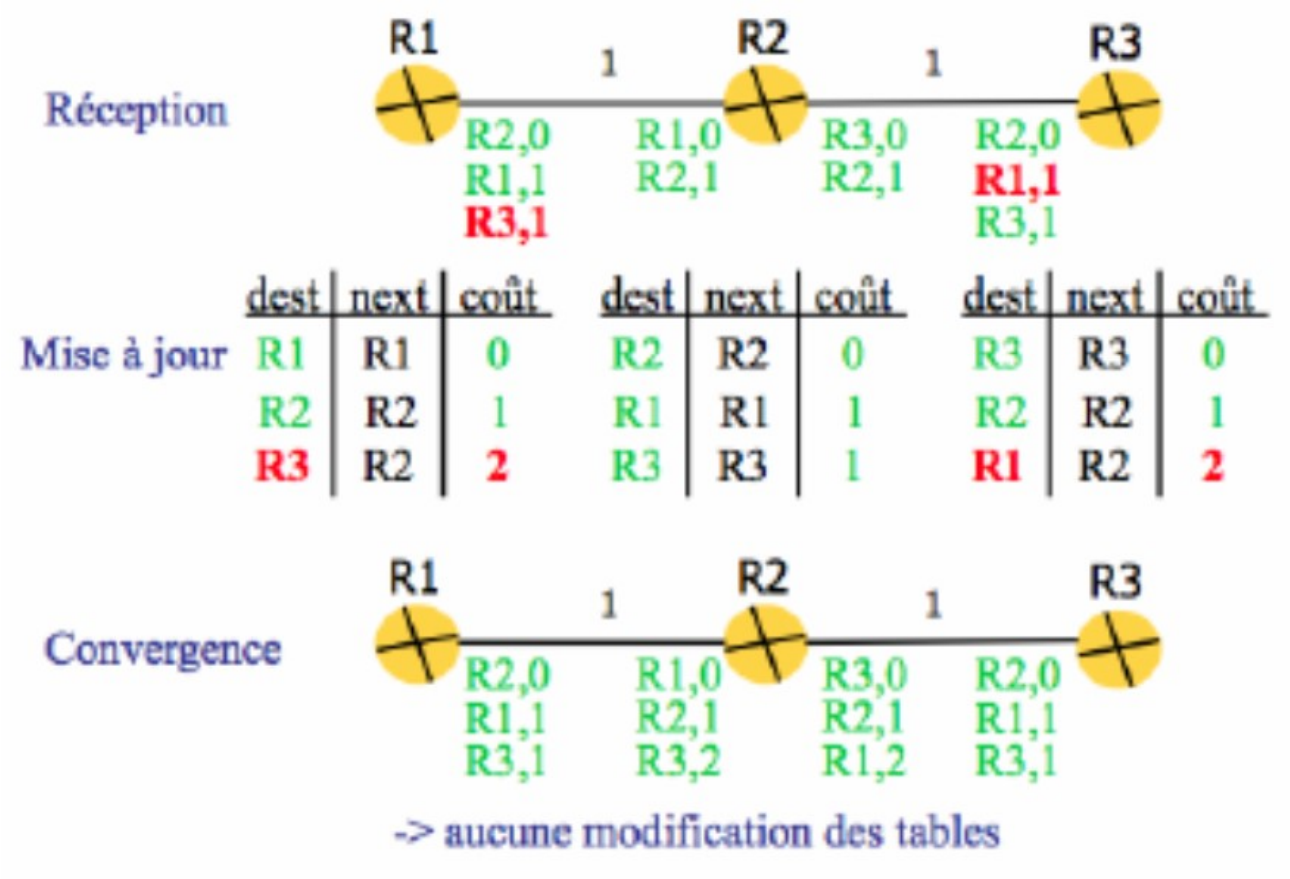
Exemple (2)



Etat initial	dest	next	coût
R1	R1	0
R2	R2	0
R3	R3	0

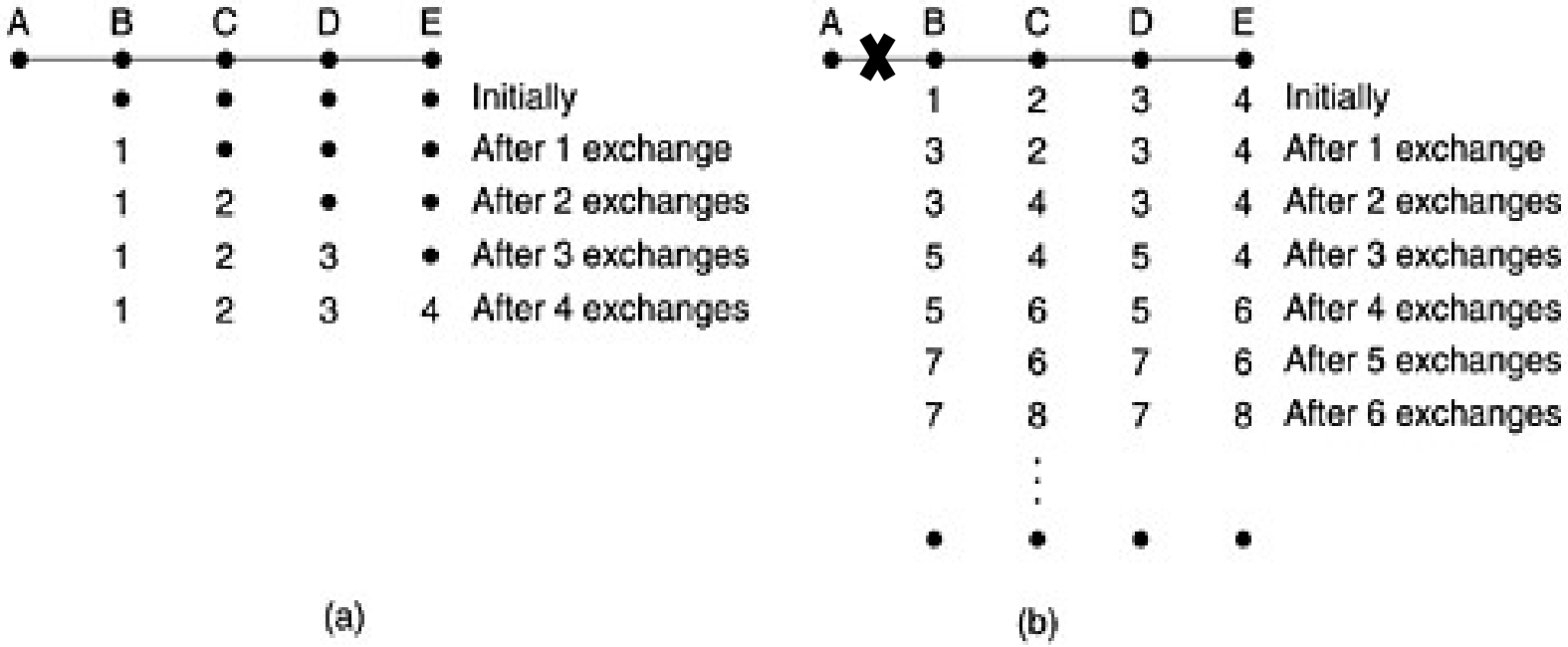
Mise à jour	dest	next	coût
R1	R1	0
R2	R2	1
R2	R2	0
R1	R1	1
R3	R3	1
R3	R3	0
R2	R2	1

Exemple (2)


Réception
R1
1
R2
1
R3
R2,0
R1,1
R3,1
R1,0
R2,1
R3,0
R2,1
R2,0
R1,1
R3,1
Mise à jour
dest | next | coût
R1 | R1 | 0
R2 | R2 | 1
R3 | R2 | 2
dest | next | coût
R2 | R2 | 0
R1 | R1 | 1
R3 | R3 | 1
dest | next | coût
R3 | R3 | 0
R2 | R2 | 1
R1 | R2 | 2
Convergence
R1
1
R2
1
R3
R2,0
R1,1
R3,1
R1,0
R2,1
R3,2
R3,0
R2,1
R1,2
R2,0
R1,1
R3,1
-> aucune modification des tables

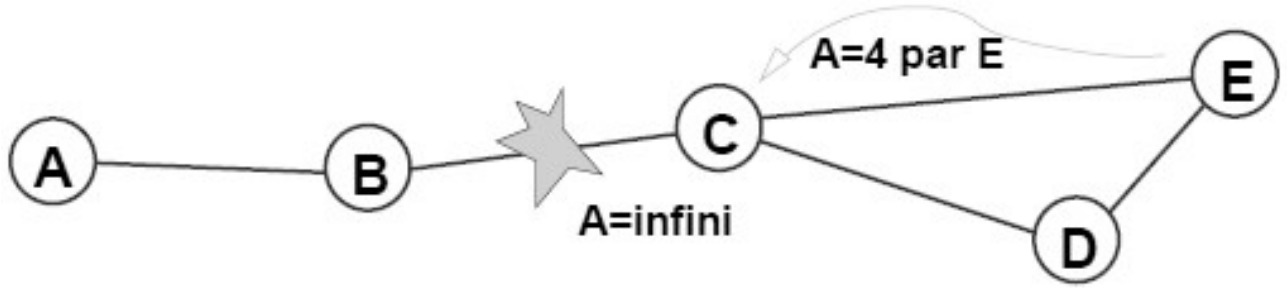
Exemple (3)

- En cas de panne de A : divergence des distances. . .



Problème des algorithmes de routage à vecteur de distance

- Plusieurs problèmes sont apparus avec ces algorithmes :
  - convergence lente
  - risques de boucle
    - interdire aux nœuds de signaler qu'ils connaissent une destination au routeur par lequel ils l'ont apprise (horizon coupé : split horizon)
    - limiter la valeur de l'infini



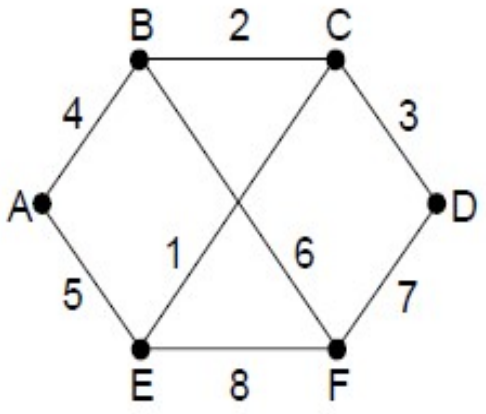
  - envoi de vecteurs avec tous les réseaux de la table de routage
    - taille de réseau limitée

Algorithmes à état des liens

- Comment s'adapter à des réseaux importants tout en évitant la propagation des informations de proche en proche ?
  - Connaitre son voisinage
    - envoi de paquets de détection sur les liaisons
    - détermine le coût de chaque lien qui lui est raccordé
  - Construire une synthèse de l'info locale
  - Diffuser l'info locale à tous les routeurs (par inondation)
    - Les routeurs doivent recevoir les messages de tous les routeurs
    - Besoin d'une distribution fiable (numéro de séquence, âge de la connexion)
  - Construire un graphe représentant le réseau (matrice de coûts)
  - Calculer le plus court chemin vers tous les routeurs
    - Algorithme de Dijkstra

Construction des paquets (LSP)

- Le paquet diffusé contient :
  - Routeur source
  - Numéro de séquence (incrémenté à chaque envoi)
    - détection des nouvelles infos
  - Age ou date : décrémenté par les routeurs avec le temps
    - détection des infos déjà connues
  - Distance entre voisins et source



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

Organisation de gros réseaux : Internet

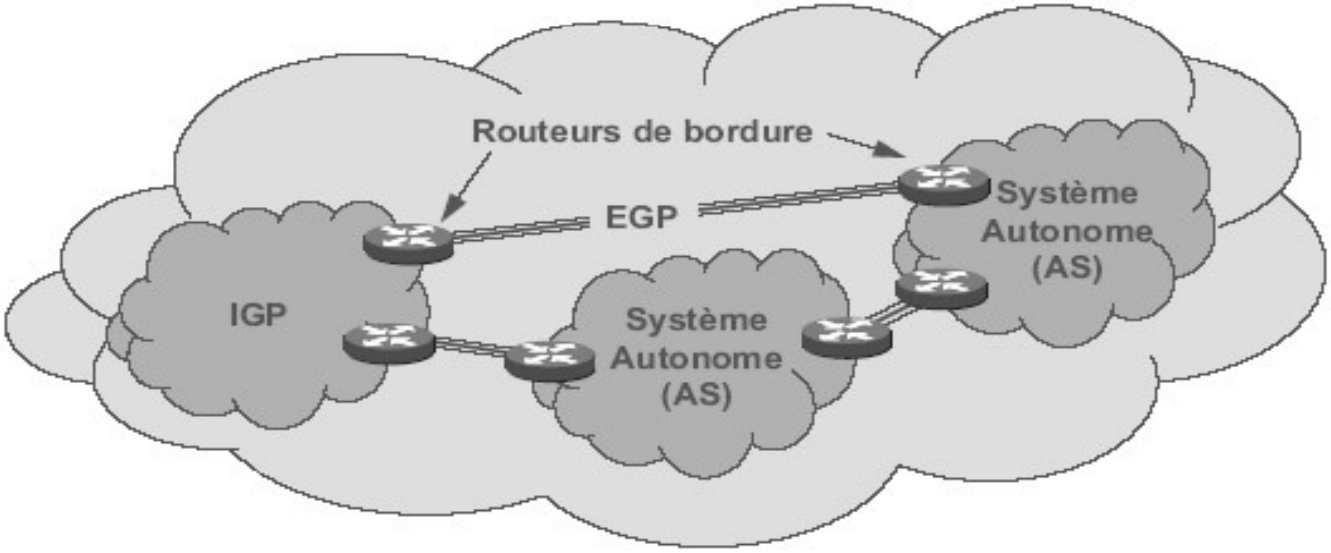
- Interconnexion de réseaux de différents opérateurs
- Chaque opérateur se débrouille pour router ses propres informations en interne.
- Protocole commun d'information de routage entre les réseaux des différents opérateurs
- AS (Autonomous System) : ensemble de réseaux contrôlés par une seule autorité

Routage hiérarchique

- Deux catégories de protocole :
  - Protocoles de routage intradomaine (IGP : Interior Gateway Protocols)
    - Protocoles de routage utilisés à l'intérieur d'un AS
      - basés sur la route la plus efficace (le plus court chemin)
      - RIP-2, EIGRP, IS-IS, OSPF
  - Protocoles de routage interdomaine (EGP : Exterior Gateway Protocols)
    - Routage entre AS
      - basés sur les accords commerciaux ou politiques entre les systèmes autonomes
      - il n'y en a qu'un : BGP-4

Routage hiérarchique

- Notons que les machines d'accès à l'interréseau, dits routeurs de bordure
  - mettent en œuvre les deux types de protocoles, un protocole intradomaine sur leur lien intradomaine, et un protocole interdomaine sur le lien interréseaux



Exemple

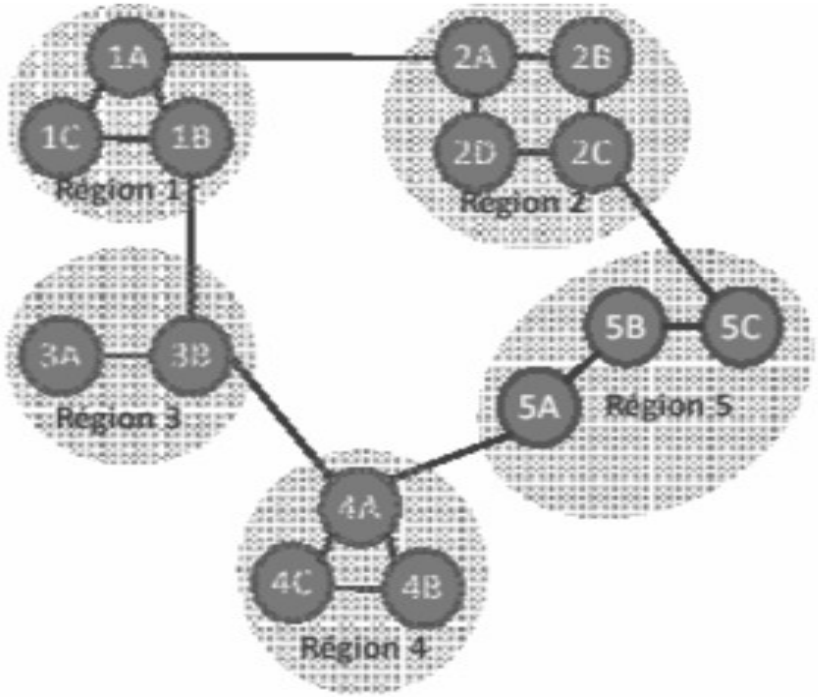


Table de routage de **1C** sans hiérarchie

Dest	Next Hop	Sauts (hops)
1A	**1A**	1
1B	*1B*	1
1C	-	0
2A	**1A**	2
2B	**1A**	3
2C	**1A**	4
2D	**1A**	3
3A	*1B*	3
3B	*1B*	2
4A	*1B*	3
4B	*1B*	4
4C	*1B*	4
5A	*1B*	4
5B	*1B*	5
5C	**1A**	5

Table de routage de **1C** avec hiérarchie

Dest	Next Hop	Sauts (hops)
1A	**1A**	1
1B	*1B*	1
1C	-	0
2	**1A**	2
3	*1B*	2
4	*1B*	3
5	*1B*	4

Économie d'espace !