TCP/IP

IP : Adressage & Routage

Pascal Mérindol & Julian Del Fiore

merindol@unistra.fr
http://www-r2.u-strasbg.fr/~merindol/

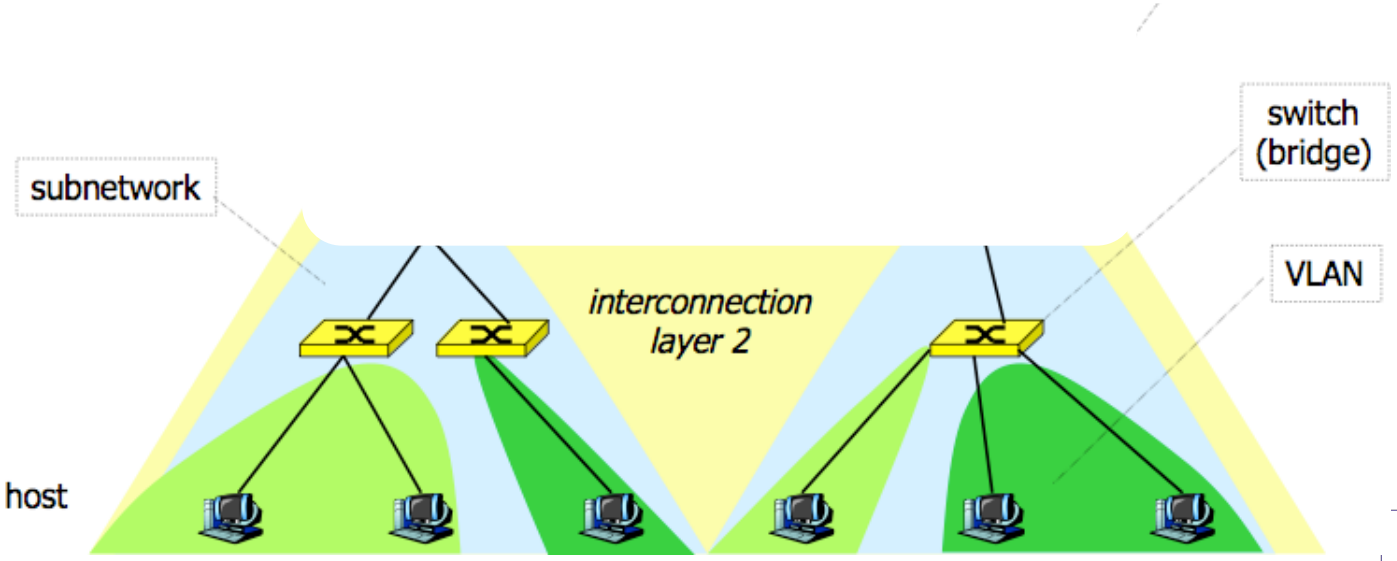
Ce cours est construit sur la base de plusieurs supports pédagogiques parmi lesquels les cours de Jean-Jacques Pansiot, Antoine Gallais et Julien Montavont. L'usage de ce support ne peut être qu'académique.

Plan

- **Généralités : vue d'ensemble, définitions, etc.**
- **Formats des paquets (PDU de niveau 3) et adressage IP**
- **Plan de commutation vs. plan de contrôle**
- **Routage**
  - Etat des liens : OSPF
  - (Vecteurs de distance : RIP)

Vue d'ensemble (couche «MAC»)

- **Inter-connexion des hôtes au niveau 2 :** les ponts (~switchs, commutateur)
  - Ethernet, Token Ring, WiFi, ...
  - STP, VLAN, ...



© A. Duda

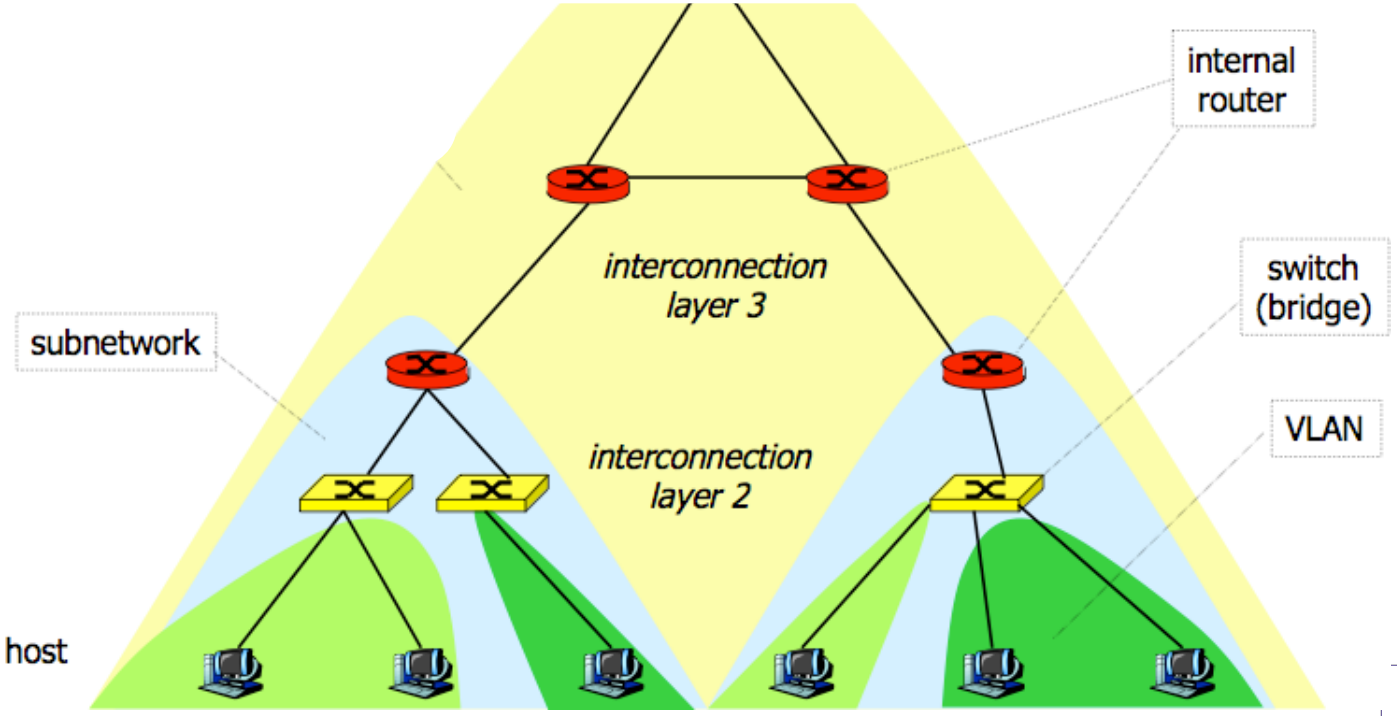
Configuration IP hôte

- manuel/statique (ifconfig & ipconfig) ou dynamique (DHCP)
  - ➡adresse IP, *S* et masque *M* (/ interface) ~ sous réseau local
  - ➡adresse IP, *G* du routeur par défaut (gateway) ~ sortie vers l'extérieur
  - ➡adresse Ethernet *sss* (/ MAC) connue statiquement ~ local + gateway
    - table ARP construite dynamiquement

```
/* IP destination = D (fournie par couche applicative)*/
Si ( D ∧ M = S ∧ M ) /* D appartient au réseau local : M = D ∧ M = S ∧ M*/
alors si D ∉ Table ARP
alors envoyer requête ARP(D) en broadcast Ethernet
     si réponse ARP(D,ddd): mettre (D, ddd) dans le cache ARP & envoyer trame
     sinon échec /* pas de réponse ARP */
si (D, d) ∈ table ARP
alors envoyer paquet dans une trame ethernet source s et destination d
sinon /* D n'appartient pas au sous-réseau local */
envoyer le paquet dans une trame ethernet source sss et destination ggg
/* ggg est l'adresse ethernet de G contenue dans la table ARP et apprise par ARP */
```

Vue d'ensemble (couche IP / intra-domaine)

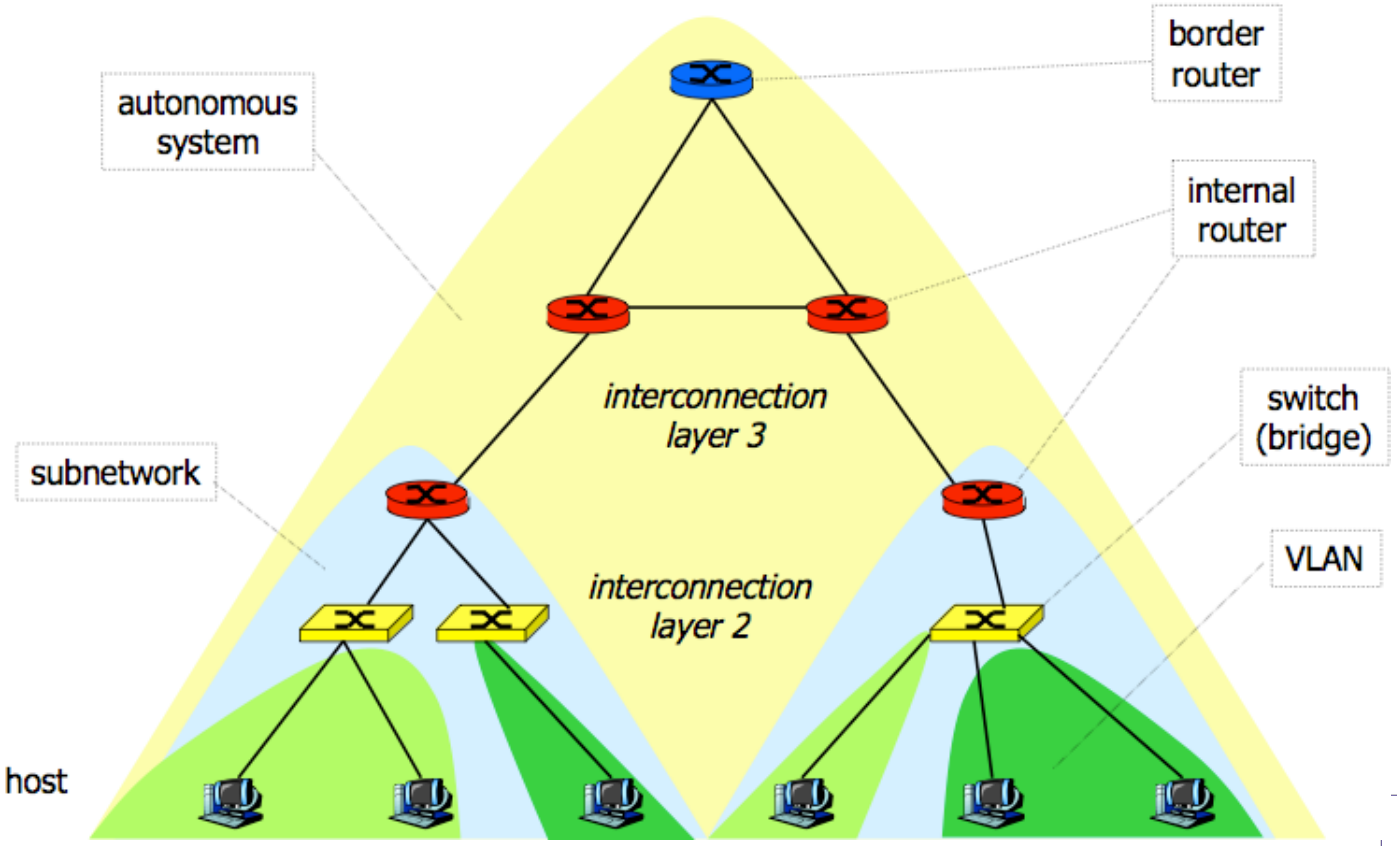
- **Inter-connection de LAN au niveau 3 :** les routeurs (backbone + access)
  - RIP, OSPF, IS-IS, ...



© A. Duda

Vue d'ensemble (couche IP / inter-domaine)

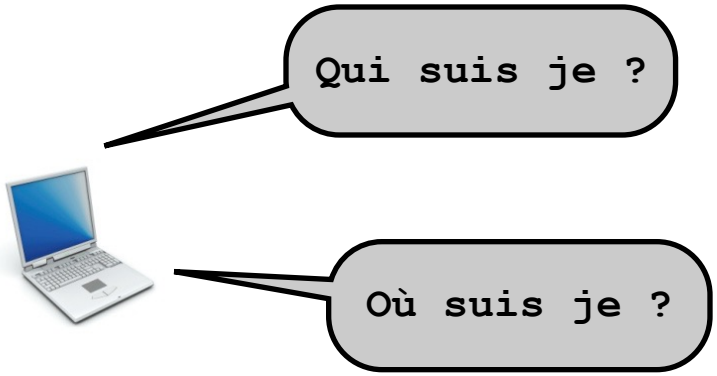
- **Inter-connection AS :** les routeurs de bordures (ASBR)
  - EGP = BGP (E-BGP + I-BGP)



© A. Duda

Identifier et localiser les hôtes

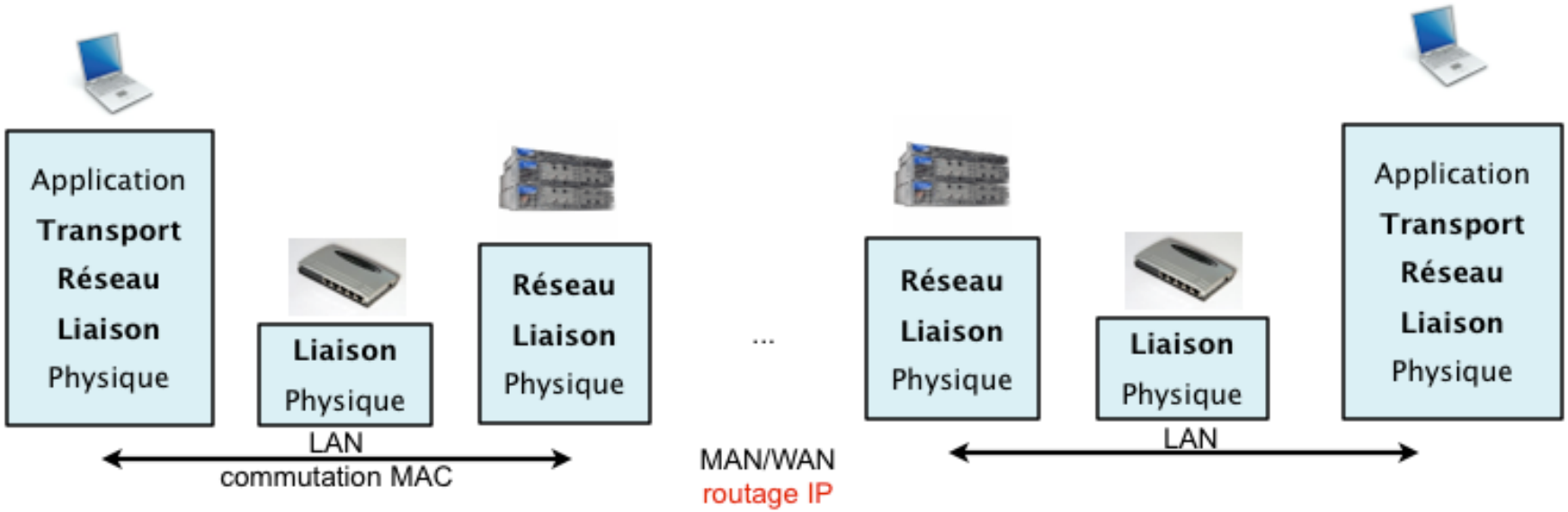
- **Adressage IPv4/v6**
  - Dans sa forme actuelle, une adresse IP assure ces deux fonctions
    - Locator Identifier Separation Protocol (LISP) ?
  - Comment organiser cette adressage ?
    - notions de plan d'adressage :
      - distribution des adresses IP
      - sous-réseaux IP
  - Quel lien avec la localisation ?
    - «interaction» adressage / routage



Acheminer les données

- **Protocoles de routage (*signalisation*) et commutation (*forwarding*)**
  - Plusieurs échelles !
    - niveau 2 (LAN) => commutation très basique : distance faible (~1km)
    - au niveau 3 :
      - MAN - routage intra-domaine (RIP, OSPF,...)
        - distances moyennes (ville/pays)
        - objectif : optimiser les délais d'acheminement, etc.
      - WAN - routage inter-domaine (BGP)
        - distances élevées (continent/ la terre entière...)
        - objectif : se faire de l'argent !
        - contrainte : passage à l'échelle...

Objectif : inter-connecter des LAN



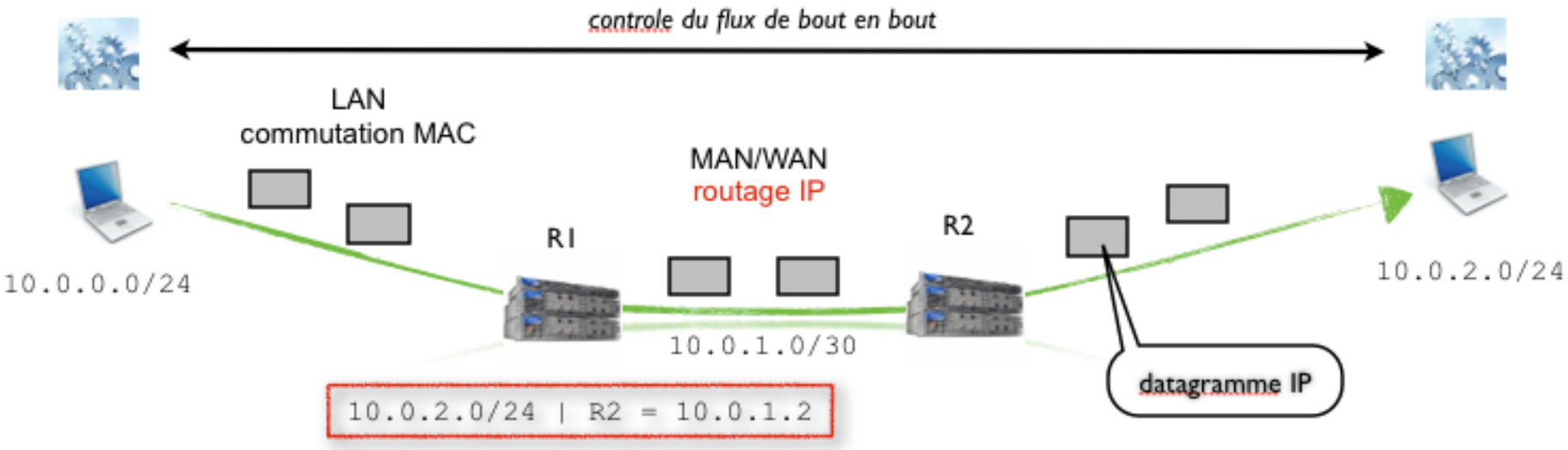
traceroute

```
sudo ./lft -A 203.222.38.67
TTL LFT trace to sl-gw1-hk-.sprintlink.net (203.222.38.67):80/tcp
1 [2259] api-rc1-ge-0-2-0-2.u-strasbg.fr (130.79.91.253) 0.9/0.8ms
2 [2259] crc-rc1-ge-0-1-0-0.u-strasbg.fr (130.79.20.13) 1.0/2.3ms
** [neglected] no reply packets received from TTL 3
4 [2200] te0-2-0-0-besancon-rtr-011.noc.renater.fr (193.51.189.122) 11.7/13.6ms
5 [2200] te0-1-0-0-dijon-rtr-011.noc.renater.fr (193.51.189.114) 14.4/12.0ms
6 [2200] te0-3-2-0-lyon1-rtr-001.noc.renater.fr (193.51.189.141) 10.0/16.7ms
7 [3356] xe-8-0-0.edge5.Paris1.Level3.net (212.73.207.173) 76.4/15.9ms
8 [3356] ae-33-51.ebr1.Paris1.Level3.net (4.69.139.193) 16.5/16.2ms
9 [3356] ae-2-2.ebr1.London1.Level3.net (4.69.142.106) 23.4/23.2ms
```

```
10 [3356] ae-11-51.car1.London1.Level3.net (4.69.139.66) 23.8/23.4ms
11 [1239] sl-bb21-lon-10-0-0.sprintlink.net (213.206.131.21) 23.8/23.1ms
12 [1239] sl-bb22-lon-3-0-0.sprintlink.net (213.206.129.153) 23.6/23.6ms
13 [1239] sl-bb20-bru-14-0-0.sprintlink.net (213.206.129.42) 32.6/32.7ms
14 [1239] sl-bb21-bru-15-0-0.sprintlink.net (80.66.128.42) 32.8/32.5ms
15 [1239] sl-bb20-ams-14-0-0.sprintlink.net (213.206.129.45) 32.9/32.2ms
16 [1239] sl-bb21-ham-6-0-0.sprintlink.net (213.206.129.145) 38.6/39.0ms
17 [1239] sl-bb21-cop-13-0-0.sprintlink.net (213.206.129.57) 43.8/43.7ms
18 [1239] sl-bb21-sto-14-0-0.sprintlink.net (213.206.129.34) 52.6/52.1ms
19 [1239] sl-bb20-hk-11-1-3.sprintlink.net (213.206.129.131) 283.9ms
```

IP = service offert à la couche transport

- ✓en mode NON connecté
- ✓le contrôle de flux (pertes & congestions) est gérée au niveau de TCP
- ✓adapté aux services asynchrones (données) ou aux transactions (requêtes/réponses)
- ✓difficulté pour garantir la qualité de service (QoS) ➥ «best effort»
- ✓paquets/datagrammes acheminés indépendamment
- ✓table de routage avec une entrée par destination (préfixe IP)
- ✓robustesse (résistance aux pannes), convergence ?



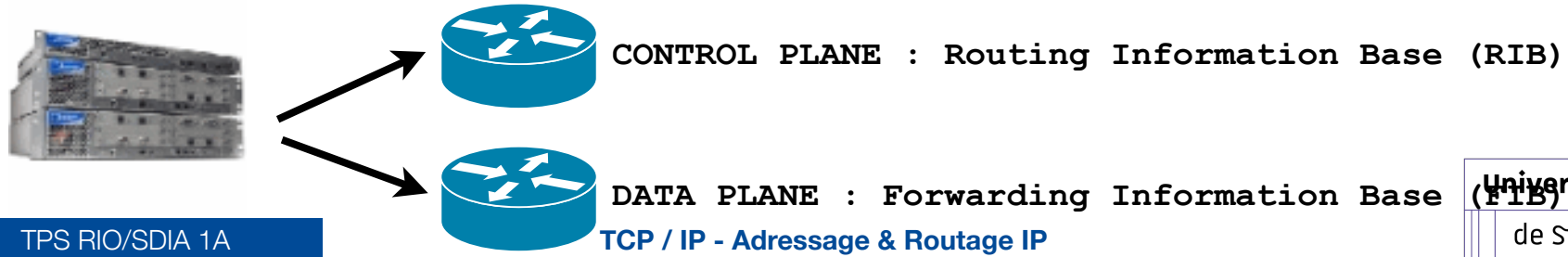
IP : Internet Protocol, 1981 RFC 791

✓envoi de datagrammes (paquets IP)
- ➡de bout en bout à travers des (sous-)réseaux interconnectés (Internet)
- ➡(sous-)réseaux hétérogènes (débit, taille trames, adressage)

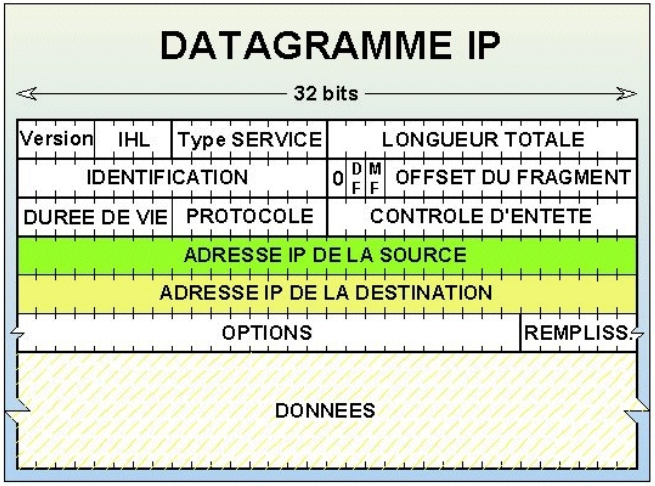
✓IP = couche unificatrice
- ➡IP au dessus de tout protocole (LAN, WAN, satellite, connecté ou non, …)
- ➡règles d'encapsulation du paquet IP pour chaque type de liaison

✓fonctionnement simple (hors fragmentation)
- ➡**dans le plan de données (DATA PLANE)**
- ➡pas de retransmission (non fiable)
- ➡déséquencement possible
- ➡pas ou peu de contrôle de flux/congestion
- ➡modèle de service usuel : « **best effort** »

✓**complexité** dans le **plan de contrôle/signalisation (CONTROL PLANE)**
- ➡protocoles de routage : apprentissage, calculs, etc ...



Format Paquet IP



Format de l'entête (IP header)

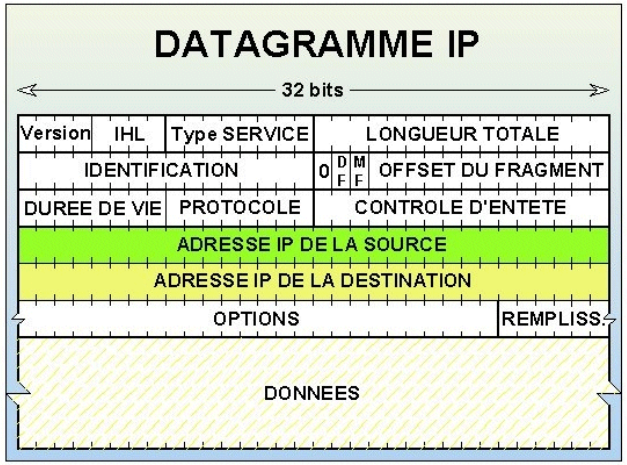
- ✓Entête obligatoire = 20 octets
- ✓+ Options facultatives
- ✓+ Données (total < 2^{16} octets : 64K)

➡ Fonctions IP ⇔ champs de l'entête

	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31
1	version	Lg_ent	ToS/DSCP,	ECN	Longueur totale			
2	identification				Flags (3b), Offset fragment (13 bits)			
3	TTL		protocole		total de contrôle entête			
4	Adresse IP source							
5	Adresse IP destination							
6+	Eventuelles options + bourrage : N * 32 bits, N ≥ 0							

© Pascal Terrioll

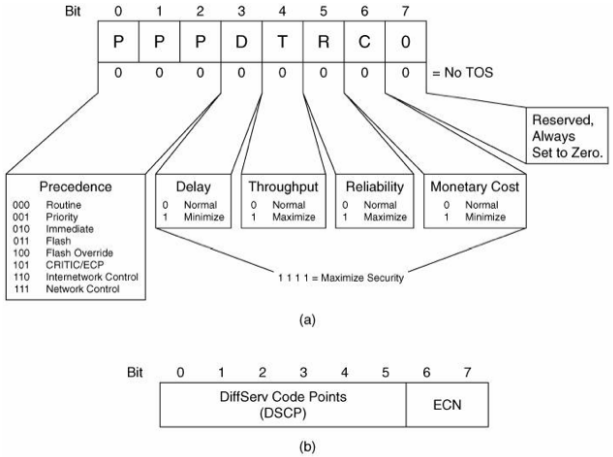
Zoom sur l'entête IP



- IPv4 Header
  - ✓version : n° de version courante (4)
  - ✓IHL - *Internet Header Length* : longueur en-tête en mots de 32 bits (sans options 20 octets = 5 mots)
  - ✓type de service (ToS) : priorité du paquet et type de routage souhaité (non utilisé par défaut)
    - ➡Utilisé en pratique pour le champ DSCP (*diffserv*) et pour les bits ECN
  - ✓longueur totale : du paquet ou du fragment en nombre d'octets
  - ✓identification, flags, offset fragment : **fragmentation/réassemblage**
  - ✓durée de vie (TTL) : durée de vie maximale d'un paquet en nombre de secondes
    - ➡en pratique décrémenté de 1 dans chaque routeur traversé
    - ➡si TTL = 0 et paquet non arrivé ➥ *message ICMP Time-to-live exceeded in transit*
  - ✓protocole : n° du protocole de transport qui récupère le paquet IP au bout du chemin : TCP(6), UDP(17), DCCP(33), ICMP(1), IGMP(2), OSPF (89)
  - ✓total de contrôle d'en-tête (checksum) : contrôle de redondance sur l'en-tête
  - ✓adresse IP source (4 octets) unicast
  - ✓adresse IP destination (4 octets) unicast, multicast, broadcast
  - ✓options IP (éventuelles)
  - ✓bourrage : permet que la longueur de l'en-tête soit un multiple de 32 bits

Type of Service (ToS)

- Signification d'origine
  - ✓priorité :
    - ➡valeurs codées de 0 à 7
    - ➡*111 - Network Control, 110 - Internetwork Control, 101 - CRITIC/ECP, 100 - Flash Override,*
    - ➡*011 - Flash, 010 - Immediate, 001 - Priority, 000 - Routine*
  - ✓bits D, T, R : qualité de service d'acheminement désirée
    - ➡D = 1 : demande un délai d'acheminement court
    - ➡T = 1 : demande un débit de transmission élevé
    - ➡R = 1 : demande une grande fiabilité



- En pratique ToS souvent ignoré par les routeurs
  - ✓Maintenant utilisé par Diffserv (voir RFC 2474)
    - ➡architecture à Différentiation de services
    - ➡le champ DSCP code la classe de service dans ToS
  - ✓Bits 6 et 7 maintenant utilisés pour indication de congestion ECN
    - ➡ECN : Explicit Congestion Notification (voir RFC 3168)

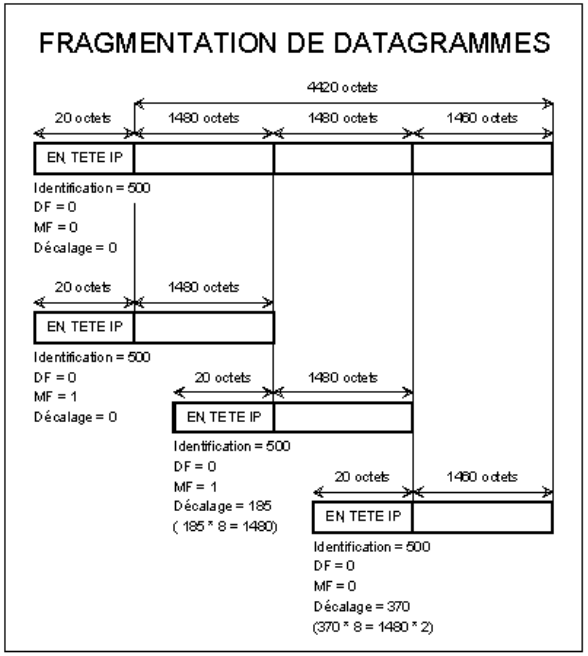
Options IP

- ***source routing* (SR) :** Indique la liste des routeurs à utiliser le long de la route
- ***route recording* (RR) :** Enregistre les routeurs utilisés le long de la route
  - Limite sur la taille des options (Record_Route : 9 routeurs)
- **Router Alert :** Force une lecture plus attentive de l'entête du paquet
- **Options souvent ignorées :** Coût de traitement
  - Traitement rapide des paquets avec en-tête de longueur fixe (sans option)

➡ **Slow path (lookup approfondi) ≠ Fast path (lookup direct)**

DF : Don't fragment ! - MF : More Fragments !
- Offset : position en octet

Fragmentation et ré-assemblage



- **Arrivée d'un paquet**
  - MF=0 && offset=0 : Paquet non fragmenté
    - Livrer paquet couche supérieure (TCP, UDP,…)
  - Autre combinaison
    - Si 1er fragment avec cet identifiant (offset=0)
      - ➡ Allocation d'un buffer, armement d'un timeout
    - Sinon : Insérer fragment dans buffer à position offset
    - Si paquet complet /* test sur MF */
      - Livrer le paquet, libérer le buffer, désarmer le timeout
- **Expiration du timeout :** Libération du buffer
  - ➡ Envoi d'un message d'erreur ICMP : /* Fragment reassembly time exceeded */

Path MTU

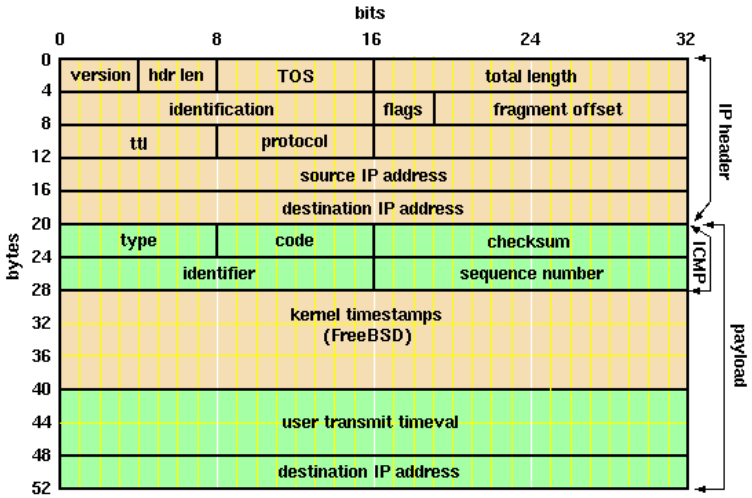
http://tools.ietf.org/html/rfc1191
http://tools.ietf.org/html/rfc1981

- **Objectif :** Eviter la fragmentation
- ***Path MTU discovery***
  - PMTU = Minimum des MTU le long de la route
  - Paquets de taille décroissante émis avec DF= 1
    - DF = 1 et paquet trop gros : drop du routeur
      - ➡ Envoi message d'erreur ICMP à la source (avec MTU)
  - Dès que l'on ne reçoit plus d'erreur ICMP, on a trouvé le PMTU

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type = 3  |     Code = 4  |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             unused = 0        |         Next-Hop MTU          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Internet Header + 64 bits of Original Datagram Data      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Plusieurs protocoles "juste au dessus" d'IP

- **Exemple :** Internet Control Message Protocol **(**ICMP, RFC 792)
  - Numéro de Protocole = 1
  - Messages d'erreur
    - Destination inaccessible
    - Temps dépassé
    - ...
  - Requêtes/réponses
    - Ex : ping



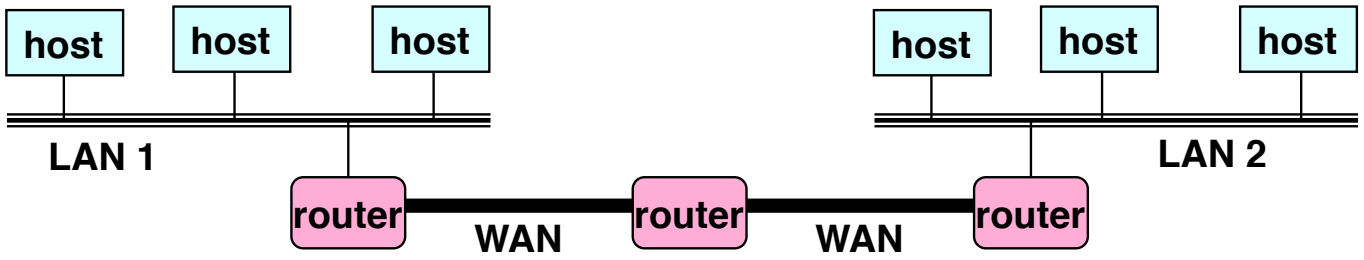
Les adresses IP : qu'est-ce que c'est ?

- **Adresses IP**
  - Notation en décimale pointée
  - Notions de préfixes et sous-réseaux IP pour l'agrégation
- **Allocation d'adresse**
  - Adresses de classe
  - Classless InterDomain Routing (CIDR)
- ***Commutation de paquet IP***
  - Principe du *Longest-prefix match forwarding*
  - Adresse IP = «index» des tables de routage

Adresses IPv4

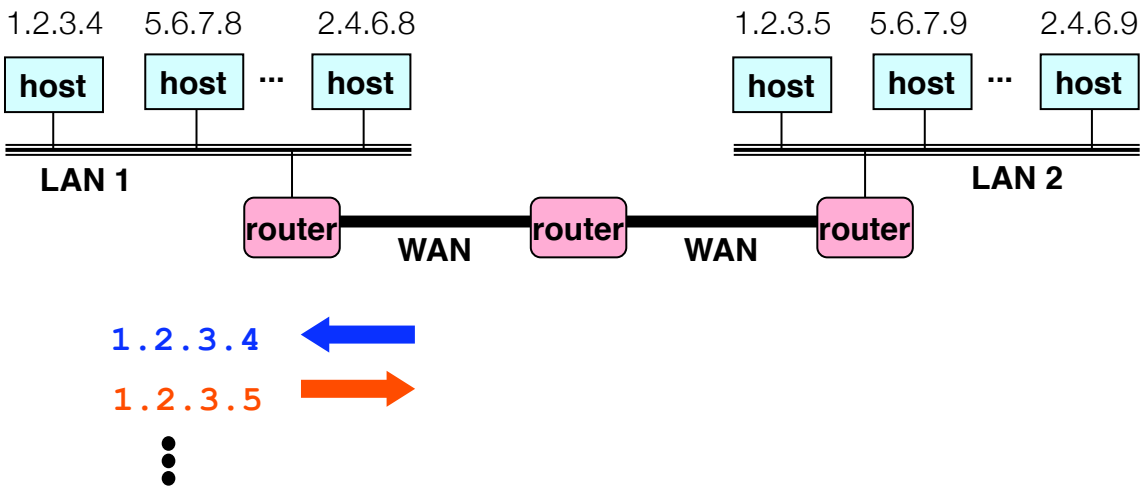
- **Codée sur 32 bits :** Notation décimale pointée
  - 11000000.10101000.00000000.00000001 <=> 192.168.0.1
- **Une adresse ~ une interface**
- **Internet = Inter-réseaux**
  - Objectif : inter-connecter des **réseaux locaux** (étendus) entre eux

  ➡ Adressage d'un groupe d'hôtes (i.e. réseau)?



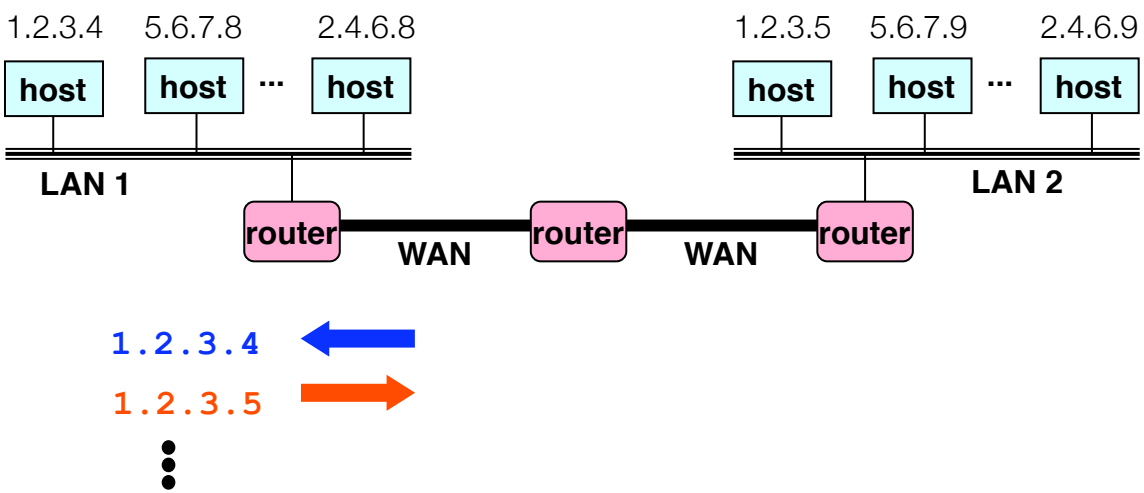
Problème : Le passage à l'échelle

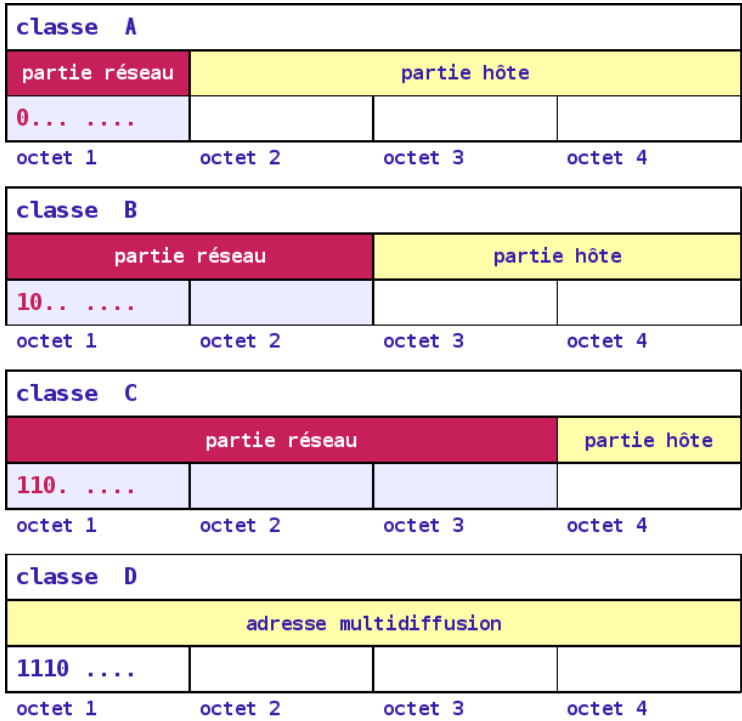
- **Objectif :** Limiter le nombre d'informations figurant dans les tables



Problème : Le passage à l'échelle

- **Objectif :** Limiter le nombre d'informations figurant dans les tables
- **Comment ? :** Introduire de la hiérarchie
  - @ : Parties réseau et hôte
  - Pour acheminer : information relatives au niveau dans la hiérarchie





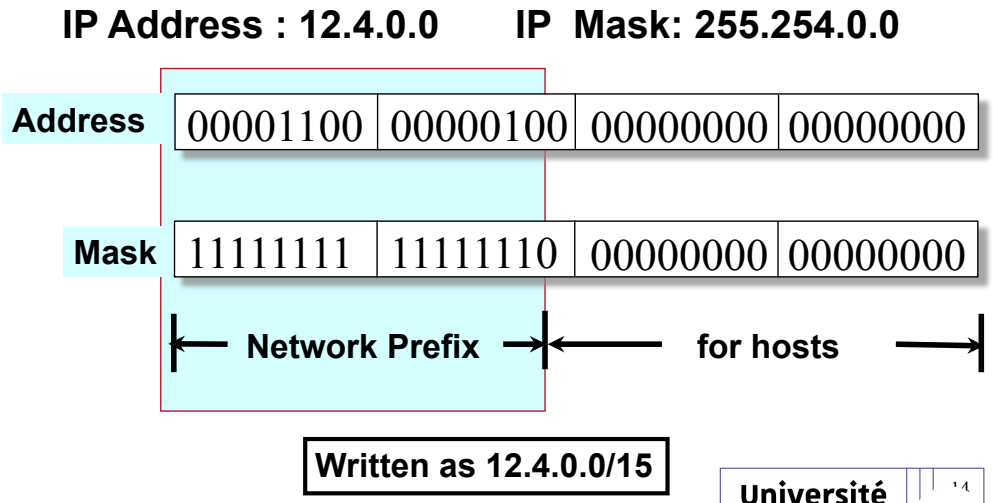
Allocation d'adresse

- **Jusqu'en 1994 :** Blocs de tailles fixes
  - Classe A (/8) : 0*.h.h.h -> Très grands blocs (2^24 machines)
  - Classe B (/16) : 10*.r.h.h -> Grands blocs (2^16 machines)
  - Classe C (/24) : 110*.r.r.h -> Petits blocs (2^8 machines) -> les + utiles en pratique
  - Classe D : 1110* : Groupes Multicast
  - Classe E : 11110* : Expérimentations

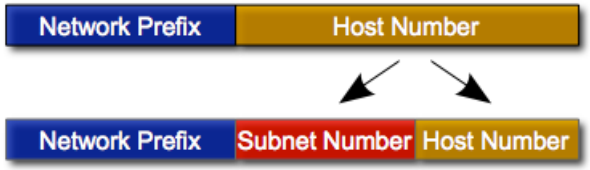
Allocation d'adresse

- **Jusqu'en 1994 :** Blocs de tailles fixes
  - Classe A (/8) : 0*.h.h.h -> Très grands blocs (2^24 machines)
  - Classe B (/16) : 10*.r.h.h -> Grands blocs (2^16 machines)
  - Classe C (/24) : 110*.r.r.h -> Petits blocs (2^8 machines) -> les + utiles en pratique
  - Classe D : 1110* : Groupes Multicast
  - Classe E : 11110* : Expérimentations

- **Apparition de CIDR en 1994**
  - Préfixes variables
  - Meilleure flexibilité :)
  - Table de routage plus grande :(



Notion de sous-réseaux



- Adresse découpée en 3 parties :
  - ✓ **réseau *R* | sous-réseau *S* | hôte *H***
- permet un découpage plus fin : sous-réseau de 2^i adresses
- découpage ne peut être «deviné» à partir de l'adresse
- nécessite un **masque** de (sous-)-réseau : ***netmask***
  - ✓les bits à 1 désignent réseau + sous-réseau : R+S
  - ✓ceux à 0 le numéro de l'hôte dans le sous-réseau : H
  - masque formé de *n* « 1 » contigus à gauche
    - ✓peut être représenté par un préfixe

      A.B.C.D/23 ⇔ A.B.C.D 255.255.254.0
- masque ou préfixe doit être **configuré** pour chaque interface
  - ✓permet déterminer si une IP ∈ même sous-réseau

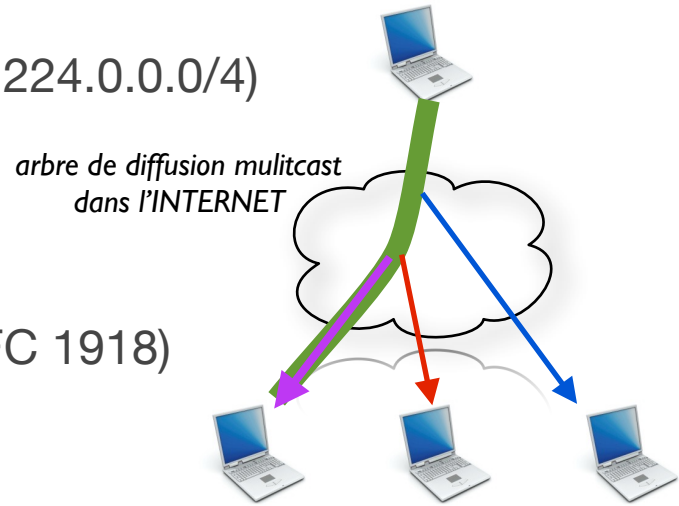
IP Subnet Masking

/24	/25	/26	/27	/28	/29	/30
...0	...128	...192	...224	...240	...248	...252
00000000	10000000	11000000	11100000	11110000	11111000	11111100
0-255	0-127	0-63	0-31	0-15	0-7	0-3
						4-7
					8-15	8-11
						12-15
				16-31	16-23	16-19
						20-23
					24-31	24-27
						28-31
			32-63	32-47	32-39	32-35
						36-39
					40-47	40-43
						44-47
				48-63	48-55	48-51
						52-55
					56-63	56-59
						60-63
		64-127	64-95	64-79	64-71	64-67
						68-71
					72-79	72-75
						76-79
				80-95	80-87	80-83
						84-87
					88-95	88-91
						92-95
			96-127	96-111	96-103	96-99
						100-103
					104-111	104-107
						108-111
				112-127	112-119	112-115
						116-119
					120-127	120-123
						124-127
	128-255	128-191	128-159	128-143	128-135	128-131
						132-135
					136-143	136-139
						140-143
				144-159	144-151	144-147
						148-151
					152-159	152-155
						156-159
			160-191	160-175	160-167	160-163
						164-167
					168-175	168-171
						172-175
				176-191	176-183	176-179
						180-183
					184-191	184-187
						188-191
		192-255	192-223	192-207	192-199	192-195
						196-199
					200-207	200-203
						204-207
				208-223	208-215	208-211
						212-215
					216-223	216-219
						220-223
			224-255	224-239	224-231	224-227
						228-231
					232-239	232-235
						236-239
				240-255	240-247	240-243
						244-247
					248-255	248-251
						252-255

Adresses spéciales

- **Broadcast 255.255.255.255 :** Toutes les machines du sous-réseau
  - Broadcast dans un (sous)-réseau spécifié : Tous les bits *hôte* mis à 1
    - A.B.90.0/23 ➥ A.B.91.255

- **Multicast :** @multicast = groupe de machines (ex : 224.0.0.0/4)
  - Nécessite un protocole d'adhésion (IGMP)



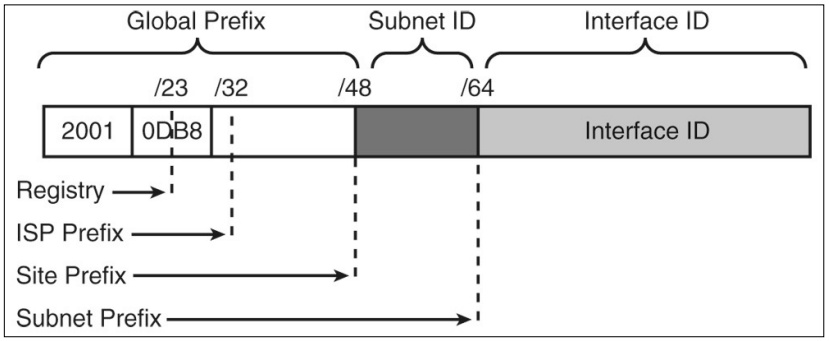
- **Adresses privées :** non routable publiquement (RFC 1918)
  - ex : 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - Ré-utilisation locales sur divers sites
- **Spécifiques au lien (*local link*) :** Non routables à travers internet
  - Adresses configurées automatiquement

Adressage IPv6

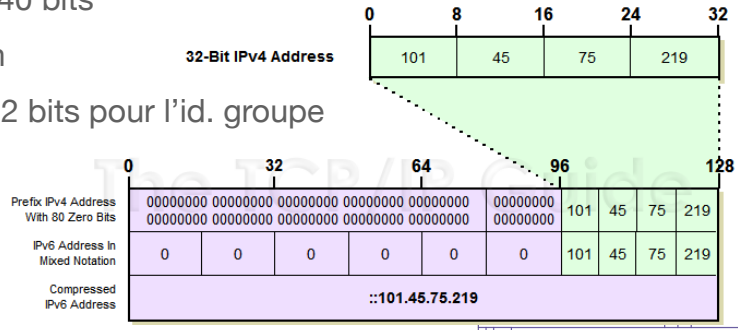
IPv4: 10.1.2.3

IPv6: 2002:0A01:0203::/48

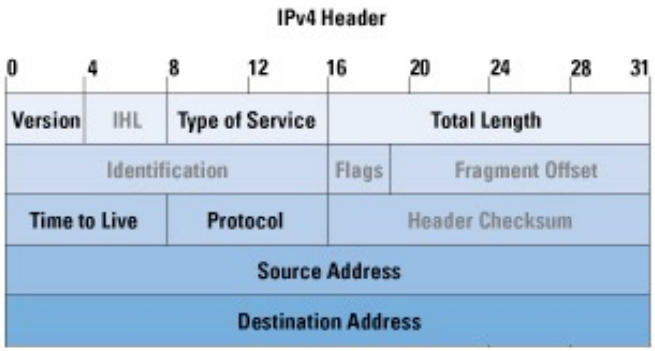
- Codée sur 128 bits (667 millions de milliards d'adresses par millimètre carré de surface terrestre !)
  - ✓notation hexadécimale 2001:0db8:0000:85a3:0000:0000:ac1f:8001
  - ✓possibilité de compresser l'écriture 2001:db8:0:85a3::ac1f:8001
    - ➡«::» : pour désigner la plus longue suite de 0
    - ➡«0» : pour résumer une suite de 0 consécutif sur 2 octets *(entre deux «:» consécutifs)*

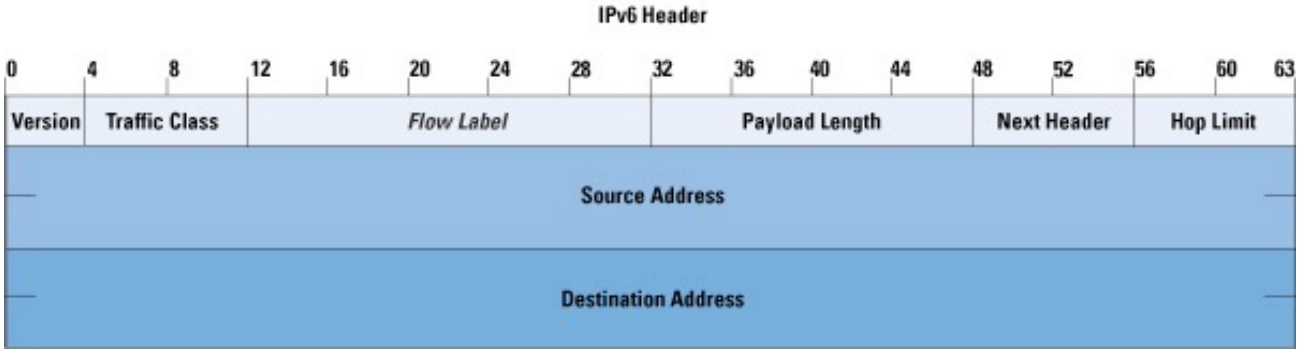


- 4 familles d'adresses
  - ✓unicast globale **2000::/3 :** «ouverte à la réservation» **2001::/16** , «6to4» **2002::/16 , ...**
  - ✓locale unique **fc00::/7 :** ~ adresses privées IPv6, ID globale sur 40 bits
  - ✓lien local **fe80::/10 :** /64 en pratique (54 zéros), unique sur un lien
  - ✓multicast **ff00::/8** : remplace le broadcast (ARP v4 ➡ NDP v6), 112 bits pour l'id. groupe
  - ✓+ @ réservées **::/8**
    - ➡::/128 : non spécifié, quand l'hôte ne connait pas son IP
    - ➡::1/128 : loopback
    - ➡::ffff:0:0/96 : pour contenir une IPv4 dans une structure IPv6



Format paquet : de IPv4 à IPv6





- ✓ Augmentation de 2^{32} (~ 4×10^{9}) à 2^{128} (~ $3{,}4\times10^{38}$) du nombre d'adresses IP
- ✓ Mécanismes de configuration et de renumérotation automatique
- ✓ IPsec, QoS et le multicast : spécification native d'IPv6
- ✓ Simplification des en-têtes facilite le routage.

© Pascal Kenngl

Acheminer les paquets

- **Chaque routeur possède une table de routage**
  - @ destination <-> interfaces sortantes du routeur

- **Réception d'un paquet**
  - Extraire l'@ destination de l'entête IP
  - L'indexer dans la table
  - Déterminer l'interface sortante
  - Réexpédier le paquet par cette interface

- **Routeur suivant, etc.**

Le routage : de la RIB vers la FIB

- Dans l'Internet d'aujourd'hui en pratique :
  - ➡chaque routeur (re)construit en permanence sa RIB puis sa FIB associée
  - ➡un paquet est commuté au saut par saut (*hop by hop*)

    interrogation des FIB successives

    ➥ aucun routeur ne connaît la route exacte d'un paquet

    => risque boucle ? ➥ TTL

  ✓Si changement de RIB

    ➡2 paquets successifs (même flux <SRC,DST,portS,portD,...>)

    ➡routes différentes

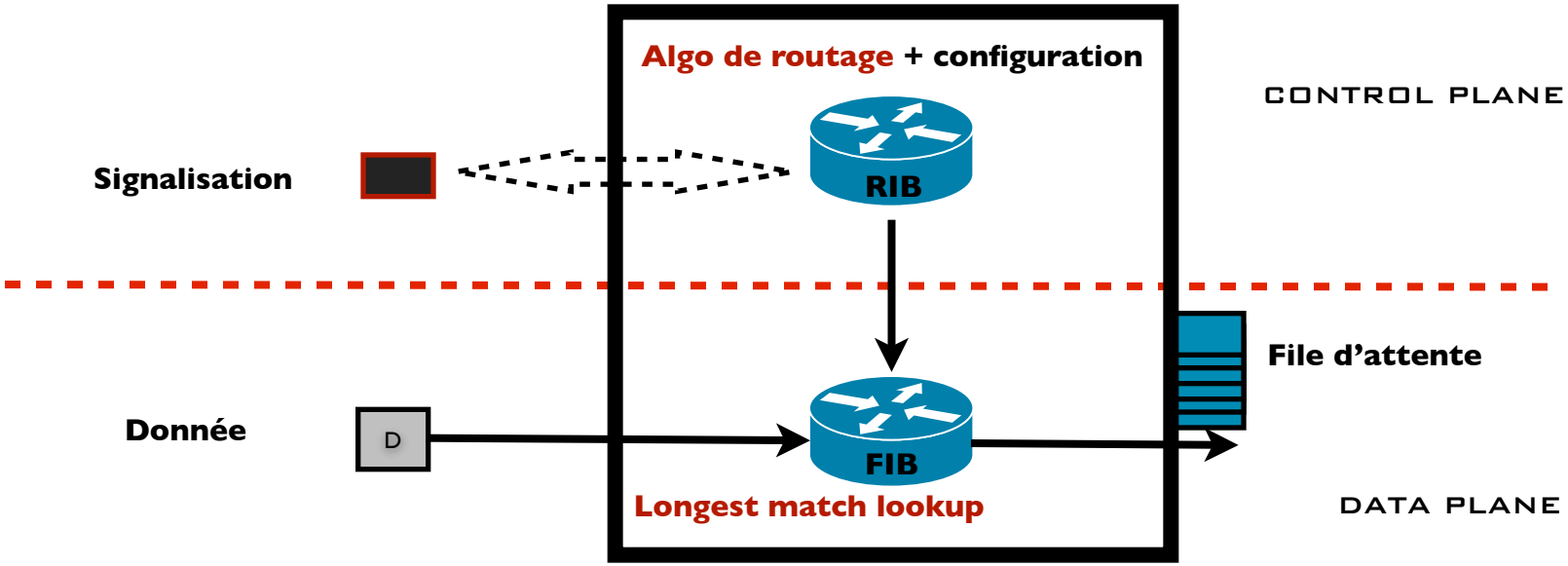
  ✓Si file d'attente (droptail, RED, DRR, WFQ, CBQ ...) pleine ou politique spécifique => paquet jeté :(

  ✓Si TTL dépassé || paquet trop gros pour sortie (et option DF) || pas de route

    ➡ paquet jeté + erreur ICMP

Acheminer les paquets

- **1) Obtention d'informations :** Topologie, ressources (ex : débits des liens)
- **2) Sortie pour un préfixe** : Table de routage (RIB : Routing information base)
- **3) Acheminement** : Table de commutation (FIB : Forwarding Information Base)



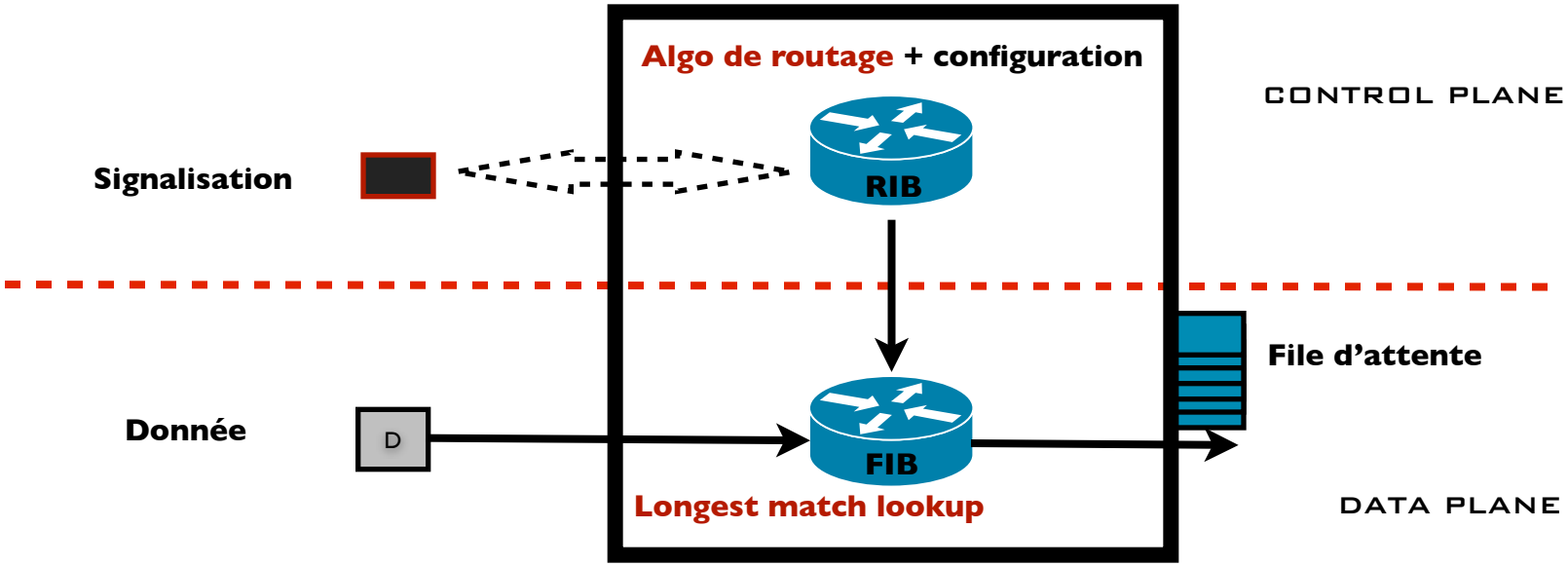
- **Plan de données**
  - 4) commuter les paquets !

© Pascal Mérindol

Acheminer les paquets

- **1) Obtention d'informations :** Topologie, ressources (ex : débits des liens)
- **2) Sortie pour un préfixe** : Table de routage (RIB : Routing Information base)
- **3) Acheminement** : Table de commutation (FIB : Forwarding Information Base)

Plan de contrôle / signalisation :
Protocoles de routage



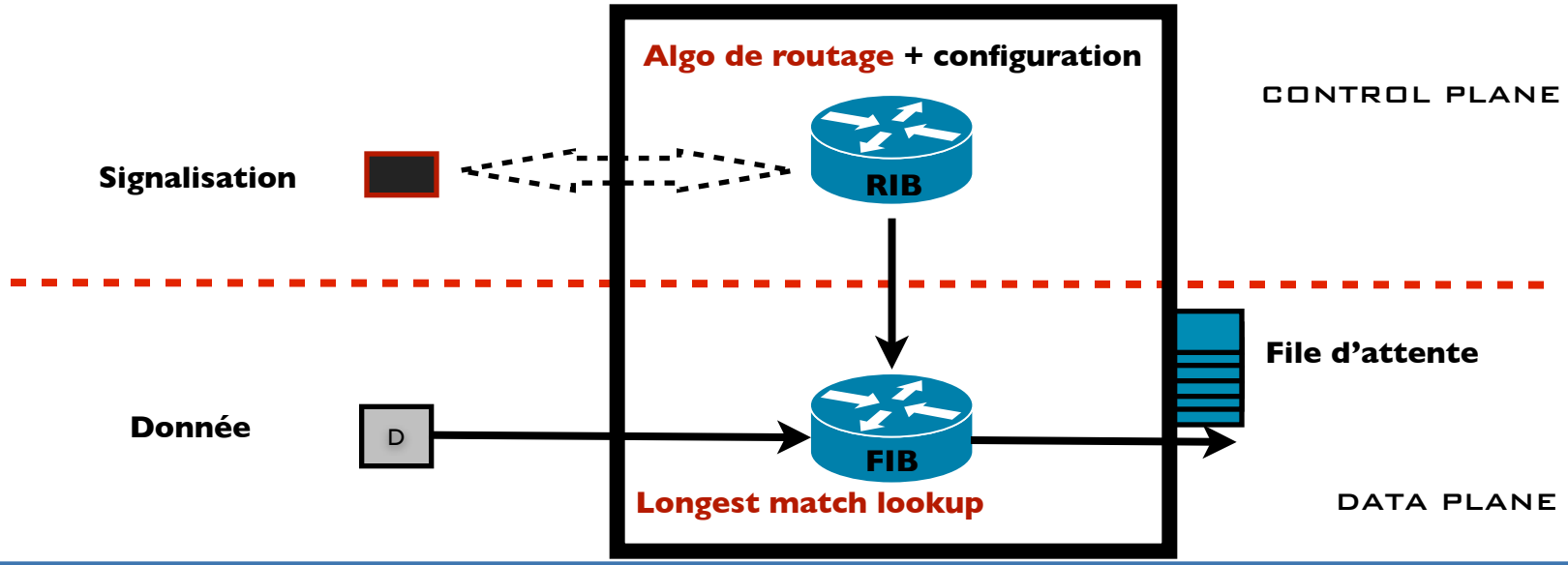
© Pascal Mérindol

- **Plan de données**
  - 4) commuter les paquets !

Acheminer les paquets

- **1) Obtention d'informations :** Topologie, ressources (ex : débits des liens)
- **2) Sortie pour un préfixe** : Table de routage (RIB : Routing Information base)
- **3) Acheminement** : Table de commutation (FIB : Forwarding Information Base)

Plan de contrôle / signalisation :
Protocoles de routage

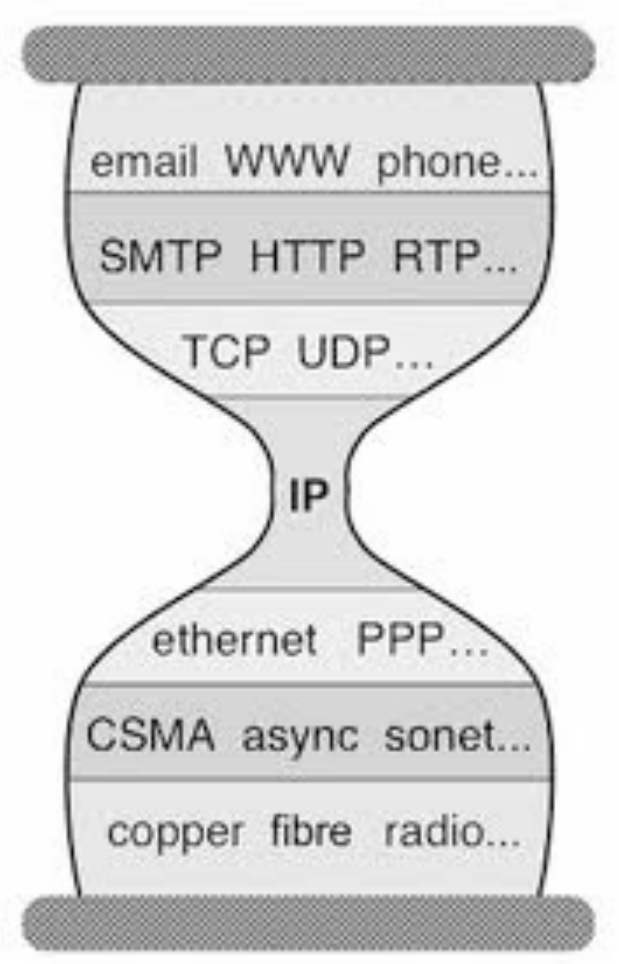


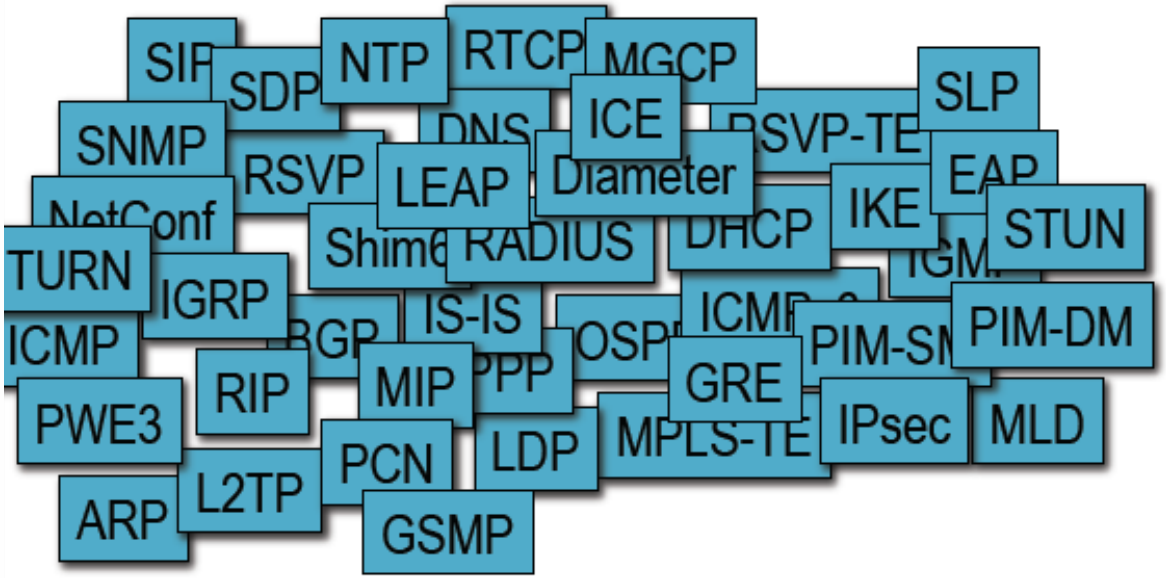
© Pascal Mérindol

- **Plan de données**
  - 4) commuter les paquets !

Plan de données / commutation

Data plan vs. Control plan

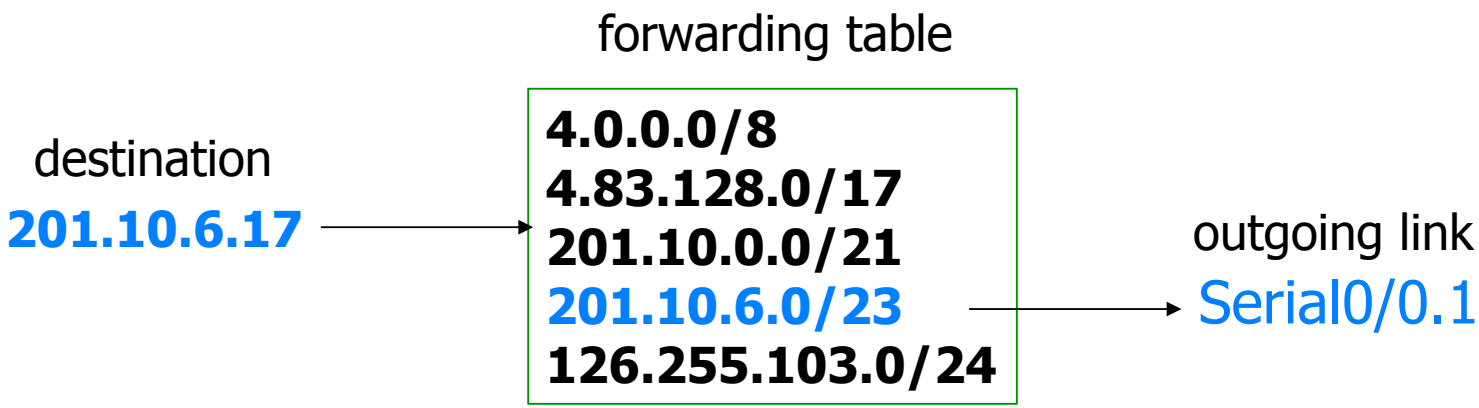




Commuter les paquets = *lookup FIB*

- **1 entrée / @ IP :** Chercher l'entrée correspondante à l'@ dans la table
- **1 entrée / préfixe /24 :** Chercher l'entrée correspondante aux 24 1ers bits
- **1 entrée par classe d'@ :** Repérer en fonction des 1ers bits (A=0*, B=10*, etc.)

- **Avec CIDR ?**... IP 201.10.6.17 couverte par 201.10.0.0/21 et 201.10.6.0/23
  - ➡ Associer le plus long/spécifique préfixe



Lookup FIB

- **Contrainte : Rapidité de la recherche**
  - Paquet de 1000 bits sur réseau 10Gb/s < 100 nanosec
  - ➡ Trouver l'interface de sortie avant l'arrivée du paquet suivant

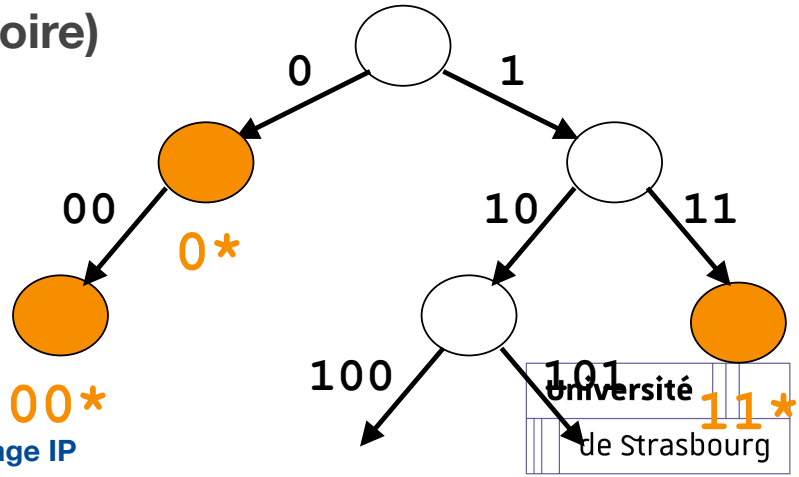
- **Principe du *Longest Prefix Match***
  - Pour une @ dest D : trouver plus long préfixe P correspondant à D
  - Plus long = plus spécifique
  - ➡ Plus complexe qu'un exact match
  - ➡ Arbres binaires - radix/patricia tree - bloom filter - architecture physique/logiciel dédiée ...

Performances FIB

- **Algorithme naïf :** Scan des entrées de la table une à une
  - ➡ Complexité linéaire
  - ➡ Rappels : ~ 300,000 entrées et quelques nanosecondes par paquet

- **Algorithmes évolués :** Organisation de la table selon un arbre
  - *Patricia tree :* 1 bit/niveau -> certains noeuds de l'arbre sont **valides**

- **Améliorations matérielles (ex : accès mémoire)**



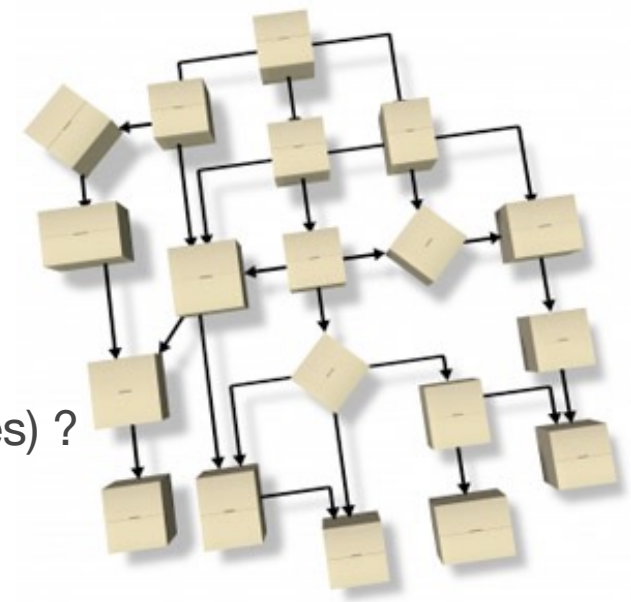
Le routage ~ Construire la RIB avec de la signalisation

- **Objectifs :**
  - Assurer l'acheminement des paquets (de @src à @dest, sans boucle)
  - Minimiser les délais de transmission
  - Maximiser les débits
  - Prévenir les congestions
  - Assurer l'équité entre utilisateurs (ou QoS)
  - Optimiser utilisation des ressources du réseau
  - Gérer les défaillances du réseau (pannes liens/routeurs ...)
  - S'adapter aux modifications du réseau (topologique + charge ?)
  - Stabilité (éviter les oscillations)

Choix d'une métrique

- **«Opposition» de certains critères**
  - Equité et optimisation
  - max-min, min-max, max-min-fair_share ...

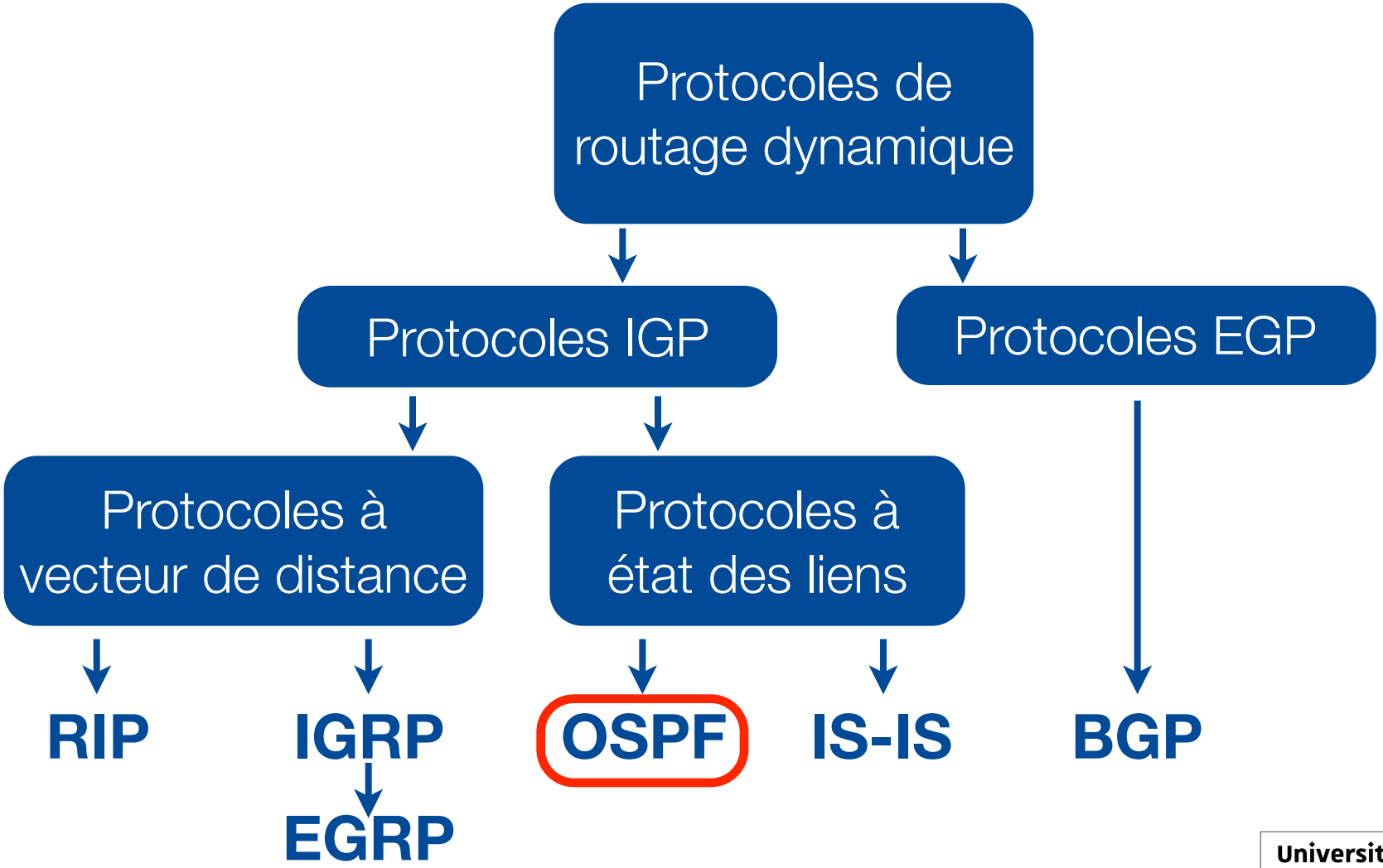
- **Compromis :**
  - Pondération entre métriques ?
  - Minimiser le nombre de sauts (#routeurs traversés) ?
  - ...



Routage intra-domaine ~ trouver les plus court chemins

- **Selon quelle métrique ?** Statique ou dynamique voire adaptative ?
  - Statiques : indépendantes de l'état du réseau ! => routes statiques
  - Dynamiques = dépendantes de l'état du graphe : sauts, capacités, coûts, etc.
  - Adaptatives = dépendantes de l'état résiduel du réseau : charge, délais, etc.
  - dynamique/adaptatif => RIB/FIB adaptées (et messages de signalisation)
- **De quelle façon ? (~ quelle signalisation & quel algorithme ?)**
  - Vecteur de distances : RIP (IETF), IGRP et EIGRP (Cisco)
    - Diffusion des adjacences (+distances associés) => *Bellman-Ford* distribué
  - Vecteur de chemins : BGP (IETF)
    - Tie-break sur le chemin sans boucle le plus court vers chaque AS
  - Etat des liens (EL) : OSPF (IETF), IS-IS (ISO)
    - Innondation EL => Base de données (graphe dirigé et valué) => *Dijkstra*

Routage IP : Classification



Algorithme vs. Protocole ?

Ingrédients (pour 4 personnes) :
- 1 petit chou rouge ou la 1/2 d'un gros
- 50 g de margarine
- sel et poivre
- 1 cuillère à soupe de vinaigre
- 2 pommes (reinettes idéalement)
- un peu de noix de muscade râpée
- 1 cuillère à soupe de cassonade
- 1/2 verre d'eau

Préparation :
Couper le chou rouge en fines lamelles et ôter les plus grosses côtes.
Les placer dans l'évier rempli d'eau et laver 2 fois. Egoutter dans une passoire.
Dans la cocotte-minute faire fondre la margarine à feu doux, puis ajouter le chou, sel et poivre. Activer la cuisson et mélanger quelques minutes.
Ajouter le vinaigre, les pommes coupées en lamelles, un peu de muscade, la cassonade. Enfin l'eau.
Fermer la cocotte. Quand la pression est atteinte, cuire à feu doux 25 minutes.

Routage à état des liens, l’exemple OSPF

NETWORK RESEARCH TEAM

Routage intra-domaine à état des liens

- **Objectif :** Vision «globale» du réseau => arbre/DAG enraciné
- **Exemples :** OSPF (IETF), IS-IS (ISO) et PNNI (routage dans ATM)

- **Sur chaque routeur**
  - **Etat local :** @ locale, liens sortants et poids associées
    - **Poids des liens :** bande passante, délai, combinaison, etc.
  - **Signalisation (messages) :** Hello, Keep alive, LSA (Link State Advertissement).
  - **Graphe +/- partiel du réseau :** Base de données des états des liens
  - **Algorithme :** *Dijkstra*
  - **Table de routage :** Commutation niveau IP (data plan)

Routage intra-domaine à état des liens

- **Changement d'état d'un lien local**
  - Diffusion de proche en proche (flooding) du nouvel LS à tout le réseau
    - Fiabilité : Transfert état des liens acquittés, numéro de séquence
    - LSDB «synchrone» sur les routeurs (~ temps de diffusion/traitement des LS)

- **Réception d'une annonce d'état de lien (*Link State Advertisement* - LSA)**
  - Calcul des plus courts chemins du routeur vers le reste du «monde»
    - Dijkstra : Arbre des plus court chemins (SPT) enraciné sur le routeur
    - Si ex-aequo alors lecture lexicographique des id. ou ECMP

- **Chemin vers D :** prochain saut = 1er fils dans le SPT conduisant à D

Open Shortest Path First : Historique

- **1987 :** Formation du groupe de travail à l'IETF
- **1990 :** OSPF version 1 http://tools.ietf.org/html/rfc1131
- **1991 :** OSPF version 2 http://tools.ietf.org/html/rfc1247 → http://tools.ietf.org/html/rfc1583
- **1992 :** Recommendation de l'IESG pour OSPF comme IGP dans l'Internet
  http://tools.ietf.org/html/rfc1371

```
This memo presents motivation, rationale and other surrounding
background information leading to the IESG's recommendation to the
IAB for a single "common IGP" for the IP portions of the Internet.
```

- **1997 et 1998 :** MAJ de OSPF version 2
  http://tools.ietf.org/html/rfc2178 → http://tools.ietf.org/html/rfc2328
- **1999 :** OSPF version 3 (IPv6)
  http://tools.ietf.org/html/rfc2740 → http://tools.ietf.org/html/rfc5340

OSPF : Les bases du protocole

- **Différents types de «sous-réseaux» :** feuilles ou transit
- **Distinction des adjacences :** point à multi-point ou point à point

- **Messages *hello* :** à intervalles réguliers -> adjacences
- **Etats des liens :** *Link-state advertisements* (LSA) -> Diffusion de proche en proche
- **Base de données distribuée du réseau :** *Link-State Database* (LSDB)
- **Changement topologique :** Envoi de LSA et re-calcul des plus courts chemins

- **Calcul des plus courts chemins :** Algorithme de *Dijkstra*
  - Shortest Path First (SPF)

  ➡ Sur un routeur : Arbre/DAG des plus courts chemins enraciné sur lui-même

OSPF : plusieurs types d'adjacences

- **Point-to-point :** Le sous réseau correspond à un lien point à point (deux routeurs liés directement)

- **Broadcast multi-access (point-to-multipoint) :** Le sous-réseau comporte plus de deux routeurs qui peuvent tous communiquer entre eux, une adresse broadcast est disponible. Par exemple, il peut s'agir d'un switch Ethernet central : une telle connectivité est traité comme une collection de liasons point à point.

- **NBMA (Non-broadcast multi-access) :** Le sous-réseau est constitué de routeurs qui peuvent communiquer entre eux mais il n'existe pas d'adresse broadcast. Dans ce type de réseau, OSPF émule le type broadcast en répliquant les LSA à tous les voisins adjacents

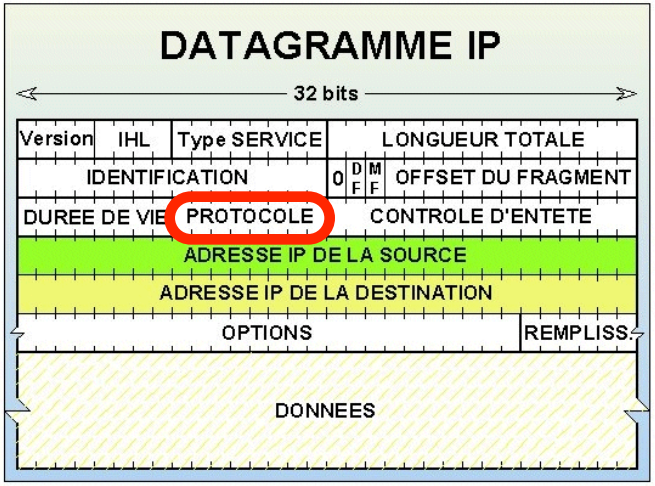
OSPF : Le protocole

- **Juste au-dessus d'IP (protocole n° 89)**

- **Diffusion multicast de la signalisation**
  - 224.0.0.5 : Tous les routeurs d'un lien
  - 224.0.0.6 : Tous les routeurs désignés (DR) et de secours (BDR)

- **Messages**
  - **Hello :** établissement et maintenance des adjacences, élections DR et BDR
  - **Database description :** Diffusion de la topologie entre routeurs
  - **Link state :** Requêtes, MAJ et acquittements des LSA

Rappels sur l'encapsulation : Paquet IP



✓Entête obligatoire = 20 octets
✓+ Options facultatives
✓+ Données (total < 2^{16} octets : 64K)

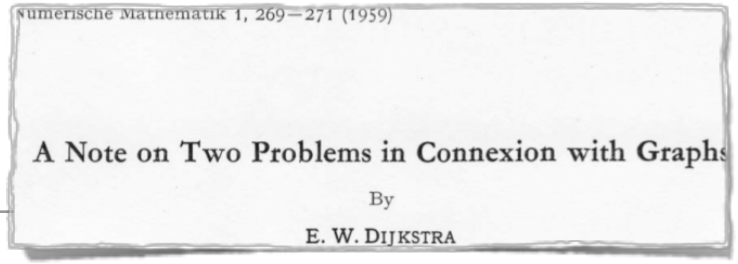
➡ Fonctions IP ⇔ champs de l'entête

Format de l'entête (IP header)

	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31
1	version	Lg_ent	ToS/DSCP,	ECN	Longueur totale			
2	identification				Flags (3b), Offset fragment (13 bits)			
3	TTL		protocole		total de contrôle entête			
4	Adresse IP source							
5	Adresse IP destination							
6+	Eventuelles options + bourrage : N * 32 bits, N ≥ 0							

© Pascal Terrioll

OSPF : L'algorithme de Dijkstra


Numerische Mathematik 1, 269—271 (1959)

A Note on Two Problems in Connexion with Graphs

By

E. W. DIJKSTRA

```
begin integer array from, to [1:N-1]; real array length [1:N-1]; real len, minlen; integer k, h, minh, p;
      COLOUR ONE POINT RED AND THE REMAINING ONES BLUE:
            k:= 1;
      CONSTRUCT THE SET OF ULTRAVIOLET BRANCHES:
            h:= 1; while h < N do begin from[h]:= N; to[h]:= h; length[h]:= distance[N,h]; h:= h + 1 end;
      while k < N do
      begin SELECT SHORTEST ULTRAVIOLET BRANCH AND CALL ITS BLUE ENDPOINT P:
                  minh:= k; minlen:= length[k]; h:= k + 1;
                  while h < N do begin len:= length[h]; if len < minlen do begin minlen:= len; minh:= h end; h:= h + 1 end;
                  p:= to[minh];
            COLOUR IT AND POINT P RED:
                  if k ≠ minh do begin h:= from[k]; from[k]:= from[minh]; from[minh]:= h;
                                       h:= to[k]; to[k]:= to[minh]; to[minh]:= h;
                                       len:= length[k]; length[k]:= length[minh]; length[minh]:= len
                              end;
                  k:= k + 1;
            ADJUST FOR EACH BLUE POINT B BY COMPARING WITH THE BRANCH BP:
                  h:= k;
                  while h < N do
                  begin len:= distance[p, to[h]]; if len < length[h] do begin length[h]:= len; from[h]:= p end; h:= h + 1 end
      end;
      h:= 1; while h < N do begin print(from[h]); print(to[h]); newline; h:= h + 1 end
end
```

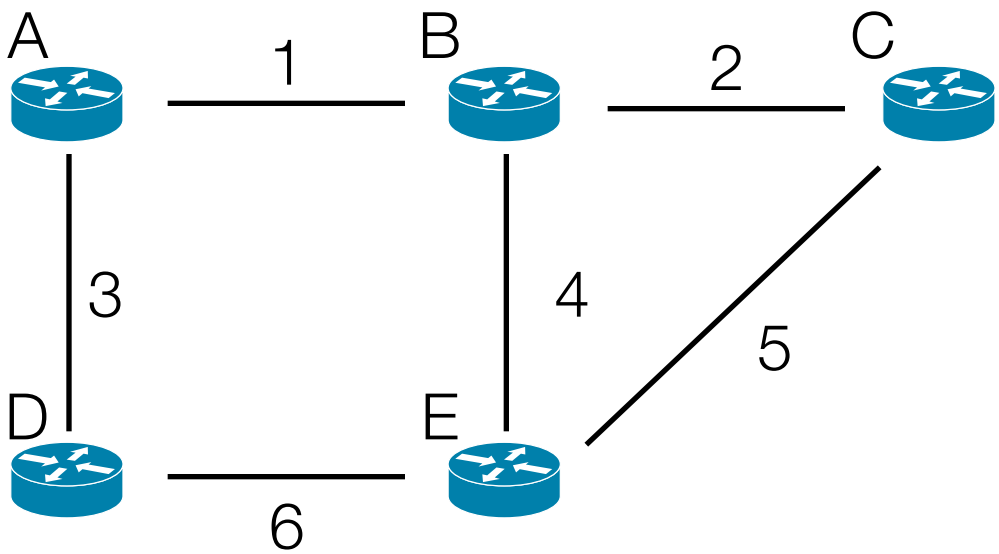
OSPF : L'algorithme SPF de Dijkstra

- **Shortest Path First :** *E. W. Dijkstra*

```
function Dijkstra(Graph, source):
    for each vertex v in Graph:                 // Initializations
        dist[v] := infinity ;                   // Unknown distance function from source to v
        previous[v] := undefined ;              // Previous node in optimal path from source
    end for ;
    dist[source] := 0 ;                         // Distance from source to source
    Q := the set of all nodes in Graph ;
    // All nodes in the graph are unoptimized - thus are in Q
    while Q is not empty:                       // The main loop
        u := vertex in Q with smallest distance in dist[] ;
        if dist[u] = infinity:
            break ;                             // all remaining vertices are inaccessible from source
        end if ;
        remove u from Q ;
        for each neighbor v of u:               // where v has not yet been removed from Q.
            alt := dist[u] + dist_between(u, v) ;
            if alt < dist[v]:                   // Relax (u,v,a)
                dist[v] := alt ;
                previous[v] := u ;
                decrease-key v in Q;            // Reorder v in the Queue
            end if ;
        end for ;
    end while ;
    return dist[] ;
end Dijkstra.
```

OSPF : LSDB

- **Message de A vers C**
  - au saut par saut : A puis B



A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

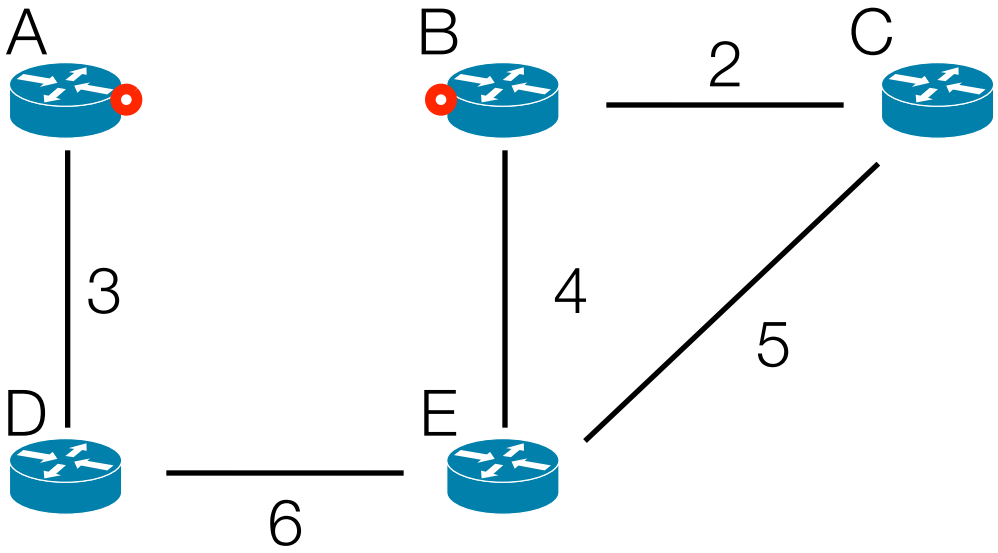
OSPF : Diffusion des LSA de proche en proche

- **Rupture d'un lien :** Détectée par A et B
  - Envoi d'un message de MAJ de A vers D :

A	B	1	∞

  - Envoi d'un message de MAJ de B vers C et E :

B	A	1	∞



OSPF : Innondation des LSA (flooding)

- **Distinction de messages :** Ajout d'un numéro de séquence
- **Réception du message :** Recherche de la ligne correspondante dans la LSDB

- **Si pas de ligne :** Ajout et «broadcast*» du message
- **Si ligne :** Comparaison des numéros de séquence
  - Si LSDB < MSG : MAJ de la LSDB et broadcast du message
  - Si LSDB > MSG : message sur l'interface d'entrée avec valeur de la LSDB
  - Si LSDB = MSG, NOP

- **Remarque :** «broadcast*» = toutes les interfaces sauf celle d'entrée du LSA

OSPF : *Bringing up adjacencies*

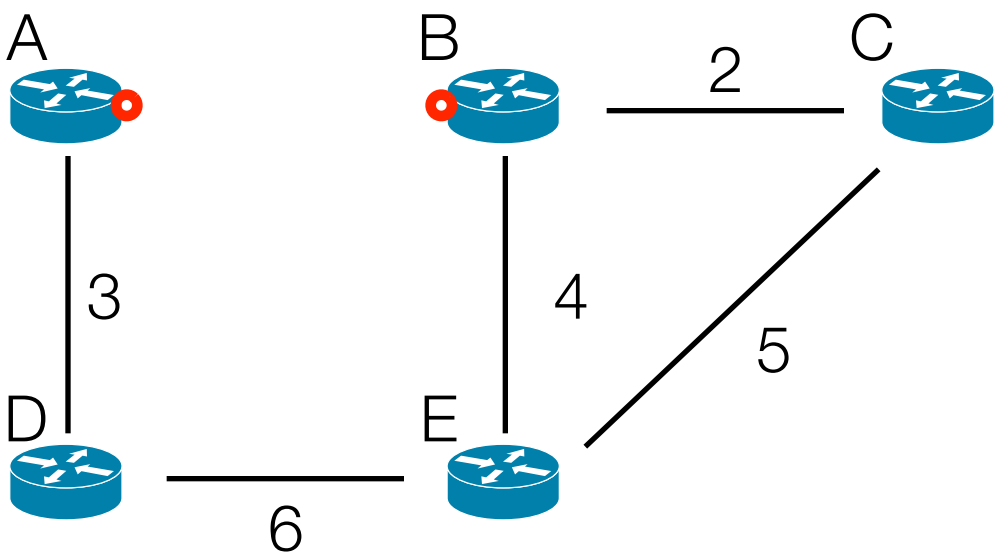
- **Conservation des liens avec numéro le plus récent**
  - Paquets *data description*

- **Synchronisation**
  - Etape 1 : Echange des paquets *data description*
    - ➡ Liste des entrées "intéressantes" (lien avec numéro ≠, lien inexistant)
  - Etape 2 : Demande aux routeurs voisins leurs listes respectives
    - Paquets *link state request*
      - ➡ Entrées mises à jour et broadcast normal

OSPF : Flooding

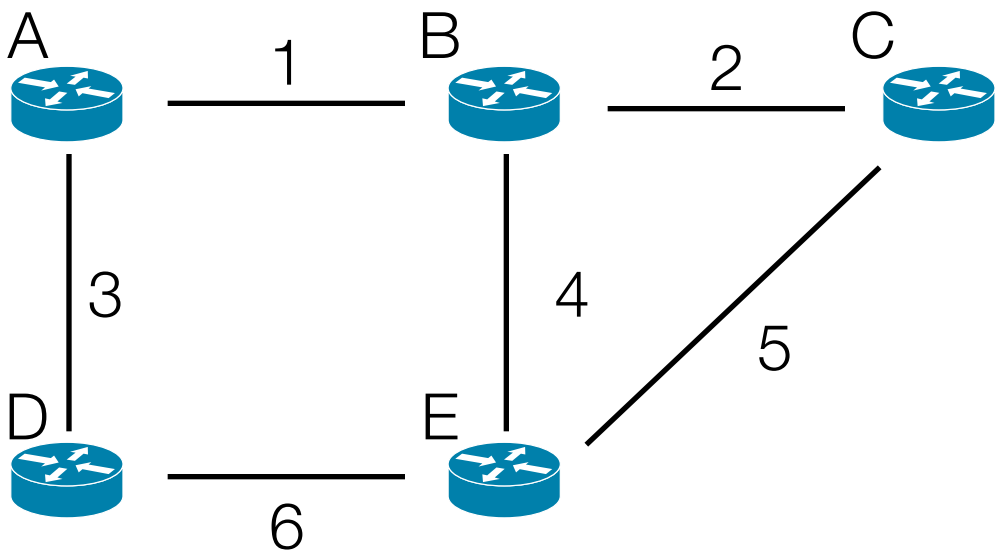
A	B	1	∞	**2**

- **Message de A :** ajout d'un numéro
  - D reçoit, MAJ sa LSDB et transfère à E
  - E reçoit, MAJ sa LSDB et transfère à B et C
  - B et C reçoivent, MAJ leurs LSDB et renvoient : ni C, ni B, ni E ne font de MAJ



OSPF : LSDB

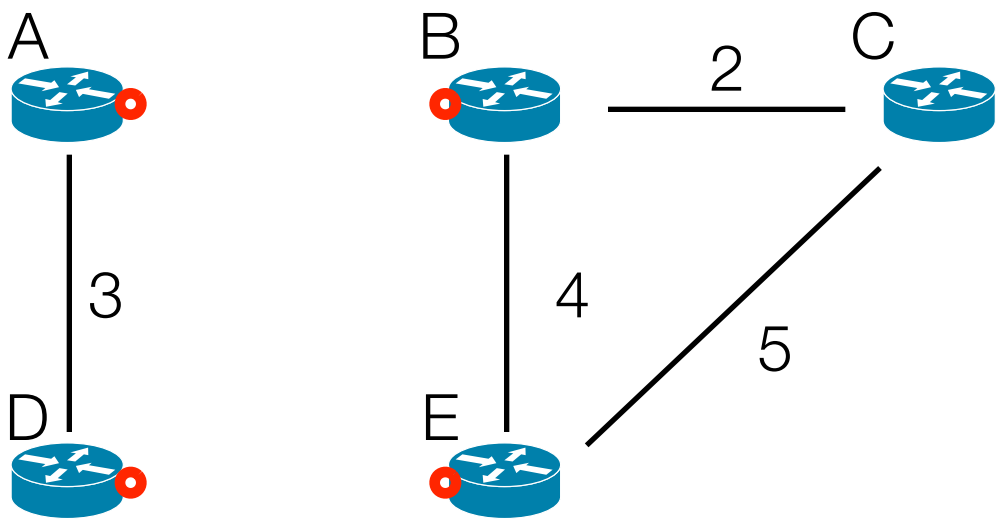
- **B a aussi envoyé son message :** LSDB MAJ



A	B	1	∞	**2**
A	D	3	1	2
B	A	1	∞	**2**
B	C	2	1	2
B	E	4	1	2
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

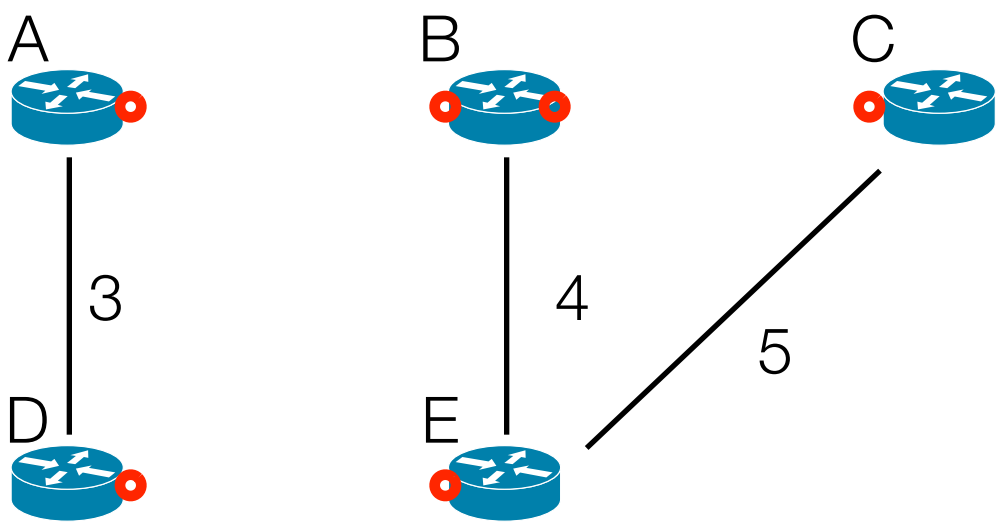
OSPF : Joindre les deux bouts

- **Ruptures de 2 liens :** Détectées par A, B, D et E
  - Création de 2 LSDB différentes (A et D d'un côté, B, C et E de l'autre)
  - A et D détectent que B, C et E ne sont plus joignables et réciproquement



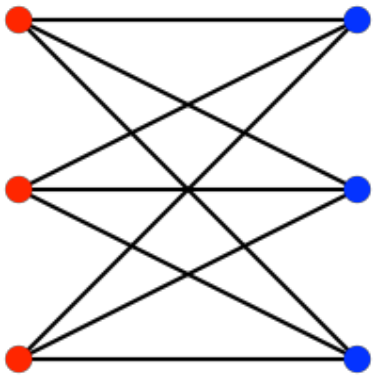
OSPF : Joindre les deux bouts

- **Problème :** Le lien 2 disparaît également
  - La LSDB côté, B, C et E change



A	B	1	∞	**2**
A	D	3	1	1
B	A	1	∞	**2**
B	C	2	∞	**2**
B	E	4	1	1
C	B	2	∞	**2**
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	∞	**2**

Spécifités d'OSPF : 2 types de noeuds



- **Routeur ou Réseau => graphe biparti**
- **Lien entre 2 noeuds orientés**
  - Bidirectionnel : Chaque direction annoncée par un routeur différent
  - Etat du lien annoncé par
    - Un routeur : origine du lien si l'origine est un routeur
      - LSA type routeur, poids positif
    - Un routeur désigné (élu) du réseau si l'origine est un réseau
      - LSA type réseau, poids nul

Spécifités d'OSPF : type réseau et type routeur

- **Chaque routeur**
  - Maintient un **LSA de type routeur**
    - Liste de tous les liens adjacents (sortant) du routeur
      - Interfaces physiques : lien vers LAN
      - Vue logique, *Hello* : lien P-à-P vers autre routeur
  - Pour chaque réseau dont il est responsable :
    - il maintient un **LSA de type réseau**
      - Liste de tous les routeurs connectés à ce réseau
        - Appris grâce messages *Hello*

Spécifités d'OSPF : Routage par zone

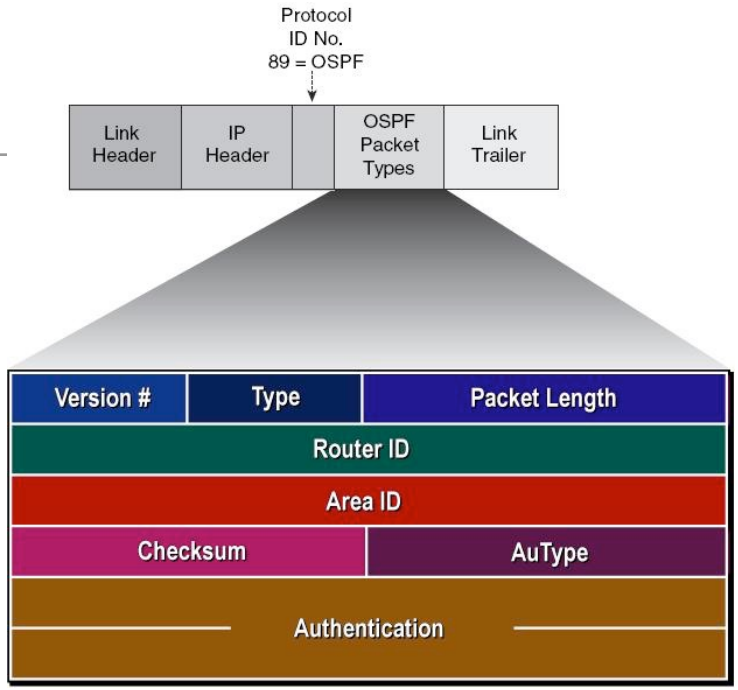
- **Diviser les grands réseaux en zones**
  - Diminution de la LSDB d'une zone
  - Diminution des coûts de calcul : MAJ suite à un changement dans la zone
    - Dijkstra : $O(n^2 + m)$, (n : #routeurs, m : #liens)

- **Notion d'aire**
  - Une aire *backbone* (aire 0) : tout le trafic *inter-area* y passe
  - Calculs de routes indépendants dans chaque aire : 1 LSDB/aire
  - Export d'agrégation de routes vers les autres aires

- **+ Routage externe ~ inter-action BGP**

OSPF : Format des paquets

OSPF version	OSPF pkt type
OSPF length	
Source OSPF Router ID	
OSPF Area ID	
Packet Checksum	
Authentication type	
Authentication data	



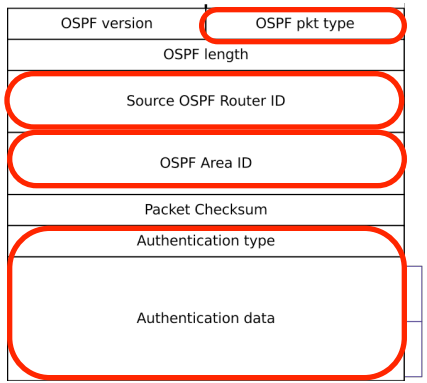
OSPF : 5 messages de signalisation

- **Type**
  - Hello : Découverte des voisins et maintien des adjacences
  - Database Description (DBD) : Description des LSA
  - Link State Request (LSR) : Requête d'un ou + LSA
  - Link State Update (LSU) : Mise à jour d'un ou + LSA
  - Link State Acknowledgement (LSAck) : Acquittement d'un LSU

Etablissement et maintient des adjacences

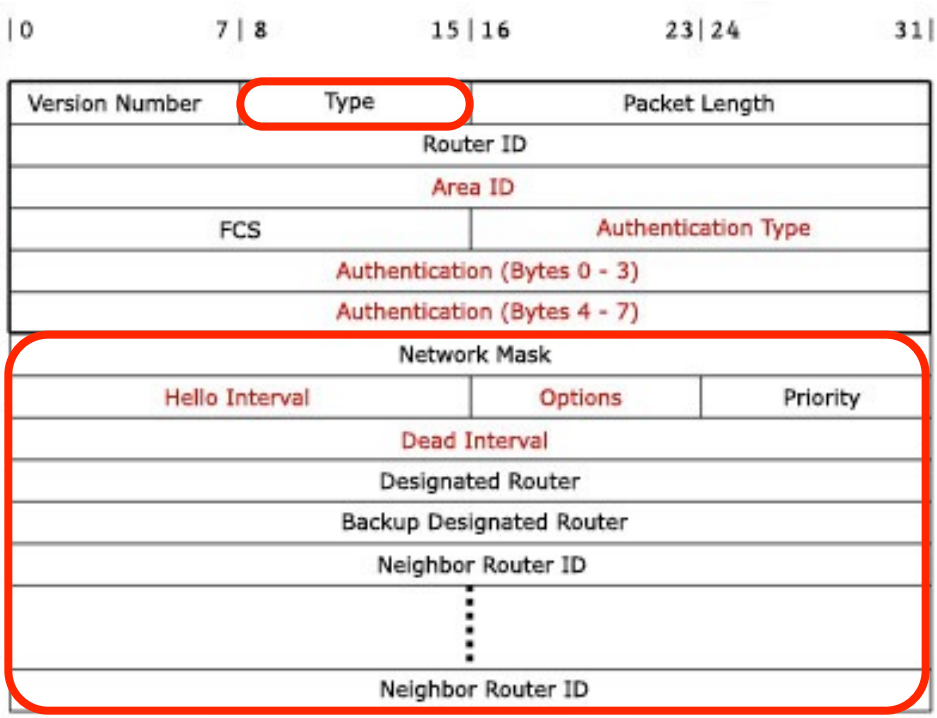
Synchro LSDB

- **Source OSPF router ID :** Plus petite @IP du routeur
- **OSPF Area ID :** Défini par l'administrateur (*backbone* -> 0)
- **Authentication :** Idem que RIPv2



OSPF : Messages *Hello*

- **Envois périodiques :** Période *Hello_Interval* (par défaut 10s pour un LAN)
- **Multicast :** ***ALL-OSPF-ROUTERS*** = 224.0.0.5
- **Options :** Bit E
  - Si capable de gérer LSA hors-AS
  - Stub area : E=0
- **Priority :** Election DR (0 si hors-jeu)
- **Router dead interval**
  - 40s
- **DR/BDR :** 0.0.0.0 si inconnu
- **Contenu :** IDs des voisins connus



Hello message OSPFv2

OSPF : Messages *DataBase Description* (DBD)

- **Interface MTU :** Taille max sur l'interface
- **DD sequence number**
- **Relation Maître-esclave :** bit M/S (master si 1)

Interface MTU	0	0	0	0	0	0	I	M	M S
DD sequence number									
An LSA Header									
...									

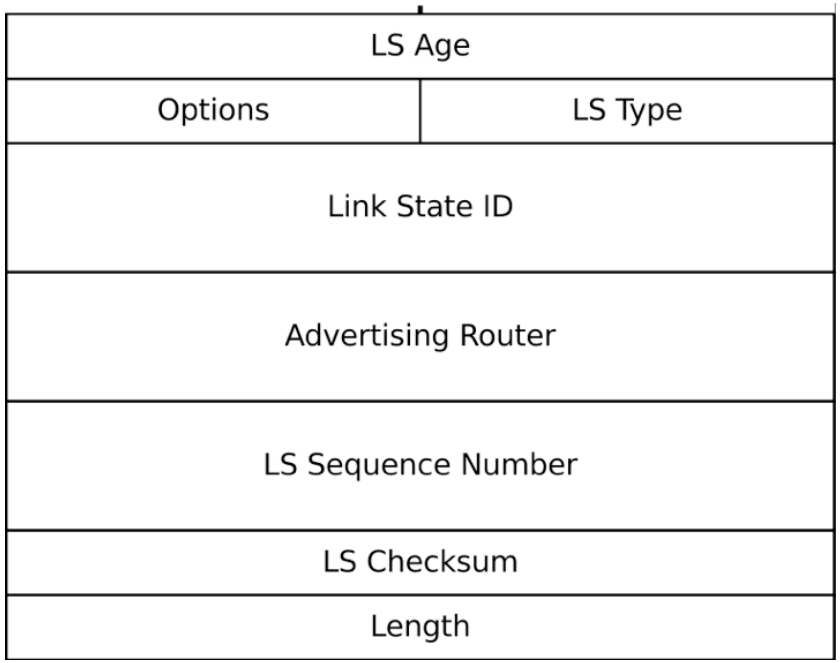
OSPF : Messages *DataBase Description* (DBD)

- **Interface MTU :** Taille max sur l'interface
- **DD sequence number**
- **Relation Maître-esclave :** bit M/S (master si 1)

Interface MTU	0	0	0	0	0	0	I	M	M S
DD sequence number									
An LSA Header									
...									

OSPF : *Header* LSA

- **LS age**
  - Temps écoulé (en sec) depuis l'"origine" du LSA
- **Options (bit E)**
- **LS Type**
  - 7 types différents
- **Link State Id (dépend du type)**
  - Portion du réseau décrite par le LSA
- **Advertising router**
  - Routeur à l'origine du LSA
- **LS sequence number**
  - Détection des LSA plus récents



OSPF : Messages *Link state request* (LSR)

- **Requête pour une entrée de la LSDB**
- **Entrée LS type, Link state ID et Advertising router**
  - ➡ Identification de l'entrée concernée

0	LS Type
Link State ID	
Advertising Router	
...	

- **Header :** LS sequence number, LS checksum et LS age
  - ➡ Entrée "en cours"

LS Age	
Options	LS Type
Link State ID	
Advertising Router	
LS Sequence Number	
LS Checksum	
Length	

OSPF : Messages *Link state update* (LSU)

- **Flooding**
  - Multicast/broadcast sur les réseaux physiques
  - Acquittés par les messages *LSA-Acks*
- **#LSAs :** Nombre de LSA à suivre
- **LSAs :** Formats ≠ selon le type de LSA

LSAs
LSAs
...

OSPF : Messages *Link state update* (LSU)

- **Flooding**
  - Multicast/broadcast sur les réseaux physiques
  - Acquittés par les messages *LSA-Acks*
- **#LSAs :** Nombre de LSA à suivre
- **LSAs :** Formats ≠ selon le type de LSA

LSAs
LSAs
...

OSPF : 7 types de LSA

- **Type 1 (router) :**
  - Etat des interfaces (diffusé par un routeur) **-> intra-aire**
- **Type 2 (network) :**
  - Routeurs attachés à un sous-réseau (diffusé par un DR) **-> intra-aire LAN**
- **Type 3 (summary) :**
  - Résumé de route par un ABR **-> ABR / inter-aire**
- **Type 4 (inter-area summary) :**
  - Route vers l'ASBR (généré par un ABR) **-> ASBR / bordure IGP**
- **Type 5 (external) :**
  - Route externe (diffusé par un ASBR) **-> inter-aire / BGP**
- **Type 6 (multicast group membership) :** cf. MOSPF http://tools.ietf.org/html/rfc1584
- **Type 7 (external NSSA) :** Route externe générée par un ASBR d'un NSSA

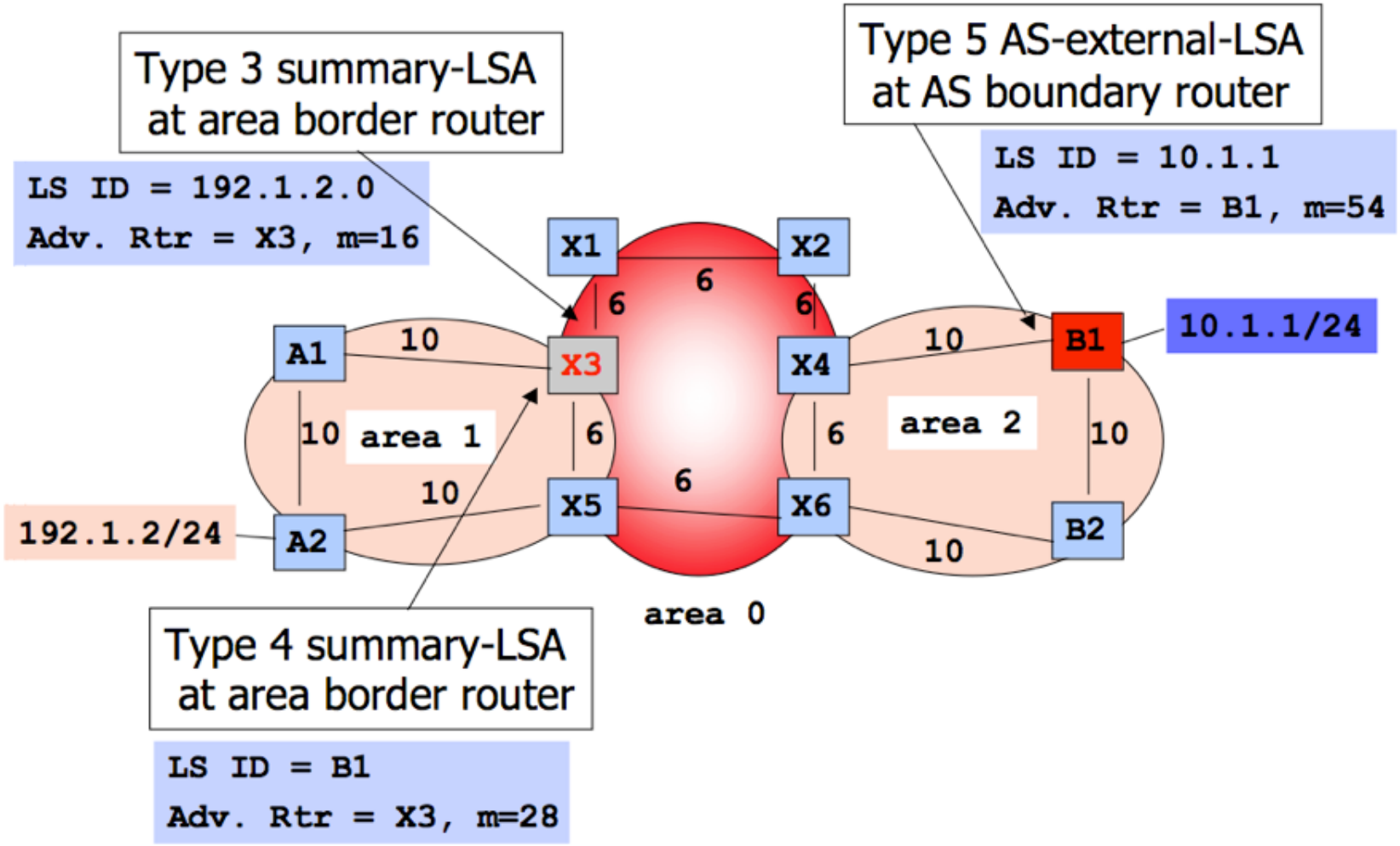
OSPF : 7 types de LSA

- **Type 1 (router) :**
  - Etat des interfaces (diffusé par un routeur) **-> intra-aire**
- **Type 2 (network) :**
  - Routeurs attachés à un sous-réseau (diffusé par un DR) **-> intra-aire LAN**
- **Type 3 (summary) :**
  - Résumé de route par un ABR **-> ABR / inter-aire**
- **Type 4 (inter-area summary) :**
  - Route vers l'ASBR (généré par un ABR) **-> ASBR / bordure IGP**
- **Type 5 (external) :**
  - Route externe (diffusé par un ASBR) **-> inter-aire / BGP**
- **Type 6 (multicast group membership) :** cf. MOSPF http://tools.ietf.org/html/rfc1584
- **Type 7 (external NSSA) :** Route externe générée par un ASBR d'un NSSA

OSPF : 7 types de LSA

- **Type 1 (router) :**
  - Etat des interfaces (diffusé par un routeur) **-> intra-aire**
- **Type 2 (network) :**
  - Routeurs attachés à un sous-réseau (diffusé par un DR) **-> intra-aire LAN**
- **Type 3 (summary) :**
  - Résumé de route par un ABR **-> ABR / inter-aire**
- **Type 4 (inter-area summary) :**
  - Route vers l'ASBR (généré par un ABR) **-> ASBR / bordure IGP**
- **Type 5 (external) :**
  - Route externe (diffusé par un ASBR) **-> inter-aire / BGP**
- **Type 6 (multicast group membership) :** cf. MOSPF http://tools.ietf.org/html/rfc1584
- **Type 7 (external NSSA) :** Route externe générée par un ASBR d'un NSSA

OSPF : 7 types de LSA, qqs exemples



© J.-Y. Le Boudec

Les différents formats de LSA

LS Age	
Options	LS Type
Link State ID	
Advertising Router	
LS Sequence Number	
LS Checksum	
Length	

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |    Options    |    LS type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

+ ???

=> 7 types possibles

Les différents formats de LSA

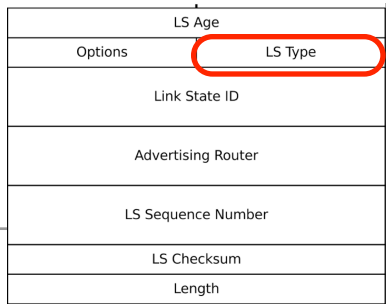
LS Age	
Options	LS Type
Link State ID	
Advertising Router	
LS Sequence Number	
LS Checksum	
Length	

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |    Options    |    LS type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

+ ???

=> 7 types possibles

Les différents formats de LSA




```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |    Options    |    LS type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

+ ???

=> 7 types possibles

OSPF : Format LSA type 1 (*router*)

http://freesoft.org/CIE/RFC/1583/

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0     |V|E|B|        0        |            # links            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Link ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link Data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     # TOS     |         TOS 0 metric          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TOS      |        0      |            metric             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TOS      |        0      |            metric             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Link ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link Data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- **V|E|B : Virtual link, External (ASBR) and Border (BR)**
- **#links : nombre de liens décrit dans ce LSA**
- **Type : type de lien**

Type	Description
1	Point-to-point connection to another router
2	Connection to a transit network
3	Connection to a stub network
4	Virtual link

OSPF : Format LSA type 1

Type	Link ID
1	Neighboring router's Router ID
2	IP address of Designated Router
3	IP network/subnet number
4	Neighboring router's Router ID

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    0    |V|E|B|        0      |            # links            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link ID                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Data                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     # TOS     |         TOS 0 metric          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TOS      |        0      |            metric             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TOS      |        0      |            metric             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link ID                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Data                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- **Link ID : type d'objet connecté**
- **Link Data : IP ou masque en fct du type**
- **#TOS : nombre de métrique pour ce lien**
- **TOS 0 metric : poids du lien pour la métrique par défaut**
- **...puis métriques suivantes... (+/- obsolète)**
- **les poids peuvent etre configurés selon divers critères et de manière statique (manuel) ou automatique (+/- adaptatif)**

OSPF (format LSA) : ToS et métrique

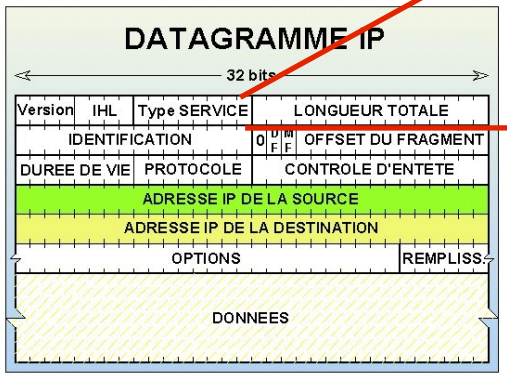
OSPF encoding	RFC 1349 TOS values
0	0000 normal service
2	0001 minimize monetary cost
4	0010 maximize reliability
6	0011
8	0100 maximize throughput
10	0101
12	0110
14	0111
16	1000 minimize delay
18	1001
20	1010
22	1011
24	1100
26	1101
28	1110
30	1111

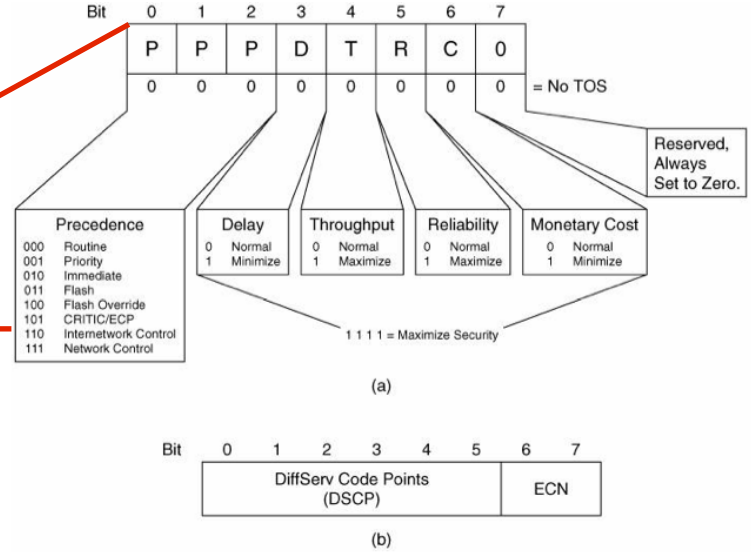
D = Delay

T = Throughput

R = Reliability

OSPF encoding	D	T	R
0	0	0	0
4	0	0	1
8	0	1	0
12	0	1	1
16	1	0	0
20	1	0	1
24	1	1	0
28	1	1	1





OSPF (format LSA) : ToS et métrique

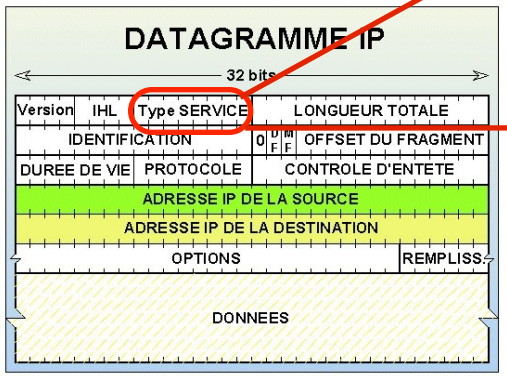
OSPF encoding	RFC 1349 TOS values
0	0000 normal service
2	0001 minimize monetary cost
4	0010 maximize reliability
6	0011
8	0100 maximize throughput
10	0101
12	0110
14	0111
16	1000 minimize delay
18	1001
20	1010
22	1011
24	1100
26	1101
28	1110
30	1111

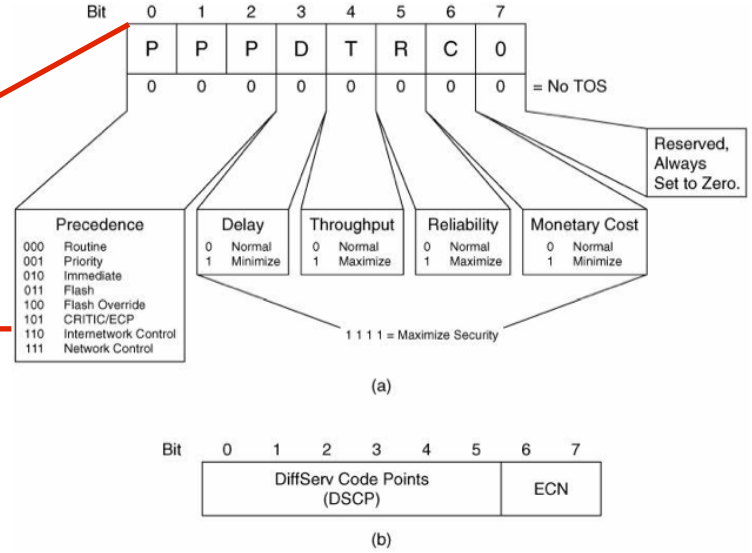
D = Delay

T = Throughput

R = Reliability

OSPF encoding	D	T	R
0	0	0	0
4	0	0	1
8	0	1	0
12	0	1	1
16	1	0	0
20	1	0	1
24	1	1	0
28	1	1	1



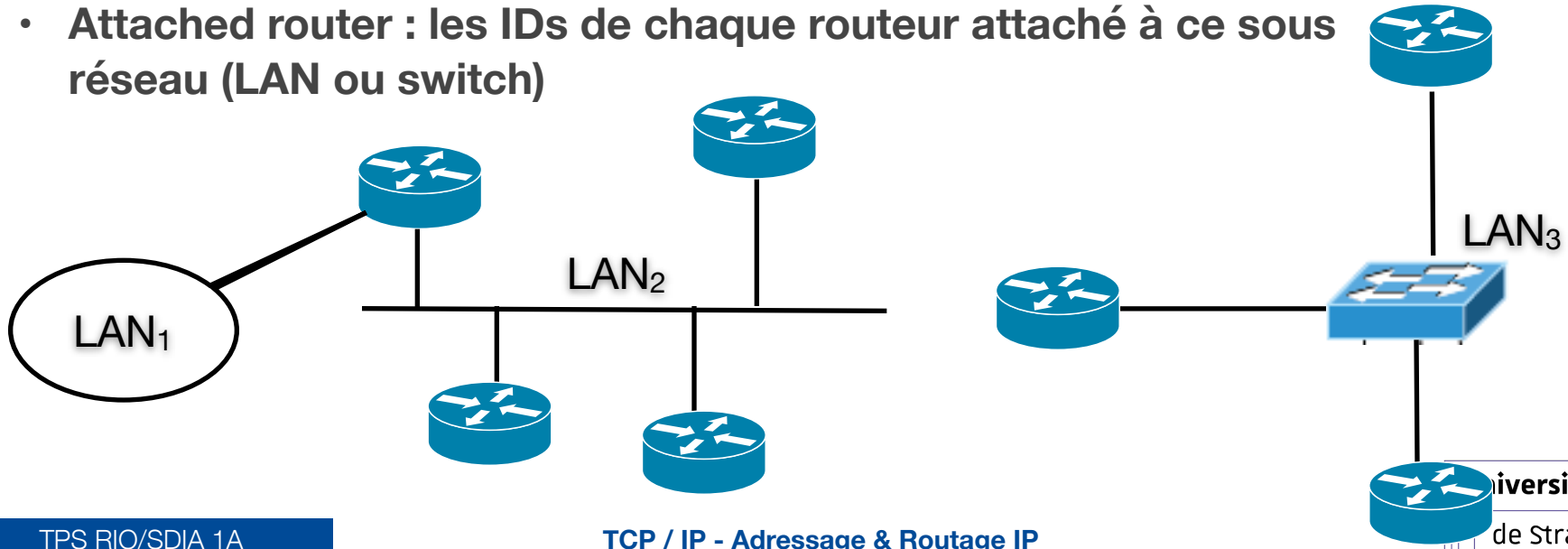


OSPF : Format LSA type 2 (*network*)

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Network Mask                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Attached Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
```

0xff000000
=
réseau de classe A

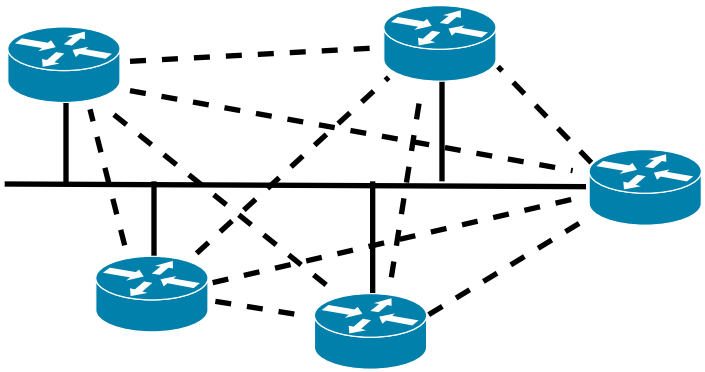
- **Network mask : le masque de sous réseau IP pour ce réseau feuille ou de transit**
- **Attached router : les IDs de chaque routeur attaché à ce sous réseau (LAN ou switch)**



OSPF : Routeurs désignés

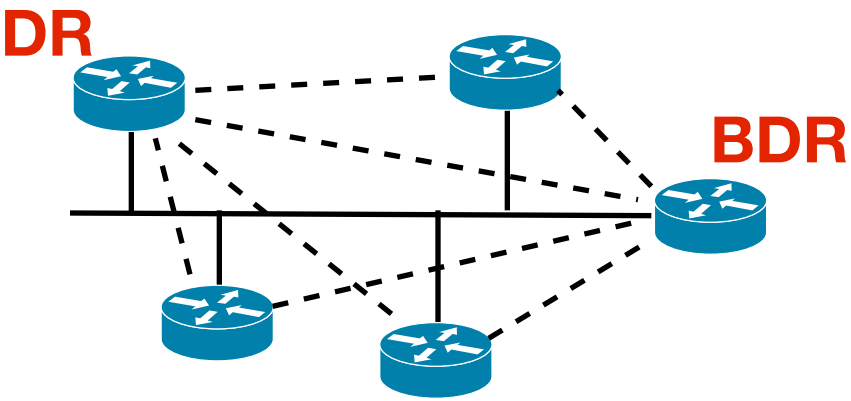
Exemple animé : http://www.visualland.net/view.php?cid=924

- **Par réseau multi-routeurs (ex : LAN)**
  - Elections : Designated Router (DR) + Backup Designated Router (BDR)
  - Critère : Priorité (configurable) et identité du routeur
  - Résultat : Annoncé dans messages *Hello*
  - Validité : Jusqu'à disparition DR / BDR
  - ➡ Le DR s'occupe de collecter/distribuer les LSA



A plat : n.(n-1)/2 liens

=> ex : avec 10 routeurs,17 au lieu de 45 !



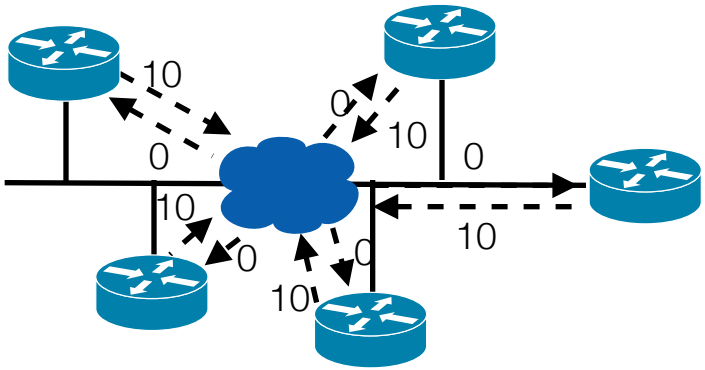
Avec DR : (n-1) liens

Avec DR et BDR : (2n-3) liens

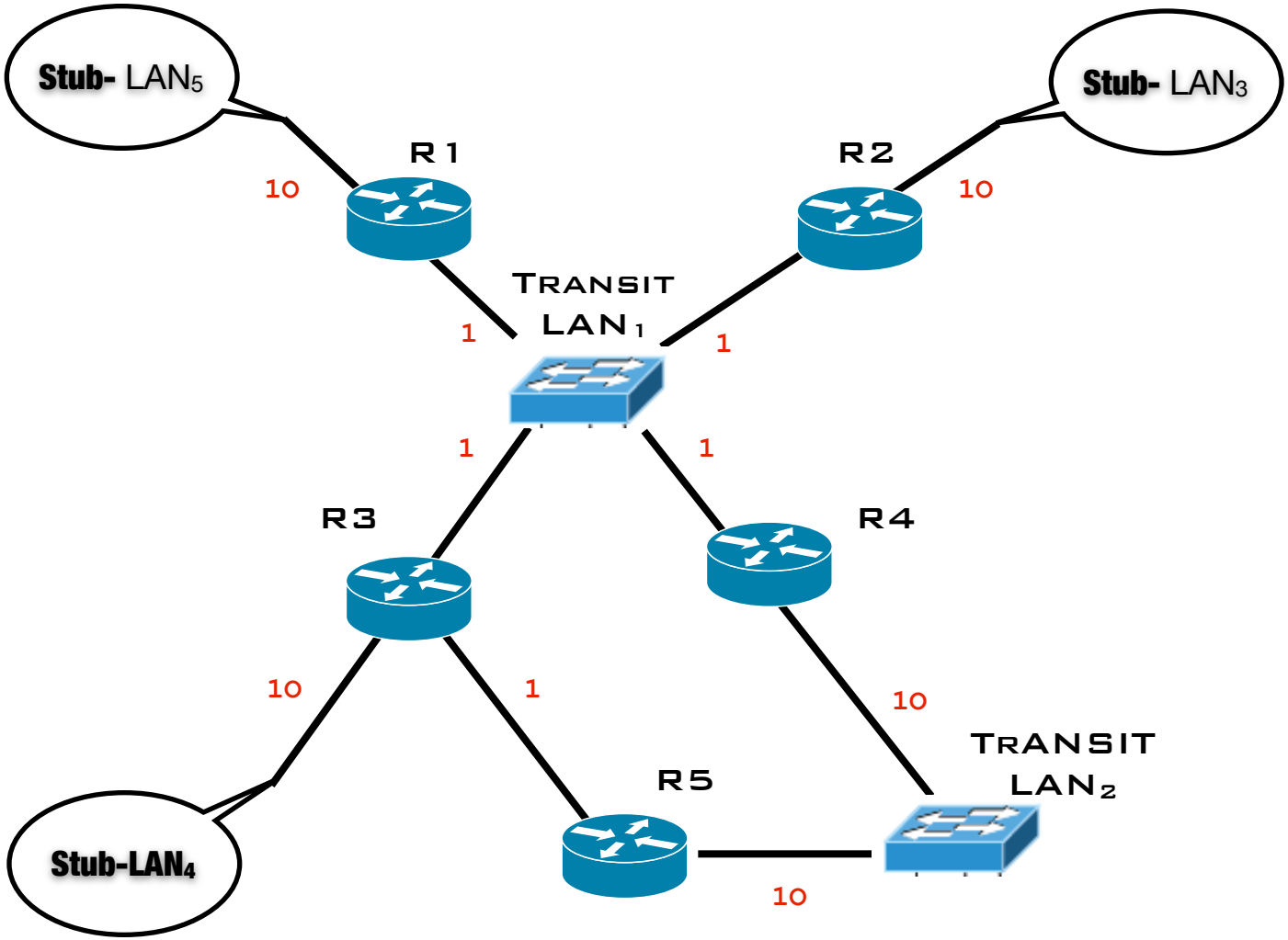
OSPF : Réseaux virtuels

- **LAN représenté comme réseau virtuel**
  - Sens routeurs -> réseau virtuel : coût réel
  - Sens réseau virtuel -> routeurs : coût nul

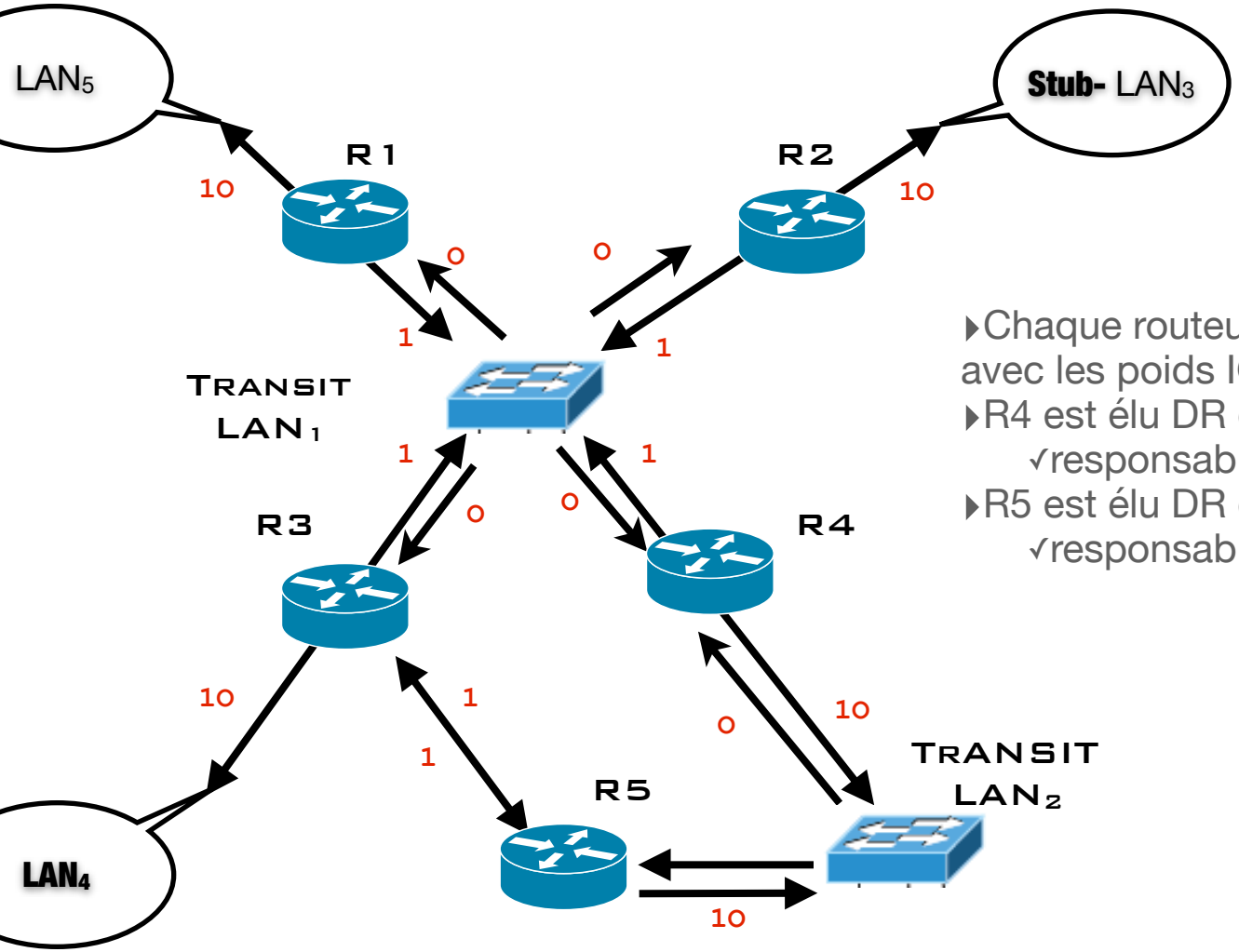
- **Cas des réseaux NBMA (Non Broadcast, Multiple Access)**
  - Circuits virtuels établis par le DR (et le BDR) vers tous les hôtes
  - DR envoie copies des MAJ sur chaque circuit



OSPF : un exemple mono-aire

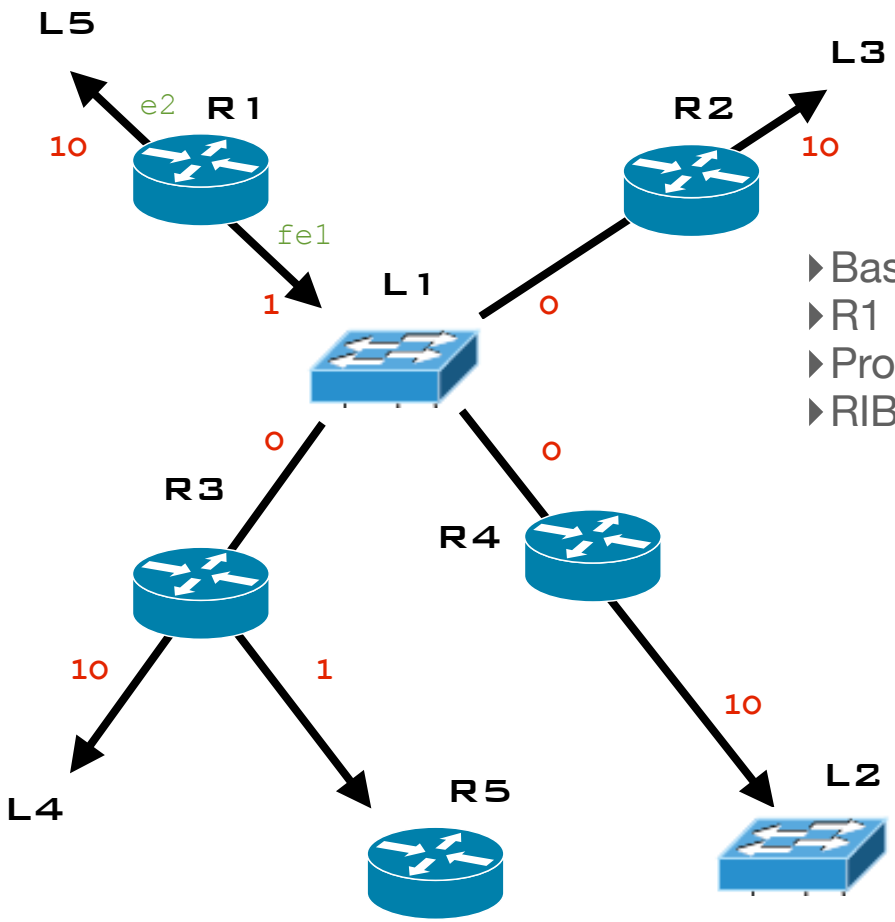


OSPF : valuation et LAN de transit



- Chaque routeur annonce ses liens sortants avec les poids IGP
- R4 est élu DR de L1 (R1,R2,R3 & R4) :
  - ✓responsable annonce des 4 états de L1
- R5 est élu DR de L2 (R4 & R5) :
  - ✓responsable annonce des 2 états de L2

OSPF : table de routage

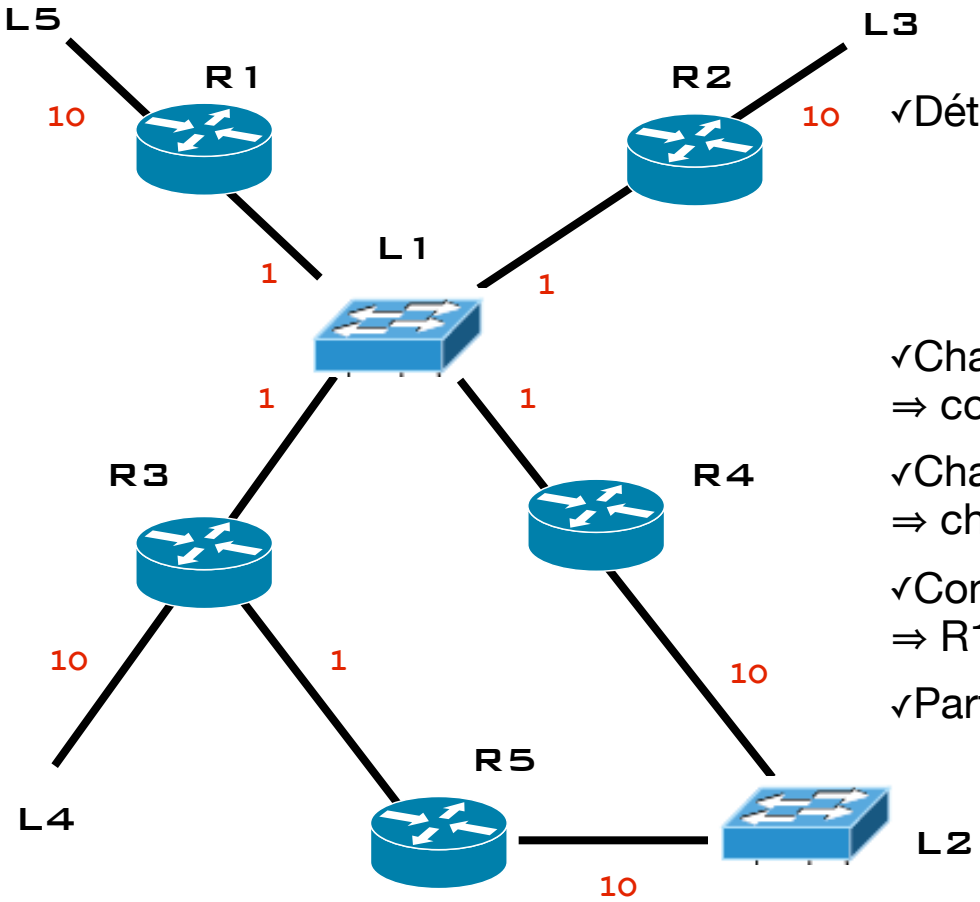


- Base états des liens ~ graphe ~ liste d'arrête
- R1 calcule ses chemins
- Prochain saut = fils dans l'arbre
- RIB contient un prochain saut par destination

- RIB sur R1

➡L1	fe1	-	1
➡L2	fe1	@R4	11
➡L3	fe1	@R2	11
➡L4	fe1	@R3	11
➡L5	e2	-	10

OSPF : changement topologique (coupure R4-L2)



✓Détection

➡ par R4 ➥ annonce R4→L1(1)
disparition R4 → L2(10)

➡ par R5 (DR_{L2}) annonce L2→R5(0)
disparition L2→ R4(0)

✓Chaque routeur diffuse ces annonces «actives»
⇒ coupure découverte implicitement

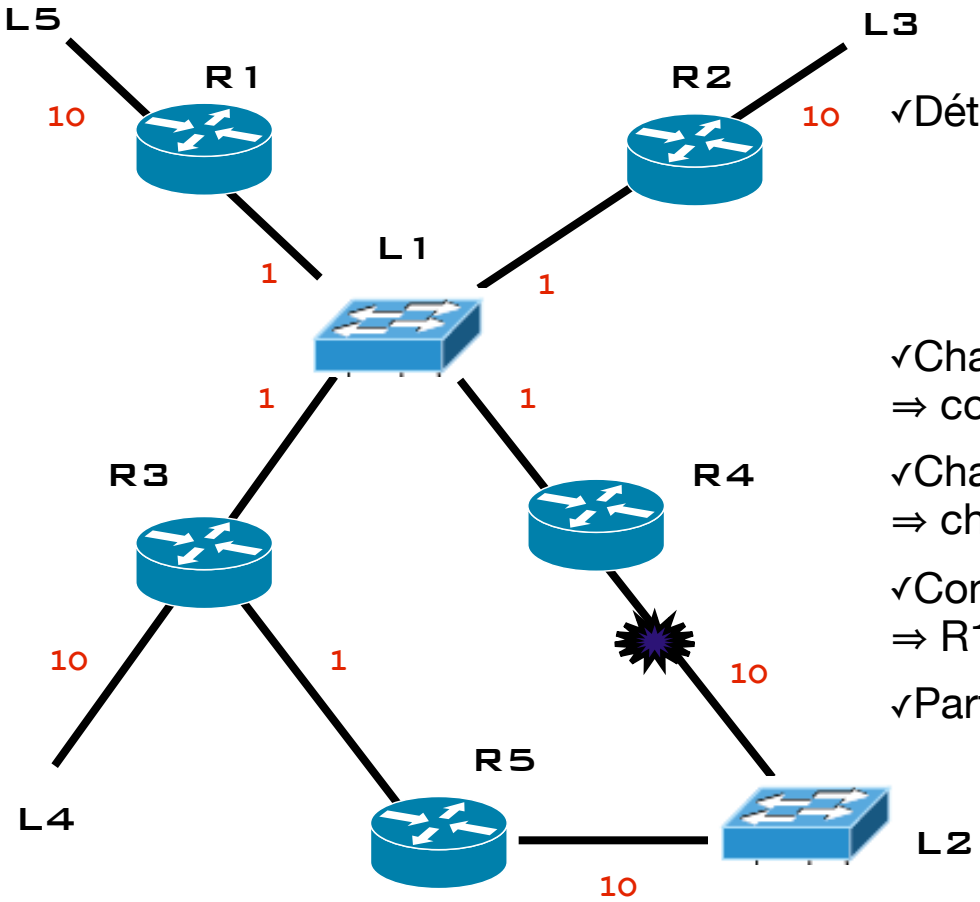
✓Changement LSDB
⇒ chaque routeur recalcule Dijkstra et RIB & FIB

✓Convergence : voir si R4 recalcule avant R1 & R3 ?
⇒ R1 ➥ R4 ➥ R3 ➥ R4 ... ?

✓Partitionnement possible : k-connexité ?

✓Détection (physique, logique)
➥ Réaction (convergence IGP, fast-reroute : IP-FRR, MPLS-FFR, ...) ?
accélérer la réponse à une panne par tous les moyens ! prévention ..!

OSPF : changement topologique (coupure R4-L2)



✓Détection

➡ par R4 ➡ annonce R4→L1(1)
disparition R4 → L2(10)

➡ par R5 (DR_{L2}) annonce L2→R5(0)
disparition L2→ R4(0)

✓Chaque routeur diffuse ces annonces «actives»
⇒ coupure découverte implicitement

✓Changement LSDB
⇒ chaque routeur recalcule Dijkstra et RIB & FIB

✓Convergence : voir si R4 recalcule avant R1 & R3 ?
⇒ R1 ➡ R4 ➡ R3 ➡ R4 ... ?

✓Partitionnement possible : k-connexité ?

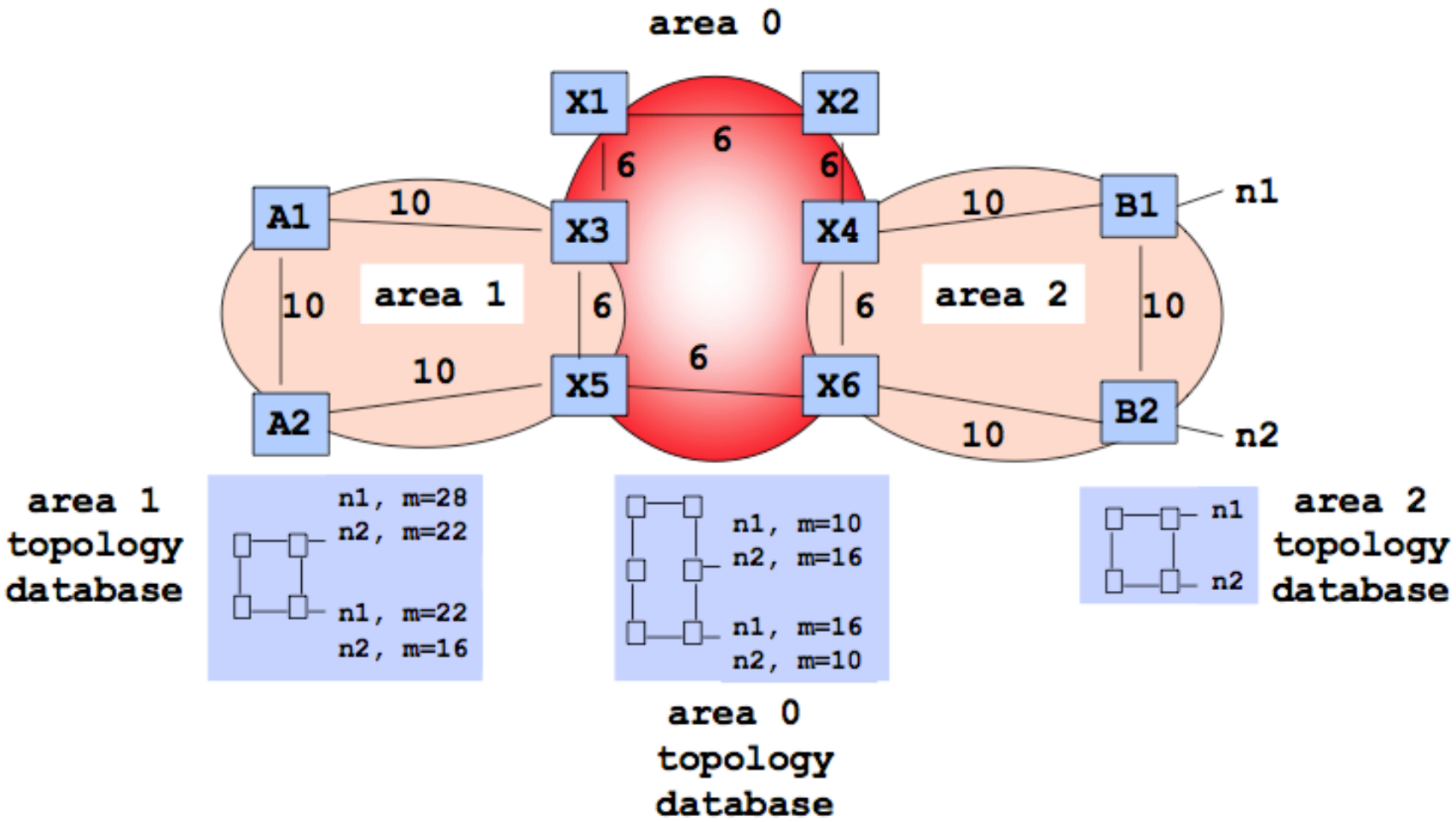
✓Détection (physique, logique)
➡ Réaction (convergence IGP, fast-reroute : IP-FRR, MPLS-FFR, ...) ?
accélérer la réponse à une panne par tous les moyens ! prévention ..!

OSPF : Format LSA types 3 et 4 (**summary**)

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Network Mask                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     TOS       |                  metric                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- **Network mask : peut etre utilisé pour une route par défaut (type 3 avec réseau feuille - stub network) ou pour indiquer le masque de sous réseau du réseau cible (ce champ est à 0 pour le type 4)**
- **TOS : Type de Service (cf. slides précédents)**
- **metric : distance pour cette route (cf. slides précédents)**

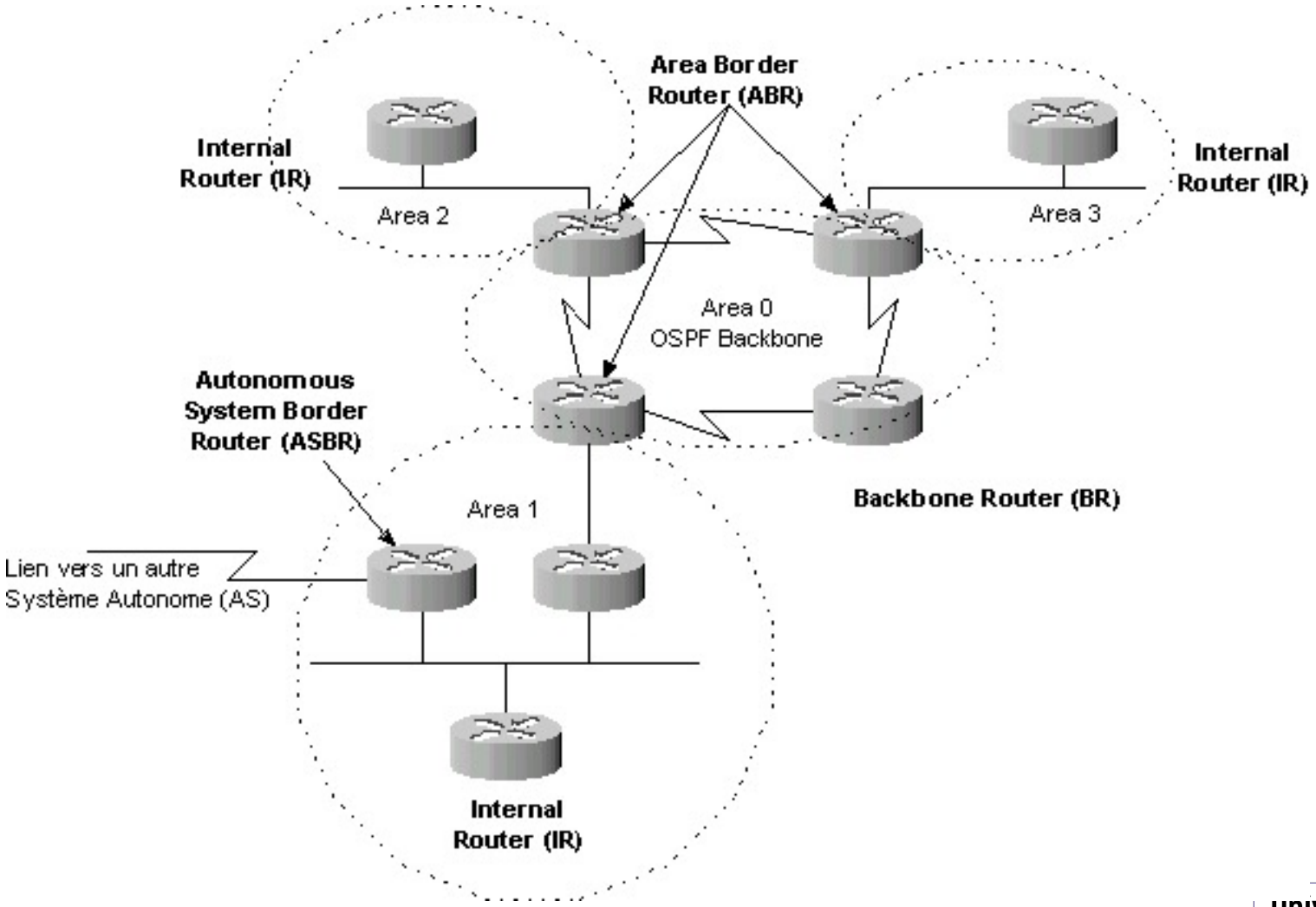
OSPF et zones : un exemple



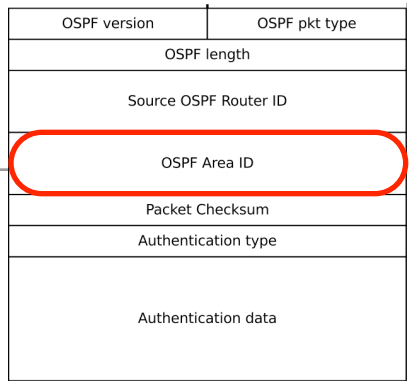
OSPF : 4 types de routeurs

- **Routeur interne :** Routeur dont toutes les interfaces sont dans la même aire
- **Routeur backbone :** Routeur dont au moins une interface appartient à l'aire 0
- **Area Border Router (ABR) :** Routeur disposant d'interfaces dans des aires ≠
  - Résumés des LSA pour ses zones et distribution dans l'aire 0
  - ABR = Routeur backbone
- **Autonomous System Boundary Router (ASBR)**
  - Routeur injectant dans OSPF des routes vers l'extérieur
    - Statiques ou provenant d'autres protocoles de routage (ex : BGP)

OSPF : 4 types de routeurs

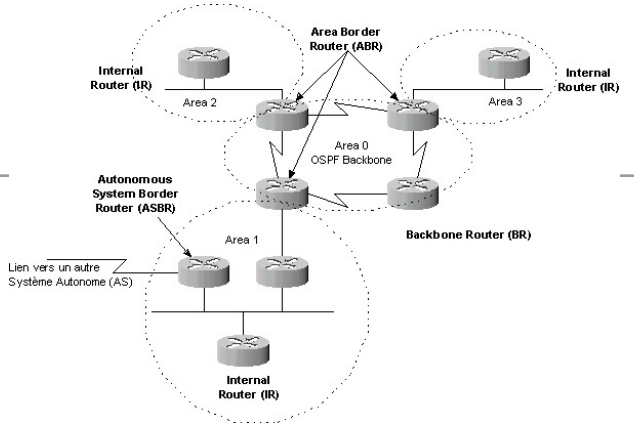


OSPF : 6 types d'aires



- **Aire 0 :** Aire backbone (l'épine dorsale du réseau)
- **Stub area :** Aire vers laquelle ne sont pas propagées les LSA de type 5 (routes externes)
- **Totally stub area :** Aucun LSA de type 3, 4 ou 5 n'y est propagé, à l'exception d'une route par défaut
- **Not-so-stubby area (NSSA) :** Type de stub area qui permet l'injection de routes externes via un LSA de type 7. Le type 7 sera converti en type 5 quand il sera transmis en dehors de l'aire
- **Totally NSSA :** NSSA sans LSA 3 et 4 à l'exception d'une route par défaut
- **Aire normale :** Toute autre aire que les précédentes

OSPF : Routage inter-aires

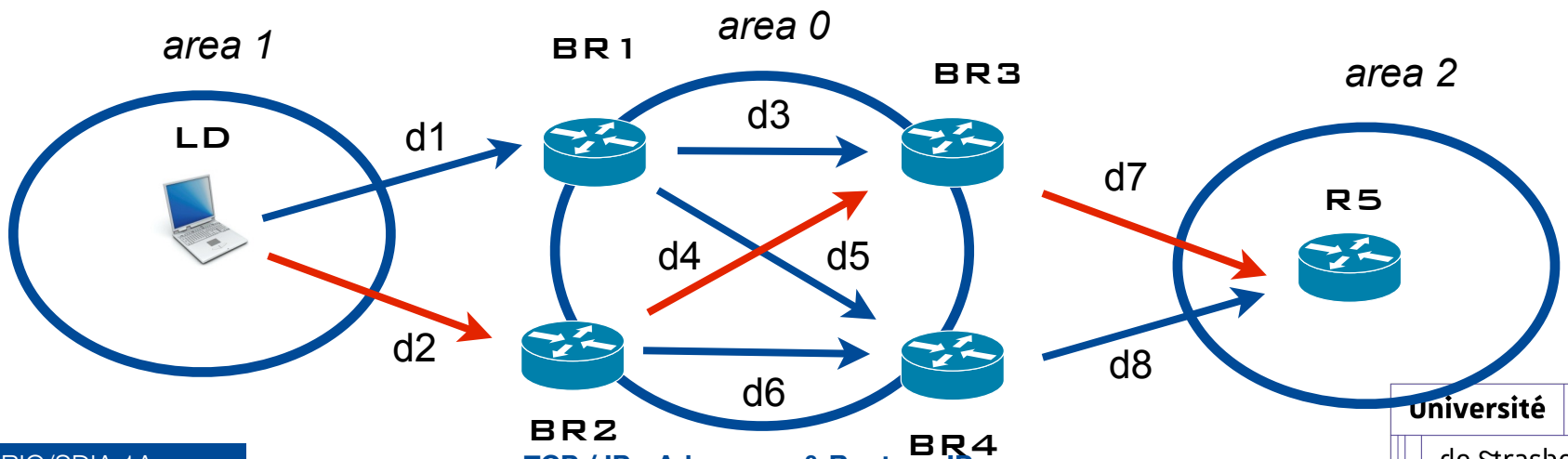


- **Principe simplifié**
  - BR calcule RIB zones adjacentes
    - Diffuse résumé dans autres zones adjacentes
    - Résumés toutes zones envoyés aux autres BR
  - Chaque BR calcule ses meilleures routes
    - Pour les zones non adjacentes
  - Envoi du résumé dans zones non backbone
    - Remarque : Inutile dans zone backbone car déjà connu

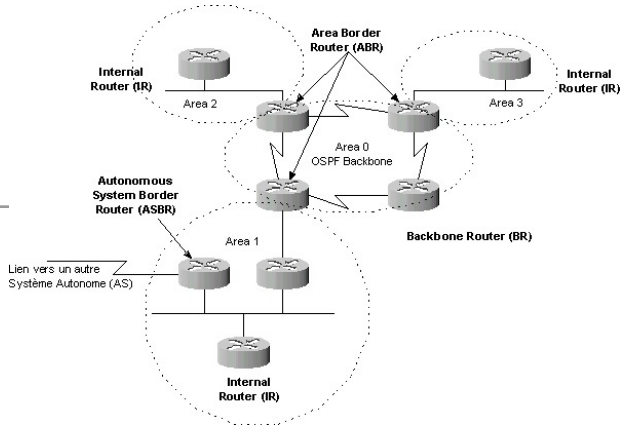
OSPF : Routage inter-aires

- **Entre deux aires :** routeur frontière (*border router* - BR)

- **Chaque BR :** Processus intra-zone indépendant selon la zone de l'interface
  - LSA/LSU de type 1 et 2 non diffusés entre zones
    - Type 3 : Préfixe interne au domaine ➥ pour routage inter-zones
    - Type 4 : Routeur frontière du domaine ➥ pour routage inter-domaine



OSPF : Routage inter-aires

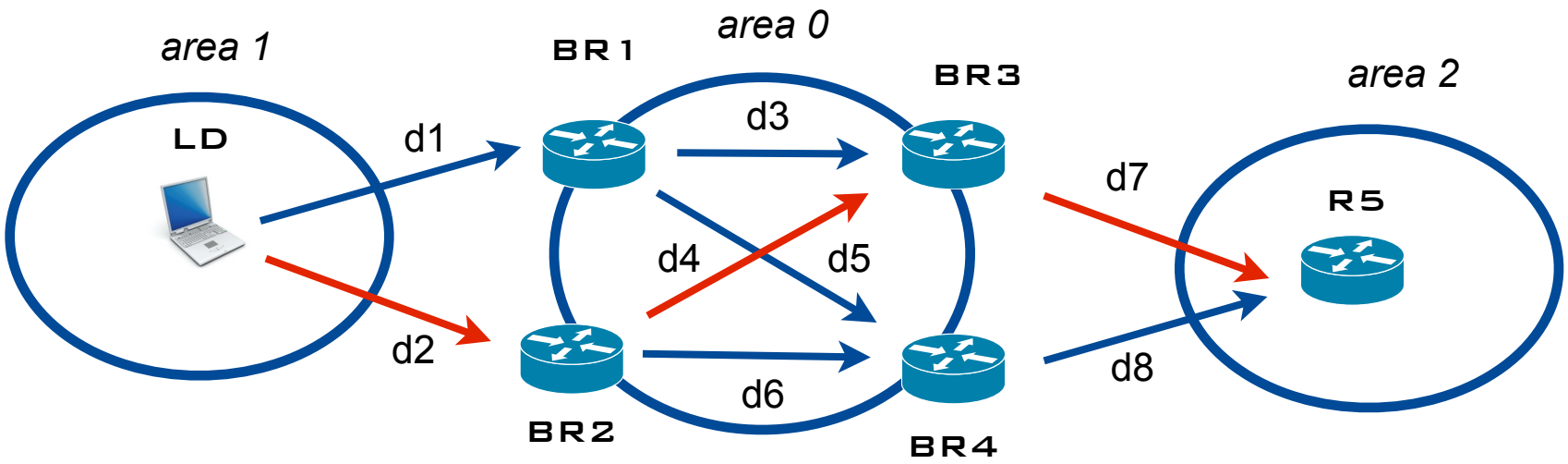


- **Route inter-zones :** Passe toujours via *area* 0

- **Route entre 2 réseaux d'une zone :** ne passe que par *area* 0
  - ➡ Pas toujours la plus courte / liens virtuels pour compenser ?

- **Calcul de la RIB et diffusion LSA mixé :**
  - Diffusion LSA intra-zone
  - Calcul RIB intra-zone
  - Diffusion LSA résumé (inter-zone)
  - Calcul RIB inter-zone
  - Pour *Dijkstra* : noeuds et arêtes intra traités «avant» noeuds et arêtes inter

OSPF : exemple de routage inter-aires



- **BR1 :** envoie résumé BR1-LD dans aire 0
- **BR2 :** envoie résumé BR2-LD dans aire 0
- **BR3 :** calcule route vers N = min(d3+d1, **d4+d2**) = **d9** et envoie dans aire 2
- **BR4 :** calcule route vers N = min(d5+d1, d6+d2) = d10 et envoie dans aire 2
- **R5 :** calcule route vers N = min (**d7+d9**, d8+d10) : **route inter-zone**

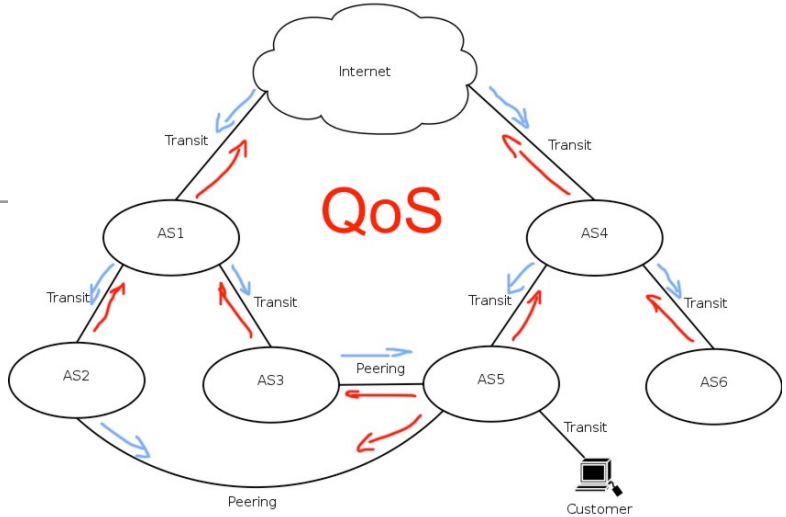
OSPF : Format LSA type 5 (**`AS-external`**)

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Network Mask                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E|     TOS     |                  metric                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Forwarding address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      External Route Tag                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- **Network mask : Indique le masque de sous réseau du réseau/AS cible**
- **bit E : Type de métrique externe (à 1 : non comparable | à 0 : comparable)**
- **Forwarding address : Adresse IP de l'ASBR origine (0.0.0.0) ou d'un autre ASBR**
- **External Route Tag : pour la communication iBGP**

Routage externe

➡4 relations et 3 niveaux (Tier-1/Transit/Stub) :
- ✓P2C : provider to customer
- ✓C2P : customer to provider
- ✓P2P : peering
- ✓S2S : sibling



Modèle

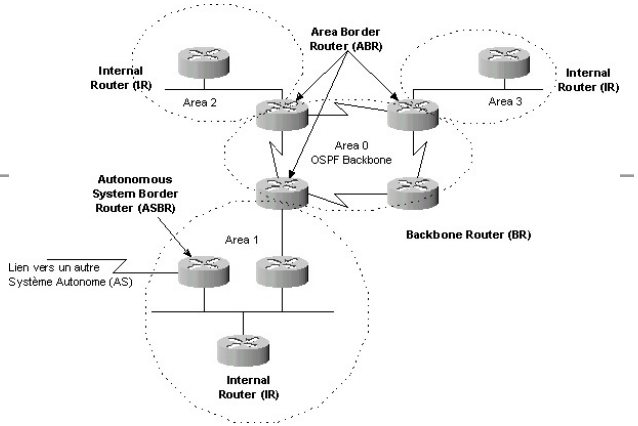
✓Internet découpé en **domaines** ou **Autonomous Systems AS**

✓1 domaine = réseau d'une seule entité
- opérateur, entreprise, …

✓1 routeur appartient à un seul AS
- frontière passe entre les routeurs
- annonces échangées via **BGP : Border Gateway Protocol**
- choix de la meilleure route sur plusieurs critères (**LOCAL_PREF, AS_PATH,**...)

➡politique de routage (pas forcément route la plus courte)

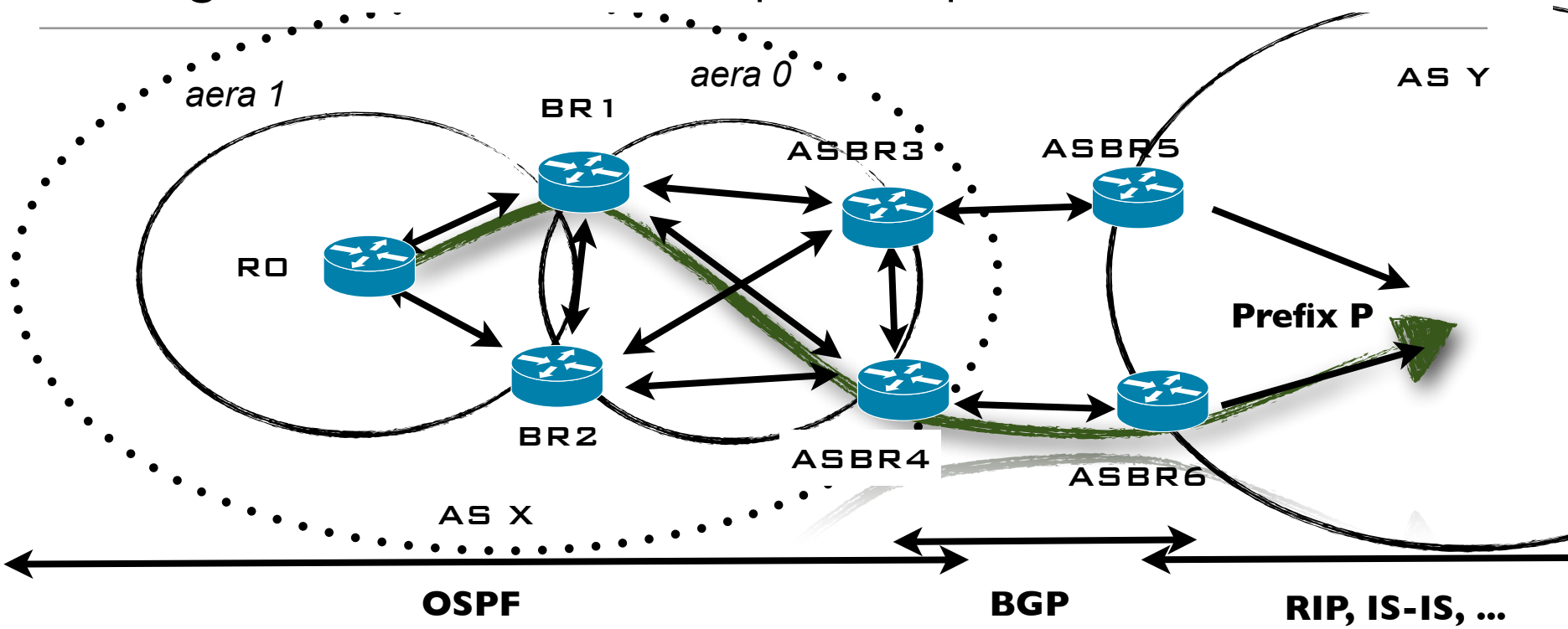
BGP & redistribution OSPF



- Un ou plusieurs ASBR (/ eBGP)
  - ➡AS **border router** : routeur frontière d'AS
  - ➡ASBR apprend routes externes
    - par ex. via BGP : Border Gateway Protocol
      - – redistribution des routes
  - ✓construit & envoie des LSA/LSU externes (type 5)
  - ✓Ces LSA/LSU externes sont diffusés dans toutes les zones
  - ✓Les BR diffusent des LSA/LSU résumé type 4 = routes vers les ASBR
  - ✓Remarque : redistribution OSPF ⇒ BGP

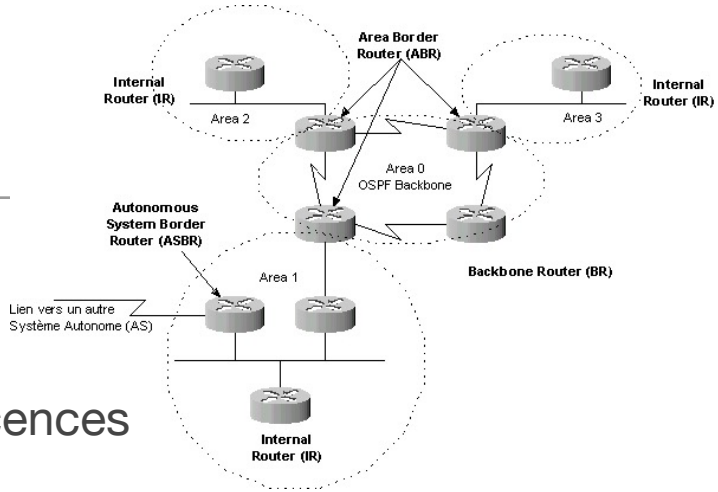
- Routeur construit RIB externe
  - ➡après RIB intra-zone
  - ➡après RIB inter-zone (dont routes vers les ASBR)

Routage externe : un exemple simple



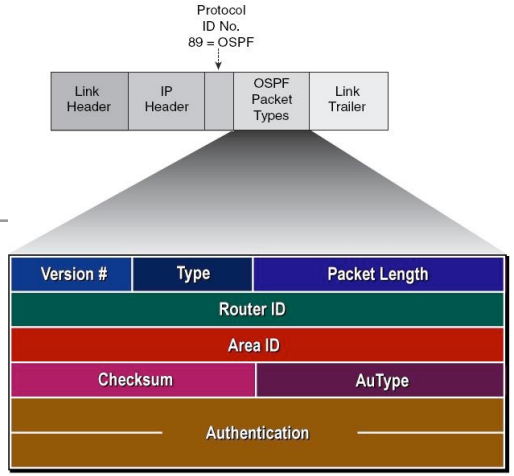
- ✓ASBR5 annonce à ASBR3 une route vers le préfixe P (via BGP)
- ✓ASBR3 diffuse dans tout réseau LSA/LSU type 5 pour P (quelle métrique ?)
- ✓ASBR6 annonce à ASBR4 une route vers P (via BGP)
- ✓ASBR4 diffuse dans tout le réseau LSA/LSU type 5 pour P (quelle métrique ?)
- ✓BR1 annonce dans zone 1 routes vers ASBR3 et ASBR4 (résumé type 4: route vers ASBR)
- ✓BR2 annonce dans zone 1 routes vers ASBR3 et ASBR4 (idem)
- ✓BR0 calcule sa meilleure route vers ASBR3 et ASBR4 (routage inter-zone)
- ✓BR0 calcule sa meilleure route vers P : min (d(BR0,ASBR3)+ d(ASBR3,P), d(BR0,ASBR4) +d(ASBR4,P))

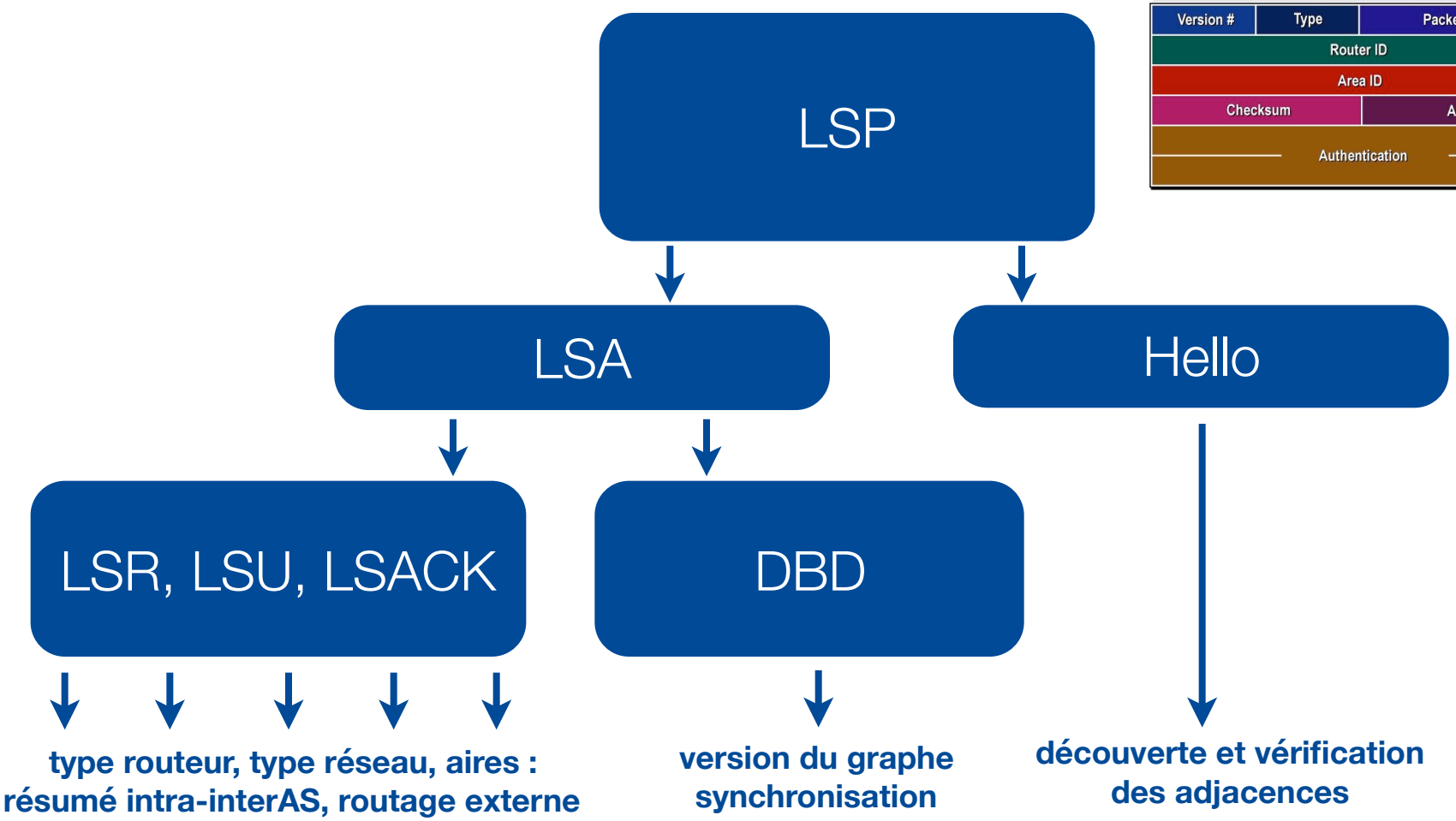
OSPF v2, résumé



- **Trois phases Data plane**
  - Découvrir les voisins et établir/surveiller adjacences
  - A chaque changement adjacence
    - Calculer LSDB, diffuser nouveaux LSA
  - A chaque changement LSDB
    - Calculer RIB intra-aire et RIB inter-aires (destinations hors aire)
  - Et diffuser LSA inter-aires
    - Calculer RIB externe (destinations/préfixes hors domaine)

OSPF v2, résumé messages LSP





OSPF : Le bilan

- **Avantages :** Pas de comptage à l'infini, pas de limite sur le nombre de sauts, convergence plus efficace, ...

- **Trafic de signalisation :** Seuls messages périodiques = *Hellos*
- **Synchronisation d'une LSDB :** Boucles transitoires «rapidement» éliminées
- **Réseaux hétérogènes :** Broadcast ou non
- **Grands réseaux :** Organisation hiérarchique, division en aires
- **Différents sous-protocoles :** Hello, Flooding LSA, inter-action BGP, etc.

- **Inconvénient :** plus complexe à mettre en oeuvre que RIP...

Autres solutions

- **IS-IS**
  - Conçu au sein de l'ISO
  - Débat OSPF vs. IS-IS
  - Actuellement : déployé dans les très grands AS

"A camel is a horse designed by a committee"

- RFC 1142 - IS-IS protocol specification (IETF) - Note: this is a copy of DP 10589 (Draft Proposal) and differs in many significant details from the final version of ISO/IEC 10589
- RFC 1195 - Use of OSI IS-IS for Routing in TCP/IP and Dual Environments

Source : Wikipedia