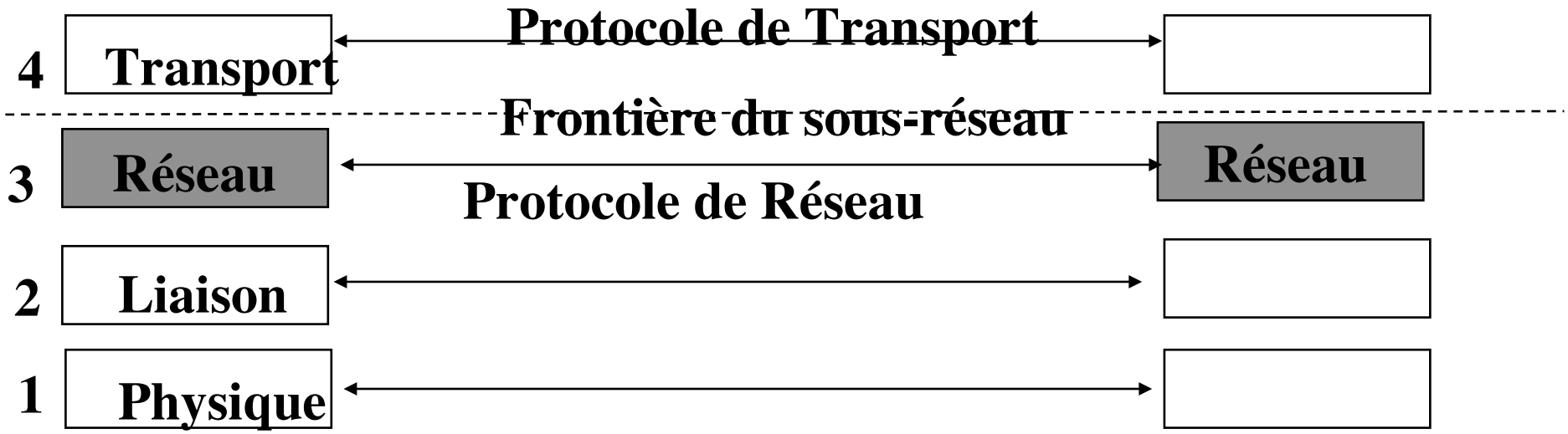
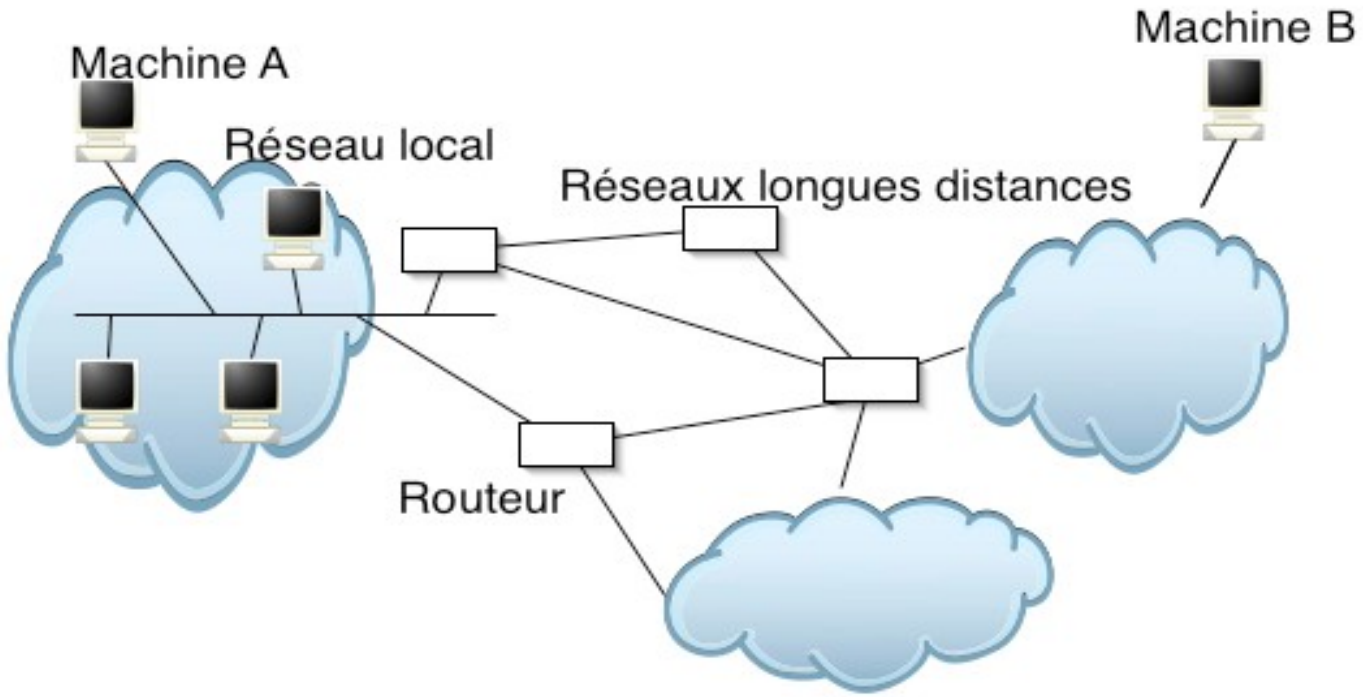
La couche réseau



Plan du chapitre 6

- **Fonctions de la couche réseau**
- **Les techniques de routage**
- **Un exemple de routage distribué: le routage dans IP**

Fonctions de la couche réseau



FONCTIONS DE LA COUCHE RÉSEAU

- **Interfaces: sous-réseau de communication et transport**
- **Service1: le routage**

  → déterminer le chemin (= la route) des paquets à travers le réseau, pas à pas, de leur source vers leur destination
  - **Si source et destination sont dans des sous-réseaux différents**

    → routage réseau niveau 3 pour aller de routeur en routeur jusqu'à la dernière passerelle réseau sur le LAN de destination.
  - **Rappel : si source et destination sont dans le même sous-réseau de communication (i.e. LAN) → pas de routage mais appel à ARP niveau 2 pour faire transmission niveau 2 en point à point au travers du LAN).**
- **Service2: la fragmentation**

  → quand réseaux hétérogènes de PDU ≠

Routage : principe

Problématique :

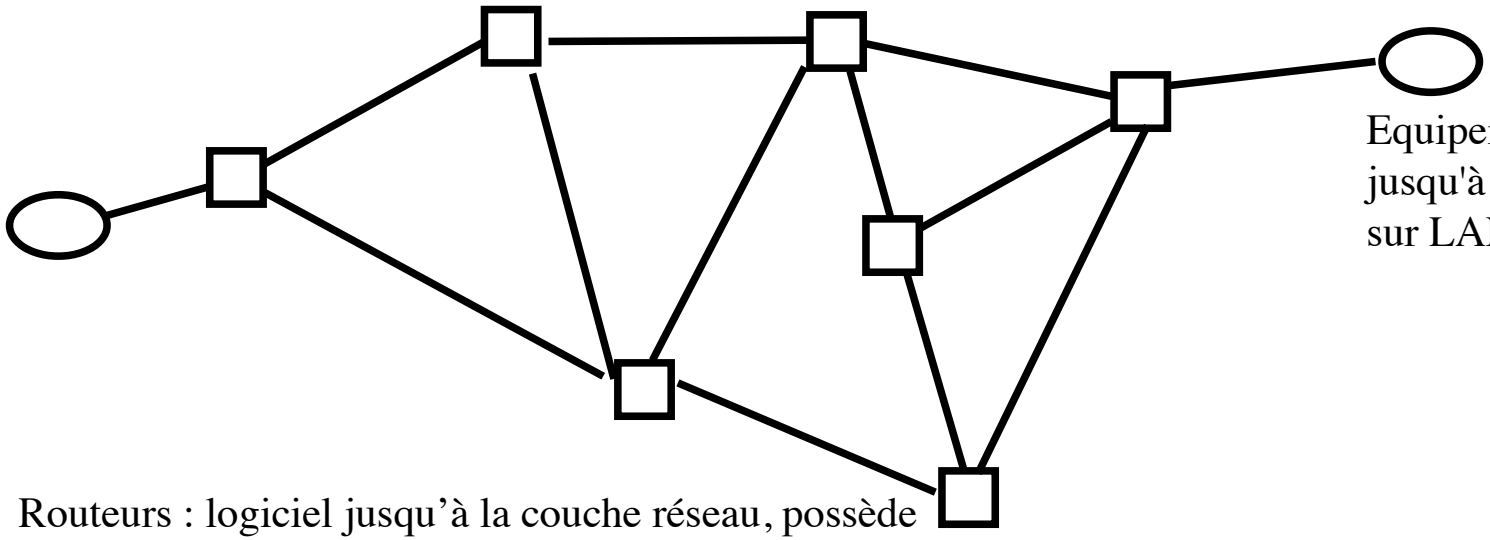
- **Différentes routes possibles : comment décider de la route à prendre ?**

  → Chaque routeur doit décider sur laquelle de ses interfaces de sortie envoyer les paquets et donc quelle est la prochaine interface de routeur qui doit recevoir le paquet.

Deux fonctions distinctes constituent la fonction de routage :

→ Déterminer le réseau voisin où émettre grâce à @destination du paquet

→ Maintenir la table de routage [@ réseau destination ; prochain routeur (next hop)]

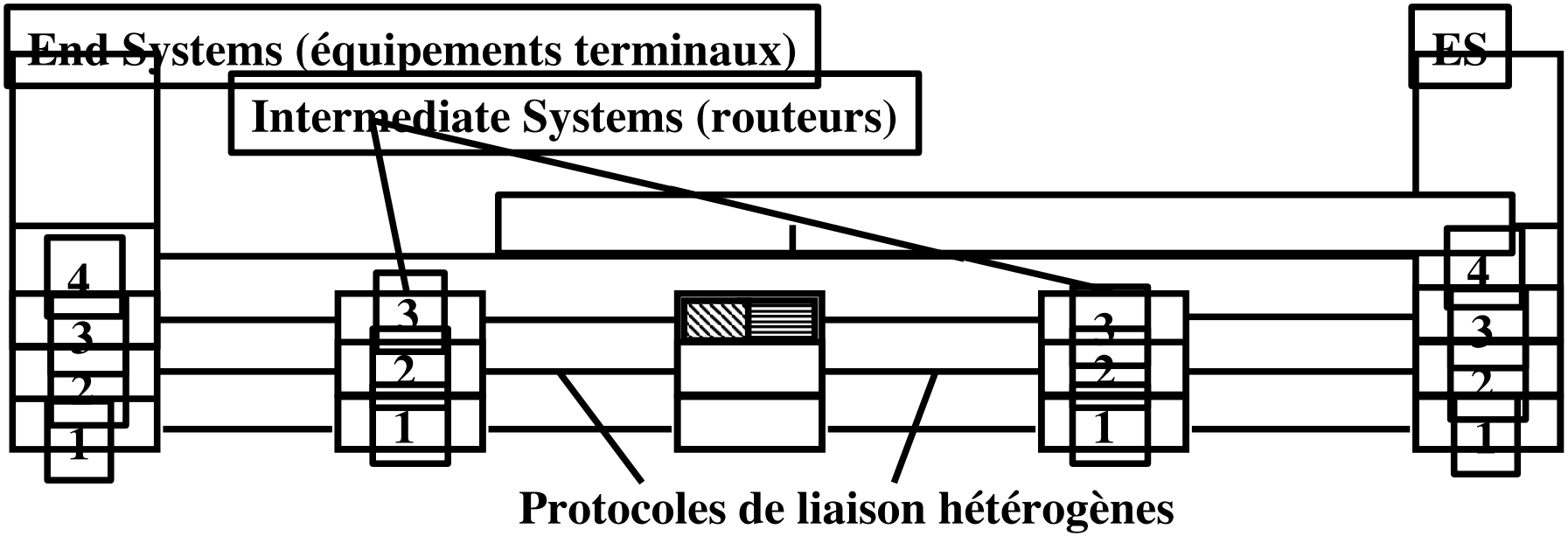


Equipement terminal : logiciel jusqu'à la couche transport/app, sur LAN ou modem.

Routeurs : logiciel jusqu'à la couche réseau, possède un certain nombre de canaux en entrée et en sortie

Architecture Générale

- **Les routeurs servent à interconnecter les réseaux hétérogènes de l'Internet**
  → interfaces réseaux ≠
  exemple : réseau LAN Ethernet et un WAN FO Token-Ring



Le routage : classification des algorithmes

- **Objectifs :**
  - **Déterminer des chemins dans un graphe (ie. entre machines de l'Internet)**

    → Exactitude, simplicité, équité (vis-à-vis des différents utilisateurs par exemple), optimisation, souplesse, sécurité, stabilité...

- **Centralisés, décentralisés**
  - **Centralisés : la route est calculée par un nœud particulier (ou plusieurs mais de manière identique)**
  - **Décentralisés : chaque nœud calcule la route**

- **Statique, dynamique**
  - **Statique : la route n'est changée que lorsqu'il y a un changement dans la topologie du réseau → mise à jour des routes manuelle par exemple**
  - **Dynamique : le choix de la route s'adapte plus ou moins rapidement à des variations du réseaux (charge, pannes de liens et de nœuds, …) → mise à jour automatique régulière**

Informations pour le routage

- **Calcul local ou global pour déterminer le chemin**
  - **global au réseau : connaissance de l'ensemble du réseau**
  - **local au noeud: connaissance partielle du réseau**

- **Fonction à calculer**
  - **La plupart des algorithmes associent un coût à chaque chemin appelé métrique**
  - **Ce coût fait intervenir des paramètres : temps de propagation, débit, charge moyenne mesurée … options politiques comme interdiction de certaines routes ou machines.**

- **Objectifs privilégiés par le routage**
  - **Plus court chemins**
  - **Minimisation du temps moyen d'attente global, du délai**
  - **Sécurité, coût, répartition de charge, robustesse**
  - **Débit ...**

Le nommage dans les réseaux

- **Identificateur : « adresse »**
- **Homogèneïté Source / Destination**
  → Unicité des adresses
- **Groupe d'adresses, adresse de diffusion**
- **Exemple IP : choix d'un système hiérarchique**
  - **Exemple similaire : Adresses postales**
    - **Nom Prémon**
    - **Numéro Rue**
    - **CP Ville**
    - **Pays**
- **Cf vision graphe des distances pas géographique**

Nommage dans Internet: adresses IP

- **IPv4 : @IP sur 4 octets**
  - **» Ex : 192.168.21.2**
- **2 parties distinctes: @du réseau local – @machine**
  - **» Ex : 192.168.10 – .2**
- **@réseau : @du réseau local – 00…0**
  - **» Ex : 192.168.10.0**
- **@broadcast : @du réseau local – 11…1**
  - **» Ex : 192.168.10.255**
- **Masque réseau : pour traiter facilement les @réseau**
  - **» « 1 » sur les bits @réseau**
  - **» Ex : 255.255.255.0**
- **IPv6 sur 16 octets**
  - **» Ex : 276A:9B3A::0CD8:1234**

Ex d'adresses IP

- **Exemple IPv4 :**
  - » **195.0.0.1 et 195.0.0.254 : 2 adresses (de classe C) de deux machines appartenant au même réseau local 195.0.0.0**
  - » **partie machine à 0 interdite : c'est le réseau lui même**
  - » **partie machine à 255 interdite : c'est l'adresse de broadcast / diffusion à tous**
- **Adresses particulières:**
  - **127: boucle locale 127.0.0.0 à 127.255.255.255**
  - **0.0.0.0 : Utilisé dans les protocoles comme adresse par défaut**

- **Attention : ne pas confondre !!!!**
  - **Adresses IP, couche 3, affectées par logiciel (dans l'OS), visibilité dans l'Internet**
    - » **4 octets représentés avec x dans [0..9] : xxx.xxx.xxx .xxx**
  - **Adresses MAC, couche 2, inscrite dans le matériel (carte réseau), visibilité uniquement dans le LAN**
    - » **6 octets représentés avec Y dans [0..F] : yy:yy:yy:yy:yy:yy**

Répartition @ Internet.v4

- **Historiquement 5 différentes classes définies par les 4 premiers bits:**
  - » **classe A: (0) réseaux 0 à 127 ; partie machines: 3 derniers octets**
  - » **classe B: (10) réseaux 128.x à 191.x ; partie machines: 2 derniers octets soit 65536-2=65534 @machines**
  - » **classe C: (110) réseaux 192.x.x à 223.x.x ; partie machines: le dernier octet soit 256-2=254 @machines**
  - » **Classe D: (1110) réseaux 224 à 239 ; @Multicast : messages destinés à un sous-ensemble de réseaux**
  - » **Classe E: (11110) réseaux expérimentaux 240 à 247 ; pas utilisés en environnements de production**
- **Cela introduit trop de gaspillage d'@ :**
  - » **Ex : réseau à 300 machines, 65234 @ non utilisées !**

Optimisation CIDR
Classless Internet Domain Routing

- **Routage de domaine Internet sans classe (CIDR)**
  - **→ généralisation des classes d'@ au bit près (au lieu octet)**
  - **→ ajuste aux besoins précis (exemple FAI) : économie @**
  - **Utilisation d'un masque de réseau au bit près**
  - **@IP/xx bits de la partie réseau**
    - **Ex : 192.168.10.64/28, 32-28 bits pour 16-2=14 machines**
    - **Ex: Un FAI se voit attribuer un /25 sur 204.21.128.0.**
      notation CIDR 204.21.128.0/25→ @réseau sur 25 bits
      masque réseau 255.255.255.128
      @machines de 204.21.128.1 à 204.21.128.126
      sous-réseau de 128-2 machines

Mecanisme avec masque @IP

- **Masque de réseau (Netmask) :**
  - **Détermine la partie @réseau de @IP : bits à 1 sur la partie réseau et élimine la partie @machine bits à 0 sur la partie machine**
    - **→ ET logique bit à bit détermine la partie @réseau**
      - **» Ex: Classe C le Netmask par défaut 255.255.255.0**
      - **» En CIDR 192.0.0.64/26 : Netmask = 255.255.255.192**
  - **CIDR permet la récursion :**
    - **» plusieurs sous-sous-réseaux par modification du netmask**
    - **» Ex : 192.0.0.64/26 donne 4 sous-reseaux /24 : 192.0.0.64/28 192.0.0.80/28 192.0.0.96/28 192.0.0.112/28**

Diffusion @broadcast

Pour s'adresser à toutes les machines d'un réseau local : partie machine à 111111111...

- **Ex: avec 192.0.0.0/24 → 192.0.0.255 (netmask 255.255.255.0)**
- **Ex: avec 192.0.0.64/26→192.0.0.127 (netmask 255.255.255.192)**
- **255.255.255.255 : broadcast sur le sous réseau entier**
- **cf. convention identique pour @Niveau2 Ethernet : FF:FF:FF:FF:FF:FF (en Hexa)**

Réseaux privés

- La RFC 1597 et la RFC 1918 réservent des plages @IP aux réseaux privés et non routées sur Internet :
  - 10.0.0.0 à 10.255.255.255
  - 172.16.0.0 à 172.31.255.255
  - 192.168.0.0 à 192.168.255.255
- Ce sont les @ à utiliser en local dans les TP etc.

Fragmentation IP

Les réseaux n'ont pas tous les mêmes capacités, buffers, traitement, couche liaison, … cf IoT, téléphonie.

- **Déterminer la MTU (maximum transmission unit)**
- **Découper la payload en fragments de taille ≤ MTU**
- **Déterminer le numéro des fragments, fragmentID**
- **Calculer les décalages/offsets**

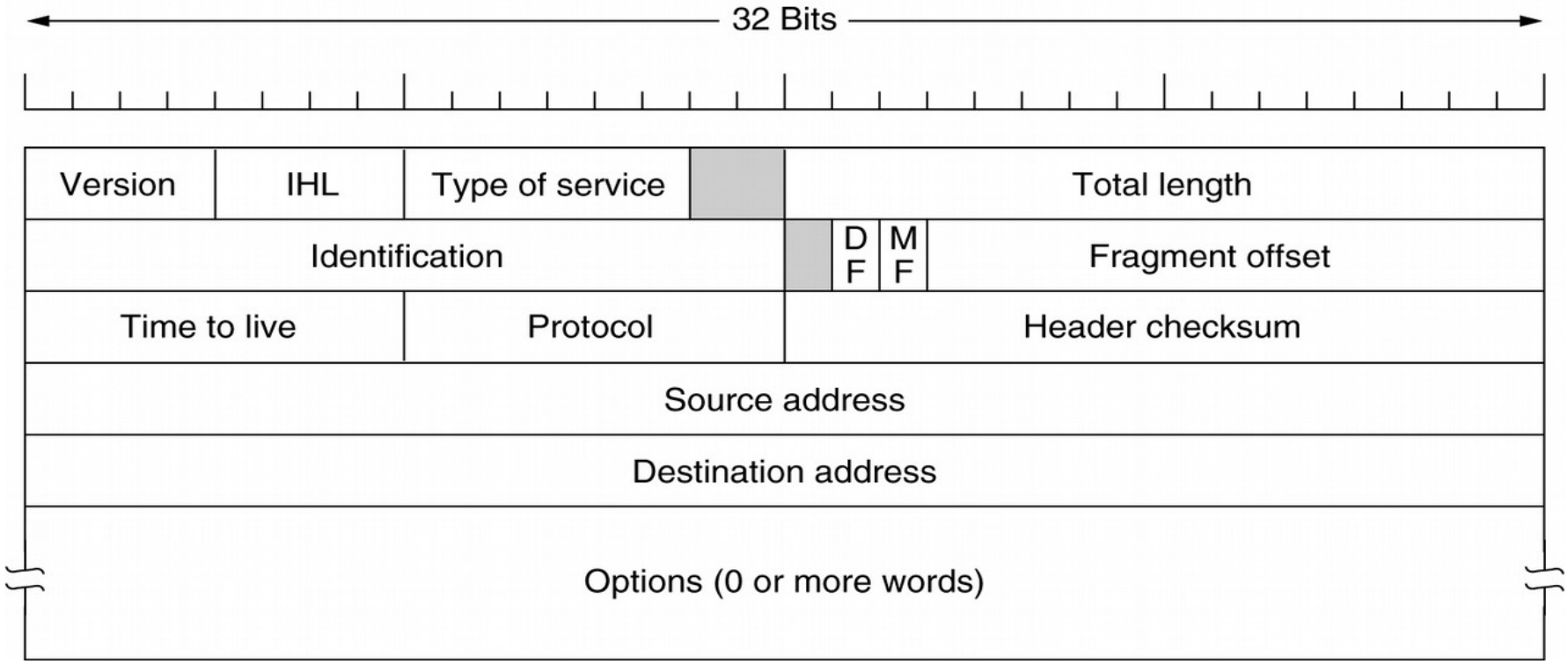
→ actuellement la plupart des sous-réseaux filaires sont Ethernet avec trames de 1500 octets max

→ l'Internet est dimensionné pour accepter MTU=1500 octets

→ très peu de fragmentation au coeur de l'Internet

→ fragmentation très présente en technologies sans fil capilaire

En-tête IP : au début de chaque paquet



The IPv4 (Internet Protocol) header.

IPv4 évolutions

- **DNS**
- **DHPC**
- **La plupart des paquets issus de Ethernet**
- **Utilisation du champ option très rare**
- **Plaques continentales**
- **Gros providers, CDN, ...**

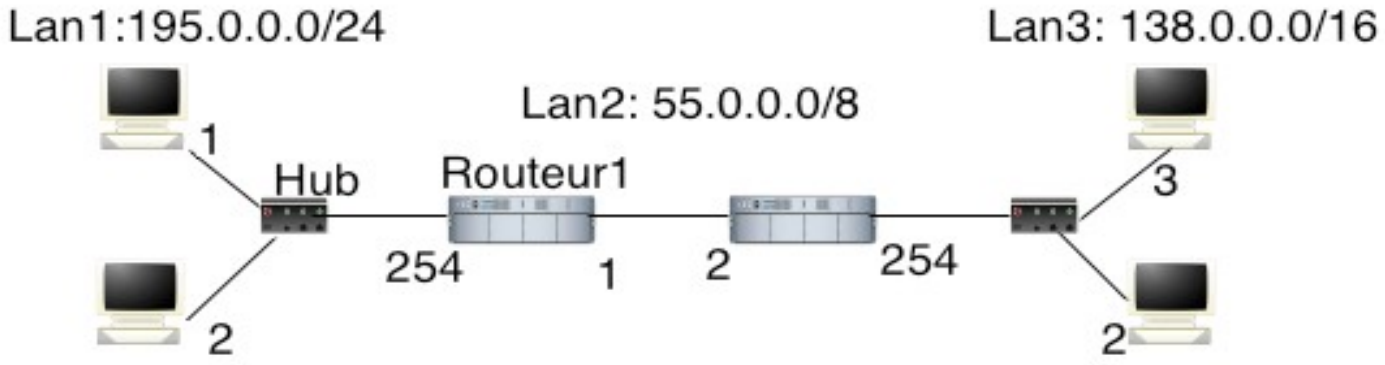
Le routage dans Internet

- **Le routage se décide en regardant la partie réseau de l'adresse**
- **Rappel : la résolution dans le sous-reseau LAN, niveau liaison, se fait par le protocole (ARP Address resolution protocol) à l'arrivée sur ce réseau local**
- **Pas de correspondance adresse-emplacement géographique jusqu'en 93**
- **Depuis des efforts sont faits pour simplifier le routage**

La décision dans IP grâce à une table de routage:

- **Table de routage :**
  - **Adresse destination (que la partie @réseau), netmask, @IP routeur voisin, coût (métrique)**
- **Parcourir la table de routage à l'arrivée d'un paquet :**
  - **Pour chaque ligne de la table de routage (adresse destination, netmask, adresse routeur voisin, coût) faire**
    - **Si adresse destination du paquet AND netmask = adresse destination alors envoyer le paquet au routeur voisin correspondant**

Exemple

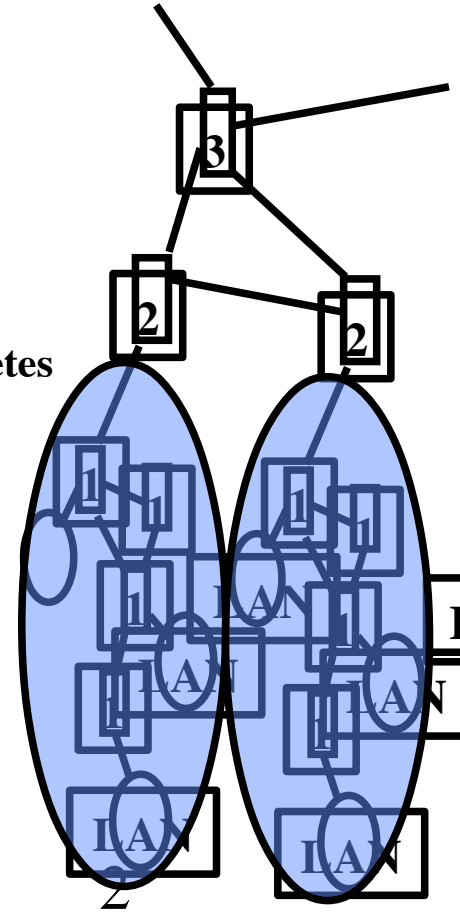


- **Exemple de tables de routage :**
  - **Routeur 1:**
    - **195.0.0.0 255.255.255.0 direct**
    - **55.0.0.0 255.0.0.0 direct**
    - **138.0.0.0 255.255.0.0 55.0.0.2**
  - **Machine d'adresse 195.0.0.1:**
    - **195.0.0.0 255.255.255.0 direct**
    - **55.0.0.0 255.0.0.0 195.0.0.254**
    - **138.0.0.0 255.255.0.0 195.0.0.254**

- **Que se passe t-il lors d'un ping de 195.0.0.1 vers 138.0.0.2 ?**

Internet : un routage hiérarchique

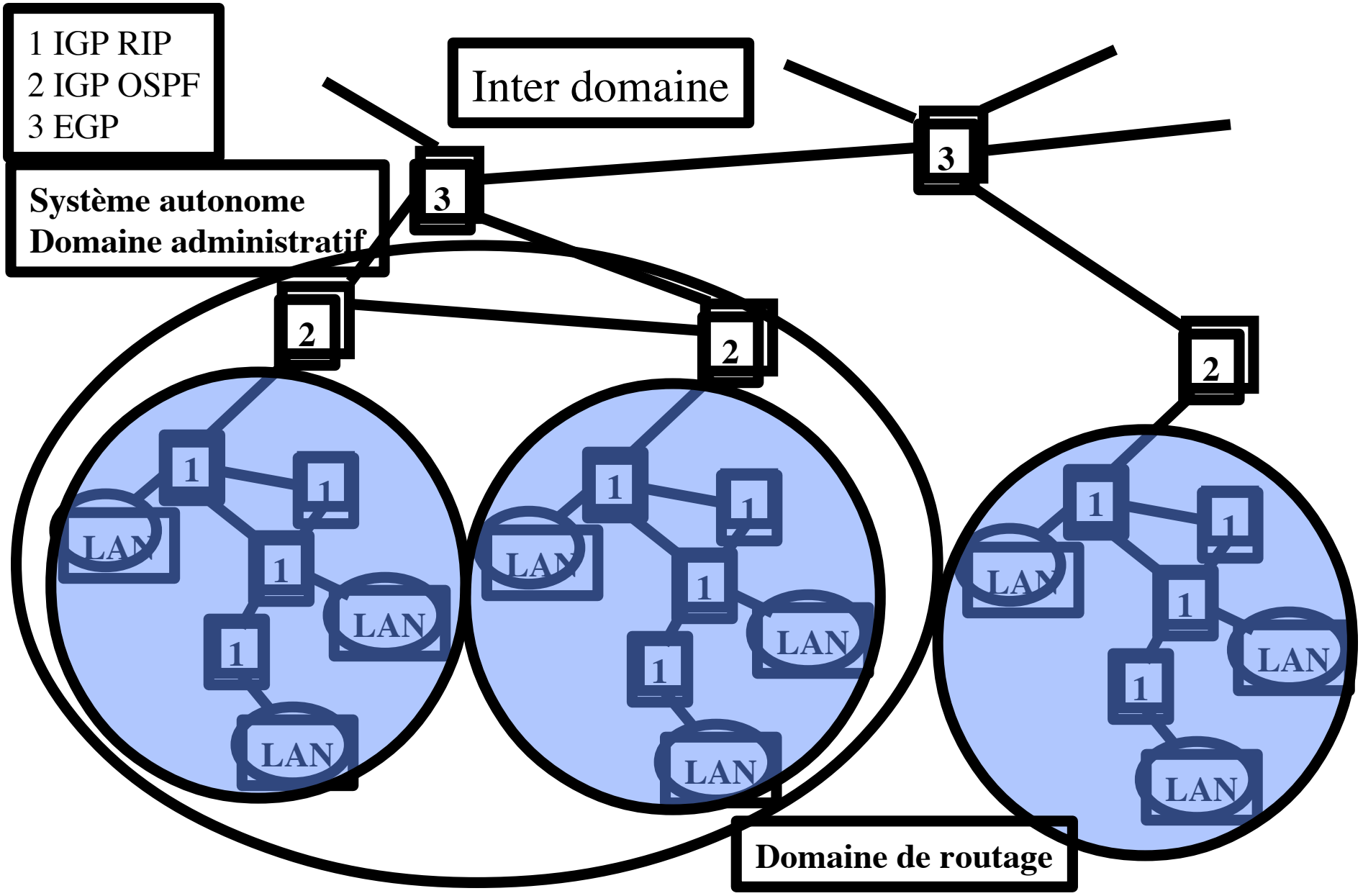
- Problème si réseau grand → table de routage grossit → pb mémoire et calcul
- Solution :
  - Regroupement par « zone » des routeurs
  - Malheureusement adresse Internet IPv4 ne donne pas la zone
- Chaque table de routage contient le moyen de se diriger
  - Vers les réseaux des routeurs de sa zone
  - Vers au moins un routeur de niveau supérieur : ligne nommée « default »
  - NB : au sommet les routeurs possèdent des tables de routage quasi-complètes
- Cette hierarchie peut avoir beaucoup de niveaux
  - Routage pas obligatoirement suivant chemin optimal
  - Choix des métriques
- Dans Internet trois types de routages
  - suivant le niveau se trouve le routeur



Organisation dans Internet

- **Système autonome (Internal Gateway Protocol IGP):**
  - **Géré par une autorité administrative**
  - **Décide des protocoles de routage → domaine de routage**
  - **Attribue les adresses IP**

  2 types pour les systèmes autonomes
  - **Niveau 1: Réseau à une échelle petite/moyenne jusqu'aux équipements terminaux**
    - **Ex: Université, grosse entreprise.**
    - **Beaucoup de variations , peu d'adresses à connaître → dynamique**
  - **Niveau 2: regroupement de réseaux terminaux (stub networks)**
    - » **Ex: Fournisseurs d'accès, routeurs inter-universitaire (RENATER)**
      - **Une ville, une région**
      - **Peu de variation, moins de chemins, mais plus d'adresses → plus statique**
- **Routeur Inter-domaine (External Gateway EGP):**
  - **un pays et inter pays**
  - **« plaques continentales »**


1 IGP RIP
2 IGP OSPF
3 EGP
Inter domaine
Système autonome
Domaine administratif
3
3
2
2
2
1
1
1
1
1
LAN
LAN
LAN
1
1
1
1
1
LAN
LAN
LAN
1
1
1
1
1
LAN
LAN
LAN
Domaine de routage

Protocoles de routage IP

- **Dans les systèmes autonomes : IGP Interior Gateway Protocol**

Intra-Domaine

- **Topologie très variable → Dynamique**
- **Information Locale & calcul léger → faible surcharge**
- **Vision partielle du réseau → Type Distance-Vector**

→ protocole RIP (Routing Internet Protocol)

Inter-domaines

- Topologie plutôt stable/fixe mais plus d'adresses à connaître

- Diffuse ces adresses aux autres routeurs de même niveau

- Métrique opérationnelle : débit, charge ...

- Information globale (plus stable): Type état de lien,

→ protocole OSPF (Open Shortest Path First)

- **Inter domaine: EGP Exterior Gateway Protocol**

- **Topologie fixe, Critères stratégiques et politiques**

→ protocole BGP (Border Gateway Protocol)

Exemple le protocole de routage RIP

- **Type vecteur de distances :**
- **les routeurs n'ont pas une vue globale du réseau, ils s'échangent le vecteur des distances qu'ils connaissent entre voisins**
- **Un des premiers protocoles utilisé dans Internet pour le routage interne au domaine, toujours très utilisé (RIPv2 permet CIDR)**
- **Inter domaine et intra domaine possible**
- **Dynamique**
  - **Adaptation progressive aux modifications du réseau**
- **Distribué et local:**
  - **Routeurs n'ont qu'une vision locale du réseau**
  - **Echange uniquement entre routeurs voisins**
  - **Contenu des échanges les tables de routage (vecteur de distances)**

RIP principe

- **Contenu des lignes de la Table de routage**
  - **adresse destination/ masque réseau/ adresse routeur voisin/ Coût**
- **Caractéristiques**
  - **Un seul chemin par destination, pas de mémorisation des autres chemins possibles**
  - **Coût: le nombre de réseaux traversés (sauts) pour arriver à destination (+1 pour chaque routeur)**
  - **RIP détermine le meilleur chemin à mettre dans la table de routage en fonction de ce coût**
  - **On peut hiérarchiser les routeurs en choisissant l'adresse du routeur par défaut dans les tables de routages**
  - **Un routeur ne possède ainsi dans sa table que les adresses de son « niveau »**

RIP mise à jour des tables

- **Mise à jour des information de routage**
  - **Messages protocole RIP entre voisins**
  - **Table de routage locale modifiée en fonction des informations reçues des routeurs voisins**
  - **Envois par les démons de routage RIP (in.routed sous Solaris, routed sous freeBSD) dans chaque routeur**
  - **Un paquet RIP contient un vecteur de lignes (la table de routage)**

    [adresse réseau, masque, coût]
  - **Les paquets sont émis en broadcast régulièrement par chaque demons RIP**
  - **RIP utilise UDP comme protocole de transport**
- **Algorithme distribué pour construire des meilleurs chemins dans un graphe (Bellman-Ford) fait les mises à jour.**

RIP Algorithme

Chaque routeur envoie à tous ses voisins périodiquement (timer 30s) le contenu de sa table de routage: vecteur (@destination, cout)

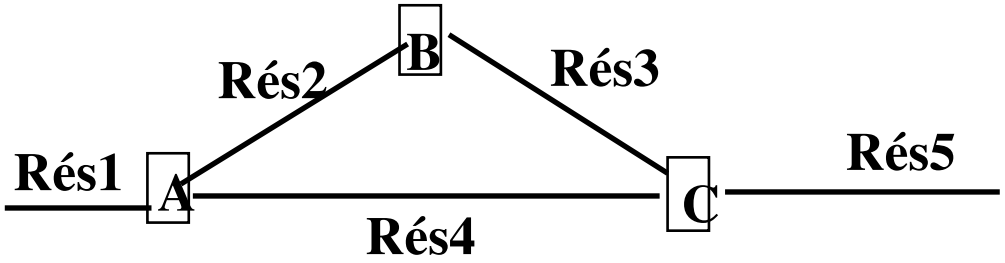
- **A la réception d'un paquet RIP arrivant d'un routeur voisin @routeur pour chaque destination (@destination, cout) contenue dans la paquet faire**
  - **Si @destination inconnue**
    - **rajouter dans la table de routage (@destination, @routeur,cout +1)**
    - **armer timer**
  - **Si @destination connue dans la table (@destination, @routeur_voisin, cout_présent))**
    - **Si @routeur_voisin = @routeur alors**
      - **changer table (@destination, @routeur, cout + 1)**
      - **relancer timer**
    - **Si @routeur_voisin ≠ @routeur et cout+ 1 < cout_présent alors**
      - **changer table**
      - **relancer timer**
- **Si sonnerie d'un timer (2mn 30) supprimer de la table la destination correspondante**

RIP initialisation

Au départ les tables de routage des routeurs sont initialisées avec l'ensemble des adresses des réseaux auxquels le routeur est connecté (cf. cartes interface) et une @default (si disponible).

- **Le coût minimum (1) est alors associé à ces adresses de réseaux destinations.**
- **Pour les équipements terminaux/hôtes seule la partie réception de l'algorithme est effectuée car ils ne peuvent pas apporter de changement.**

RIP exemple: Soit les réseaux Rés1 à Rés5 interconnectés par les routeurs A, B, C



- On suppose que l'algorithme de routage de IP a été activé sur les noeuds de ce réseau. Le coût associé à un chemin est le nombre de liens à traverser.

Table de routage de A: Adresse dest.	Voisin	coût
Rés1	direct	1
Rés2	direct	1
Rés3	B	2
Rés4	direct	1
Rés5	C	2

. Paquets RIP envoyés par A vers B et C

RIP fonctionnement

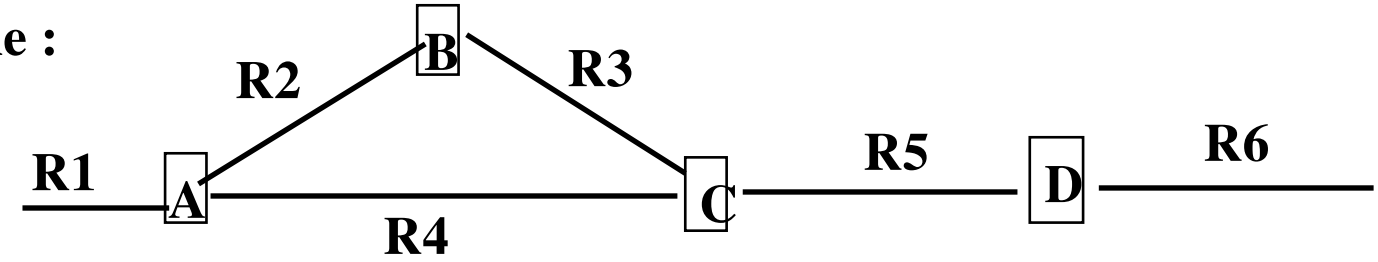
- **Pour arriver à un état stable il faut que les informations se propagent de proche en proche**
- **Le temps de stabilisation dépend du «diamètre» du réseau x 30s**
- **Au départ les tables de routage ne contiennent que les réseaux sur lesquels se trouvent les routeurs (ils connaissent leurs propres adresses Internet)**
- **Après (diamètre réseau x 30s) l'information globale est présente partout**

- **Adaptation dynamique aux modifications du réseau :**
  - **Panne d'une machine ou coupure de lien → changement de route**
  - **Attention la convergence n'est pas instantanée.**

- **Exemple: Le lien entre A et C vient d'être coupé, expliquez ce qui va se passer et donnez les nouvelles tables de routages.**

Problème de la valeur infinie

- **L'algorithme peut aussi être mis en défaut dans le cas de boucle dans le réseau**
- **Exemple :**



- **Si la ligne entre C et D est coupée, un scénario possible :**
  - **C élimine de sa table de routage R6 (plus d'info RIP de D)**
  - **Puis A élimine de sa table de routage R6 car C ne lui envoie plus d'info sur R6 .**
  - **B n'a pas encore éliminé R6 de sa table de routage (timer non synchronisé)**
  - **B envoie à A qu'il peut accéder à R6 avec un coût de 3**
  - **A va donc envoyer à C qu'il peut accéder a R6 avec un coût de 4**
  - **Cela tourne en rond en augmentant de 1 à chaque coup. La distance de A et B et C vers R6 va augmenter jusqu'à l'infini ou la valeur maximum choisie dans le protocole (15).**

- **Ce problème peut survenir dès que le réseau possède des boucles**
- **Dans RIP on retient « l'émetteur » d'une route inscrite dans la table afin de permettre les mises à jours « négatives »**
- **Le fait de limiter l'infini à un entier relativement petit limite les dégâts. Mais il doit être supérieure au nombre maximum de sauts dans le réseau complet. (RIP : 15)**
- **Pour remédier complètement au problème il faudrait avoir une vision globale du réseau**
- **Pour améliorer le protocole deux techniques supplémentaires:**
  - **Retenir des chemins:**
    - **Dans le cas où un routeur supprime de sa table une destination, pendant une durée transitoire (le temps que l'information issue du problème ait effectué le tour de la boucle éventuelle), un routeur s'interdit de prendre en compte une route pour cette destination**
    - **Après la durée transitoire on reprend l'algorithme classique**
  - **Route empoisonnée**
    - **Dans le cas précédent si la boucle est trop grande, ça ne marche pas si l'on limite l'état transitoire**
    - **Le routeur mémorise le dernier coût pour une destination, si le prochain coût (après timeout) est largement supérieur à celui mémorisé, il l'ignore.**

Routage intra-domaine entre routeurs de niveau 2

- **Type « link state » (Etat des liens)**
- **OSPF: Open Shortest Path First**
  - **Dynamique : réagit au changement du réseau**
  - **Métrique variable: nombre de saut, délais, charge des routeurs ....**
    - » **Répartition de charge**
  - **Global et distribué: Chaque routeur possède une topologie complète du réseau.**

- **Chaque routeur possède une structure de données:**
  - **graphe orienté + poids sur les arcs (métrique)**
  - **Chaque routeur calcule les plus courts chemins (Djistra) de lui vers tous les autres**

OSPF

- **Pour construire ce graphe différents types de paquets sont émis régulièrement par chaque routeur vers ses voisins:**
  - **Hello: apparition d'un nouveau routeur**
  - **Mise à jour des liens: envoie l'état des liens que le routeur connaît**
  - **Acquittement**
  - **Demande d'état de lien**
  - **Description de lien**
- **Complexe car il est impératif que tous les routeurs aient la même vision du réseau**
- **Utilise l'algorithme des plus courts chemins dans un graphe (Djistra)**

Internet Protocol v6

IPv6 (RFC 2460) = the next Internet Protocol

- **Complete redesign of IP addressing : Hierarchical 128-bit address with decoupled host identifier**
- **Stateless auto-configuration: ND, ICMP, default**
- **Simple routing and address management**
- **Flow management**
- **Mobility**
- **Multicast, anycast**
- **Security**

IPv4 vs. IPv6 Addressing

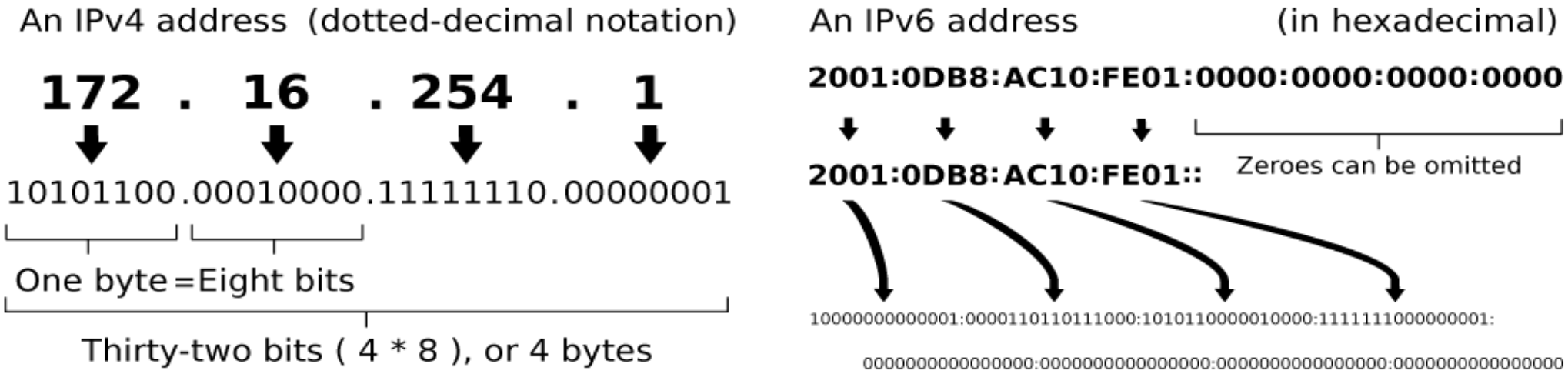


Image source: Indeterminant (Wikipeida) GFDL

Address Space Comparison

- **2^128 (IPv6) = 2^96 x 2^32 (IPv4) > 10^38**
- **2^10 = 1024 > 1k**
- **2^20 > 1M**
- **2^30 > 1G**
- **Surface Terre 510x10^6 km² = 510x10^12 m² = 510x10^24 mm²**
  → 10^16 @ par mm²
- **Nombre Avogadro 6,022x10^23 mol^-1**
- **Observable Universe**
  - **10^54 kg**
  - **5×10^80 atoms**
  - **volume 10^81 m^3**

IPv6

- **Bientôt IPV6 partout**
  - **Saturation des adresses IPv4**
  - **Adresses sur 16 octets, permet 2^128 machines**
  - **Ex: 2001:0db8:0000:85a3:0000:0000:ac1f:8001**
    2001:0db8:0000:85a3::ac1f:8001
  - **Compatible avec Ipv4**
    2001:0db8:0000:85a3:0000:0000:<@IPv4>:<@IPv4>
  - **Entete à taille fixe (cf next header) → Traitement plus simple**
  - **Peut "utiliser" @MAC, zones géographiques, etc.**
    <zone>:0db8:0000:85a3:0000:<@MAC>:<@MAC>:<@MAC>
  - **Autoconf : dhcp, ND, ...**
  - **Securité: IPsec**

Hierarchical Address space

- **Simple routing with mask and prefix length**
- **Field 1: 48b public topology network prefix**
- **Field 2: 16b local topology subnet number**
- **Field 3: 64b Interface id (cf MAC@ 48b) host number**

IPv4 vs. IPv6 Header

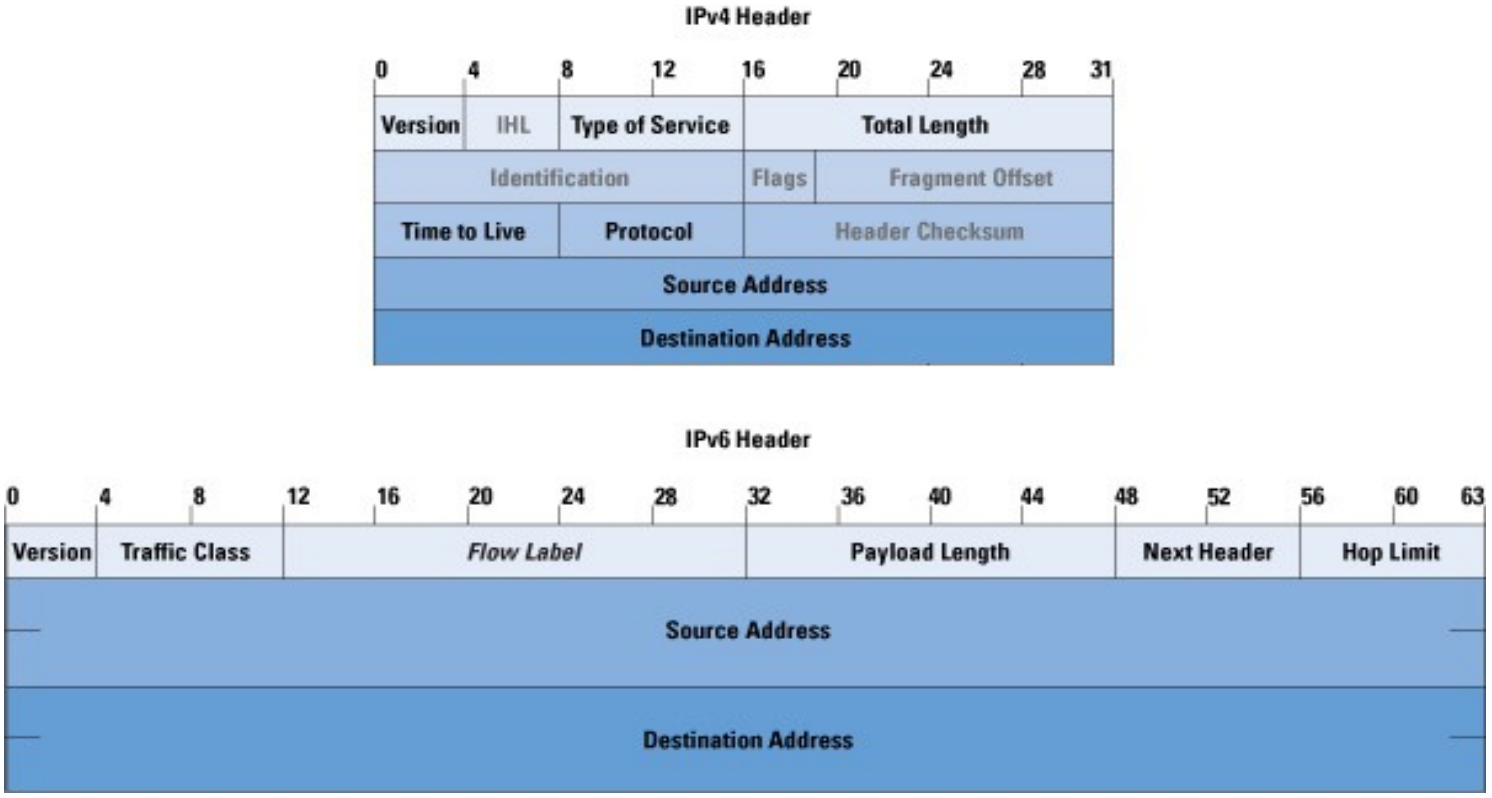


Image source: Bino1000, Mkim (Wikipeida) GFDL