IPP: Internet

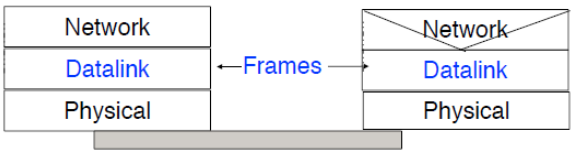
Résumé

Marc de Burlet

01/06/2016

Table des matières

DataLink Layer



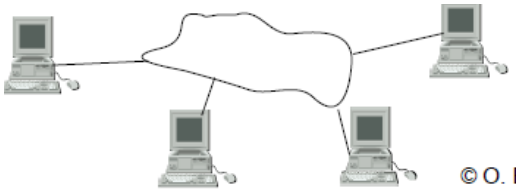
Types de dataLink Layer

Wan :

- PPP et HDLC
- Echange fiable de frames entre 2 hôtes reliés par le même lien

Lan:

- Ethernet, "Token ring", FDDI, WiFi et Wimax
- Echange de frames entre des hôtes reliés au même LAN
- Zone géographique limitée



Services de la dataLink Layer

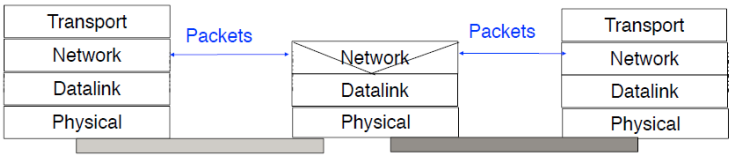
Service non fiable sans connection

- Transmission de frames entre des hôtes directement connectés au layer physique ou au même LAN
- Les frames peuvent être perdus (mais souvent détecté)
- La plupart des dataLink layers ont une taille de frame maximum.

Service fiable ou non avec connection

- Transmission de frames entre des hôtes directement connectés au layer physique ou au même LAN

Network Layer



- Chaque hôte/routeur est identifé grâce à son **adresse de couche réseau** qui est indépendant de l'adresse de la couche dataLink.
- La couche réseau envoit des paquets d'une source à sa destination au travers de plusieurs routeurs.
- Le service de la couche réseau doit être complètement indépendant du service fourni par la couche dataLink.
- Les utilisateurs de la couche réseau ne doivent pas connaître son implémentation pour pouvoir envoyer des paquets.

But: Permet d'envoyer des paquets d'une source jusqu'à une destination aux travers de réseaux et de routeurs.

Services: service non fiable sans connection ou service fiable avec connection

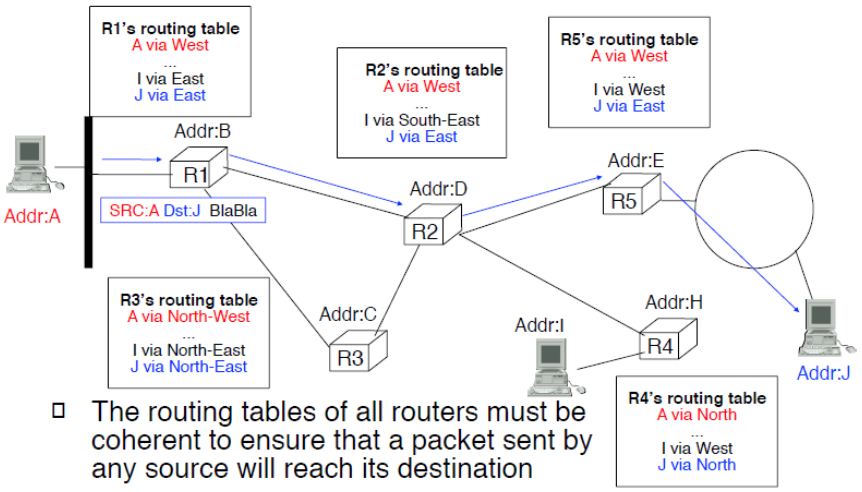
Organisation interne de la couche réseau

Datagrams

- Sert à fournir un service sans connection.
- Chaque routeur/hôte est identifié par une adresse
- L'information est divisée en paquets
- Chaque paquet contient:
  - L'adresse de la source
  - L'adresse de la destination
  - Le Payload

Comportement du routeur:
A l'arrivée d'un paquet, le routeur regarde l'adresse de destination et la table de routage pour décider où le paquet doit être renvoyé. Chaque routeur doit prendre une décision de forwarding (hop-by-hop forwarding)

Exemple: IP (IPv4 et IPv6), CLNP, IPX



Circuits virtuels

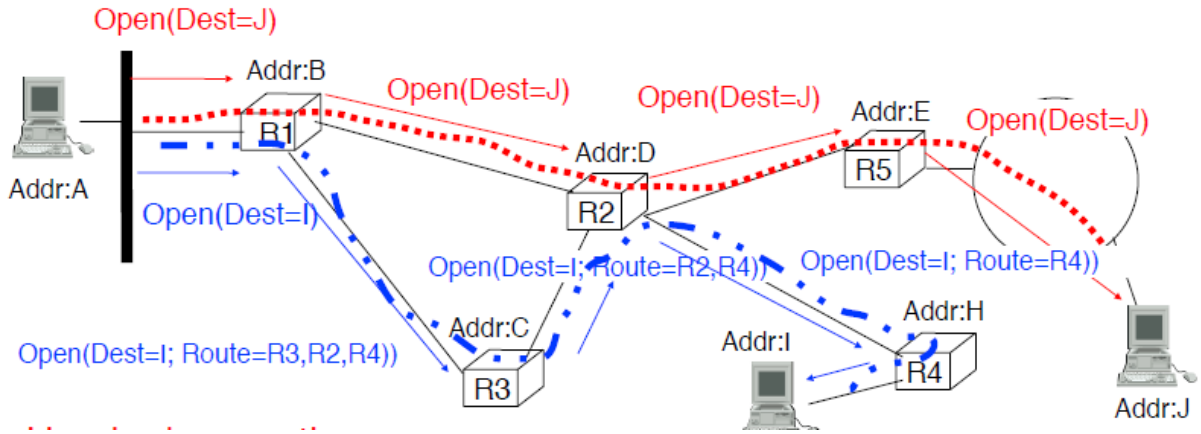
- Sert à fournir un service orienté connection.

But: Garder au plus simple le forwarding aux routeurs. (Consulter la table de routage pour chaque paquet coûte cher en performance)

⇨ On crée donc un circuit virtuel qui lie la source à la destination avant d'envoyer les paquets.
- Tous les paquets suivront le même chemin

Exemple: ATM, X.25, Frame Relay, MPLS, gMPLS

Etablissement d'un circuit virtuel



En rouge:

- Hop-by-hop routing
  - Chaque routeur consulte sa table de routage pour renvoyer les paquets

En bleu:

- Routage explicite
  - La source (ou le 1er routeur) indique un circuit virtuel que le paquet va emprunter pour arriver à destination

Transmission des paquets

Les paquets contiennent:

- **identifiant de circuit virtuel**
  - Unique pour un lien donné
    - Plus facile à gérer et pas besoin de concordance
    - identifiant doit parfois être changé en fonction du lien
  - Pour modifier l'identifiant:
    - Chaque routeur possède une "**label forwarding table**" qu'il met à jour à chaque fois qu'un circuit virtuel est établi.
- leur **payload**

Label forwarding table

Inport	Inlabel	Outport	Outlabel
West	L1	East	L1
West	L2	East	L3
N-W	L1	North	L1

Routage

- Les routeurs servent de relais entre les dataLink layers
- L'unité de transmission est le **paquet**

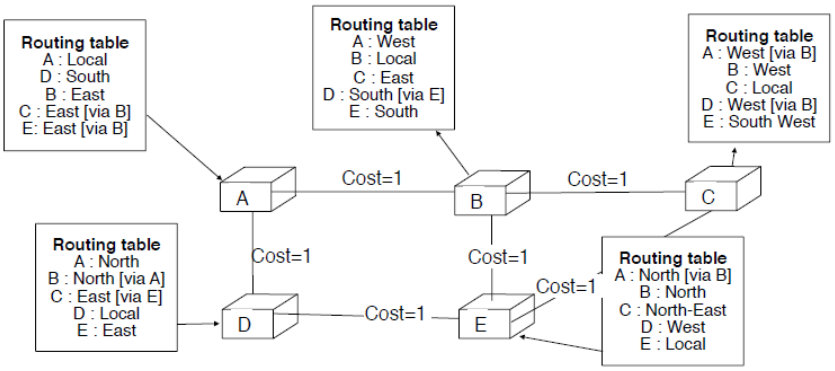
Deux mécanismes sont utilisés dans la couche réseau:

- Forwarding
  - Algorithme utilisé par chaque routeur pour déterminer, grâce à sa table de routage, sur quelle interface les paquets doivent être envoyés pour atteindre sa destination ou suivre son circuit virtuel.
- Routage
  - Algorithme qui distribue à chaque routeur les informations nécessaires pour leur permettre de construire leur table de routage.

Selection du chemin le plus court

Principe:

- Associer un coût à chaque lien
- Chaque routeur choisit le chemin le moins cher.



Static Routing

Principe:

- Hardcoder la table de routage dans chaque routeur.

Avantages:

- Facile à utiliser sur des petits réseaux
- On peut optimiser les tables de routages

Désavantages:

- Ne s'adapte pas dynamiquement à la charge du réseau
- Ne gère pas les erreurs de liens ou de routeur

Dynamic Routing

Principe:

- Les routeurs échanges des messages et utilisent un algorithme distribué pour construire leur table de routage

Avantage:

- Adapte facilement les tables de routages en fonction des évenements.

Désavantages:

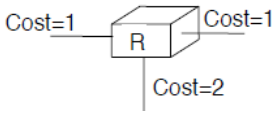
- Plus complexe à mettre en place

Méthodes de routage les plus courantes:

- Distance vector routing
- Link state routing

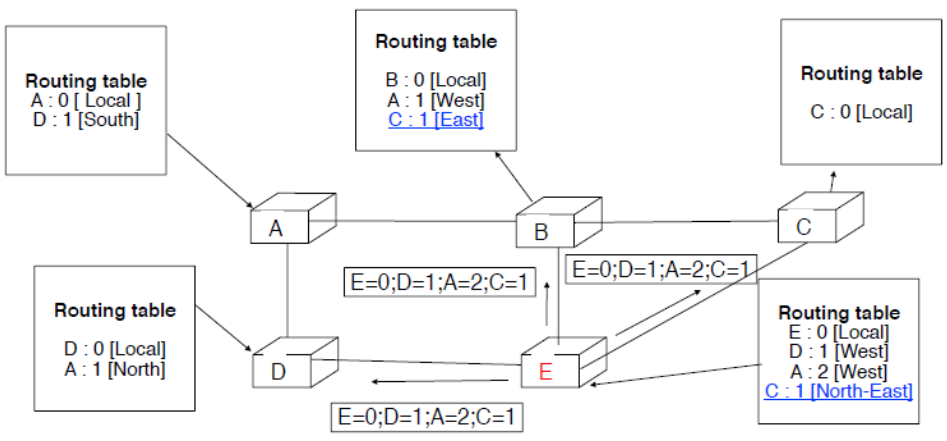
Distance vector routing

Principe:

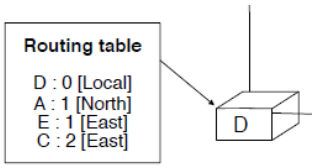


- Configuration du routeur:
  - Coût pour chaque lien
- Au lancement, un routeur ne connait que lui-même.
- Chaque routeur envoit périodiquement à ses voisins un vecteur qui contient pour chaque destination qu'il connait:
  - L'adresse de destination
  - La distance entre le routeur et la destination
    - coût total du chemin le plus court pour atteindre la destination.
- Chaque routeur met à jour sa table de routage en fonction des informations qu'il reçoit de ses voisins.

Exemple:



- **E** envoit à ses voisins un vecteur avec les infos qu'il connaît
- Les voisins vont mettre à jour leur table de routage



Cassure d'un lien

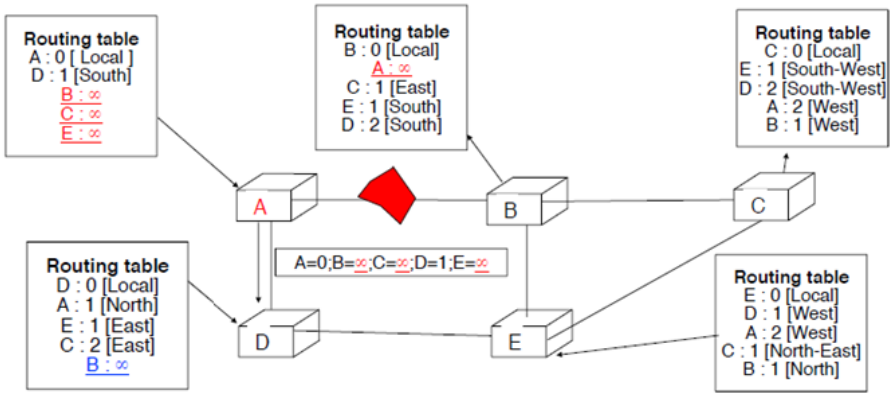
Détection

- Compter sur l'echec d'information de la couche datalink ou de la couche physique
  - Rapide et fiable
  - Pas supporté par toutes les couches datalink/physique

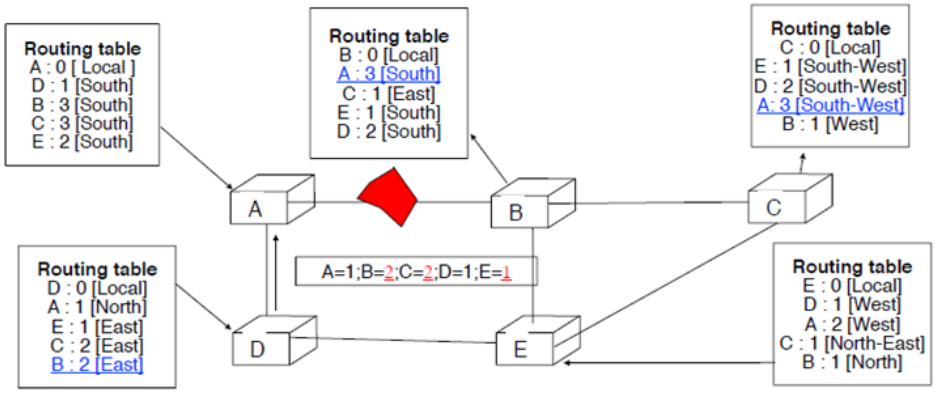
- Chaque routeur doit envoyer à interval régulier son vecteur de distance
  - Si un routeur ne reçoit pas un vecteur d'un de ses voisins pendant un certain temps, il considère que ce routeur voisin n'est plus disponible.

Mise à jour des tables de routage

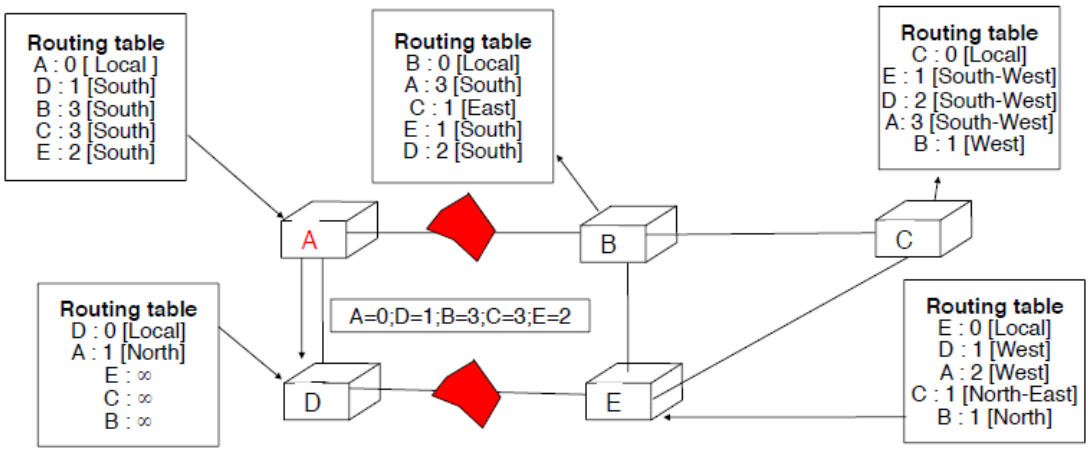
- On remplace le coût des chemins qui passent par le lien cassé par un coût ∞.



- Les tables de routeurs seront remises à jour grâce aux nouveaux vecteurs reçus.



Cassure d'un second lien



Problème:

- **A** envoit son vecteur avant **D.**
- **D** va alors remettre à jour sa table avec les informations de **A.**
- **D** renvoit son vecteur et **A** remet sa table à jour ...

⇨ Comptage à l'infini

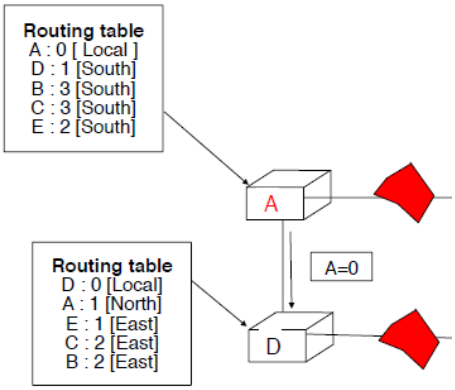
Solution: split horizon

Split Horizon

On envoit un vecteur différent pour chaque voisin. Chaque vecteur ne contriendra pas les informations concernant directement le destinataire.

Exemple:

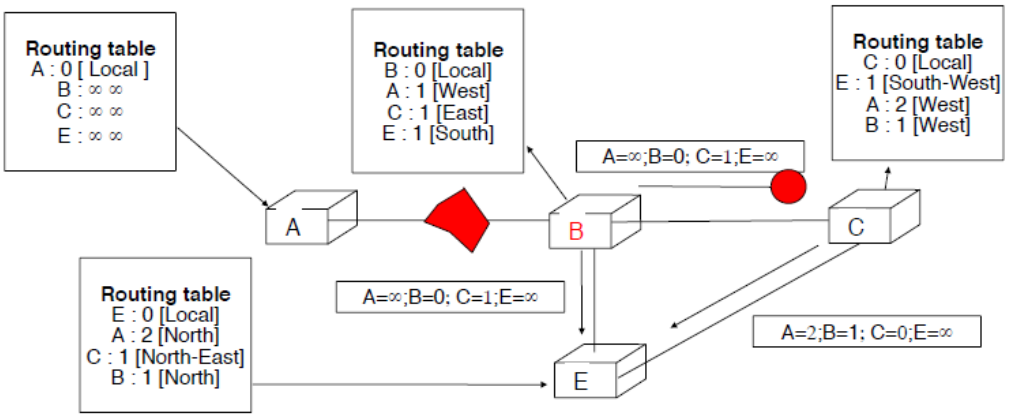
A n'envoit pas les infos concernant **D,B,C** et **E** car ces informations concernent directement **E**.



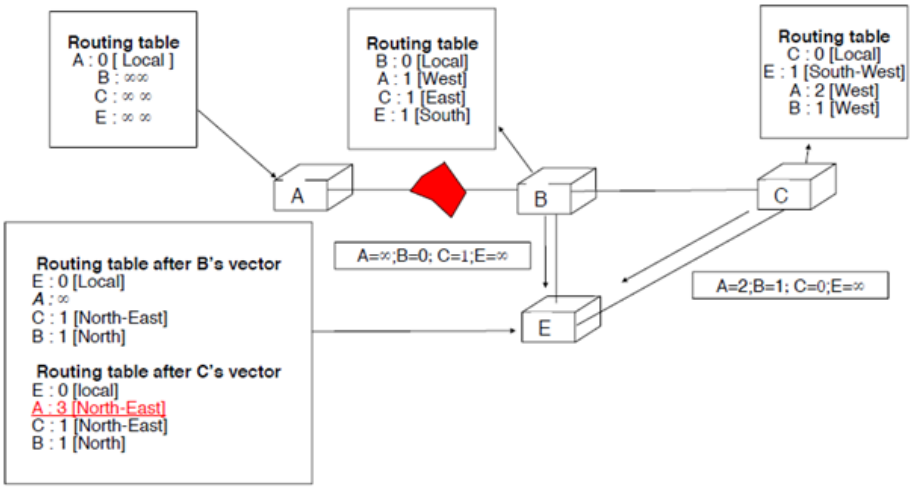
Amélioration:

On envoit un coût infini pour les informations concernant directement le voisin auquel on envoit le message. (**split horizon avec empoisonnement**)

Limitation du split horizon



- **B** envoit ses vecteurs mais **C** ne le reçoit pas et ne met donc pas à jour les informations concernant **A** qui est injoignable



- **E** reçoit le vecteur de **B** et place **A** en "injoignable" mais il reçoit le vecteur de **C** qui lui dit que **A** est joignable via **C**.
- **E** va envoyer ses vecteurs
- **B** pensera que **A** est joignable via **E**

⇨ On retombe sur le problème de comptage à l'infini.

En pratique on va ajouter une borne maximum, qui permet d'enlever une liaison si le coup d'un trajet dépasse cette borne.

Link State Routing

Principe

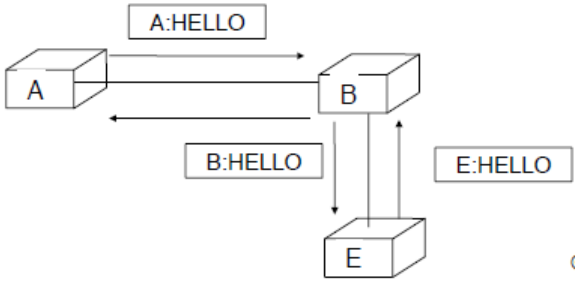
On envoit une carte du réseau complet au lieu d'un résumé.

- Chaque routeur doit découvrir ses voisins.
- Chaque lien doit avoir un coût.
- Chaque routeur envoit sa topologie locale à tous les routeurs et traite les informations reçues des autres routeurs.
- Les routeurs construisent un graphe du réseau en utilisant l'algorithme de Dijkstra afin de trouver les chemins les plus courts.

Hello packets

- Par configuration manuelle
  - Pas fiable et compliqué

- En utilisant les "Hello packets"
  - Toutes les N secondes, chaque routeur envoit un Hello packet à chacun des liens avec son adresse
  - Les voisins repondent en envoyant leur adresse
  - La transmission périodique permet de vérifier que le lien est toujours actif et de détecter les erreurs.



Coût des liens

Chaque lien possède un coût mais il faut déterminer ce coût.

- Unité de coût
  - Solution la plus simple mais applicable uniquement pour les réseaux homogènes

- En fonction de la bande passante
  - coût plus cher pour une bande passante faible et inversement.

- En fonction du délai
  - Souvent utilisé pour éviter les liens satellites

- Coût basé sur des mesures
  - On utilise Hello pour mesurer le coût

- Permet de mesurer la charge des liens mais peu changer la topologie du réseau si les mesures ne sont pas stables

Assemblage de la topologie du réseau

- En recevant les Hello packets, chaque routeur construit sa partie locale de la carte du réseau
- Chaque routeur résume sa topologie locale dans un "Link state packet" (**LSP**) qui contient:
    - l'identification du routeur
    - ses pairs (l'identification de ses voisins ainsi que le coût des liens)

Ces paquets (LSP) sont envoyés:

- Lorsqu'il y a un modification de topologie
    - Permet d'avertir les autres routeurs du changement
- Toutes les N secondes

- Chaque routeur envoie sa topologie locale dans des LSP sur tous ses liens.
- Quand un routeur reçoit un LSP, il le forward à tous ses liens sauf celui dont il a reçu le paquet.

➢ On risque un LSP flooding

LSP flooding

Les routeurs qui reçoivent un LSP, le forward à ses autres liens mais il faut s'arrêter à un moment pour éviter un LSP flooding.

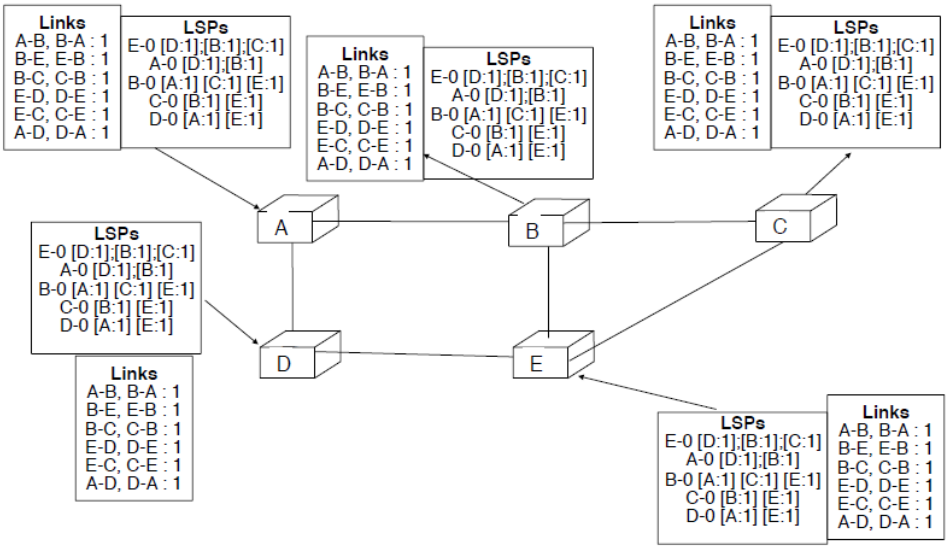
Solution:

- On place dans le LSP:
    - Un numéro de séquence lors de la création du LSP
    - L'adresse du créateur du LSP
    - L'adresse des voisins du créateur

- Chaque routeur doit conserver le dernier LSP reçu par chaque routeur du réseau.

- Un LSP n'est traité que si il est plus récent que le dernier LSP reçu par ce routeur

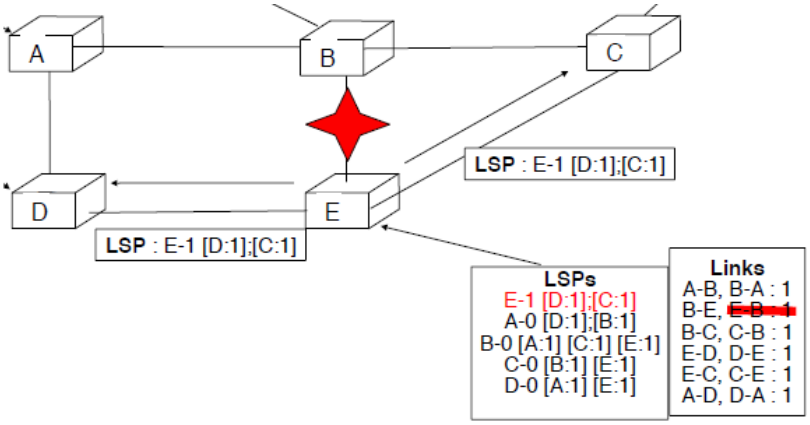
Exemple de topologie complete



Remarques:

- Chaque routeurs contient les liens dans les deux sens
- Chaque routeur contient le dernier LSP envoyé par chaque routeur.

Cassure d'un lien



Pour gérer la cassure d'un lien, on utilise le principe de **two-way connectivity check.**

- Un lien n'est alors utilisable que si les deux directions ont été annoncées.

Erreurs de routeur

- Si un routeur **crash**, ses interfaces ne sont plus utilisables et ne réponderont plus aux Hello packets
- Si un routeur **reboot**, il va envoyer un LSP avec le numéro de séquence 0 et les autres routeurs ne mettront pas à jour leur table à jour si ils ont un LSP plus âgé.

Solution:

On attribue un âge à chaque LSP (même ceux stoqués dans la LSDB) et on décrémente régulièrement cet âge.

- Si l'âge = 0, on supprime ce LSP.
- Il faut envoyer régulièrement son propre LSP avec un âge > 0 pour s'assurer de rester dans le réseau

Améliorations du LSP flooding

- Eviter d'envoyer deux fois le même LSP sur un lien et laisse du temps aux autres routeurs de traiter le LSP
    - Réduit le nombre de LSB envoyés
    - Mais augmente le temps de flooding

- Flooding fiable
    - Placer un CRC dans les LSP pour détecter les erreurs
    - Une confirmation sur chaque lien pour les LSP échangés sur ce lien
        - Chaque transmission est protégée par un timer

- Synchro/echange de Link state Databases
    - Les routeurs peuvent comparer le contenu de leur LSDB et echanger seulement les LSP manquants des voisins
        - Pratique lorsqu'un router boot et veux rapidement recevoir tous les LSP du réseau

Algorithme de Dijkstra

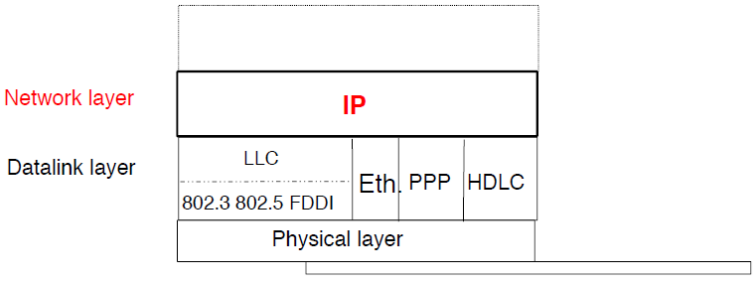
https://www.youtube.com/watch?v=rHylCtXtdNs

IP : Internet Protocol

IP version 4

Internet network layer:

- Fournit un service non fiable sans connection



Principe:

Mode datagram

- Chaque hôte est identifié par son adresse IP (encodée sur 32 bits)

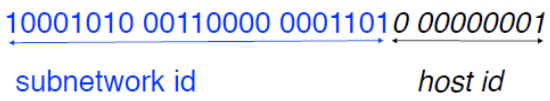
10001010 00110000 00011010 00000001
138 . 48 . 26 . 1

- Chaque hôte sait comment joindre au moins un routeur
- Les routeurs savent joindre d'autres routeurs

Adresses IP

Allocation hiérarchique des adresses IP

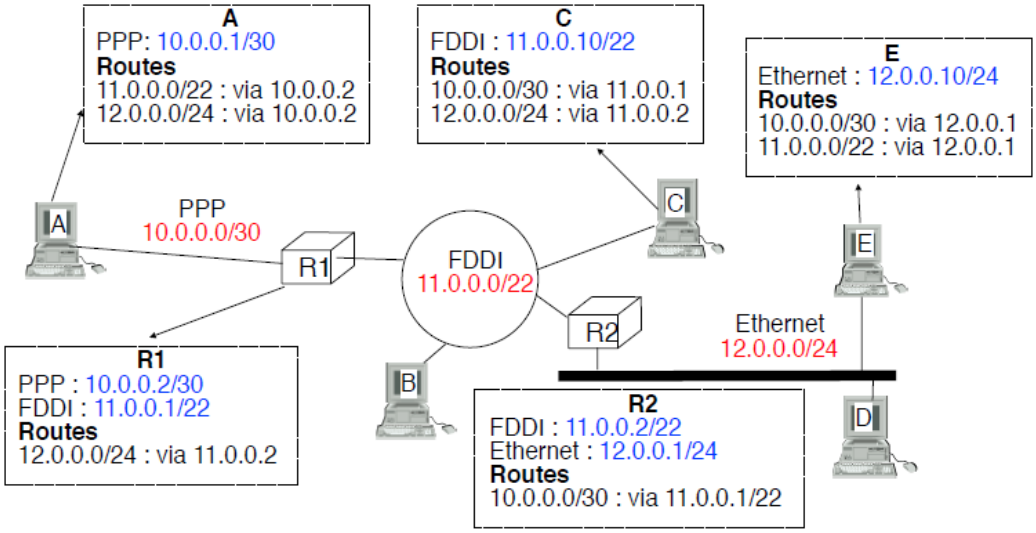
- Une adresse par interface
- Chaque adresse est composée de 2 parties
  - l'identifiant subnetwork
    - Les **M** premiers bits
  - l'identifiant de matériel dans le subnetwork
    - les 32 – **M** bits



Notation 138.48.26.1/23 or 138.48.26.1 255.255.254.0

Tous les hôtes qui appartiennent au même sous-réseau peuvent échanger directement des frames via la couche datalink.

Exemple d'adressage IP:



Désavantages des sous-réseaux:

- La plupart des sous-réseaux ne sont pas complètement remplis.
- Un réseau de campus requiert plus d'adresse IP que le nombre d'hôtes attachés au réseau.

La plupart des adresses sont alouées par l'**IANA** et les registres nationaux RIPE, ARIN,...

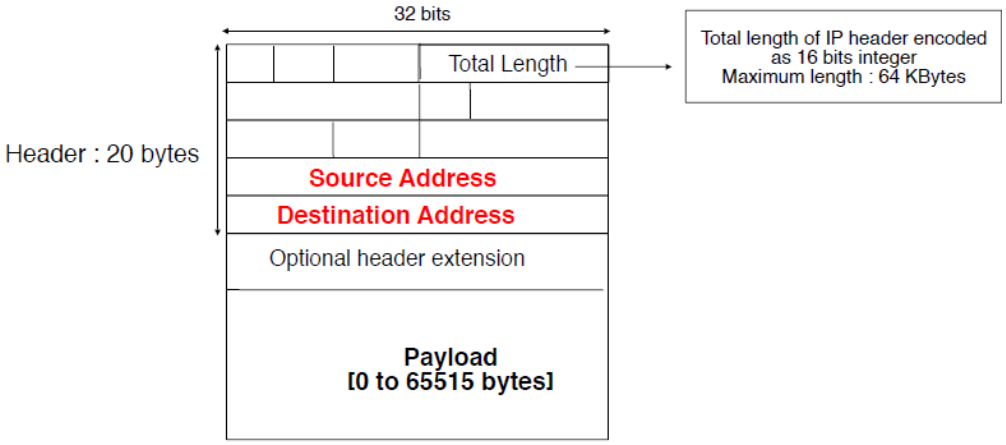
Adresse spéciales:

- 127.0.0.1
  - localhost
- 10.0.0.0/8, 172.16.0.0/12 et 192.168.0.0/16
  - Utilisées pour des réseaux privés (pas directement reliés à Internet)

- 218.0.0.0/8 – 223.0.0.0/8 et 240.0.0.0/8 – 255.0.0.0/8
    - Utilisée pour d'autres utilisations
- 224.0.0.0/8 - 239.0.0.0/8
    - IP multicast
- 255.255.255.255
    - adresse de broadcast
- 0.0.0.0
    - lors du boot, avant de connaître son adresse

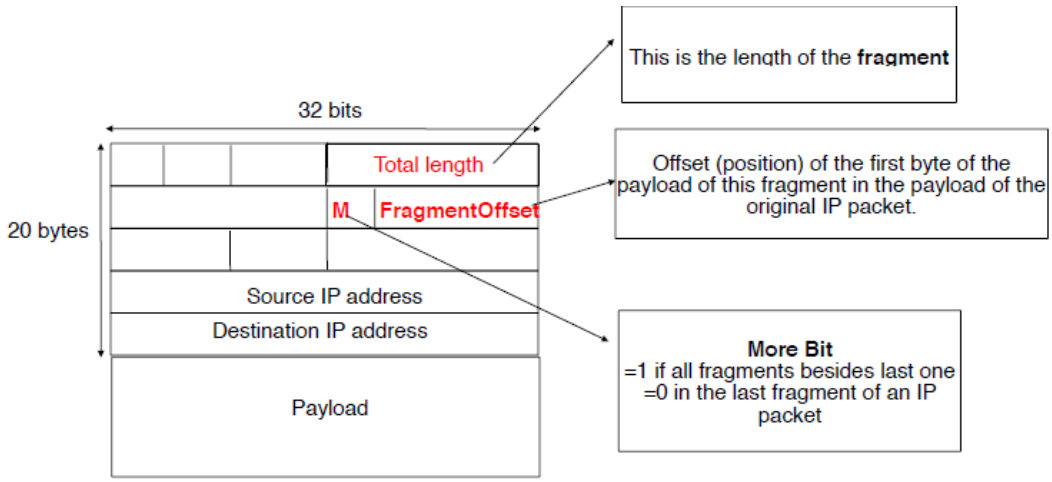
IP paquets

Format:



Fragmentation des paquets

Pour envoyer des longs paquets, les hôtes et les routeurs peuvent **fragmenter** le payload. Chaque fragment sera un **paquet complet** (header complet recopié + payload (1,2,...). C'est l'hôte de **destination qui réassemblera** les paquets.



1 Format d'un fragment de paquet

Problèmes de réassemblage

- Si un fragment est perdu
  - Le paquet ne sera pas reconstruit et les fragments reçu seront rejetés
- Mauvais ordonnancement
  - L'Offset révient cela pour un même paquet mais si il peut y avoir plusieurs paquets fragmentés et donc plusieurs fragments avec le même offset
  - ➢ Chaque fragment doit contenir un identifiant du paquet dont il provient

La fragmentation de paquets peut provoquer des erreurs, pour éviter cela, les routeurs indiquent la taille maximum des paquets qu'ils supportent pour que la source des paquets envoit directment des paquets de la bonne taille.

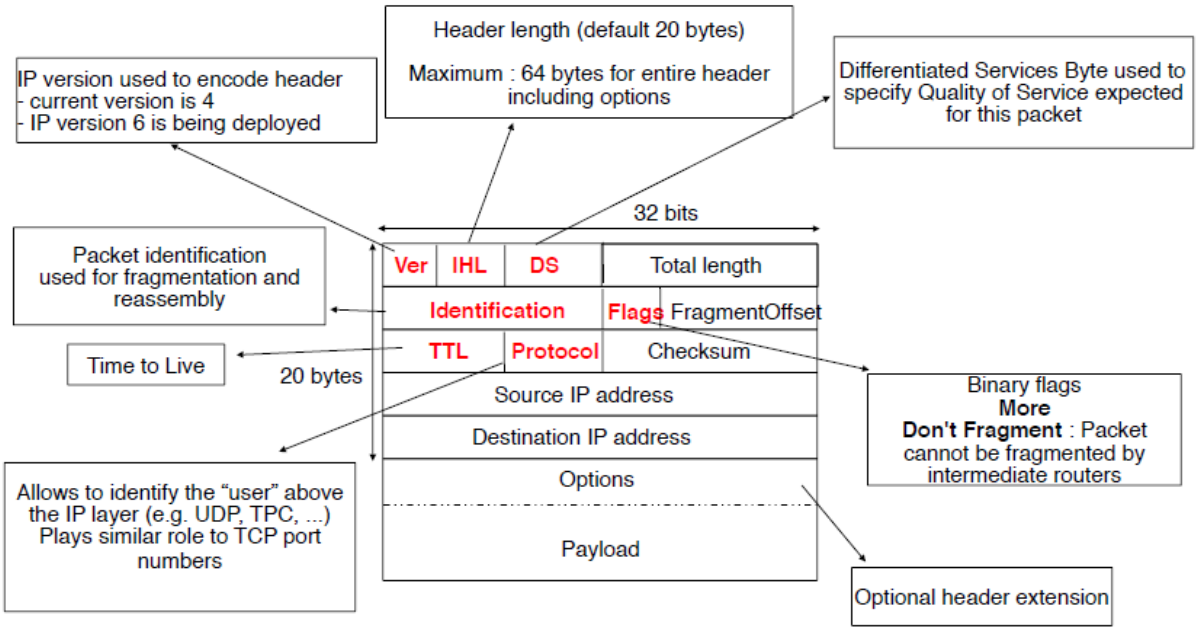
Erreurs de transmission

- Dans le contenu du paquet (**Payload**)
  - Certaines appli peuvent continuer malgré l'erreur
  - IP: aucune détection d'erreur de transmission dans le payload

- Dans le **header**
  - Peut provoquer plus de problème (mauvais destinataire,...)
  - ➢ Il y a donc un **checksum** pour détecter les erreurs dans le header
    - checksum de 16 bits (comme TCP/UDP)
    - Chaque routeur et chaque hôte vérifie le checksum des paquets qu'il reçoit et il écarte ce paquet si il trouve une erreur de checksum

- On peut encore avoir des **boucles** lorsque les tables de routage se mettent à jour.
  - ➢ On rajoute donc un **Time-to-Live** dans le header du paquet. Ce **TTL** correspond au nombre d'intermédiaires par lesquels le paquet passer.
    - Généralement TTL = 32 ou 64
    - Chaque routeur regarde le TTL
      - Si TTL = 1, le paquet est écarté et la source est prévenue
      - Si TTL > 1, le paquet est forwardé et le TTL est décrémenté d'au moins 1

Format du header d'un paquet IP



Options:

- **Strict**
  - Permet à la source de lister les adresses IP de **chaque** intermédiaire pour atteindre la destination
- **Loose**
  - Permet à la source de lister les adresses IP de **certains** intermédiaires pour atteindre la destination
- **Record route option**
  - permet à chaque routeur d'insérer son adresse IP dans le header
    - Rarement utilisé car la taille du header est limitée
- **Router alert**
  - permet à la source d'indiquer aux routeurs qu'il y a un traitement particulier à effectuer pour ce paquet

Il faut faire attention à ne pas dépasser les 64 bytes du header en rajoutant des options

Opérations d'un IP endhost

Création et envoi de message.

Informations requises pour un IP endhost:

- Les **adresses IP** de ses interfaces
  - Pour chaque adresse, le masque de sous-réseau permet au endhost de déterminer les adresses qui sont directement joignables par l'interface

- **Table de routage**

- Subnets directements connectés
  - Provient du masque de sous-reseau de sa propre adresse IP

- Routeur par défaut
  - Routeur utilisé pour atteindre toute adresse inconnue
  - Par convention, c'est 0.0.0.0/0

- Autres sous-réseaux connus par l'endhost
  - peut être configuré manuellement ou appris par des protocoles de routage

Configuration de l'adresse IP

Pour qu'un hôte connaisse son adresse IP:

- Par **configuration manuelle**
  - Utilisé pour bcp de petits réseaux

- Par **autoconfiguration RARP basé serveur**
  - **DHCP** (**D**ynamic **H**ost **C**onfiguration **P**rotocol)
    - Quand il s'attache à un sous-réseau, l'**endhost broadcast une requête** pour trouver un serveur DHCP
    - **DHCP répond** et l'endhost peut le contacter pour obtenir son adresse IP
    - Le serveur DHCP alloue une adresse IP pendant un certain temps et peut fournir des informations supplémentaires (subnet, routeur par défaut, DNS resolver,...)

    - Les serveurs DHCP peuvent être configurés pour toujours donner la même adresse IP à un endhost donné.
    - Les endhosts reconfirment leur allocation régulièrement.

- **Autoconfiguration sans serveur**
  - Utilisé par **IPv6**

Opérations d'un routeur IP

forward les messages reçus d'une interface vers une autre interface. Ils peuvent aussi emmètre des messages de temps à autre.

Informations requises pour un routeur: comme pour le IP endhost

<u>Opérations faites pour chaque paquet</u>

1. Vérifier que l'adresse de destination du paquet est bien une des adresses du routeur
   - Si oui, le paquet atteint sa destination

2. Query de forwarding des informations qui contient:
   - une liste des réseaux directement connectés ainsi que leurs masques

- une liste des réseaux joignables et les routeurs intermédiaires

3. Chercher la route la route la plus adaptée
    - Pour chaque route A.B.C.D/**M** via Rx
    - compare les **M** premiers bits de l'adresse de destination avec les **M** premiers bits des routes et garde la correspondance la plus longue
    - Forward le paquet via cette route

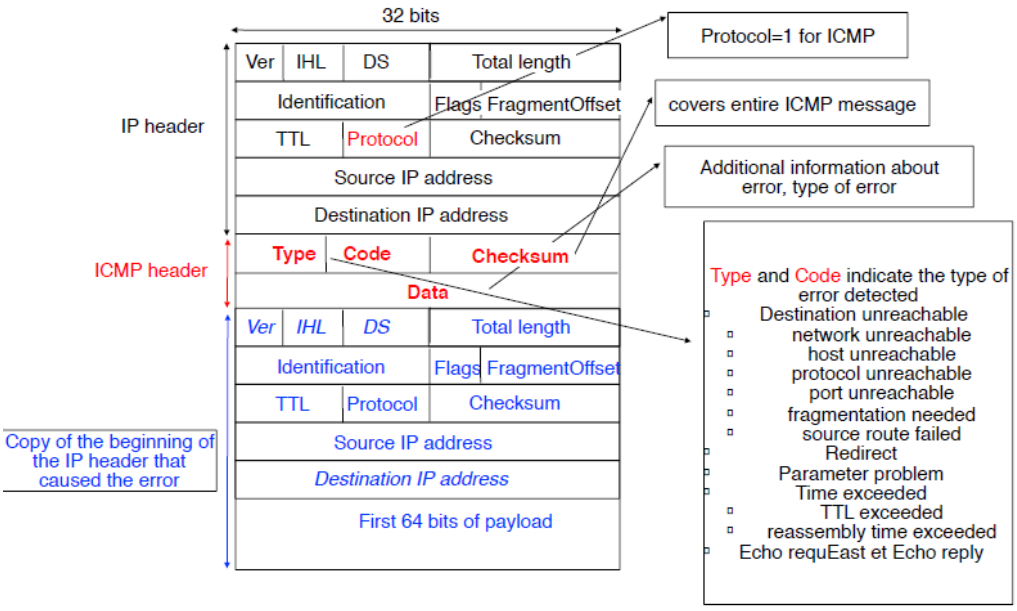
ICMP

Lorsqu'un routeur/hôte reçoit un paquet contenant une erreur, il peut :

- Ignorer et rejeter le paquet
- Envoyer un message à la source du paquet pour l'avertir du problème => ICMP

Internet **C**ontrol **M**essage **P**rotocol

- Les ICMP sont envoyés dans des paquets IP principalement par les routeurs mais parfois par des hôtes
- Pour éviter des problèmes de perfo, les routeurs/hôtes limitent la quantité des ICMP qu'ils envoient



Exemples d'usage d'un message ICMP

- **Erreur de routage**
    - Destination injoignable
    - Mauvais forwarding

- **Erreur dans le header**
    - temps écoulé
        - TTL = 0

        - utilisé par *traceroute*

    - redirection
        - pour atteindre sa destination, un autre routeur doit être utilisé et le message ICMP permet d'en avertir la source et de donner l'adresse de ce nouveau routeur

    - Fragmentation impossible

Problèmes de l'IPv4

- Fin des années 80
    - Croissance exponentielle d'Internet
- 1990
    - D'autres protocoles de réseaux existent
    - Les gouvernement poussent vers le CLNP
- 1992
    - Presque plus d'adresse IPv4

⇨ Création de l'IPv6

IP version 6

Les adresses IPv6 sont codées sur **128 bits**.
Il existe **3 types** d'adresses IPv6:

- Adresses **unicast**
    - Un identifiant pour une interface
    - Un paquet envoyé à une adresse unicast est envoyé à **l**'interface identifiée par l'adresse .

- Adresse **anycast**
    - Un identifiant pour un ensemble d'interfaces.
    - Un paquet envoyé à une adresse anycast est envoyé à l'interface la **plus proche** indentifiée par cette adresse.

- Adresses **multicast**
    - Un identifiant pour un ensemble d'interfaces
    - Un paquet envoyé à une adresse mulitcast est envoyé à **toutes** les interfaces de cette adresses

Représentation de l'adresse IPv6

- Format hexadécimal
    - FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
    - 1080:0:0:0:8:800:200C:417A

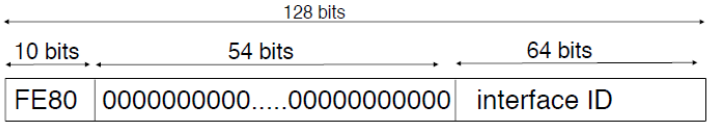
- Format hexadécimal compacte
    - Permet de simplifier les 0
    - On utilise "**::**" pour indiquer un ou plusieurs **groupes de 16 bits de 0**
        - Ne peut apparaître qu'une seule fois dans l'adresse
    - Exemples:
        - 1080:0:0:0:8:800:200C:417A = 1080::8:800:200C:417A
        - FF01:0:0:0:0:0:0:101 = FF01::101
        - 0:0:0:0:0:0:0:1 = ::1

Adresse spéciales

- Adresse non spécifiée => 0:0:0:0:0:0:0:0 ou ::
- Loopback address => 0:0:0:0:0:0:0:1 ou ::1

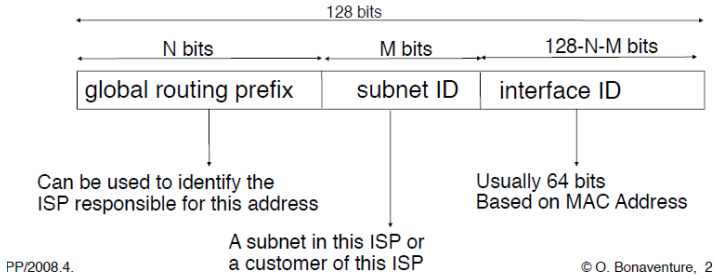
Adresse link-local IPv6

- Utilisée par les hôtes/routeurs attachés au même LAN pour échanger des paquets IPv6 quand ils n'ont pas besoin d'adresse routable globale.



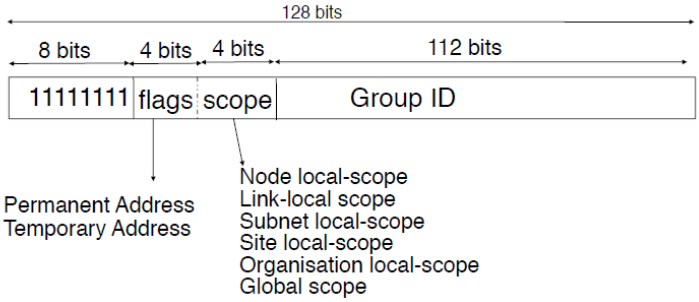
- Chaque hôte/routeur doit généré une adresse de lien local pour chacune de ses interfaces
    - ➢ Chaque hôte utilisera donc plusieurs adresses IPv6

Adresse unicast globale



Adresse multicast IPv6

Identifie un groupe de receveur.



Groupes connus:

- Tous les endsystems appartiennent automatiquement au groupe FF02::1
- Tous les routeurs appartient automatiquement au groupe FF02::2

Adresse anycast IPv6

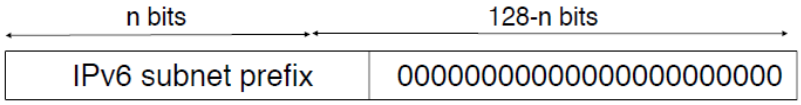
Définition: Une adresse anycast IPv6 est une adresse qui est assignée à plus d'une interface avec la propriété qu'un paquet envoyé à une adresse anycast est routé à l'interface la plus proche ayant cette adresse selon le protocole de routage de mesure de distance.

Usage:

- Plusieurs serveurs redondant utilisant la même adresse
- DNS resolver et serveur DNS

Représentation:

- Les adresses IPv6 anycast sont des adresses unicast
- Requiert une adresse de sous-réseau anycast



Allocation des adresses IPv6

- L'IANA contrôle toutes les adresses IP et délègue attribution des blocs au **R**egional **I**P Address **R**egister (**RIR**)

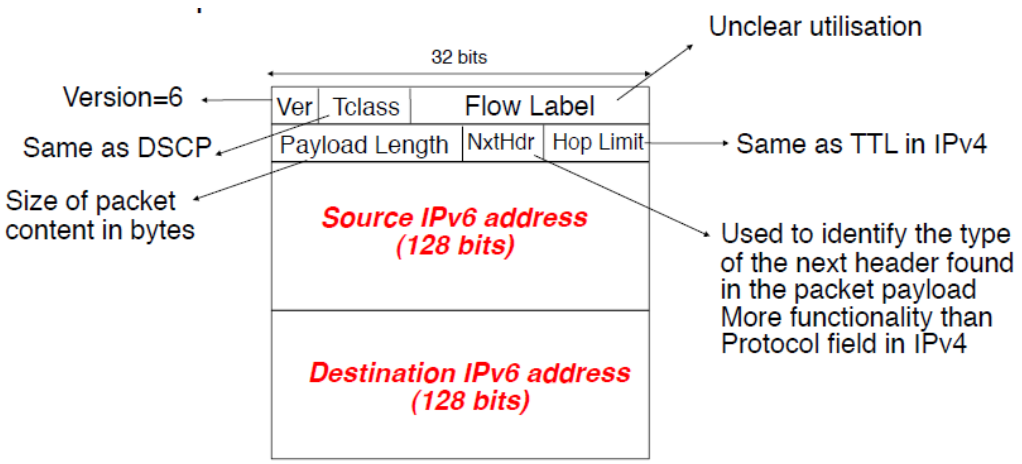
Une organisation peut allouer deux types d'adresses IPv6:

- **P**rovider **I**ndependent (**PI**) addresses
  - Généralement alloué aux ISP ou des grosses entreprises directement par des RIRs
  - Taille par défaut: /32

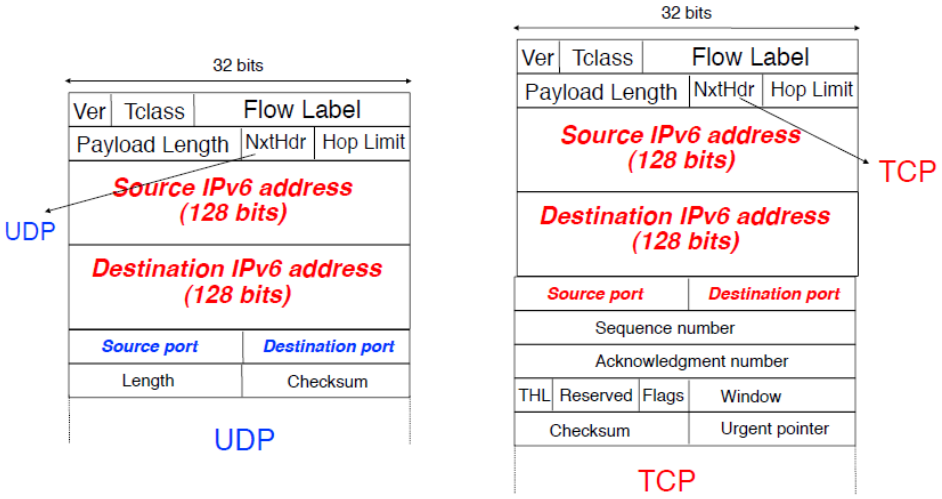
- **P**rovider **A**ggregatable (**PA**) addresses
  - Préfixes plus petits, assigné par des ISP de leur bloc
  - Tailles
    - **/48** en **général** sauf pour des grands abonnés
    - **/64** lorsqu'il n'y a **qu'un seul sous-réseau**
    - **/128** lorsqu'il n'y a **qu'un seul appareil** connecté

Les paquets IPv6

Format



Il n'y a pas de checksum dans le header des paquets IPv6 => se repose sur le checksum de la couche datalink et transport



Extension du header

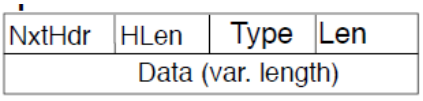
Chaque header d'option doit être encodé sur n*64 bits

Pusieurs types d'extension de header:

- Hop-by-hop Options
  - contient les informations à traiter à chaque hop
- Routing (type 0 et type 2)
  - contient des informations touchant les routeurs intermédiaires
- Fragment
  - utilisé pour la fragmentation et le réassemblage
- Destination Options
  - informations pour la destination
- Authentification
  - pour IPSec
- Encapsulating Security Payload
  - pour IPSec

Hop-by-hop et destination options

Format TLV de ces options:



- Les deux 1ers bits à gauche:
  - Gère le fait de ne pas connaître l'option
    - 00 => ignore et continue le traitement
    - 01 => écarte le paquet silencieusement
    - 10 => écarte le paquet et envoit un paquet ICMP à la source
    - 11 => écarte le paquet et envoit un paquet ICMP à a source si la destination n'est pas multicast

- Le 3ème bit
    - Permet de changer le contenu de l'option en cours de route

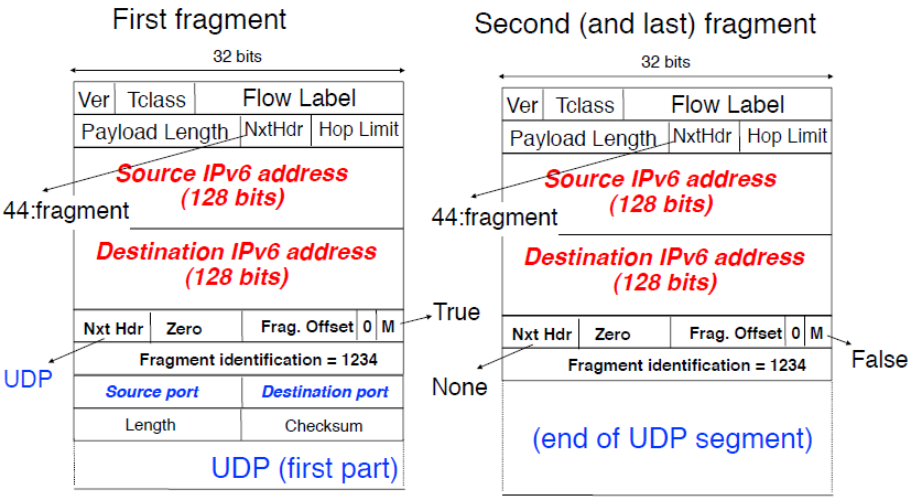
- Les 5 bits à droite
    - Type assigné par l'IANA

Fragmentation des paquets IPv6

IPv6 requiert que chaque lien dans Internet ait un MTU de 1280 octets ou plus.

⇨ La fragmentation et le réassemblage doit se faire à un niveau inférieur à IPv6

- Les routeurs **ne fragmentent pas** les paquets
- Seuls les endhosts utilisent la fragmentation et le réassemblage grâce au fragmentation header
- La découverte du PathMTU doit éviter la fragmentation la plupart du temps



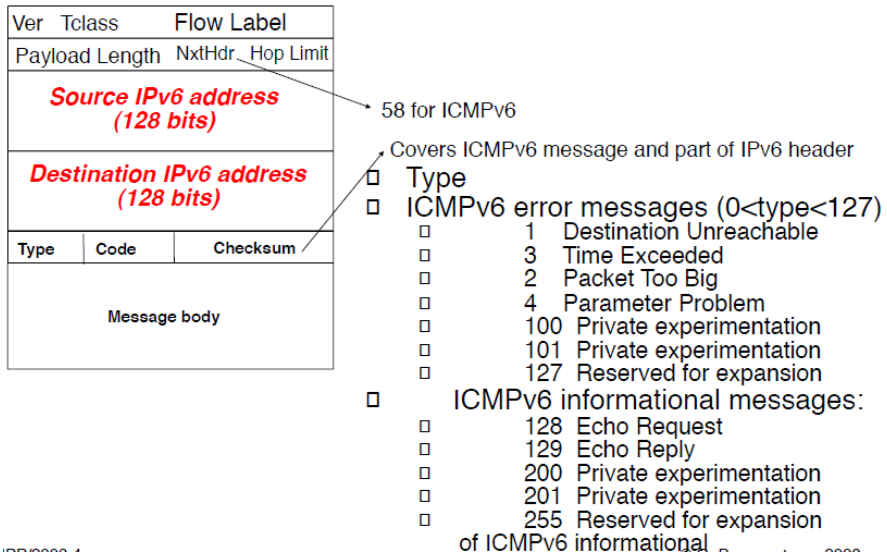
ICMPv6

Fourni les mêmes fonctions que ICMPv4 ainsi que d'autres fonctions

Types de message ICMPv6

- Destination injoignable
- Paquet trop gros
    - Utilisé pour la découverte de PathMTU
- Temps écoulé (Hop limit atteint)
    - Traceroute v6
- Echo request et echo reply
    - Pingv6
- Multicast group membership
- Router advertisments
- Découverte des voisins
- Autoconfiguration

Format:



Middleboxes

Les réseaux actuels ne comportent plus uniquement des routeurs et des hôtes. Certains appareils sont capables de traîter, analyser ou encore modifier des paquets IP.

Exemple:

- Firewall
- Network Address Translator
- Traffic shaper
- Deep Packet Inspection
- Intrusion Detection System
- Load balancer

Firewall

Principe:

- Le firewall analyse tous les headers des paquets.
- Accepte ou refuse les paquets en fonction de règles définies

On utilise un firewall pour filtrer ce qui rentre et ce qui sort du réseau. Le firewall est situé sur un routeur qui fait la liaison entre internet et un réseau privé/professionnel.

- Son rôle est donc de différencier deux mondes, le monde public et le monde privé. Il va donc analyser tout ce qui passe entre ces deux mondes, avec la possibilité d'éjecter des paquets qui ne seraient pas les biens venu.

- Mais cela peut aller encore plus loin, en effet, certains essaye même de reconstituer les messages pour savoir si on peut les laisser passer.

Network Address Translator

Il y a qu'un nombre limité d'adresses IPv4 publiques. On utilise donc plusieurs adresses privées qui seront traduites en 1 ou peu d'adresses publiques

Le **N**etwork **A**ddress **T**ranslator permet de traduire les paquets envoyés sur Internet.

- Le **NAT** consiste en une Map Adresse Interne < -- > Adresse publique
- Le NAT traduit les **adresses IP** et les numéros de **port** TCP/UDP afin de différencier plusieurs sources pour une même adresse IP

Routage dans les réseaux IP

Organisation du routage Internet

Internet est un inter-réseaux avec un grand nombre d'Autonomous Systems (**AS**).

AS:

- C'est un ensemble de routeurs managé par la même entité admnistrative
    - Exemple: Belnet, UUNET, SKYNET,...
    - +- 20 000 AS en 2007
- Les AS sont interconnectés afin de permettre la transmission de paquet de n'importe quelle source jusqu'à n'importe quelle destination

Internet est composé d'environ 30 000 domaines de routage autonomes.

Domaines

Un **domaine**:

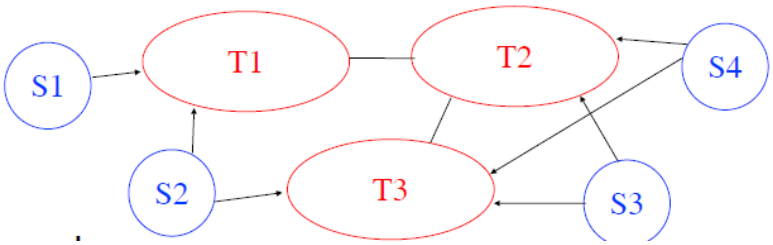
- est un ensemble de routeurs, liens, hôtes et de réseaux locaux sous lecontrôle d'une même administration
    - Peut être très grand tout comme il peut être très petit

- Les domaines sont interconnectés de plusieurs manières
    - Les interconnections de tous les domaines permettent en théorie d'envoyer des paquets n'importe où.

Types de domaines

- **Domaines de Transit**
  - Les domaines de transit permettent à des domaines externes d'utiliser leur infrastructure pour envoyer des paquets à d'autres domaines.
  - Exemple: UUNet, OpenTransit, GEANT, Internet2, RENATER, EQUANT, BT, Telia, Level3,...

- **Stup domaines**
  - Un stup domaine ne permet pas aux domaines externes d'utiliser leur infrastructure pour envoyer des paquets à d'autres domaines
  - Un stub doit être connecté à au moins 1 domaine de transit
    - Single-homed stub: connecté à 1 domaine de transit
    - Dual-homed stub: connecté à 2 domaines de transit
  - Exemple:
    - Content-rich stub domain
      - Large web servers : Yahoo, Google, MSN, TF1, BBC,...
    - Access-rich stub domain
      - ISPs providing Internet access via CATV, ADSL, ...



Internet Routing

On peut distinguer 2 types de routages pour Internet:

- **I**nterior **G**ateway **P**rotocol (**IGP**)
  - Routage des paquets IP dans chaque domaine
  - Connaît seulement la topologie de son domaine

- **E**xterior **G**ateway **P**rotocol (**EGP**)
  - Routage des paquets entre les domaines

Routage intradomaine

But:

- Permettre aux routeurs de transmettre des paquets IP par le meilleur chemin jusqu'à sa destination.
  - Le meilleur chemin est généralement le plus court (en seconde ou en hop) mais il peut être aussi le chemin le moins chargé

- Permet de trouver un chemin alternatif en cas d'erreur

Types de routage intradomaine (IGP)

- **Static routing**
  - Utilisé uniquement pour des domaines très petits

- **Distance vector routing**
  - Routing Information Protocol (**RIP**)
    - Toujours utilisé dans des petits domaines malgré ses limitations

- **Link-state routing**
  - Open Shortest Path First (**OSPF**)
    - Largement utilisé dans les réseaux d'entreprise
  - Intermediate System- Intermediate-System (**IS-IS**)
    - Utilisé par les ISP

Routing Information Protocol (RIP)

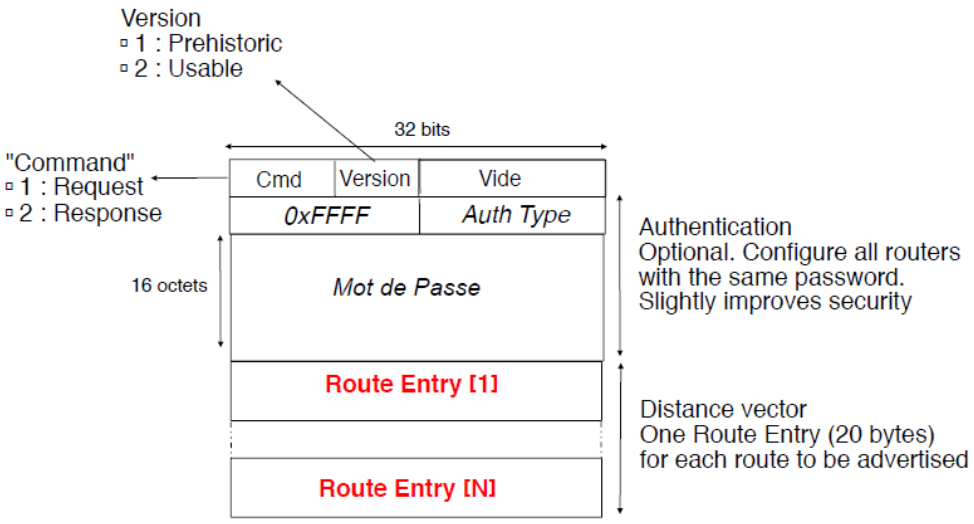
C'est un simple protocole de routage qui se base sur les vecteurs de distance.

Principe

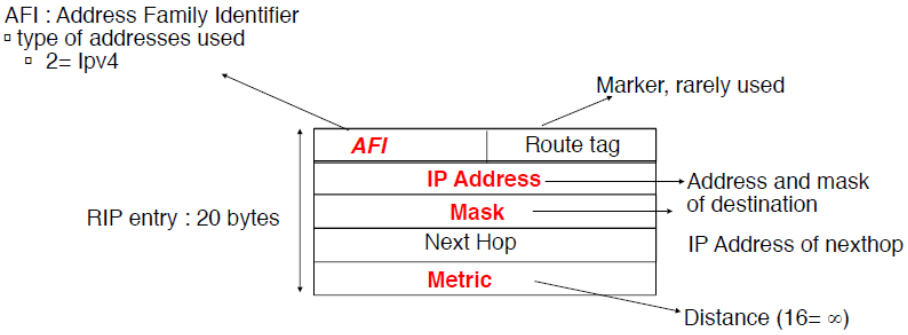
- Chaque routeur envoie periodiquement son vecteur de distance
  - Période par défaut: 30s
  - vecteur de distance envoyé dans un paquet **UDP** avec un **TTL = 1** à tous les routeurs du sous-réseau (via IP multicast)

- Extension optionnelle: envoie son vecteur de distance quand sa table de routage change et qu'on en a pas envoyé un dans les 5 dernières secondes

Format du message



Route Entry



- Route par défaut
  - adresse IP = 0.0.0.0, Mask = 0
- Chaque message RIP peut contenir jusqu'à 25 entrées de route (24 avec l'authentification)
  - Si la table de routage comporte plus de 25 entrées, il faut envoyer plusieurs messages

RIP timers

Toutes les 30s, les routeurs envoient leur vecteur de distance.

- Si il y a une coupure de courant, tous les routeurs vont reboot en même temps et auront des timer synchro

➢ On ajoute donc du random (entre 27.5s et 32.5s) dans le timer

Open Shortest Path First (OSPF)

C'est un protocole de routage link-state standardisé.

Opérations

- Démarrage du routeur
  - "Hello packets" envoyés pour découvrir les voisins

- Mise à jour des tables de routage
  - Paquets link-state
    - confirmation, num de séquence, âge
  - transmission périodique
  - transmission des changements de lien

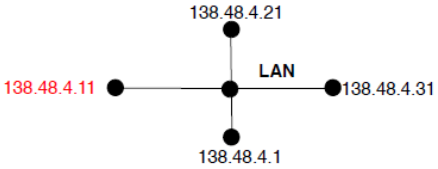
- Description de la base de données
  - fourni une liste des numeros de séquence de tous les LSP stockés dans le routeur

- Link-state request
  - Utilisé lorsqu'un routeur démarre pour demander les LSP des voisins.

Les routeurs sont souvent attachés à des LAN.
On voudrait le représenter par un graph:

- Désavantages
  - Trop de liens
  - Si il y a une coupure d'un lien dans un LAN, tous les routeurs sont déconnectés
    - Le graph ne montre pas cela

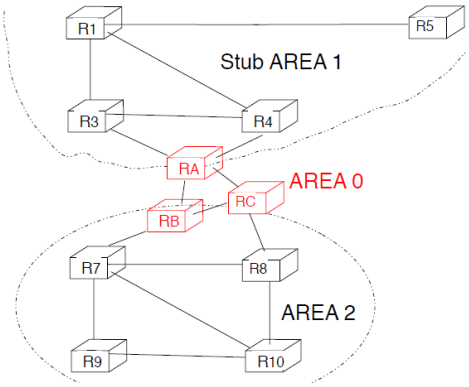
➢ On représente le LAN comme une étoile



OSPF dans des grands réseaux

On veut éviter des tables de routages trop grandes dans les routeurs OSPF.

➢ On divise le réseau en **régions**
  - Colonne vertébrale du réseau
    - Tous les routeurs qui sont reliés à au moins 2 régions du réseau sont considérés comme appartenant à la colonne vertébrale du réseau.
    - Les autres routeurs doivent être reliés à la colonne vertébrale
    - Au moins un routeur par région doit être relié à la colonne vertébrale



- Dans la région de la colonne vertébrale
  - Les routeurs échangent des LSP pour distribuer la topologie de la **région de la colonne**
  - Chaque routeur sait comment joindre les autres régions et les vecteurs de distance sont utilisés pour distribuer les routes inter-régions

- Dans les autres régions
  - Les routeurs échangent des LSP pour distribuer la topologie de sa **propre région**
  - Les routeurs ne connaissent pas la topologie des autres régions mais savent comment joindre la colonne vertébrale

Routage interdomaine

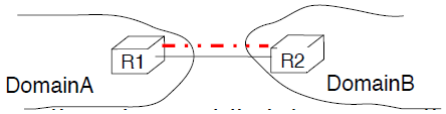
But

Permet de transmettre des paquets par le **meilleur chemin** (le moins cher) vers sa destination au travers plusieurs domaines de transit en prenant compte des règles de routage de chaque domaine sans connaître la topologie de ces domaines.

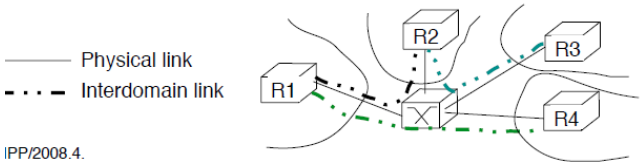
Chaque domaine peut spécifier dans ses règles de routage les domaines pour lesquels il est d'accord de fournir un service de transit ainsi que les méthodes pour choisir le meilleur chemin.

Types de liens interdomaines

- Les liens **privés**
  - Liens entre deux routeurs, appartenant aux deux domaines connectés.



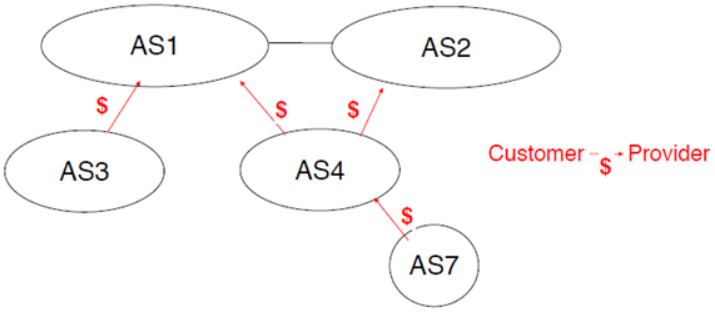
- Connection via un **point publique d'interconnetion**
  - Switch qui interconnecte des routeurs appartenant à des domaines différents



Politique de routage

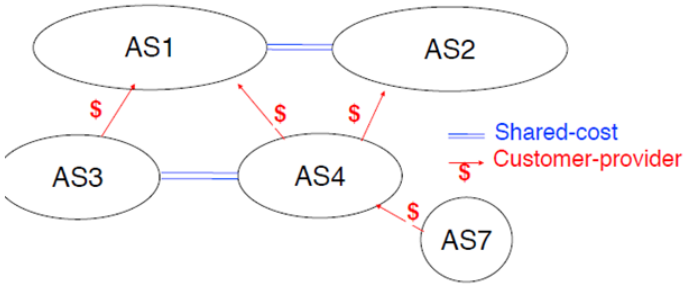
Il y a 2 politiques de routage courantes:

- **customer-provider peering**
  - Les client **c** achètent une connection Internet à un provider **p**
  - Le client envoie à son provider ses routes internes et les routes connues de ses propres clients
    - Les providers vont diffuser ses routes pour que tout le monde puisse joindre le client
  - Le provider envoie à ses clients toutes les routes connues
    - Pour que les clients puissent joidre n'importe qui sur Internet



- **shared cost peering**
  - Les domaines x et y acceptent d'échanger des paquets en utilisant un lien direct ou au travers d'un point d'interconnection
  - PeerX sends to PeerY its internal routes and the routes learned from its own customers
    - PeerY will use shared link to reach PeerX and PeerX's customers

        - PeerX's providers are not reachable via the shared link
    - Inversément avec PeerY qui envoie ses routes à PeerX



Chaque domaine spécifie sa politique de routage en définissant dans chaque routeur 2 ensembles de filtres pour chaque pair.

- **Import filter**
    - Spécifie quelles routes peuvent être accpetées par le routeur parmi toutes les routes reçues par un pair donné
- **Export filter**
    - Spécifie quelles routes peuvent être annoncée par le routeur à un pair donné.

Les filtres peuvent être définis en RPSL (Routing Policy Specification Language)

RSPL

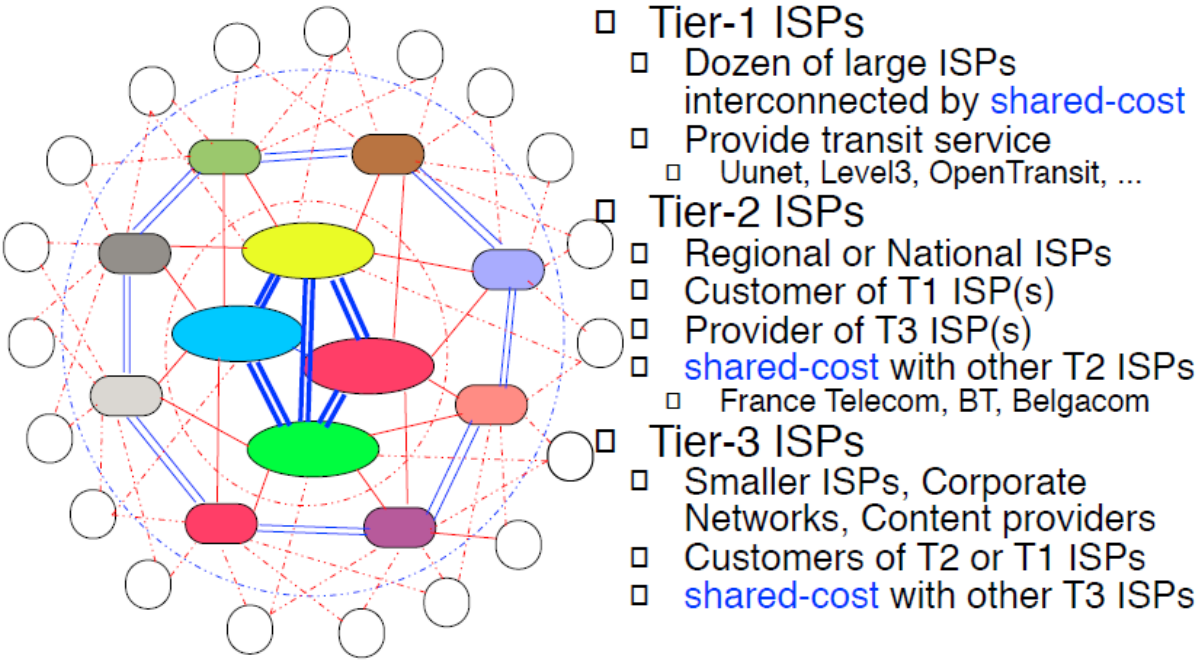
Import policy

- Syntaxe
    - **import: from** AS# **accept** list_of_AS
- Examples
    - Import: from Belgacom accept Belgacom WIN
    - Import: from Provider accept ANY

Export policy

- Syntaxe
    - **export: to** AS# **accept** list_of_AS
- Examples
    - Export: to Customer announce ANY
    - Export: to Peer announce Customer1 Customer2

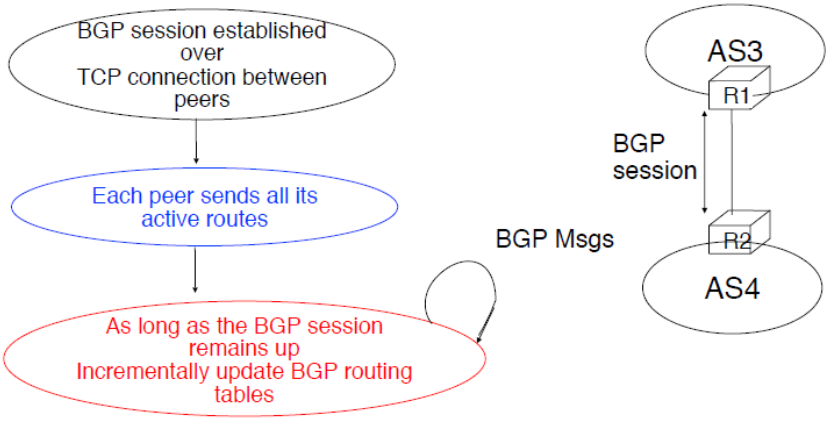
Organisation d'Internet



Border Gateway Protocol (BGP)

Principe

- Path vector protocol
  - Les routeurs BGP annoncent leur meilleure route pour chaque destination
- avec des updates incrémentales
  - Les annonces sont envoyées uniquement lorsque leur contenu change



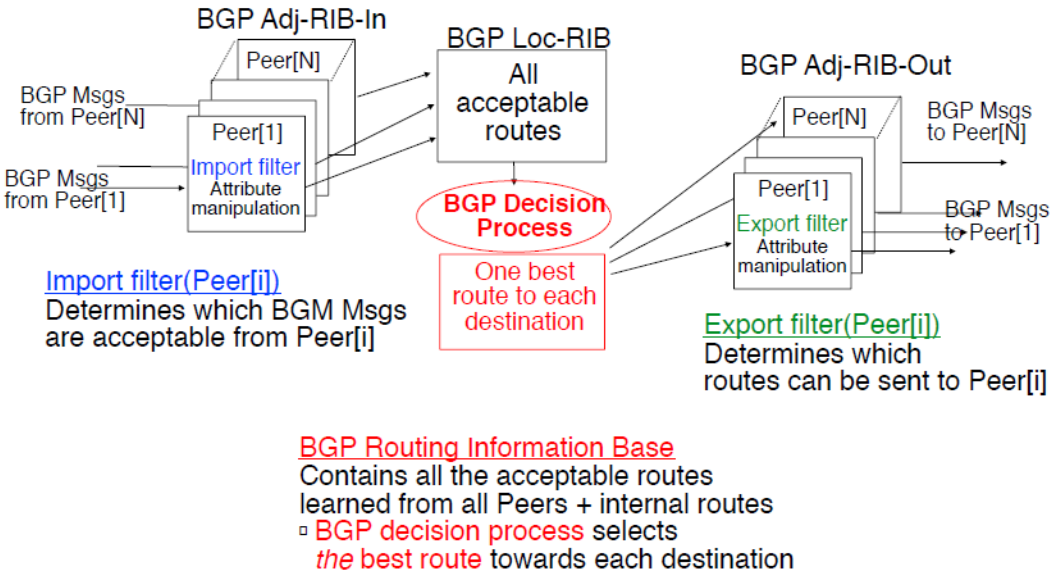
On peut considérer qu'il y a 2 types de vecteur de chemin BGP:

- **Update**
  - Utilisé pour annoncer un route au travers d'un préfixe
  - Contenu:

- adresse/préfixe de la destination
- Chemin interdomaine utilisé pour joindre la destination (AS-Path)
- NextHop (adresse du routeur annonçant la route)

- **Withdraw**
  - Utilisé pour indiquer que la route précédemment annoncée n'est plus utilisable
  - Contenu:
    - L'adresse/préfixe de la destination injoignable

Modèle conceptuel d'un routeur BGP



L'intérieur d'un server BGP est relativement complexe.

- Un routeur BGP peut être connecté à plusieurs autres routeurs frontières d'autres AS. On a donc un import filter différent pour chaque AS auquel on est relié.

- Quand le message arrive, il passe par le filtre, s'il ne passe pas, on le jette.

- Dans le cas où le message est accepté, on va stocker dans une base de données (RIB) les routes acceptées.

- Chaque information à envoyer va passer par un export filter. Si l'information ne passe pas, on n'envoie pas le message. Tout comme pour les imports filter, chaque AS possédera un export filter différent.

- La plupart du temps, les routes retenues sur un serveur BGP sont entrée manuellement, cependant, elles peuvent aussi venir d'autre routeur BGP.

- Les paquets BGP contiennent un champ "NextHop" qui indique le routeur (Adresse IP) à qui il faut s'adresser pour prendre la route indiqué.

Provenance des routes annoncées par le routeur BGP

- Apprise d'un autre routeur BGP
    - Chaque BGP routeur annoncent la meilleure route pour chacune des destination

- Static route
    - Configuration manuelle sur le routeur
        - - Il faut le faire manuellement
        - + Les annoncent sont stables

- Apprise via un protocole de routage intradomaine

Evènements pendant une session BGP

- **Ajout d'une nouvelle route au RIB**
    - Une nouvelle route est ajoutée au routeur local
        - soit static si elle a été ajoutée par configuration
        - soit dynamique si elle a été apprise par IGP
    - Réception d'un message d'update annonçant une nouvelle route ou une route modifiée

- **Retrait d'une route**
    - Retrait d'une route interne
        - route static si elle a été retirée par configuration du routeur
        - IGP déclare qu'une route intradomaine est injoignable
    - Réception d'un message **Withdraw**

- **Perte d'une session BGP**
    - Toutes les routes apprises du pair sont retirées du RIB
    - Le routeur BGP envoie à tous ses voisins un withdraw pour enlever le chemin vers le routeur perdu.

Choix d'une route en BGP

Deux variables entrent en jeu dans le choix d'un chemin ou d'un autre:

- La vitesse de transfert du lien
- Le coût du transport

Les filters jouent aussi leurs rôles:

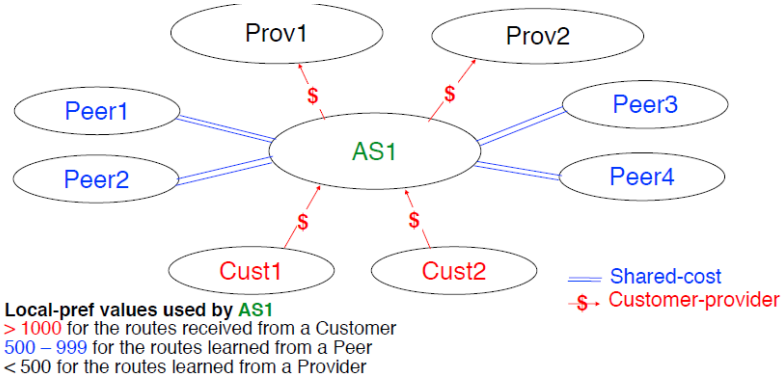
- Les imports filters filtrent les préfixes que l'on va accepter ou non, tant en entrée qu'en sortie.

Sur chaque route on va ajouter un attribut "local-pref" auquel on attribue une valeur. Lorsqu'on choisira une route, on prendra celle qui a le plus grand local-pref.

- Si plusieurs routes ont le même préfixe, on prendra celle au plus grand local-pref.

- Si plusieurs routes ont le même préfixe et le même local-pref, on choisira celle qui a le plus petite ASPath.
- L'utilisation de l'attribut local-pref risque d'engendrer des problèmes de convergences, car ces préférences sont locales à chaque domaine.

En pratique, local-pref est souvent utilisé pour renforcer les relations économiques.



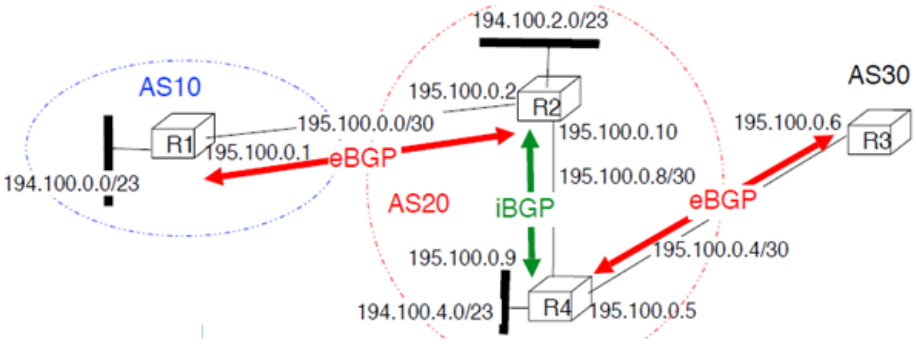
Conséquence de l'utilisation de local-pref

- Les chemins deviennent asymétriques
  - On risque donc de bloquer certains AS

Solution:

- **IGP** pour supporter les routes BGP
  - – IGP pourait ne pas supporter autant de routes
  - – IGP ne supporte pas les attributs BGP comme ASPath

- **iBGP** et **eBGP**
  - **eBGP** entre les routeurs appartenant à différents AS
  - **iBGP** entre les routeurs appartenant au même AS
    - Chaque routeur iBGP dans un AS maintient une session iBGP avec tous les autres routeurs de l'AS (full mesh)
    - Les sessions iBGP ne suivent pas forcément la topologie physique



iBGP et eBGP

Différences

- Sur une session **eBGP**, un routeur annonce uniquement la meilleure route vers chaque destination

- Sur une sessions **iBGP**, un routeur annonce uniquement les meilleurs chemins appris des sessions eBGP
    - Normalement, on n'applique pas les filtres avec iBGP
    - Connexion TCP entre deux routeurs. Les liens indirects sont aussi tolérés, contrairement aux liaisons eBGP qui sont forcément directes.
    - L'attribut local-pref se trouve uniquement dans les messages envoyés sur les sessions iBGP.