Protocoles de routage
— RIP, OSPF, BGP —

ART
Eric Fleury
Eric.Fleury@inria.fr

Remerciements

- Luc Saccavini, INRIA
- Laurent Toutain, ENST Bretagne
- Isabelle Chrisment, LORIA
- Nick McKeown, Stanford University

Références pour ce chapitre

- B.Halabi, « *Internet Routing Architectures* », Cisco Press, 1997
- Christian Huitema, « *Routing in the Internet* »,Prentice Hall, 1995, ISBN : 0131321927
- Christian Huitema, « *Et Dieu Créa l'Internet* », Eyrolles, ISBN : 2-212-08855-8
- James Kurose et Keith Ross, « *Analyse structurée des réseaux* », Pearson, 2nd édition, 2003
- Guy Pujolle, « *Les Réseaux* », ISBN : 2212110863
- J. W. Stewart, « *BGP4 Inter-Domain Routing in the Internet* », Addison-Wesley, 1999
- Andrew Tanenbaum, « *Computer Networks* », Pearson Education, ISBN : 2744070017
- BGP Case studies

Bibliographie

- J. McQuillan, I. Richer et E. Rosen, *The new routing algorithm for the ARPANET*, IEEE Transactions on Communications, COM-28(5), mai 1980.
- S. Floyd et V. Jacobson, *The synchronisation of Periodic Routing Messages*, ACM Sigcom '93 symposium, septembre 1993

Plan du cours

I. Routage unicast
   i. Principes
   ii. RIP
   iii. OSPF
   iv. BGP

– 1 – Généralités

C’est quoi le routage…

… c’est où la mer ?

Quelques rappels 3TC sur IP !

- Chaque «objet IP» est identifié par une adresse IP qui contient
  - l'adresse du réseau IP local (extraite grâce au Netmask)
  - Le numéro de la machine dans le réseau IP local
- Chaque «objet IP» est physiquement connecté :
  - à un réseau local de niveau 2 (Ethernet, liaison série, GPRS, X25, Frame Relay, ATM…)
- La communication avec d'autres «objets IP» du même réseau IP se fait directement grâce au réseau local de niveau 2
- La communication avec d'autres «objets IP» d'autres réseaux IP se fait grâce à des passerelles de niveau 3 ou Routeurs

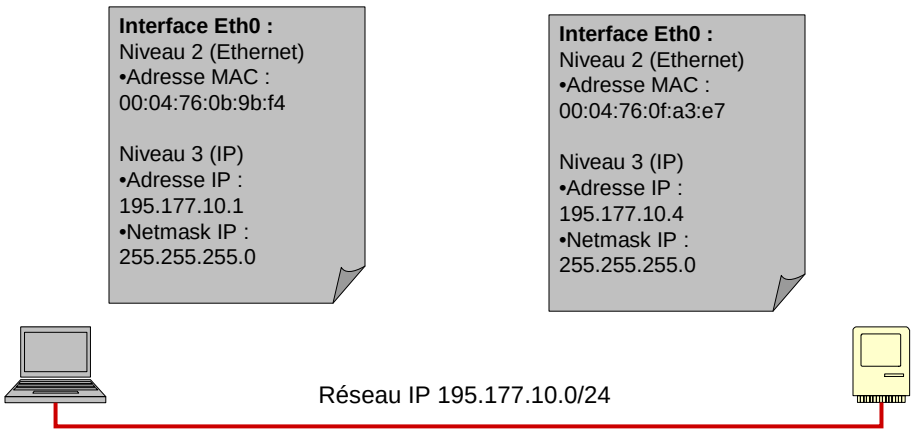
Et en pratique…

- la partie communicante de l'objet IP est désignée interface réseau (ou interface)
- notion de machine ↔ adresse IP
  - association non bijective
  - une machine peut avoir plusieurs interfaces réseau, et donc plusieurs adresses IP (cas des routeurs
  - Une interface peut avoir plusieurs adresses IP aussi…

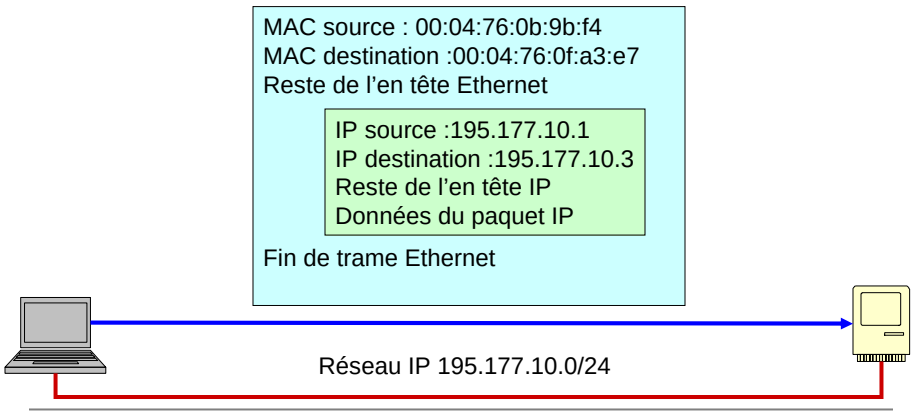
Du hyper connu sur IP

- Le protocole IP est non connecté
  - Pas de circuit virtuel entre émetteur et destinataire
  - Les paquets IP peuvent arriver dans le désordre
  - Les paquets IP peuvent ne pas arriver
  - Pas d'état dans le réseau
- Le protocole IP est dit « de bout en bout »
- Le protocole IP est dit « best effort »
  - Chaque noeud du réseau fait de son mieux pour acheminer les paquets

On reste au chaud chez soi : IP sur Ethernet (3TC toujours)

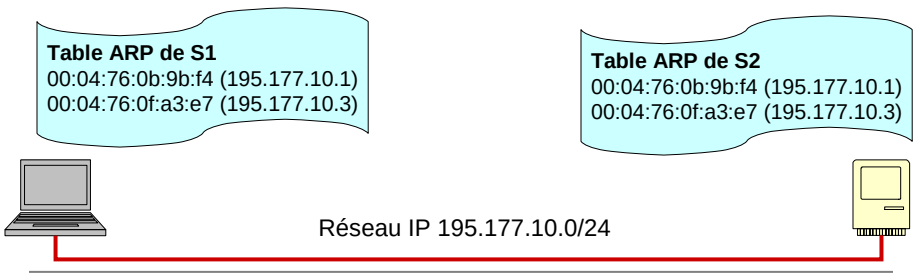


On reste au chaud chez soi : IP sur Ethernet (3TC toujours)

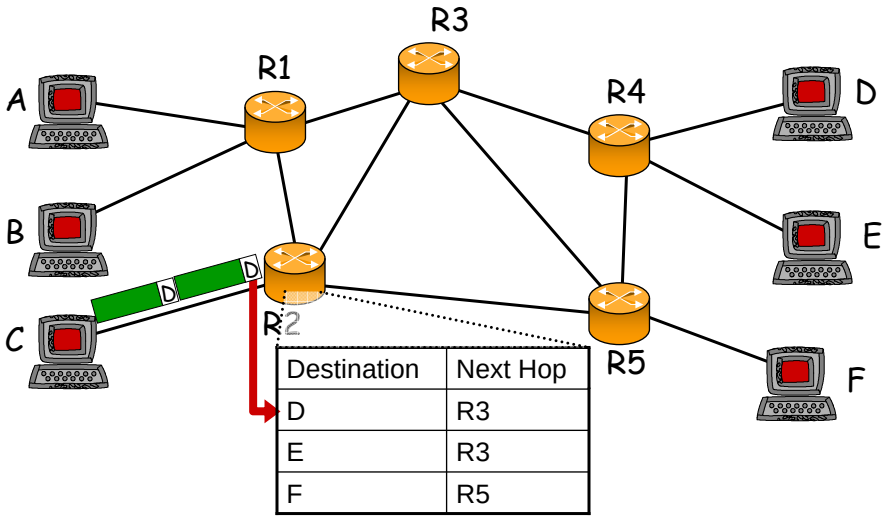


IP sur Ethernet (cont)

- Dans le cas d'Ethernet, un mécanisme spécifique : ARP (Address Resolution Protocol, RFC826) permet de créer et maintenir à jour une table de
  - correspondance entre les adresses de niveau 2 (MAC) et 3 (IP)
- Contenu des tables ARP de S1 et S2

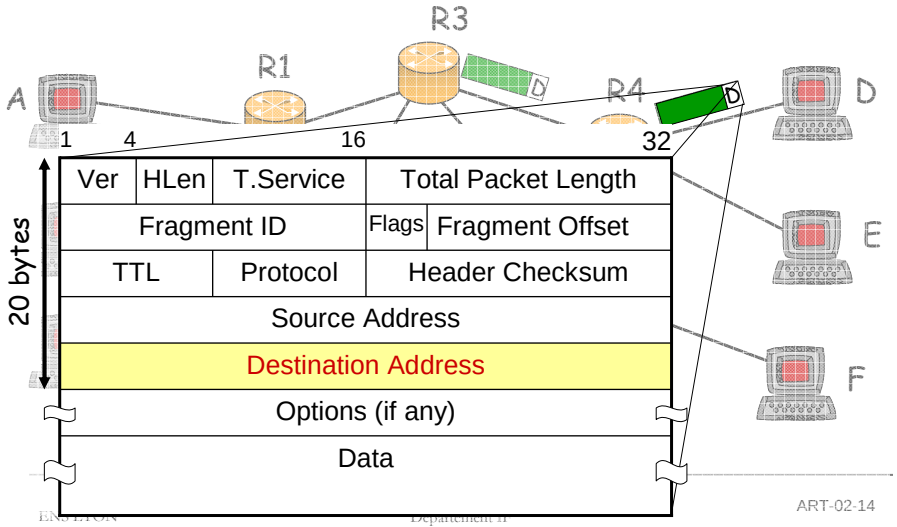


C'est quoi « router » ?

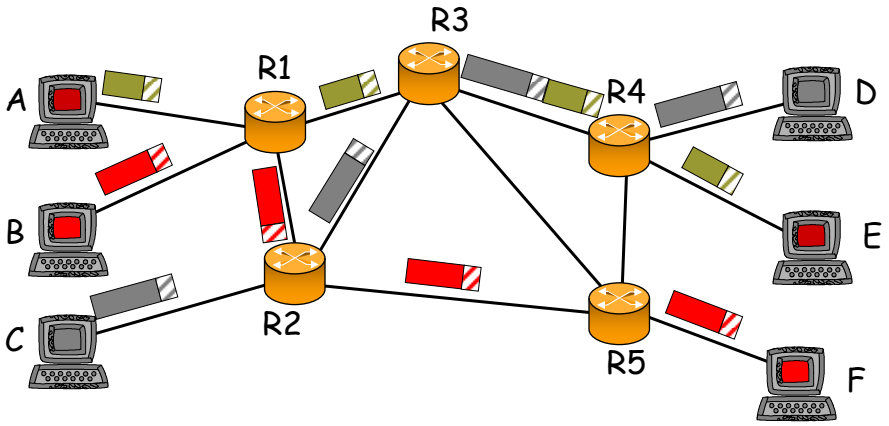


Destination	Next Hop
D	R3
E	R3
F	R5

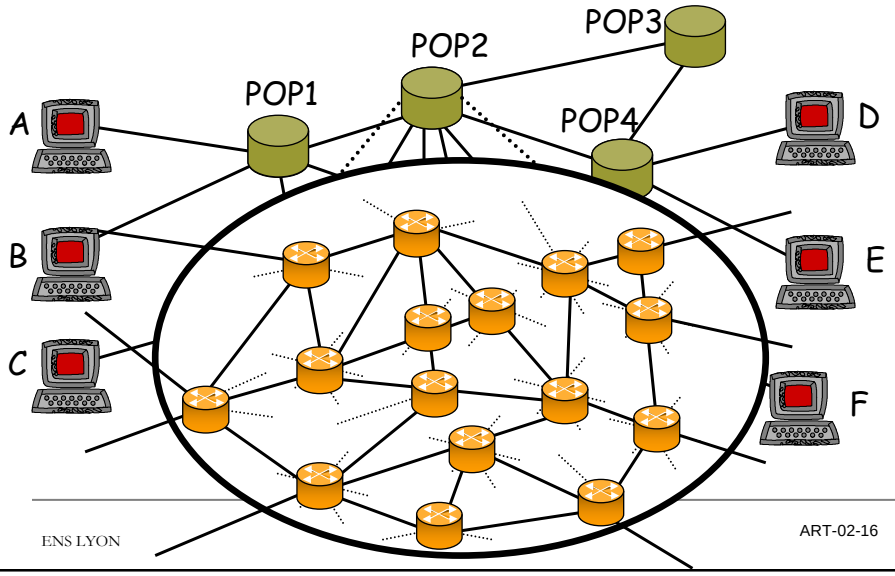
C'est quoi « router » ?



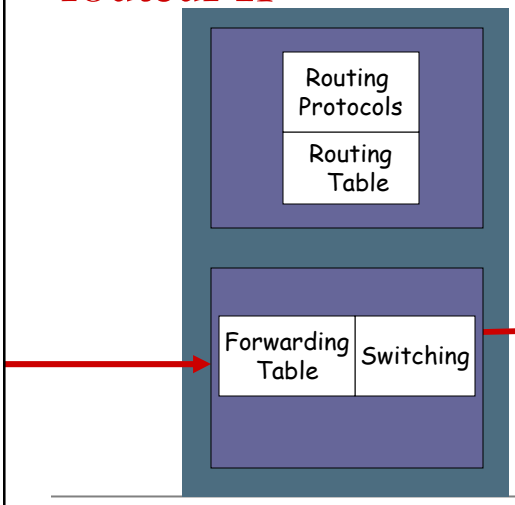
C'est quoi « router » ?



Points de Présence (POPs)



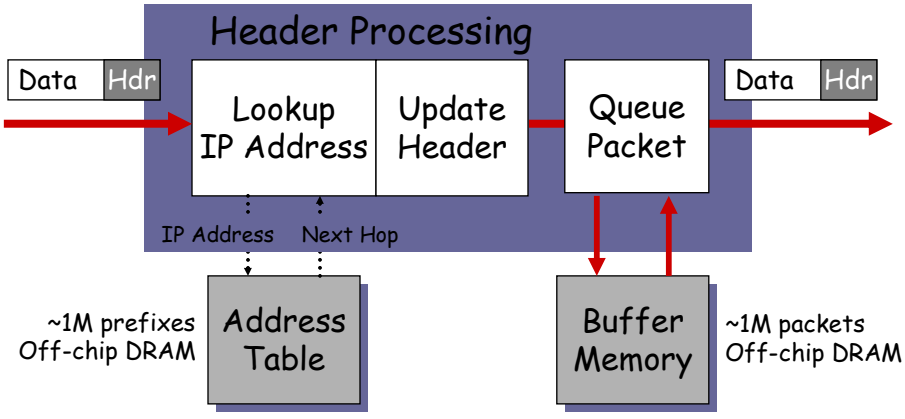
Composants architecturaux d'un routeur IP



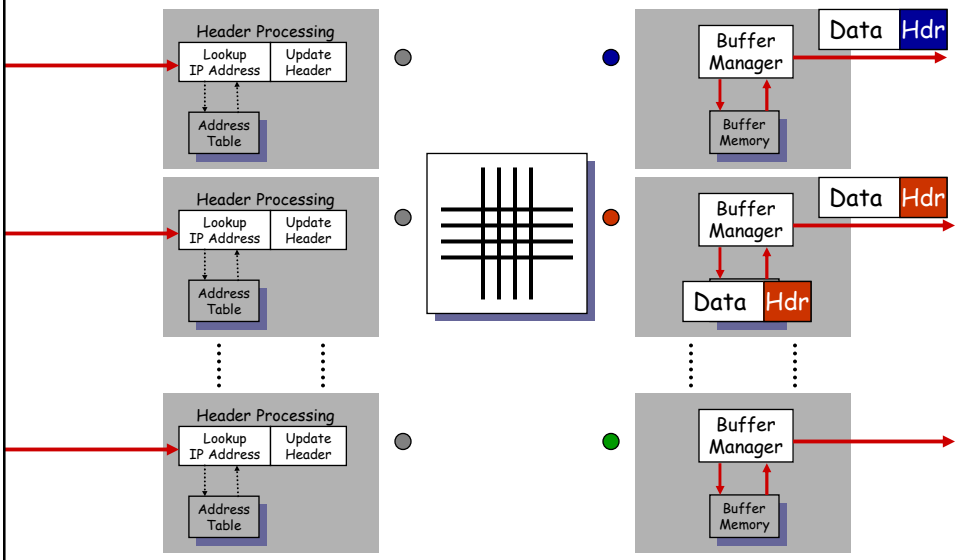
Plan de contrôle

Plan de données
per-packet processing

Architecture générique



Architecture générique



Un pas vers l'inconnu : IP sur Ethernet mais sur des réseaux différents

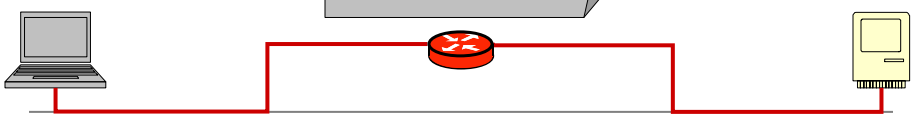
Interface Eth0 :
Niveau 2 (Ethernet)
•Adresse MAC :
00:04:76:0b:9b:f4

Niveau 3 (IP)
•Adresse IP :
195.177.10.1
•Netmask IP :
255.255.255.224

Interface eth0 :
Niveau 2 (Ethernet)
•@ MAC : 00:00:0c:fa:01:2a
Niveau 3 (IP)
•@IP : 195.177.10.62
•Netmask : 255.255.255.224
Interface eth1 :
Niveau 2(Ethernet)
•@MAC : 00:00:0c:3f::ac:0b
Niveau 3 (IP)
•@ IP : 195.177.10.126
•Netmask : 255.255.255.224

Interface Eth0 :
Niveau 2 (Ethernet)
•Adresse MAC :
00:04:76:0f:a3:e7

Niveau 3 (IP)
•Adresse IP :
195.177.10.4
•Netmask IP :
255.255.255.224



Réseau IP 195.177.10.0/27 Réseau IP 195.177.10.64/27

Couche réseau

- Transmission d'un paquet IP de S1 vers S3 via le routeur R1

MAC source : 00:04:76:0b:9b:f4
MAC destination ::00:00:0c:fa:01:2a
Reste de l'en tête Ethernet

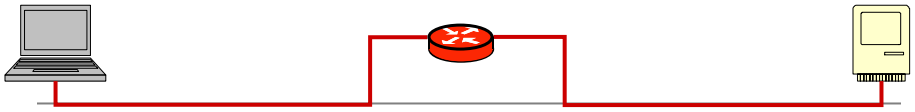
IP source :195.177.10.1
IP destination :195.177.10.65
Reste de l'en tête IP
Données du paquet IP

Fin de trame Ethernet

MAC source : :00:00:0c:3f::ac:0b
MAC destination ::00:04:76:0f:a3:e7
Reste de l'en tête Ethernet

IP source :195.177.10.1
IP destination :195.177.10.65
Reste de l'en tête IP
Données du paquet IP

Fin de trame Ethernet



Réseau IP 195.177.10.0/27 Réseau IP 195.177.10.64/27

Routage IP sur Ethernet (cont)

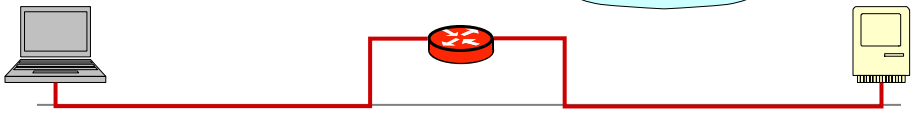
- Table de routage de S1, R1 et S2:

Table de routage de S1
195.177.10.0/27 via eth0
195.177.10.64/27 via 195.177.10.62

Erreur ds le netmask

Table de routage de R1
195.177.10.0/27 via eth0
195.177.10.64/27 via eth1

Table de routage de S2
195.177.10.64/27 via eth0
195.177.10.0/27 via 195.177.10.126



Réseau IP 195.177.10.0/27 Réseau IP 195.177.10.64/27

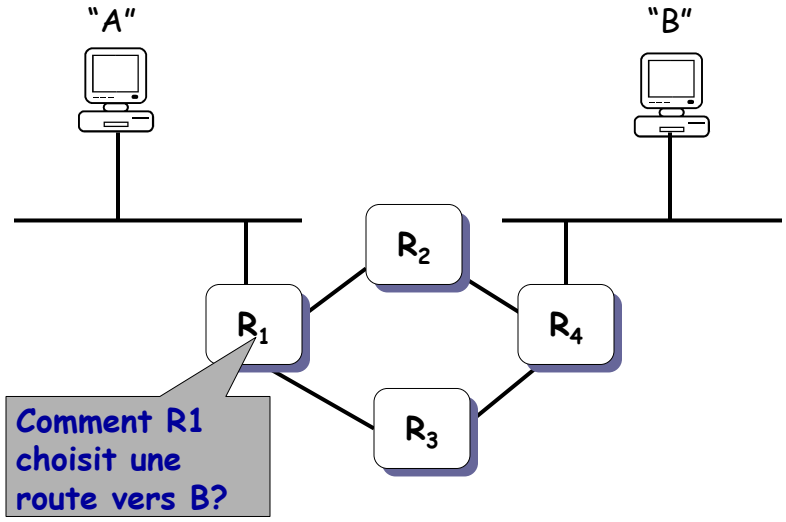
Rôle de la couche réseau

- Choix de l'itinéraire (route) entre *src* et *dst* :
  - Algorithme de routage
  - Routage par information de liens / état de liens
  - Routage à vecteur de distance
- Réexpédition (forwarding)
  - Comment les paquets sont relayés
- Établissement de l'appel (*e.g.,* ATM)
  - Aucun service de ce genre dans Internet

Fonction d'un routeur

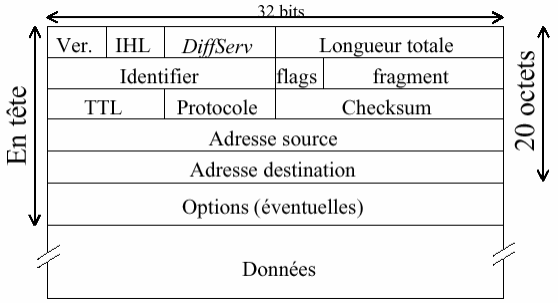
- Contrairement aux ponts, les routeurs doivent être configurés.
- Ils doivent connaître les adresses des routeurs ou des stations vers lesquels ils envoient les paquets.

Un problème simple !



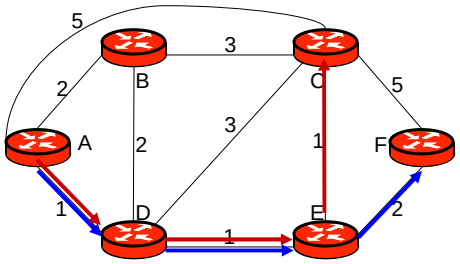
Problématique du routage IP

- Routage en fonction de l'adresse destination uniquement



- Routage de proche en proche
  - Chaque routeur prend la décision de routage qui lui paraît la meilleure (en fonction de sa table de routage)

Modélisation



- Plus « court » chemin entre A et C ?
- Plus « court » chemin entre A et F ?
  - Avez-vous testé toutes les possibilités ?
  - Combien de chemins existe-t-il entre A et F ?
  - En général dans un graphe G=(V,E) avec n=|V| et m=|E| ?

A D E C

A D E F

– 2 – Algorithmes de routage

Comment construire des routes, les maintenir…

Classification des algorithmes de routage

- **globale**
  - Connaissance globale du réseau
  - Calcul centralisé ou distribué
  - Notion d'état de liens
- **décentralisé**
  - Pas de connaissance globale
  - Par itération

Classification des algorithmes de routage

- **statiques**
  - Les parcours changent peu
  - Modification humaine
- **dynamiques**
  - Parcours s'adaptent à la topologie
- **Sensible à la charge**
  - Les coûts varient de façon dynamiques
    - McQuillan 1980
    - Huitema 1995
- **Insensible à la charge**
  - Les coûts ne reflètent pas le niveau de congestion
    - RIP, OSPF

Algorithme de routage

- **Échange automatique de tables de routage.**
- Table de routage :
  - pour aller :
    - réseau,
    - sous- réseau,
    - machine.
  - passer par :
    - attachement local,
    - routeur.
  - avec un coût de
    - nombre de sauts,
    - fonction du débit, délai...

Route statique

- Commande `route`
  - Permet d'indiquer une route
    - Vers un réseau (net) ou vers un équipement (host)
    - Ou une route par défaut (default)
  - Syntaxe
    - `route add|delete [net|host] destination|default gateway metric`
    - Sur les équipements non routeurs, une seule route par défault est définie

Route statique

- Commande `ifconfig`
  - Permet de configurer une interface en lui attribuant une adresse IP
  - Syntaxe
    - `ifconfig eth0 @IP netmask @mask broadcast @cas`
- Consultation des routes :
  - Commande `netstat`

Routage statique

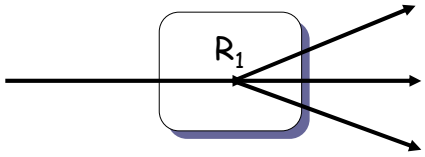
- Caractéristiques
  - Très stable (fichier de configuration)
  - Fastidieux et risque d'erreur important si grand réseau (>10 routeurs)
- Réservé aux cas simples
  - Postes de travail (une route par défaut vers le routeur le plus proche)
  - Petits réseaux (un routeur avec une route par défaut vers le FAI)
  - Pas de possibilité de gérer des routes redondantes

Problèmes du routage statique

- Statique implique
  - Mise à jour manuelle de tous les équipements réseaux
  - Difficile à maintenir en cas d'évolution du réseau
- Recommandation
  - Stations → routage statique
  - Routeurs → routage dynamique

Technique 1 : Innondation

Les routeurs retransmettent les paquet sur tous les ports



- **Avantages :**
  - Toute destination est atteignable.
  - Utile si la topologie est inconnue.
- **Inconvénients :**
  - Des routeurs reçoivent des paquets dupliqués.
  - Problème de boucles…

Routage dynamique

- Caractéristiques
  - Adaptatif à l'évolution du réseau (vie et mort des routeurs et de leurs liaisons)
  - Configuration simple (varie peu avec le nombre de routeurs)
- Objectifs d'un protocole de routage
  - Optimisation : sélection des meilleures routes
  - Élimination les boucles de routage (routes circulaires)
  - Efficacité : peu de consommation de bande passante et de CPU
  - Stabilité : convergence et reconfiguration rapides
  - Simplicité : configuration simple

Les protocoles de routage dynamique

- Les protocoles intérieurs (IGP)
  - Distance-vecteur : RIP, IGRP
  - État des liens : OSPF, IS-IS
  - Taille <100 routeurs, 1 autorité d'administration
  - Échange de routes, granularité = routeur
- Les protocoles extérieurs (EGP)
  - EGP, BGP, IDRP
  - Taille = Internet, coopération d'entités indépendantes
  - Échange d'informations de routage, granularité = AS

Algorithme de Routage
— Vecteur de Distance —

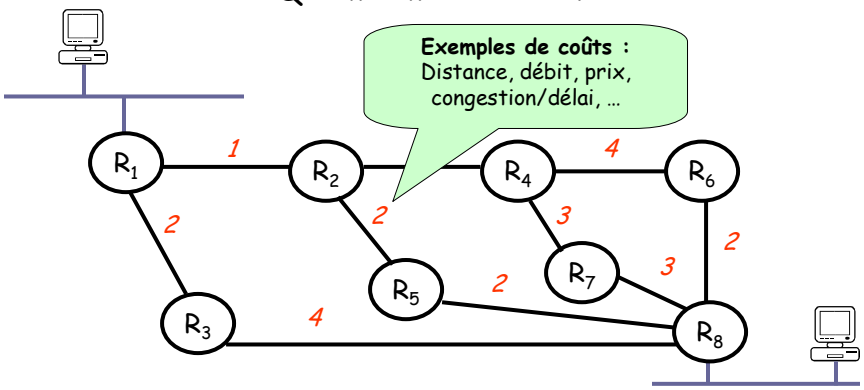
- Itératif
  - Continue tant qu'il y a des informations à échanger
- Asynchrone
  - Chaque nœud est indépendant
- Distribué
  - Aucun nœud n'a la vison complète du réseau

Vecteur de distance

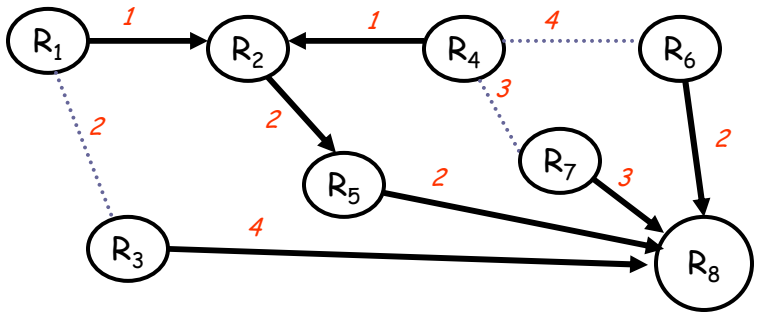
- Échange d'information entre routeurs adjacents.
  - Les routeurs diffusent vers les noeuds adjacents leur table de routage rudimentaire constituée de ses différents voisins accessibles et du coût de la liaison.
- Quand un routeur reçoit une nouvelle table, il effectue les traitements suivants pour chaque entrée de la table reçue :
  - Si l'entrée n'est pas dans sa table, il la rajoute.
  - Si le coût de la route proposée par la table plus le coût de la route pour aller jusqu'au routeur (émetteur de la table) est inférieur au coût indiqué dans sa table, sa table de routage est modifiée pour prendre en compte cette nouvelle route.
  - Sinon, il n'y a pas de changement.
- La modification d'une entrée dans la table d'un routeur engendre l'émission de la nouvelle table sur tous les ports du routeur
- Les échanges entres les routeurs continuent jusqu'à ce que l'algorithme converge

Technique 2 : Bellman-Ford Algorithm

Objectifs : Calculer une route de (R_1, …, R_7) vers R_8
Qui minimise le coût.



La solution semble évidente !

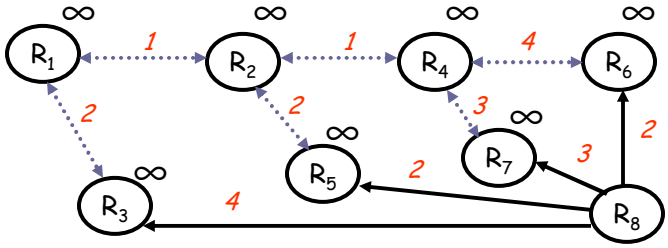


- C'est un arbre de recouvrement enraciné en R8
- L'algo de Bellman-Ford calcule cet arbre…

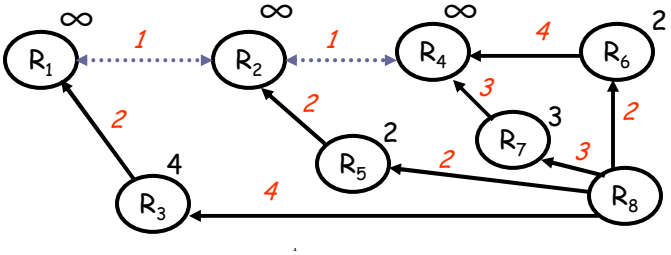
Algorithme de Bellman-Ford distribué

1. Let $\underline{X}_n = (C_1, C_2, ..., C_7)$ where: C_i = cost from R_i to R_8.
2. Set $\underline{X}_0 = (\infty, \infty, \infty, ..., \infty)$.
3. Every T seconds, router i sends C_i to its neighbors.
4. If router i is told of a lower cost path to R_8, it updates C_i. Hence, $\underline{X}_{n+1} = f(\underline{X}_n)$ where $f(.)$ determines the next step improvement.
5. If $\underline{X}_{n+1} \neq \underline{X}_n$ then goto step (3).
6. Stop.

Bellman-Ford

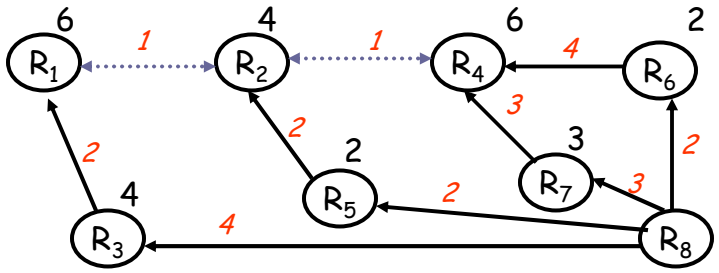


R_1	Inf
R_2	Inf
R_3	4, R_8
R_4	Inf
R_5	2, R_8
R_6	2, R_8
R_7	3, R_8



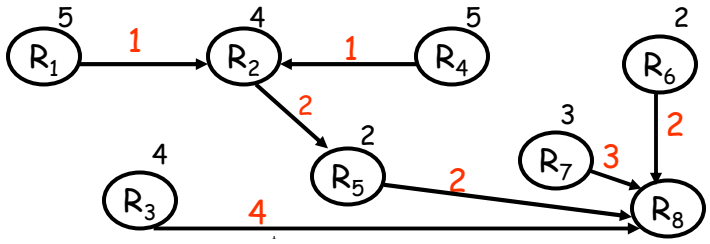
Bellman-Ford Algorithm

R_1	6, R_3
R_2	4, R_5
R_3	4, R_8
R_4	6, R_7
R_5	2, R_8
R_6	2, R_8
R_7	3, R_8



Solution

R_1	5, R_2
R_2	4, R_5
R_3	4, R_8
R_4	5, R_2
R_5	2, R_8
R_6	2, R_8
R_7	3, R_8

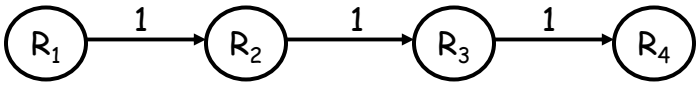


Bellman-Ford

Questions :

1. Combien de temps prend l'algorithme pour s'exécuter ?
2. Est ce qu'il converge ? Toujours ?
3. Que se passe-t-il si un coût change ?
4. Si un lien casse ?

Bellman-Ford : «Bad news travels slowly»



Calcul des distances vers R_4:

Time	R_1	R_2	R_3
0	3,R_2	2,R_3	1, R_4
1	3,R_2	2,R_3	3,R_2
2	3,R_2	4,R_3	3,R_2
3	5,R_2	4,R_3	5,R_2
…	"Counting to infinity" …		

$R_3 \longrightarrow R_4$ fails

Heuristiques pour compter à l'infini…

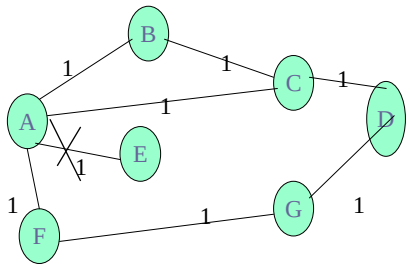
- Définir l'infini comme étant un entier « petit »,
  - *e.g.,* on s'arrête de compter à 16
- Vecteur de chemin
- L'horizon partagé
- L'horizon partagé avec retour empoisonné
- Source tracing

Vecteur de chemin

- Manque d'informations dans le vecteur
- Ex: pour aller de **R2** à **R4** , il faut passer par **R3**
- Solution : Annoter chaque entrée avec le chemin pour obtenir le coût
  - <R2, 2, « R2-R3-R4 »>
- Les vecteurs nécessitent des tables assez grandes

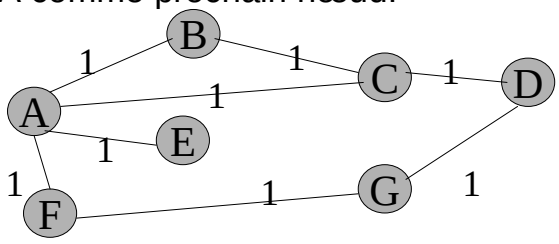
Panne d'un lien ?

- Que se passe-t-il si la Liaison de A vers E tombe en panne ?



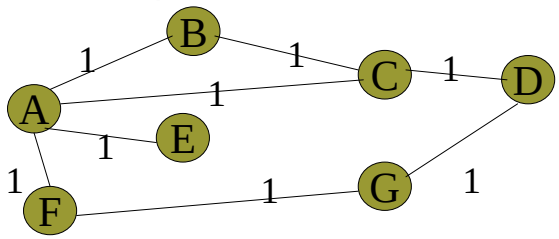
Horizon partagé

- Un routeur *u* n'annonce jamais la route à son voisin *v*, si *v* est le prochain nœud vers cette destination
  - C ne prévient pas A du coût vers E comme il utilise A comme prochain nœud.



Horizon partagé avec retour empoisonné

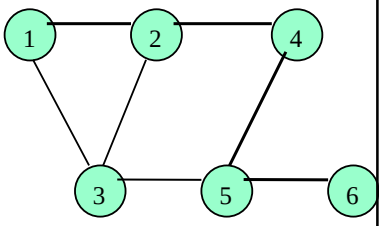
- Méthode plus agressive
- Un routeur *u* donne à son voisin *v*, si *v* est le prochain nœud vers cette destination, une route infinie
  - C prévient A que le coût vers E est infini comme il utilise A comme prochain nœud.



Source tracing

- Le vecteur de distance contient le routeur précédent la destination

Destination	Next	Last
1	-	-
2	2	1
3	3	1
4	2	2
5	2	4
6	2	5



Technique 3 : Plus courts chemins de Dijkstra

- Les routeurs émettent des messages dès que l'état d'un lien évolue (d'où le nom Link State Routing")
- Chaque routeur calcule tous les plus courts chemins de tous les nœuds vers lui-même
- A chaque étape le routeur rajoute le chemin dans son arbre
- Pour calculer, in fine, un arbre enraciné

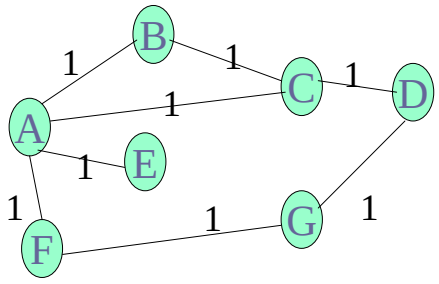
Algorithme de Routage à état de liaisons

- Stratégie: envoyer à tous les nœuds (pas seulement les voisins) l'information au sujet de ses voisins
- Distribution de la topologie du réseau et du coût de chaque liaison à tous les routeurs

État de liaisons

- Les nœuds ont une copie complète de la carte du réseau
- Les nœuds exécutent le calcul des meilleurs routes localement en utilisant cette carte
  - Plus de boucles.

Carte de réseau



A	B	1
A	C	1
A	E	1
A	F	1
B	A	1
B	C	1
C	A	1
C	B	1
C	D	1
……		

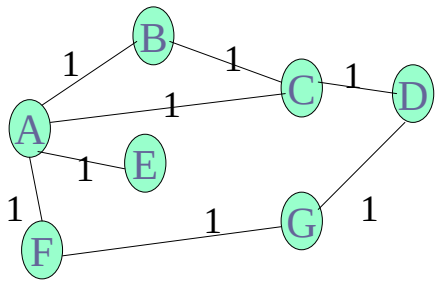
État de liaison

- Basé sur 2 mécanismes
  - Dissémination fiable de l'information (reliable flooding)
  - Pour permettre une mise à jour de la base de données
  - Calcul des routes à partir des informations locales

Dissémination de la topologie

- Chaque routeur met les informations décrivant les liaisons dans des paquets appelés LSP (Link State Packet)
- Link State Packet (LSP)
  - L'ID ou identificateur du nœud qui a créé le LSP
  - Une liste des voisins directement connecté avec le coût de la liaison vers chaque voisin
  - numéro de séquence (SEQNO)
  - time-to-live (TTL) pour ce paquet

LSP



A	B	1
A	C	1
A	E	1
A	F	1

Inondation (Flooding) fiable Algorithme

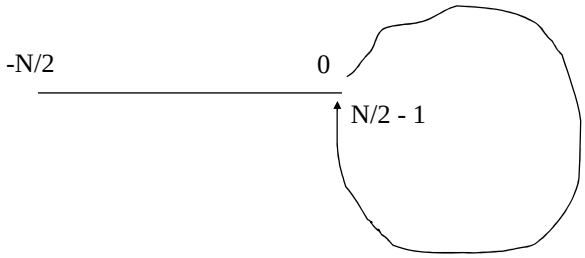
- Recevoir un LSP et chercher l'enregistrement correspondant dans la base
- Si enregistrement pas présent, l'ajouter et diffuser le LSP à tous les nœuds excepté celui de la réception
- Si enregistrement présent, comparer le numéro de séquence du paquet reçu avec celui stocké
  - Si le numéro de séquence dans la base est plus petit que celui du LSP reçu, diffuser le LSP à tous les nœuds excepté celui de la réception

Inondation (Flooding) fiable

- Générer de nouveaux LSPs périodiquement => incrémentation du numéro de séquence
  - Quel problème peut-il y avoir avec les numéros de séquence ?
- Comment choisir le numéro initial, si un routeur reboot ?
  - Lollipop sequence space [perlman 83]

Inondation (Flooding) fiable lollipop sequence space

- Partitionnement de l'espace de numérotation de taille N en 3 parties :
  - De -N/2 à 0
  - Le numéro 0
  - De 0 à N/2-1
- Quand un routeur démarre il utilise le numéro de séquence –N/2 pour son LSP et ensuite –N/2+1, -N/2/+2,….
- Quand le numéro de séquence devient positif, il entre dans la partie circulaire



Règle 1 Un LSP de numéro de séquence a est plus vieux que celui de séquence b si :

a < 0 et a < b
a > 0 , a < b et b-a < N/4
a > 0, b > 0, a >b et a-b > N/4

Règle 2 Si un routeur obtient un LSP d'un autre routeur qui a un numéro de séquence plus vieux que celui stocké dans sa base, il informe ce routeur de son numéro de séquence

Inondation (Flooding) fiable

- Que se passe-t-il en cas de panne ?
  - Détection de la panne (message HELLO)
- Comment détruire de vieux LSPs ?
  - Aging process :
    - Décrémenter le TTL pour chacun des LSPs stockés avant de diffuser une copie
    - Le détruire quand le TTL=0 et rediffuser le LSP avec un TTL de 0 pour le détruire dans les autres noeuds

Calcul de la route en théorie

- Algorithme utilisé :
  - Algorithme du plus court chemin d'abord de Dijkstra (Short Path First)
  - Théorie des graphes
    - Soit N l'ensemble des nœuds dans le graphe
    - l(i,j) poids sur l'arc entre le nœud i et j .
    - Si aucun arc entre i et j alors l(i,j) = ∞
    - C(n) détermine le coût du chemin de s vers le nœud n

Algorithme de Dijkstra

$M = \{s\}$
Pour tout n dans N - $\{s\}$
$C(n) = l(s,n)$

Tant que ($N \neq M$)

$M = M \cup \{w\}$ tel que $C(w)$ est le minimum pour tout w dans (N-M)

Pour tout n dans (N-M)
$C(n) = \text{MIN}\ (C(n), C(w)+l(w,n))$

Calcul de la route en pratique

- Algorithme du "Forward Search »
- Chaque routeur maintient deux listes :
  - Temporaire (T) et Permanent (P)
- Chaque liste contient 3 éléments:
  - Destination, Coût, Prochain nœud

Calcul des routes

1) Initialiser la liste P avec une entrée pour le nœud lui-même (s). Cette entrée a un coût de 0

2) Pour le nœud ajouté dans la liste P lors de la précédente étape et appelé Next, sélectionner son LSP

3) Pour chaque Voisin de Next, calculer le coût pour atteindre son Voisin comme la somme du coût de s à Next avec celui de Next au Voisin.

Calcul des routes (cont)

4. Deux cas peuvent se produire
   4. Si Voisin n'est présent ni dans la liste P ni dans la liste T alors ajouter <Voisin, Coût, Nexhop> dans la liste T, avec Nexthop étant la direction que je dois prendre pour atteindre Next
   5. b) Si Voisin est présent dans la liste T et le coût inférieur à celui actuellement listé pour Voisin, alors remplacer l'entrée courante par <Voisin, Coût, Nexthop> où Nexthop est la direction que je dois prendre pour atteindre Next
5. Si la liste T est vide, arrêter. Sinon prendre une entrée dans la liste T avec le coût le plus faible,la déplacer dans la liste P et retourner en 2

Choisir le coût des liaisons

- Déterminer la façon dont la charge de trafic est distribuée dans le réseau
  - Plus le coût est faible pour une liaison, plus la probabilité de choisir cette liaison dans un plus court chemin est grande
- Différentes métriques possibles
  - Métrique statique
  - Métrique dynamique de l 'ARPAnet
  - Métrique dynamique de l 'ARPAnet modifiée

Métrique statique

- Le plus simple donné un poids de 1 à chaque liaison
  - Le chemin le plus court est le chemin avec le moins de routeurs intermédiaires
  - Problème ?
- Assigner un poids différent en fonction du type de liaison
  - Problème ?

Métrique dynamique de l'ARPAnet

- Le coût d'une liaison est proportionnelle à la longueur de la file d'attente du routeur à l'entrée de la liaison.
- Problème quand le réseau est surchargé ?

Métrique dynamique de l'ARPAnet modifiée

- Prise en compte également des capacités des liaisons.
  - Quand la charge de la liaison est faible, son coût dépend entièrement de la capacité de la liaison

Et le routage dans l'Internet ?

- L'Internet est découpé en systèmes autonomes (AS) comprenant chacun un ensemble de routeurs sous une administration unique
  - Stanford (32), HP (71), MCI Worldcom (17373)
  - whois –h whois.arin.net ASN "MCI Worldcom"
- Au sein d'un AS, l'administrateur met en œuvre un protocole de routage interne (IGP)
  - Exemple d'IGPs: RIP (rfc 1058), OSPF (rfc 1247).
  - Les routeurs dans le même AS exécutent le même algorithme de routage
  - L'algorithme de routage dans à l'intérieur d'un AS est appelé intra- autonomous system routing protocol ou protocole de routage intra-AS

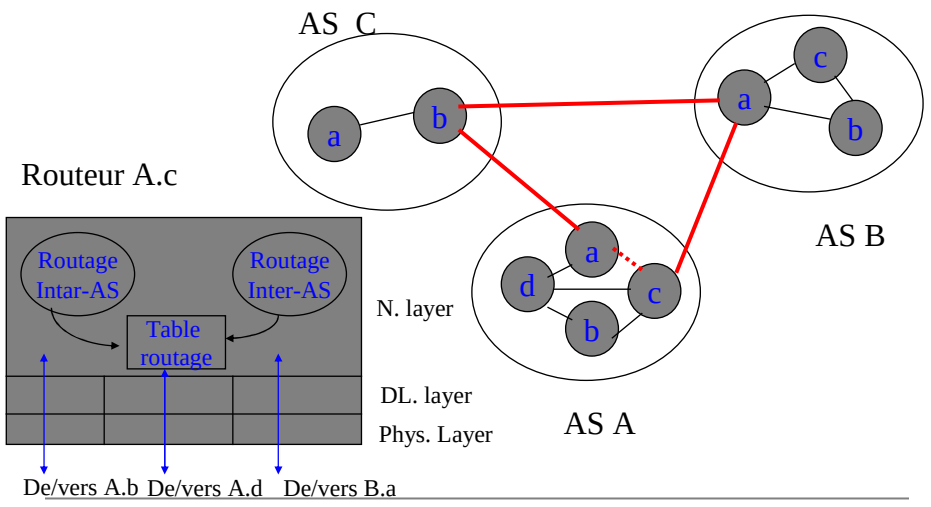
Structuration en Système Autonome

- Les systèmes autonomes sont connectés par des routeurs appelés *Gateways Routers*
- L'algorithme de routage utilisé pour déterminer la route entre les AS est appelé *inter- autonomous system routing protocol* ou protocole de routage inter-AS

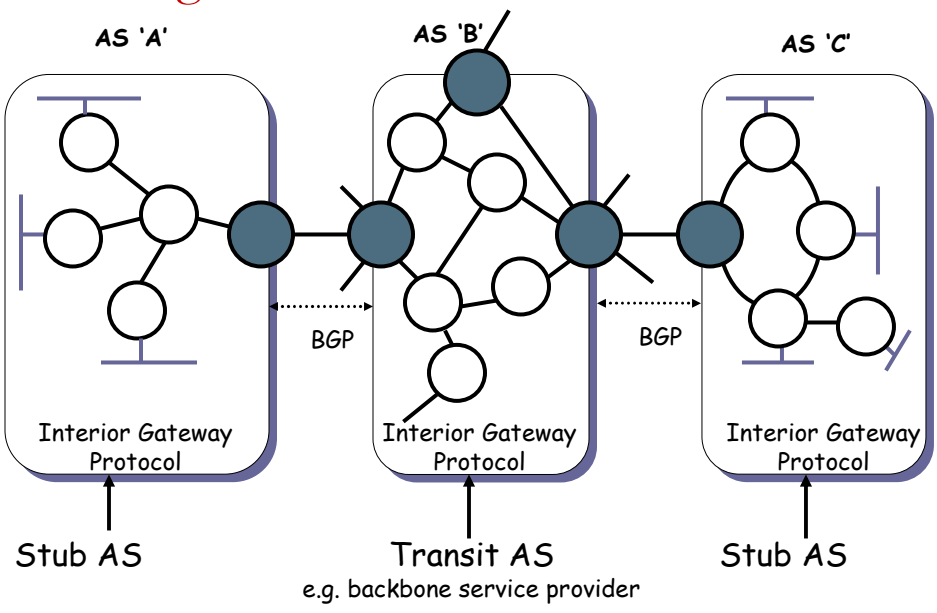
⇒ Les routeurs intra-AS ont besoin de connaître les autres routeurs intra-AS et le ou les *gateways routers*

- Entre les AS, on emploie un protocole de routage externe (EGP)
  - Exemple de BGP, BGP-4 (rfc 1771)

Systèmes autonomes



Routage dans l'Internet



Routage au sein d'un Stub AS

- Il y a un seul point de sortie, on peut donc utiliser un routage par défaut.
  - Chaque routeur connaît tous les IDs au sein d'un AS
  - Les paquets à destination des autres AS sont envoyés vers le routeur par défaut
  - Le routeur par défaut est un routeur de bordure vers le prochain AS (border gateway)
- Tables de routage assez petite dans les AS Stub

Protocoles de routage – RIP –

Routing Information Protocol

Historique de RIP

- RIPv1 : RFC1058 (06/88)
- RIPv2 : RFC1387, RFC1388 (01/93), RFC1723 (04/94)
  - Permet le routage CIDR
  - Diffusion multicast (224.0.0.9) plutôt que broadcast
  - Permet l'authentification des routeurs
- RIPng : RFC2080 (01/97), RFC2453 (11/98)
  - Adaptation pour IPv6

Contexte d'utilisation de RIP

- Usage pour des réseaux de diamètre < à 15 routeurs
- Utilisation d'une métrique fixe acceptable
  - Pas de possibilité de prendre en compte de éléments variables dans le temps
  - Une métrique composite est possible, mais elle sera statique et elle peut réduire le diamètre maximal effectif du réseau
- Temps de convergence de quelques minutes acceptable
- En IPv4 et/ou IPv6

Fonctionnement de RIP

- Basé sur l'algorithme de Belleman-Ford (type distance-vecteur)
- À chaque route (@IP+netmask) est associée une métrique (M) qui est sa distance exprimée en nombre de routeurs à traverser
- Chaque routeur envoie à ses voisins ses informations de routage (les réseaux qu'il sait router et métriques associées)
  - Toutes les 30 secondes systématiquement
- Si un routeur reçoit d'un voisin ses informations de routage
  - Il calcule les métriques locales des routes apprises (M → M+1)
  - Sélectionne les meilleures routes et en déduit sa table de routage
  - Envoie à ses voisins ses nouvelles informations de routage si elles ont changé

NB: L'algorithme cherche à produire les routes de métriques minimales, mais il est nécessaire d'avoir un mécanisme permettant de faire augmenter la métrique d'une route (cf. la propagation des routes invalides par *'poison reverse*'). Pour ce faire une route annoncée par le voisin qui est le *'next hop'* d 'une route déjà connue est toujours installée, même si sa métrique est plus importante que la route actuelle.

Fonctionnement de RIP (cont)

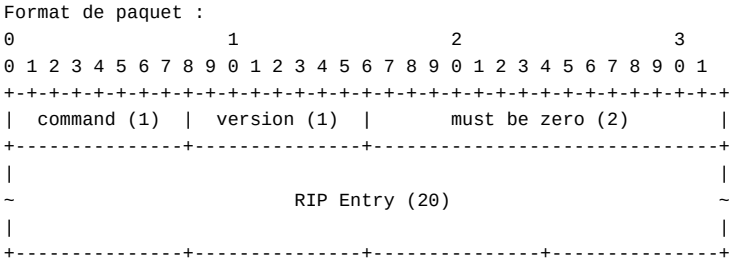
- Fonctionne au dessus des ports udp 520 (IPv4), 521 (IPv6)
- Amélioration de la convergence et de la stabilité
  - Élimination des boucles
    - *poison reverse* : les routes en provenance d'un voisin lui sont ré-annancées avec une métrique infinie
    - *split horizon* : la métrique maximum est de 15
  - Minuteurs associés
    - *routing-update* (30 secondes ± 0 à 5 secondes)
    - *route-timeout* (180 secondes)
    - *route-flush* ou *garbage-collection* (120 secondes)

Définition des timers

- **routing-update** : période maximale entre deux annonces pour un routeur.
- **route-timeout** : durée de vie associée à chacune des routes apprise par RIP. Après expiration de ce minuteur, la route est marquée comme invalide dans la table des informations RIP. Elle ne sera effacée que lorsque le minuteur route-flush expire. Ce mécanisme permet à un routeur de propager l'information de route invalide vers ses voisins (pour tenir compte d'une interface réseau qui devient inopérante par exemple). Si pendant ce temps une nouvelle route vers ce préfixe est apprise, elle remplace la route invalide.
- **route-flush** : périodicité de nettoyage de la table des informations RIP. Les routes marquées comme invalides sont effacées.

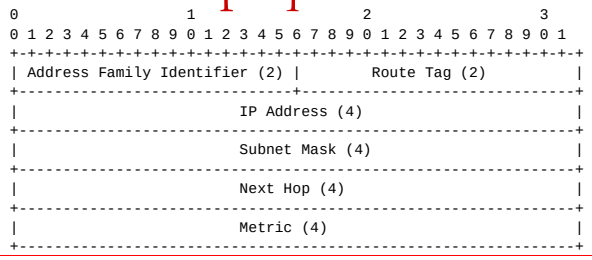
NB: Si tous les routeurs utilisaient des minuteurs *routing-update* paramétrés avec la même valeur de 30 secondes par exemple, il se produirait au bout d'un certain un phénomène de synchronisation de leurs annonces RIP. Pour éviter ce phénomène qui conduirait à des rafales de paquets et des risques de congestion cycliques, les valeurs effectives des minuteurs sont perturbées aléatoirement de 0 à 5 secondes.

Format des paquets RIPv2



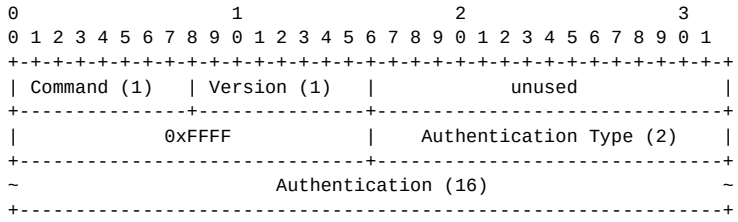
- **Commande** : indique si le paquet est une requête ou une réponse. La requête est une demande d'avoir la table des informations de routage. La réponse peut être non sollicitée (cas des émissions régulières faites par les routeurs) ou sollicitée par une requête.
- **Version** : 2 actuellement (la version 1 de RIP n'est plus utilisée)

Entrée des paquets RIPv2



- **AFI** : *(Address Family Identifier*) type de protocole
- **Route tag** : marqueur qui peut être utilisé pour distinguer les routes internes (au protocole
- (apprises par RIP) des routes apprises par d'autres protocoles (ex. OSPF).
- **Adresse du réseau** : Adresse IP donnant le préfixe
- **Masque du réseau** : champ binaire dont les bits positionnés à 1 donnent la longueur du
- préfixe
- **Adresse du routeur cible** : adresse IP où il faut router les paquets à destination du réseau
- cible
- **Métrique** : valeur de la métrique (nombre compris entre 1 et 15)
- Un préfixe est constitué de l'ensemble {adresse du réseau , masque du réseau}.
- Une route est constituée de l'ensemble des informations {AFI, tag, préfixe, adresse IP du routeur cible, métrique}.
- Les paquets de type réponse peuvent contenir jusqu'à 25 routes par paquet. S'il y a plus de 25 routes à envoyer, plusieurs paquets sont émis.

Entrée pour l'authentification



Protocoles de routage
– OSPF –

Open Shortest Path First Protocol

Historique d'OSPF

- OSPFv1 : RFC1131 (10/89) puis RFC1247 (07/91)
- OSPFv2 : RFC 2328 (04/98)
- OSPFv3 : RFC 2740 (12/99)
  - Adaptation pour IPv6

Pourquoi OSPF ?

- OSPF a été conçu pour s'affranchir des limitations de RIP
  - Possibilité de gérer des domaines de diamètre > 16
  - Amélioration du temps de convergence
  - Métrique plus sophistiquée (prise en compte des débits)
  - Meilleure possibilité d'agrégation des routes
  - Segmentation possible du domaine en aires
- Mais OSPF est aussi
  - Plus complexe (routeurs plus puissants, configuration Moins simple que RIP)

OSPF

- Pour de grands réseaux,
  - Force à structurer le réseau
  - Poids attribuable aux liens, généralement fonction du débit

$$coût = \frac{10^8}{\text{bande passante en } b/s}$$

  - Pour un Ethernet à 10 Mbit/ s le coût est de 10
- Utilise l'algorithme du Link State

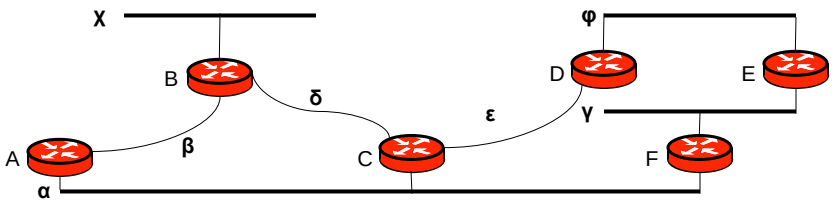
Fonctionnement d'OSPF

- Chaque routeur identifie (ou connaît par configuration) ses voisins
- S'il y a plusieurs routeurs sur un réseau, un routeur principal (et un routeur principal de secours) sont élus parmi eux
- Chaque routeur acquiert la base de donnée du routeur principal
- Chaque routeur diffuse à ses voisins (messages de type LSA)
  - La liste de ses voisins immédiats
  - Le coût (métrique) de la liaison vers chacun de ses voisins
- Chaque routeur met à jour sa base de données, ce qui lui donne une vision globale du réseau
- Chaque routeur calcule ses meilleures routes (métrique minimum) et en déduit sa table de routage

Calcul des meilleures routes

- OSPF utilise l'algorithme de Djkstra : «Shorted Path First»
- À partir de la table des informations sur l'état des liens qui est unique et partagée par tous les routeurs
- Chaque routeur construit sa vision optimum du réseau sous forme d'un arbre qui minimise les coûts des routes vers les réseaux cibles
- La construction de l'arbre se fait en choisissant toujours en premier la branche de coût minimum

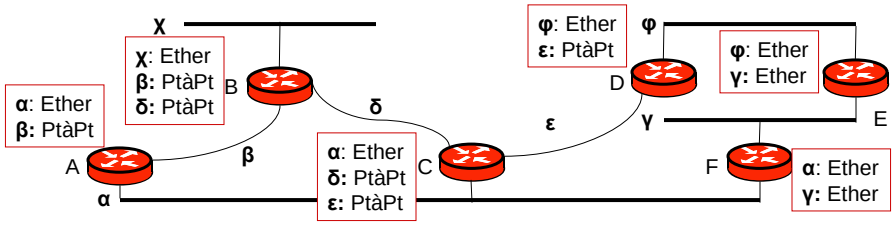
Exemple



- Chaque routeur connaît uniquement les réseaux auxquels il est connecté

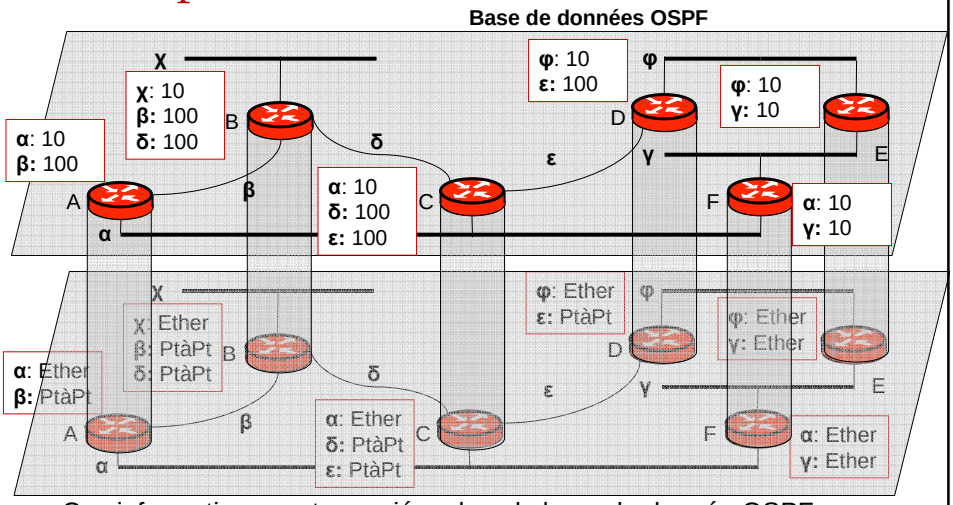
Exemple

Table de routage



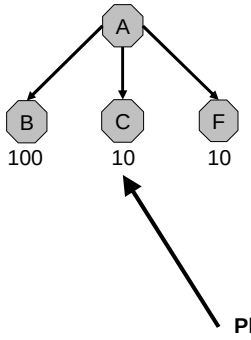
- Ces informations sont recopiées dans la base de donnée OSPF du routeur avec le coût choisi par l'administrateur

Exemple



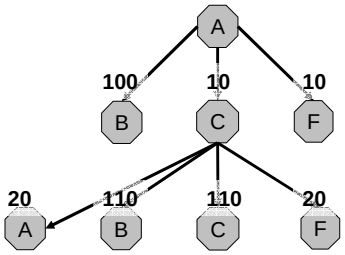
- Ces informations sont recopiées dans la base de donnée OSPF du routeur avec un coût

On déroule l'algorithme pour A



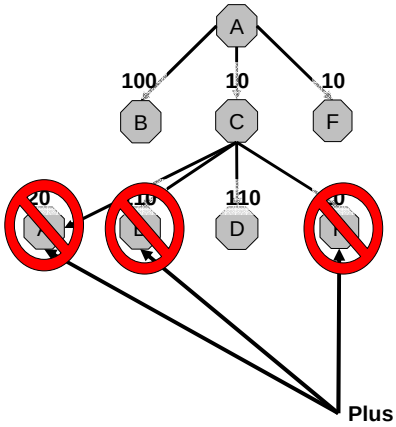
- A regarde les coûts pour ses voisins immédiats

On déroule l'algorithme pour A



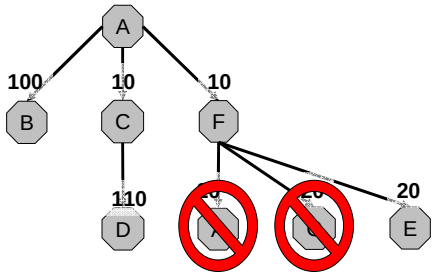
- A regarde les coûts cumulés pour les voisins de C

On déroule l'algorithme pour A

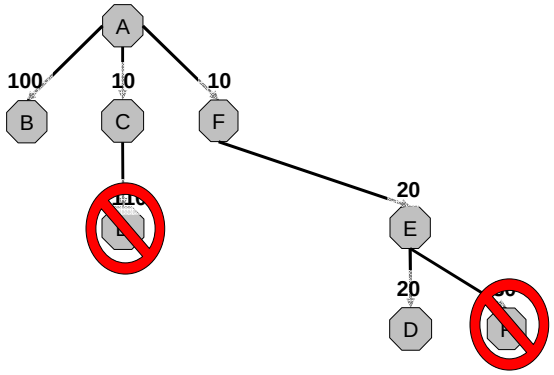


- A s'intéresse maintenant au plus petit chemin de l'arbre non encore exploré
- A calcule des coûts cumulés pour les voisins immédiats de F

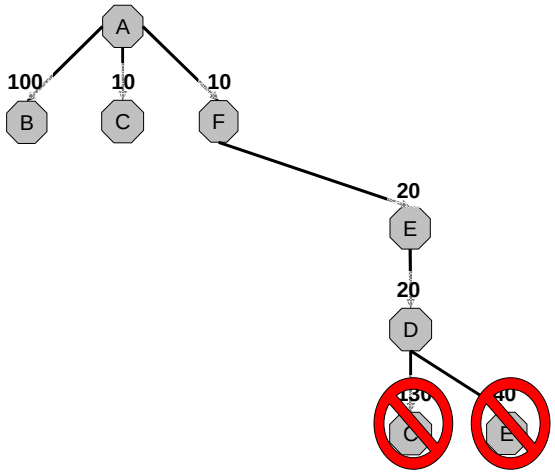
On déroule l'algorithme pour A



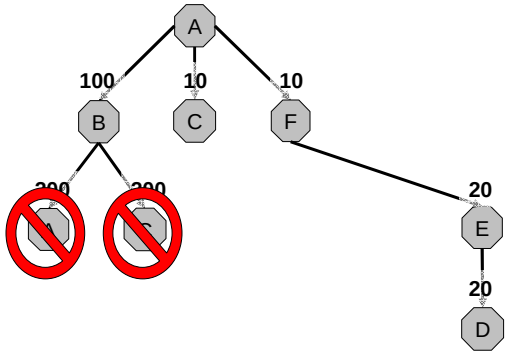
On déroule l'algorithme pour A



On déroule l'algorithme pour A

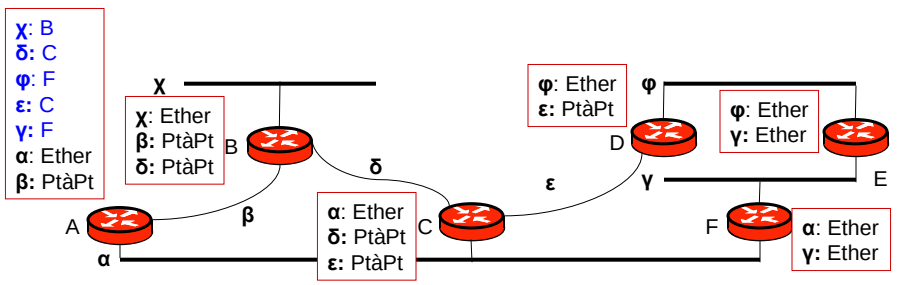


On déroule l'algorithme pour A



- A en déduit l'arbre des plus courts chemins
- A peut donc calculer sa table de routage

Mise à jour de la table de routage de A

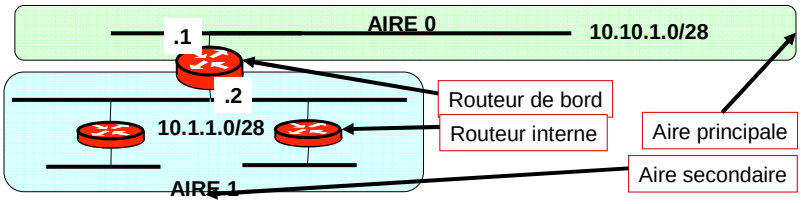


- Idem pour F…

Optimisation d'OSPF : les aires

- Dans grand domaine, chaque changement provoque une diffusion de la table de l'état des liens de tous les routeurs, ce qui provoque
  - Une consommation de bande passante importante
  - Une charge CPU importante sur les routeurs
  - Alors que la portée d'une modification reste assez localisée
- D'où l'idée de découper le domaine en aires
  - Chaque aire est plus simple, plus stable -> plus simple à traiter
- Pour garder une cohérence globale, une aire principale (*backbone*)
  - Relie toutes les aires entre elles
  - Connaît toutes les infos de routage, mais ne diffuse que des condensés (en agrégeant les routes, si c'est possible)

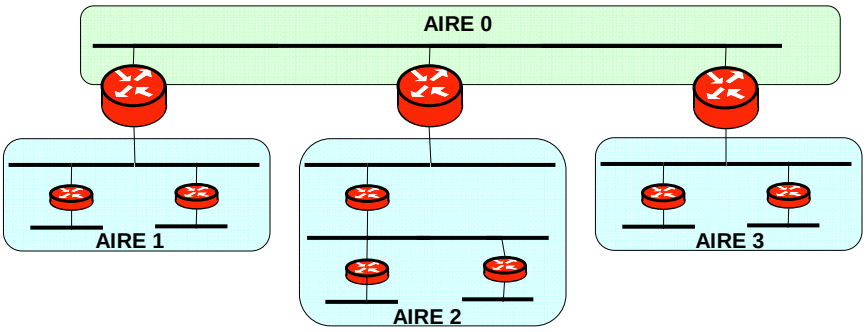
Aire principale et aires secondaires



- numéro 0 → aire ***principale***
- Il y a toujours **une et une seule** aire principale par domaine OSPF.
- Si on ne veut pas découper le domaine, toutes les interfaces de tous les routeurs sont dans l'aire 0.
- Les autres aires sont de type ***Secondaire*** et doivent être connectée à l'aire principale

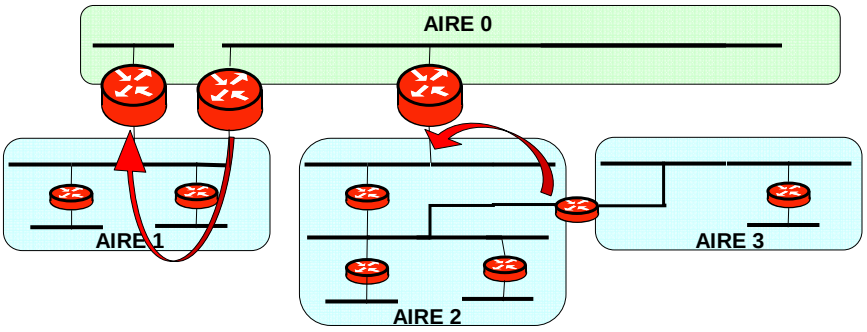
```
router OPSF 100 // router ospf <process-id>
network 10.10.1.1 0.0.0.0 area 0
network 10.1.1.2  0.0.0.0 ares 1
```


Aire principale, secondaires, terminales



- **ce sont les interfaces des routeurs (les liens) qui sont positionnés dans les différentes aires.**
- **network *<network or IP address> <mask> <area-id>***
- **L'aire 3 est une aire secondaire de type terminale (Stub). Ce type d'aire a une gestion simplifiée car l'unique point de sortie vers l'aire principale permet de gérer le routeurs en leur diffusant une route par défaut.**
- **`area <area-id> stub [no-summary]`**

Lien virtuel



- Pour connecter (logiquement) une aire terminale à l'aire principale quand la connexion physique ne peut être réalisée.
- Pour rétablir la continuité de l'aire principale quand cette dernière n'est plus assurée (il s'agit bien sûr d'une solution de dépannage !).
- On notera que les liens virtuels font dépendre le comportement du routage dans une aire de ce qui se passe dans une autre, ils augment ainsi la probabilité d'instabilité du routage.
- **`area <area-id> virtual-link <RID>`, *ares-id == tranbsit area***

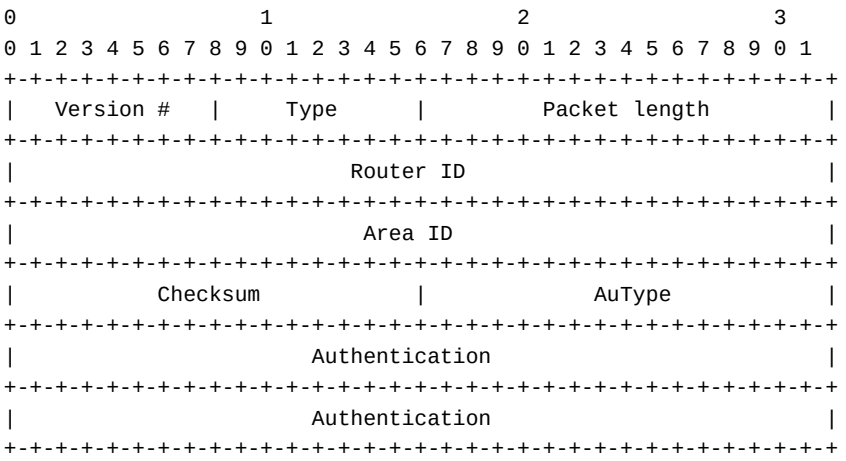
Paquets OSPF

- Hello
  - Maintenance des liens, identification des voisins
- Exchange / description
  - Échange initial des tables de routage
- Demande
- Flooding
  - Mise à jour incrémentale des tables

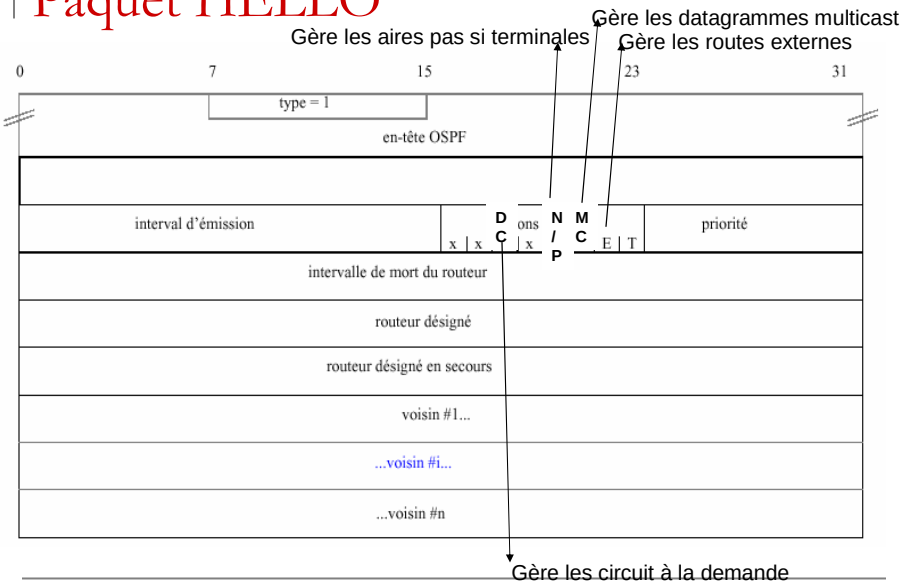
Entrée des tables OSPF database

- Router links
  - Summarizes links from advertising router
- Network links
  - Transit networks (broadcast and non-broadcast)
- Summary links
  - Summary info advertised by area border routers
- External links
  - Imported routers, typically from a EGP

Common OSPF header



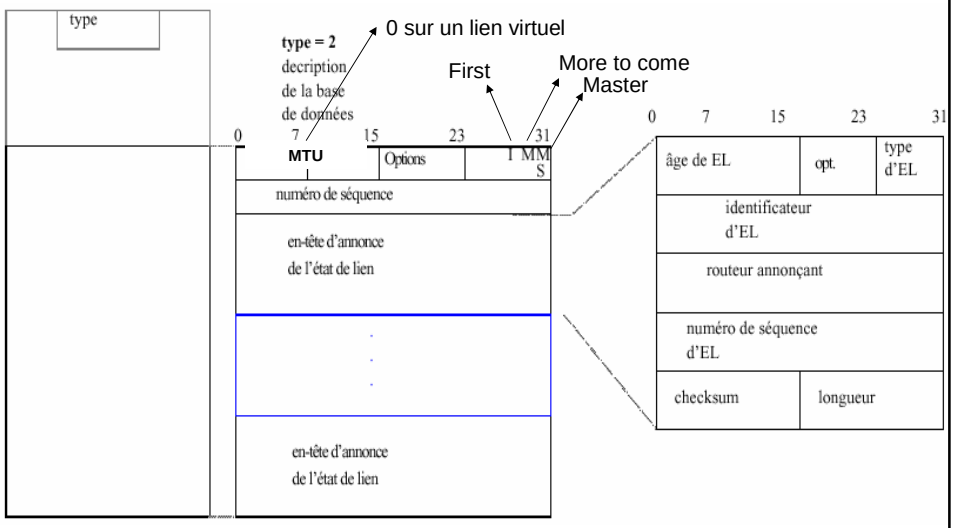
Paquet HELLO



Algorithme du *link state*

- chaque routeur identifie ses voisins immédiats.
- un routeur principal (designated router) et de secours (backup designated router) sont désignés sur chaque réseau grâce à un mécanisme d'élection.
- le routeur acquière la base de donnée du routeur désigné.
- chaque routeur construit un message contenant la liste de ses voisins immédiats ainsi que le coût associé à la liaison. Ce message sera appelé LSP pour Link State Packet.
- ce paquet est transmis à tous les autres routeurs du réseau avec un mécanisme de diffusion qui limite la propagation des messages et évite les boucles.
- chaque routeur met à jour sa base de donnée ce qui lui donne une vision globale du réseau et il peut en déduire ses tables de routage.

Paquet de description de la BD



Champ « type état de lien »

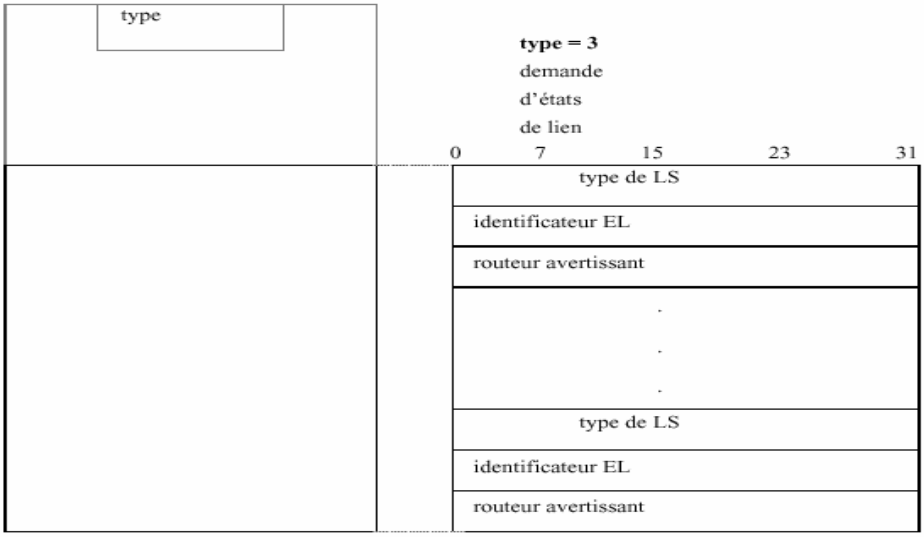
- 1 : liaison du routeur
  - (net attaché au routeur)
- 2 : liaison dans le net
  - (dans les net non diffusants)
- 3 : récapitulatif
  - (exporter l ens. des net que peut gérer un routeur)
- 4 : résumé des @ de routeur externes
  - (simplification de 3 pour les aires terminales
- 5 : réseaux externes accessibles
  - (pour info apprise par un protocole externe)

Id d'Etat de Liaison

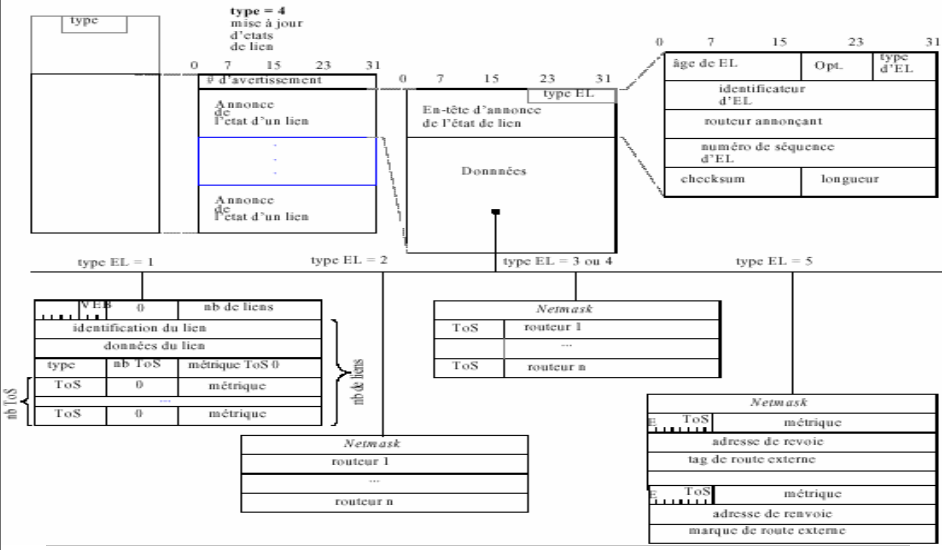
- EL = 1 : Id du routeur qui a produit le msg
- EL = 2 : @ IP du DR
- EL = 3 : @ du net / host accessible
- EL = 4 : Id du routeur de frontière
- EL = 5 : @ du net externe

- Les routeurs échangent tour à tour des DD. Si un enregistrement n'est pas dans la base, ou ancienne valeur, le routeur demande l'intégralité de l'EL (LSR).
- Info envoyé dans un Link State Update

Paquet de demande



Paquet de mise à jour



Différents types d'EL

- Type EL = 1
  - Routeur annonçant : un routeur
  - id. d'EL : id du routeur
  - Récepteur : intérieur d'une aire
  - Le routeur donne la liste des réseaux auxquels il est attaché
  - V = virtuel / E = frontière d'AS / B = frontière d'aire

Type	Nature	Id link	Data link
1	Pt-to-pt	@ IP oposite router	Num NIC
2	To a transit net	@ IP DR	@ IP router sur l'interface
3	To a terminal net	@ IP net	Netmask du net
4	Virtual link	@ IP oposite routeur	@ IP router sur l'interface

Différents types d'EL

- Type EL = 2
  - Routeur annonçant : un routeur d'un NBMA
  - id. d'EL : id du DR sur le net
  - Récepteur : les routeurs u NBMA
  - Dans un NBMA les équipements doivent être configurés pour connaître leurs voisins (pas de diffusion, X25 / FR). Ce message doit permettre aux routeurs de connaître leurs voisins.

Différents types d'EL

- Type EL = 3
  - Routeur annonçant : un routeur en bordure d'aire
  - id. d'EL : @ Ip de la liaison annoncée
  - Récepteur : les routeurs des autres aires / backbone
  - Ce msg contient la liste de tous les réseaux accessibles dans une aire, i.e., tous les net que le routeur de bordure a appris.
  - Les routeurs des autres aires ne connaissent pas le chemin pour atteindre ces réseaux mais le routeur vers lequel il faut envoyer.

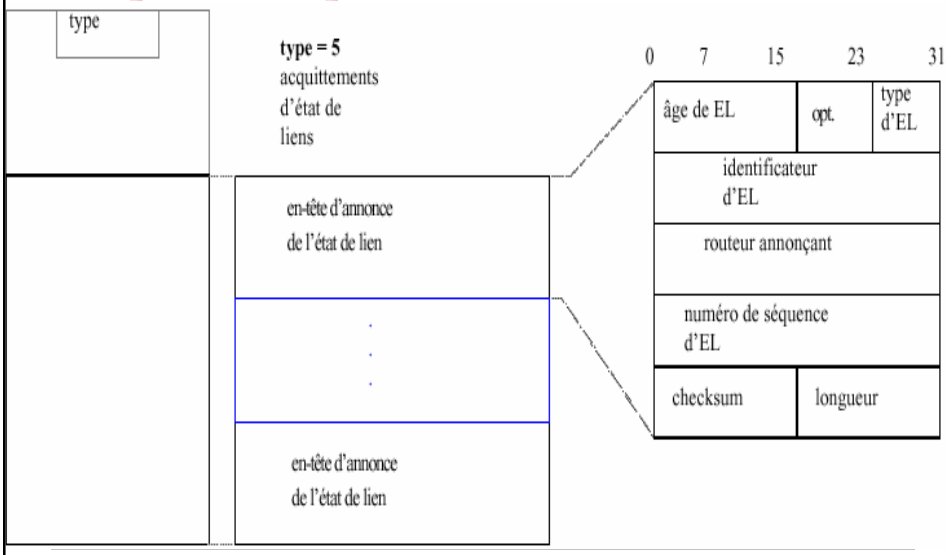
Différents types d'EL

- Type EL = 4
  - Routeur annonçant : un routeur en bordure d'aire
  - id. d'EL : Id du routeur récepteur
  - Récepteur : routeurs d'aire terminale
  - Ce msg contient un sous ensemble de la liste de tous les réseaux accessibles dans une aire.
  - Une aire terminale n'a pas à connaître toutes les infos puisqu'elle ne les rediffuse pas.

Différents types d'EL

- Type EL = 5
  - Routeur annonçant : un routeur en bordure de l'AS
  - id. d'EL : @IP de la liaison annoncée
  - Récepteur : autres routeurs de l'AS
  - Ce msg contient les routes externes apprises par le routeur de bordure de l'AS via les protocoles de routage externe.

Paquet d'acquitement



Pourquoi OSPF est "si" complexe ?

- Plusieurs bases de données
  - Routes dans le réseau
  - Table de routage vers les autres réseaux du domaine
  - Table de routage vers les autres AS
  - @ des routeurs vers les NBMA (Non Broadcast Multiple Access)

Beaucoup de fonctionnalités

- Inonder une aire avec des infos de routage
- Diffuser les tables de routage
  - Construite par un routeur frontière
  - Cacher la complexité aux autres
- Connaître tous les routeurs d'un NBMA
- Synchroniser / récupérer les informations
  - Toutes les infos ne sont pas émises en permanence
- Élection du DR
- S'assurer du fonctionnement des routeurs voisins

Fonctions / sous protocoles

	1 = Hello	2 = DD	3=req LS	4=update	5 = ack
Flooding des updates				X	X
Diffusion des tables				X	
Routeur ds NBMA			X	X	
Synch		X	X		
Élection DR	X				
Voisins	X				

Protocoles de routage
– BGP –

Border Gateway Protocol

Protocoles de routage externe

- Topologie : L'Internet est un réseau maillé entre AS complexe avec « peu » de structures
- Autonomie des AS : Chaque AS définit le coût des liens avec des philosophies/approches différentes. Il est impossible de trouver les plus courts chemins
- Confiance : des AS peuvent ne pas se faire confiance mutuellement pour propager de « bonnes routes »
- Opérateurs concurrents, nation en guerre…
- Politiques : Des AS différents ont des objectifs différents
  - Router en peu de sauts, utiliser un provider plutôt qu un autre…

Objectifs généraux de BGP

- Échanger des routes (du trafic) entre organismes indépendants
  - Opérateurs
  - Gros sites mono ou multi connectés
- Implémenter la politique de routage de chaque organisme
  - Respect des contrats passés entre organismes
  - Sûreté de fonctionnement
- Être indépendant des IGP utilisés en interne à un organisme
- Supporter un passage à l'échelle (de l'Internet)
- Minimiser le trafic induit sur les liens
- Donner une bonne stabilité au routage

BGP en quelques chiffres

N: nbr de préfixes	M: dist. Moy.	A: nbr moy. AS	K: nbr moy. de voisins	Vol init. échangé	Vol mémoire
2100	5	59	3	9000	27000
4000	10	100	6	18000	108000
10000	15	300	10	49000	490000
20000	8	400		86000	
40000	15	400		172000	
100000	20	3000	20	520000	1040000

- Volume d'information : O(N + M * A)
- Volume mémoire : O(N + M * A * K)

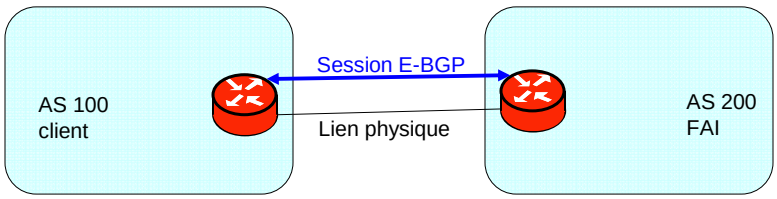
Principes généraux du protocole BGP

- Protocole de type PATH-vecteur
- Chaque entité est identifiée par un numéro d'AS
- La granularité du routage est l'AS
- Le support de la session BGP est TCP:179
  - garantie d'une bonne transmission des informations.
  - Envoi initial puis mise à jour
- Les sessions BGP sont établies entre les routeurs de bord d'AS
- Protocole point à point entre routeurs de bord d'AS
- Protocole symétrique
- Politique de routage →filtrage des routes apprises et annoncées
  - tout ou rien sur la route (annonce, prise en compte),
  - modification des attributs de la route pour modifier la préférence

ne jamais oublier qu'annoncer une route vers un réseau c'est accepter du trafic à destination de ce réseau

Exemple de connexion BGP (1)

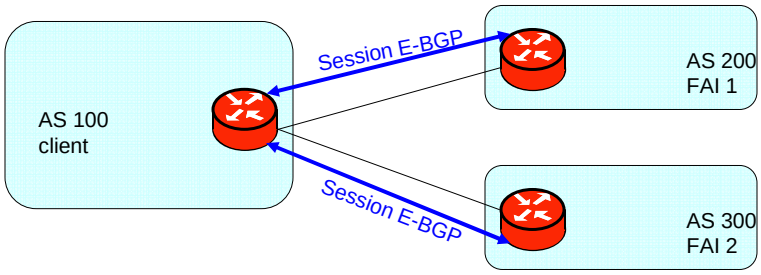
- Client connecté à un seul Fournisseur d'Accès Internet (FAI). Seuls les routeurs de bord de l'AS sont figurés.
- Les routeurs qui échangent leurs informations en BGP doivent être directement connectés (liaison point à point ou LAN partagé).
- L'utilisation de numéros d'AS privés est à éviter pour des AS terminaux (clients) car une connexion à un deuxième AS de transit (FAI) peut conduire à une configuration illégale.



AS officiels (enregistrés) : de 1 à 64511
AS privés (non-enregistrés) : de 64512 à 65535

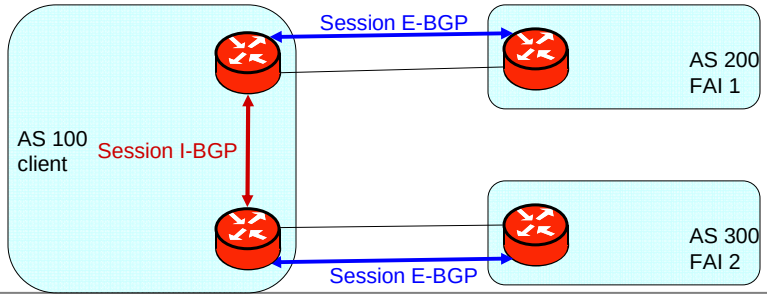
Exemple de connexion BGP (2)

- Client connecté à deux FAI
  - faire passer tout son trafic par FAI1 (AS 200) et garder sa liaison vers FAI2 (AS 300) en secours
  - Équilibrer son trafic entre FAI1 et FAI2.
- C 'est le cas typique qui amène à utiliser le protocole de routage BGP pour réagir dynamiquement en cas de défaillance d'un lien.



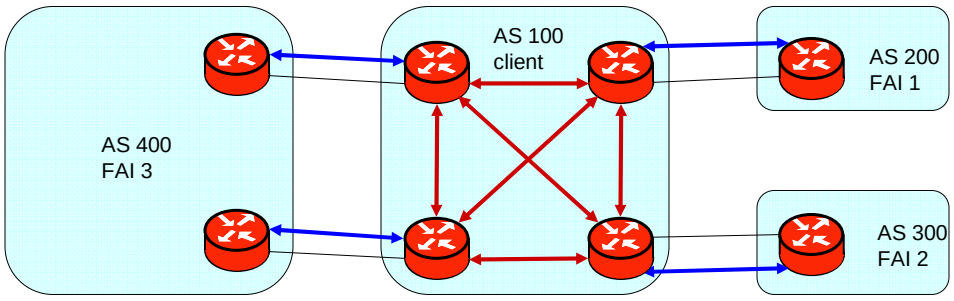
Exemple de connexion BGP (3)

- Client connecté à deux FAI par 2 routeurs
  - protection contre la défaillance de l'un d'entre eux ou de l'un de ses routeurs
- Connexion BGP entre les routeurs de bord de l'AS 100.
  - maintenir la cohérence entre les 2 routeurs qui doivent posséder les mêmes informations de routage
  - En BGP la granularité du routage est l'AS !



Exemple de connexion BGP (4)

- Client connecté à 3 FAI avec redondance sur l'un
- Maillage complet de sessions I-BGP
- Pour les autres AS, les 4 routeurs de bord de l'AS 100 sont vus, du point de vue fonctionnel comme un seul routeur (avec 4 interfaces).



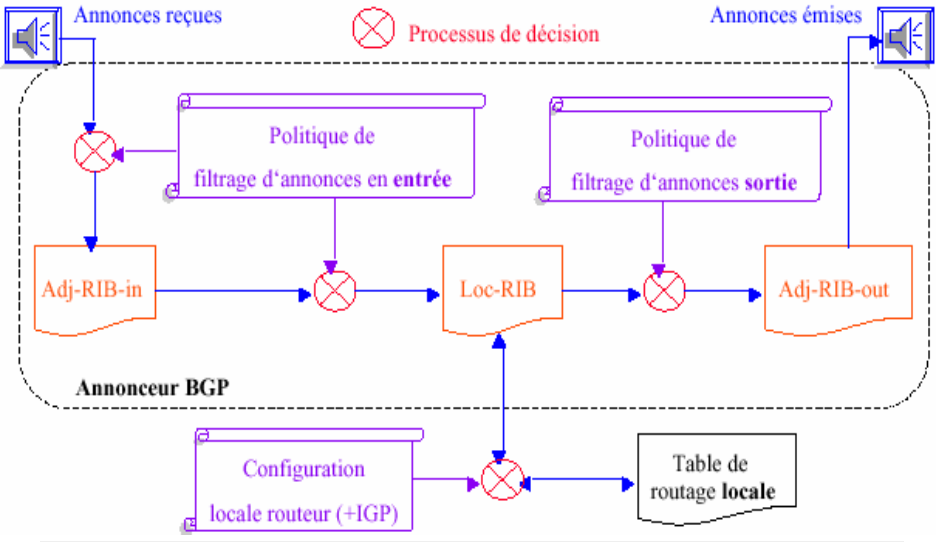
Règles pour les AS multi-connectés

- Les routeurs de bord d'un même AS échangent leurs informations de routage en I-BGP
- Les connexions en I-BGP forment un maillage complet sur les routeurs de bord d'un AS
- Ce sont les IGP internes à l'AS qui assurent et maintiennent la connectivité entre les routeurs de bord qui échangent des informations de routage en I-BGP
- Le numéro d'AS est un numéro officiel (si connexions vers 2 AS différents)
- Dans un même AS, c'est bien l'IGP (ou le routage statique) qui est responsable de la connectivité interne de l'AS.
  - Si un routeur de bord ne peut pas atteindre une route de son AS (qui lui a été annoncée par un voisin interne par exemple), il ne la propagera pas à ses voisins BGP (externes ou internes).

Les composants d'un annonceur BGP

- Une description des politiques de routage
  - entrée et sortie
- Des tables où sont stockées les informations de routage
  - En entrée : table **Adj-RIB-in**
  - En sortie : table **Adj-RIB-out**
  - En interne : table **Loc-RIB**
- Un automate implémentant le processus de décision
- Des sessions avec ses voisins pour échanger les informations de routage

Schéma fonctionnel du processus BGP



La vie du processus BGP

- Automate à 6 états, qui réagit sur 13 événements
- Il interagit avec les autres processus BGP par échange de 4 types de messages :
  - **OPEN**
  - **KEEPALIVE**
  - **NOTIFICATION**
  - **UPDATE**
- Taille des messages de 19 à 4096 octets
- Éventuellement sécurisés par MD5

Le message OPEN

- 1er message envoyé après l'ouverture de la session TCP. Informe son voisin de :
  - Sa version de BGP
  - Son numéro d'AS
  - D'un numéro identifiant le processus BGP
- Propose une valeur de temps de maintien de la session
  - Valeur suggérée : 90 secondes
  - Si 0 : maintien sans limite de durée
- Met le processus en attente d'un KEEPALIVE

Le message KEEPALIVE

- Confirme un OPEN
- Réarme le minuteur contrôlant le temps de maintien de la session
- Si temps de maintien non égal à 0
  - Est réémis toutes les 30 secondes (suggéré)
- Message de taille minimum (19 octets)
- En cas d'absence de modification de leur table de routage, les routeurs ne s'échangent plus que des messages KEEPALIVE toutes les 30 secondes, ce qui génère un trafic limité à environ 5bits/s au niveau BGP.

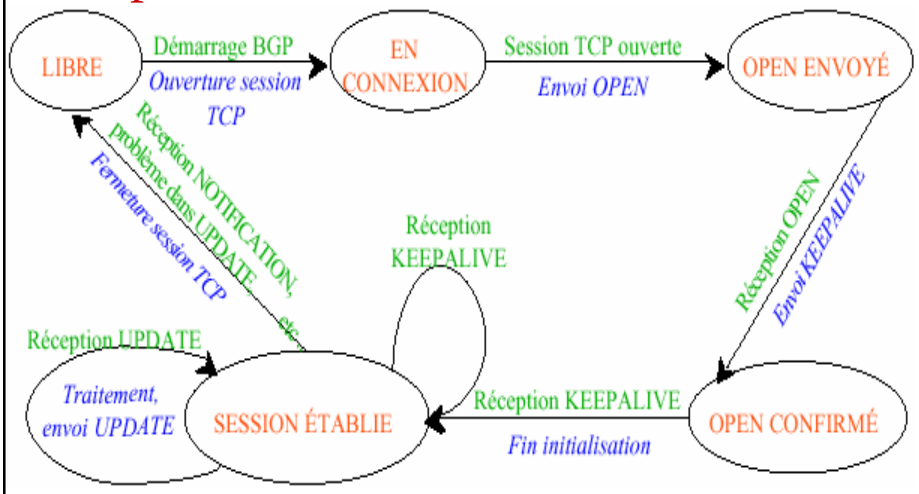
Le message NOTIFICATION

- Ferme la session BGP
- Fournit un code et un sous code renseignant sur l'erreur
- Ferme aussi la session TCP
- **Annule toutes les routes apprises par BGP**
  - peut provoquer des instabilités de routage injustifiées
  - un incident ne veut pas forcément dire que toutes les routes apprises précédemment sont devenues fausses
- Émis sur incidents :
  - Pas de KEEPALIVE pendant 90s (<*hold time*>)
  - Message incorrect
  - Problème dans le processus BGP

Le message UPDATE

- Sert à échanger les informations de routage
  - Routes à éliminer (éventuellement)
  - Ensemble des attributs de la route
  - Ensemble des réseaux accessibles (NLRI)
  - Chaque réseau est défini par (préfixe, longueur)
- Envoyé uniquement si changement
- Active le processus BGP
  - Modification des RIB (Update, politique de routage, conf.)
  - Émission d'un message UPDATE vers les autres voisins

Le processus BGP



- Automate simplifié au chemin principal

Le message UPDATE : attributs de la route

BGP doit savoir le traiter

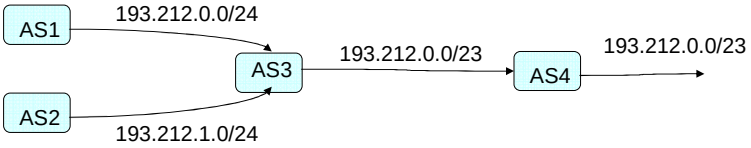
- Reconnus, obligatoires
  - **ORIGIN, AS_PATH, NEXT_HOP**
- Reconnus, non obligatoires
  - **LOCAL_PREF**, **ATOMIC_AGGREGATE**
- Optionnels, annonçables (transitifs ou non)
  - **MULTI_EXIT_DISC (MED), AGGREGATOR**
- Optionnels, non-annonçables
  - **WEIGHT (spécifique à Cisco)**

Porté illimitée

Agrégation

- Tout domaine sans route par défaut
  - Connaître toutes les routes (> 120000)
  - Dans les tables de routage IP
  - Dans les annonces BGP
- Agrégation permet de réduire le nombre de routes

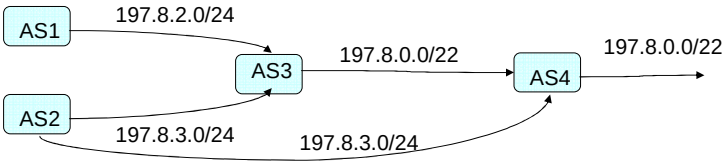
Agrégation I



- AS1 : 193.212.0.0/24
- AS2 : 193.212.1.0/24
- AS3 : 193.212.0.0/23
- AS4 : 193.212.0.0/23

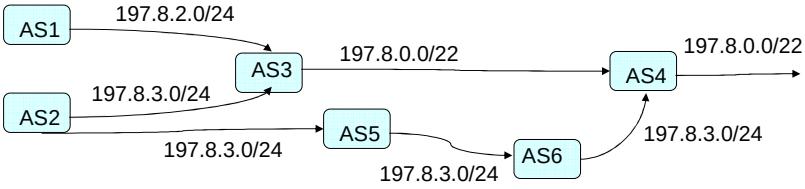
- AS_PATH 1
- AS_PATH 2
- AS_PATH 3 {1 2}
- AS_PATH 4 3 {1 2}

Agrégation II



- AS4 reçoit
  - 197.8.0.0/22
  - 197.8.3.0/24
  - AS_PATH 3 {1 2}
  - AS_PATH 2
- Les 2 routes sont injectées dans les tables de AS4
- Comment sont routés les paquets de n4 vers n2 ?

Agrégation III



- AS4 reçoit
  - 197.8.0.0/22
  - 197.8.3.0/24
  - AS_PATH 3 {1 2}
  - AS_PATH 6 5 2
- Les 2 routes sont reçues
- Seul les plus courtes sont injectées dans les tables de AS4
- Comment sont routés les paquets de n4 vers n2 ?

Les attributs de route obligatoires (1)

- ORIGIN : Donne l'origine de la route :
  - IGP (`i`) : la route est intérieure à l'AS d'origine
  - EGP (e) : la route a été apprise par **le protocole** EGP (historique car EGP non employé)
  - Incomplète (?) : l'origine de la route est inconnue ou apprise par un autre moyen (redistribution des routes statiques ou connectées dans BGP par exemple)
- **`show ip bgp`**

Les attributs de route obligatoires (2)

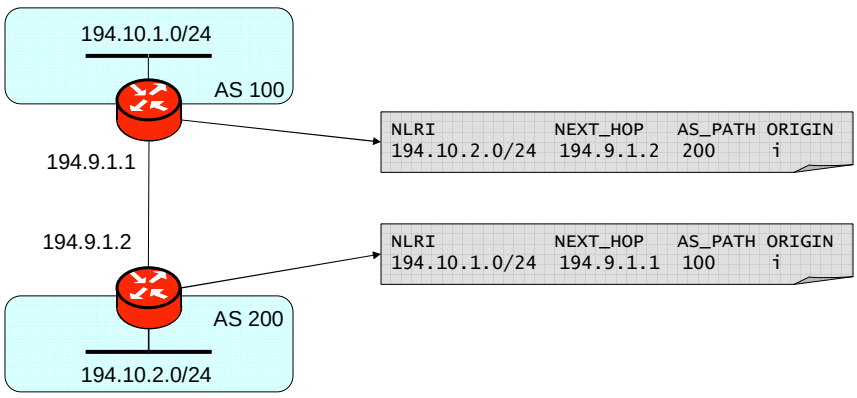
- AS_PATH
  - Donne la route sous forme d'une liste de segments d'AS
  - Les segments sont ordonnés ou non (AS_SET)

Les segments d'AS non ordonnés sont formés par un routeur qui a fait une opération d'agrégation. Ce dernier regroupe dans cet ensemble non ordonné tous les AS associés aux routes qu'il a agrégées. Cela permet aux autres routeurs de continuer à détecter d'éventuelles boucles concernant ces routes.

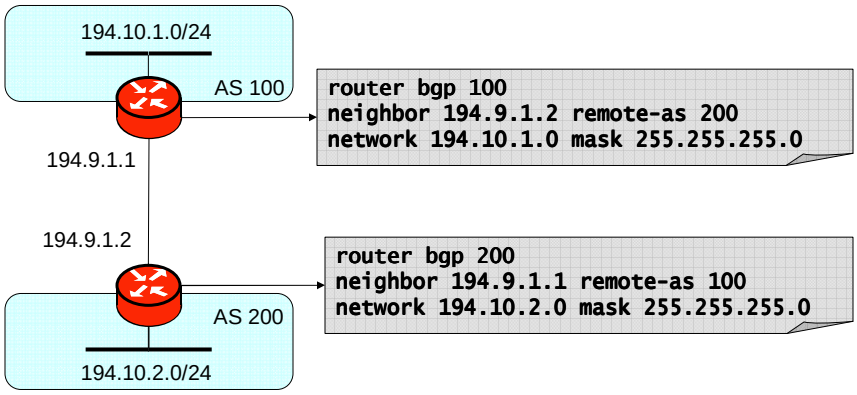
  - Chaque routeur rajoute son numéro d'AS aux AS_PATH des routes qu'il a apprises avant de les ré-annoncer

- NEXT_HOP
  - Donne l'adresse IP du prochain routeur qui devrait être utilisé (peut éviter un rebond si plusieurs routeurs BGP sont sur un même réseau local)

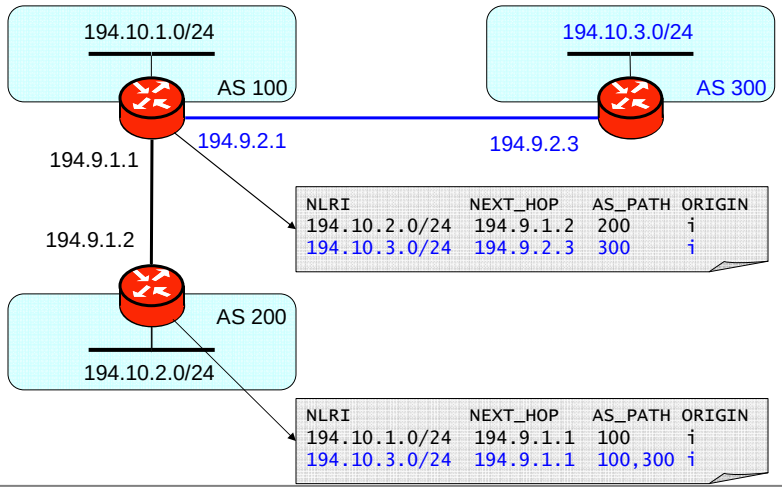
Exemple 1 : tables Adj-RIB-in



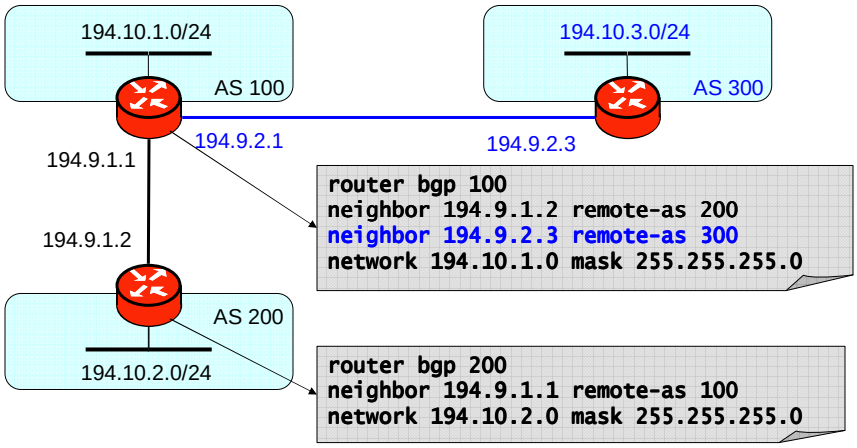
Exemple 1 : configuration sur IOS



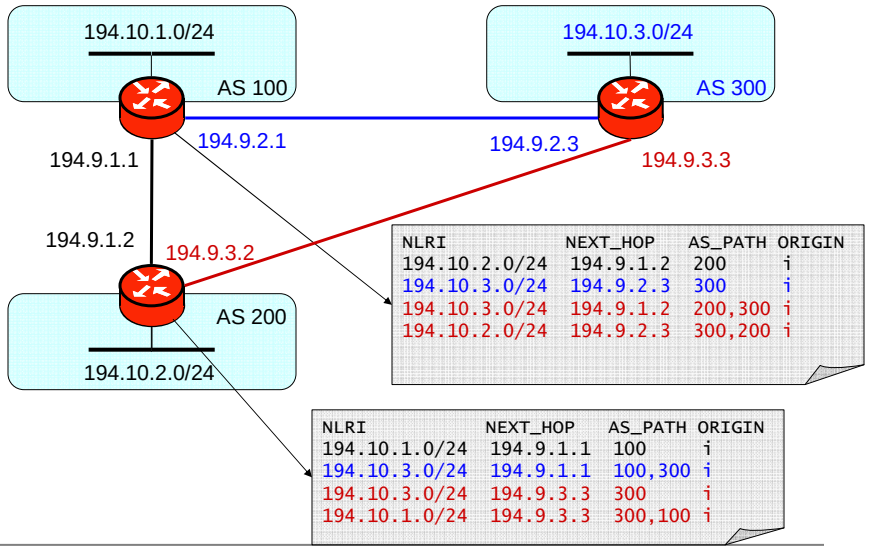
Exemple 2 : tables Adj-RIB-in



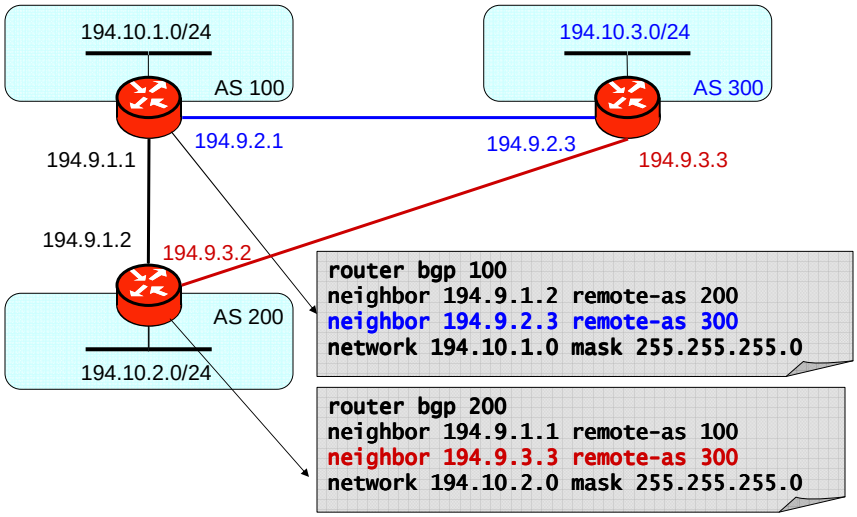
Exemple 2 : configuration sur IOS



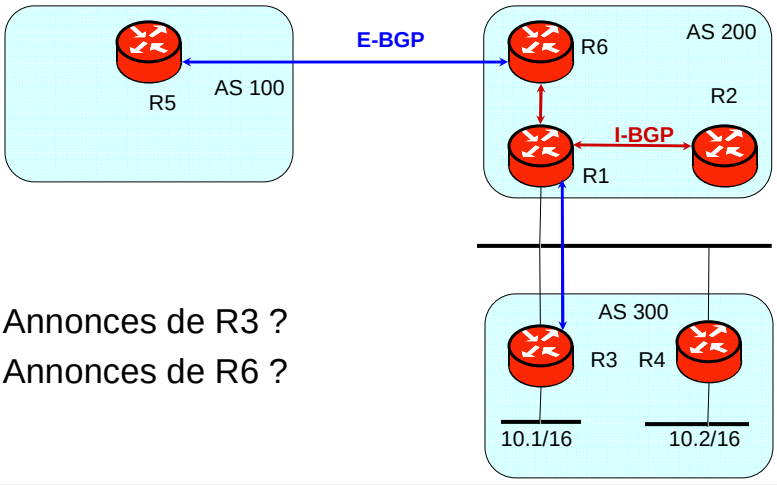
Exemple 3 : tables Adj-RIB-in



Exemple 3 : tables Adj-RIB-in



Next Hop I

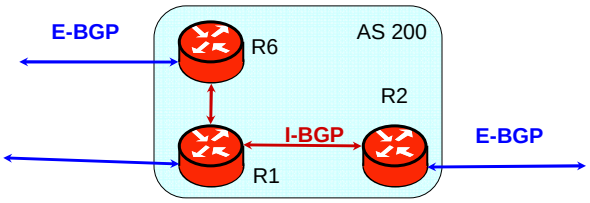


- Annonces de R3 ?
- Annonces de R6 ?

Les attributs de route optionnels (1)

- LOCAL_PREF (non transitif, discretionary)
  - Pondère la priorité donnée aux routes en interne à l'AS
  - Jamais annoncé en E-BGP (en interne donc !)
  - Pris en compte avant la longueur de AS_PATH
- ATOMIC_AGGREGATE (transitif, discretionary)
  - Indicateur d'agrégation
  - Quand des routes plus précises ne sont pas annoncées
- AGGREGATOR (transitif)
  - Donne l'AS qui a formé la route agrégée
  - L'adresse IP du routeur qui a fait l'agrégation

LOCAL-PREF



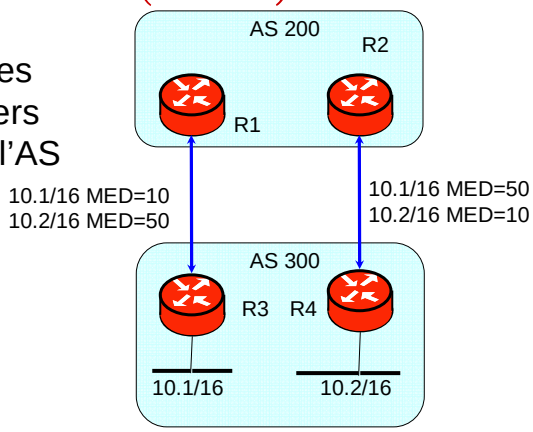
- Mis en œuvre par les routeurs à la réception de route sur E-BGP
  - Propagé sans changement par I-BGP
- R6 associe pref=100, R2 pref=10
- R1 choisit la plus grande préférence
- Bgp default local-preference *pref-value*

Les attributs de route optionnels (2)

- MULTI_EXT_DISC ou MED (non transitif)
  - Permet de discriminer les différents points de connexion d'un AS multi-connecté (plus faible valeur préférée)
- WEIGHT (non transitif, spécifique Cisco)
  - Pondère localement (au routeur) la priorité des routes BGP
- COMMUNITY (transitif)
  - Pour un ensemble de routeurs ayant une même propriété
    - no-export : pas annoncé aux voisins de la confédération
    - no-advertise : pas annoncé aux voisins BGP
    - no-export-subconfed : pas annoncé en E-BGP

MULTI-EXIT-DISC (MED) I

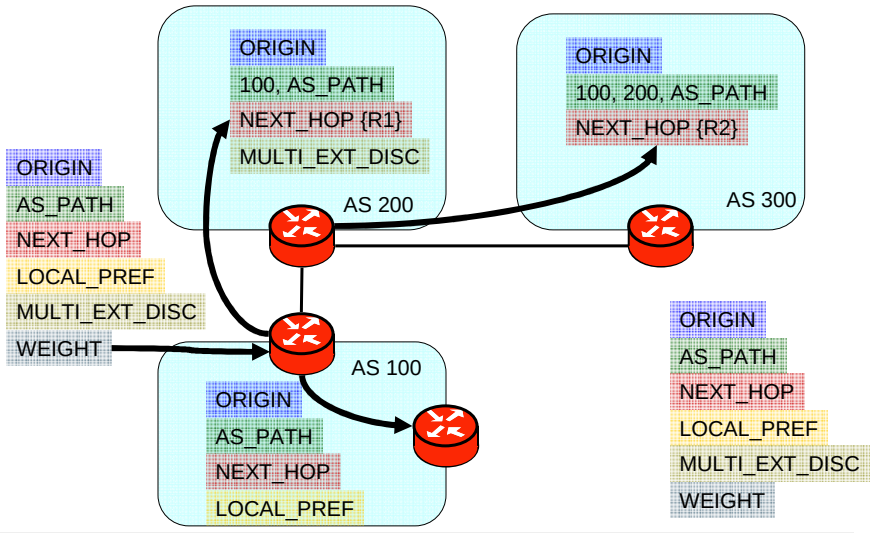
- Si AS 200 accepte les MEDs, le trafic va vers le lien privilégié par l'AS 300
- Plus petite MED



MED

- Soit ASx et ASy deux peers et tous les deux sont des IPS.
  - Pourquoi ASx n'est pas intéressé par prendre en compte les MEDs ?
- Par quel moyen ASx prend la route la plus courte vers ASy ? (Supposez que ASx emploie OSPF en interne).

La portée de quelques attributs de route



Le processus de décision

- Il est enclenché par une annonce de route
- Il se déroule en trois phases
  - Calcul du degré de préférence de chaque route apprise
  - Choix des meilleures routes à installer dans RIB-Loc
  - Choix des routes qui vont être annoncées
- Il applique aux informations de routage un traitement basé sur
  - Critères techniques : suppression boucles, optimisations…
  - Critères administratifs : application de la politique de routage de l'AS.
    - une annonce de route doit avoir son NEXT_HOP routable.
    - Une route interne n'est annoncée par un routeur que s'il sait la joindre.
    - Une route externe n'est annoncée par un routeur que s'il sait joindre le NEXT_HOP.
    - Une route dont l'attribut NEXT_HOP est l'adresse IP du voisin n'est pas annoncée à ce voisin (qui la connaît déjà!).

Le processus de décision : Critères de choix entre 2 routes

- WEIGHT (propriétaire Cisco, plus grand préféré)
- LOCAL_PREF le plus grand
- Route initiée par le processus BGP local
- AS_PATH minimum
- ORIGIN minimum (IGP -> EGP -> Incomplete)
- MULTI_EXT_DISC minimum
- Route externe préférée à une route interne (à l'AS)
- Route vers le plus proche voisin local (au sens de l'IGP)
- Route vers le routeur BGP de plus petite adresse IP

Différences entre E-BGP et I-BGP

- Une annonce reçue en I-BGP n'est pas réannoncée en I-BGP
- L'attribut LOCAL_PREF n'est annoncé qu'en I-BGP
- Seuls les voisins E-BGP doivent être directement connectés
- Les annonces I-BGP ne modifient pas l'AS_PATH
- Les annonces I-BGP ne modifient pas le NEXT_HOP
- Le MED n'est pas annoncé en I-BGP

Interaction BGP – IGP 6 Forwardind

- Redistribution
  - Routes apprises par BGP → IGP (OSPF)
    - Redistribution de BGP dans OSPF
    - OSPF propage les routes via des LSAs type 4 à tous les routeurs du nuage OSPF
- Injection
  - Route apprise par BGP sont écrites dans la table de routage du routeur.
    - Pas de propagation (cela n aide que le routeur)
- Synchronisation

Redistribution est onéreuse !

- Redistribution de BGP dans IGP
  - Grand nombre de routes /entrées dan IGP
  - Accroît le temps de convergence après panne
- Recursive table lookup
  - Next Hop n'est pas sur le lien

L'annonce des routes internes d'un AS

- Statique
  - Pas d'instabilité de routage, mais trous noirs possibles
    - `redistribute [static|connected]`
      - → ORIGIN: Incomplete
    - `network <adresse réseau>`
      - → ORIGIN: IGP
- Dynamique
  - Suit au mieux l'état du réseau, nécessite du filtrage
    - ***redistribute <paramètres de l'IGP>***
      - →ORIGIN: IGPz

La politique de routage

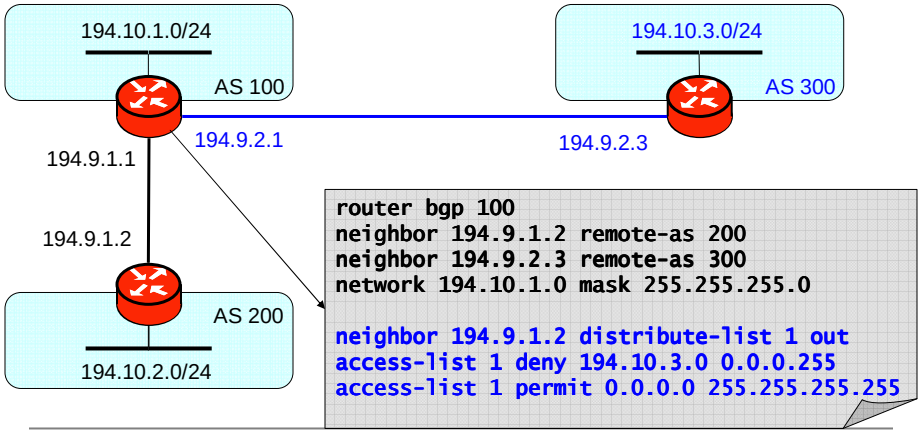
- Elle peut influencer :
  - Le traitement des routes reçues
  - Le traitement des routes annoncées
  - L'interaction avec les IGP de l'AS
- En pratique elle s'exprime par :
  - Du filtrage de réseaux
  - Du filtrage de routes (AS_PATH)
  - De la manipulation d'attributs de routes

Filtrage de routes

- Associer une *access list* à un voisin
- Neighbor *ID* distribute-list *no-of-the-list* [in/ou]
- Definir une access list
  - Bit non significatif (inverse du netmask)
  - Si pas d action en fin de liste
    - Aplpiquer le 'deny everything else'
- Access-list *No-of-the-list* [deny/permit] IP-@ *non-sig-bit*

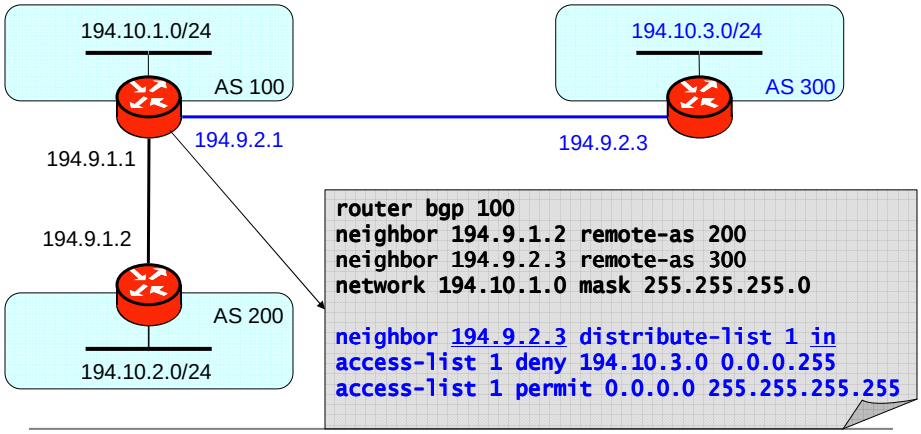
Politique de routage

- Filtrage des réseaux annoncés
  - AS100 ne veut pas servir d'AS de transit pour le réseau 194.10.3.0/24 de l'AS300



Politique de routage

- Filtrage des réseaux annoncés
  - Idem mais en plus l'AS100 ne connaît plus 194.10.3.0/24 de l'AS300



Filtrage de chemin

- Associer une *filtre list* à un voisin
- Neighbor *ID* filter-list *no-of-the-list* [in/ou]
- Définir une *filter list*
- Ip as-path access-list *No-of-the-list* [deny/permit] regexpr
- Regular expression
  - ^ début du chemin
  - $ fin du chemin
  - . Tout caractère
  - ? Un caractère

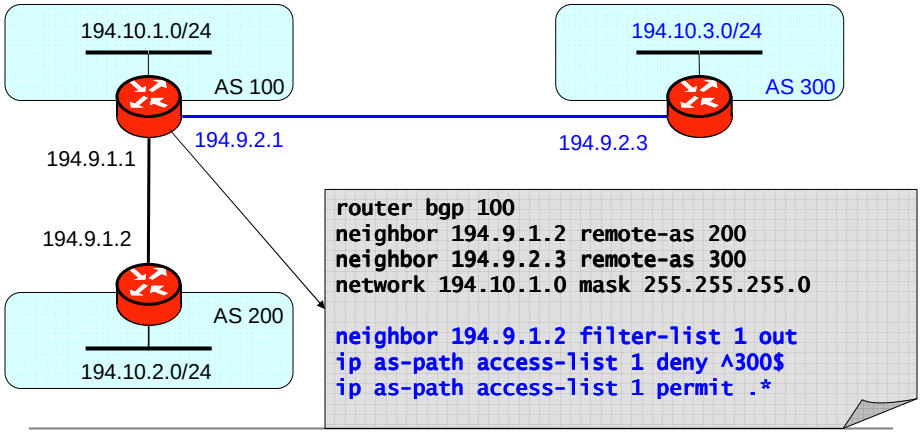
_ matches ^ $ ( ) 'space'
* toute répétition
+ au moins une répétition

Filtrage de chemins

- ^$
  - route locale seulement (AS_PATH vide)
- .*
  - toutes les routes
- ^300$
  - AS_PATH = 300
- ^300_
  - toutes les routes en provenance de 300
- _300$
  - Toutes les routes originaires de 300
- _300_
  - Toutes les routes passant par 300

Politique de routage

- Filtrage des AS_PATH annoncés
  - AS100 ne veut pas servir d'AS de transit pour tous les réseaux internes d'AS300



Route maps

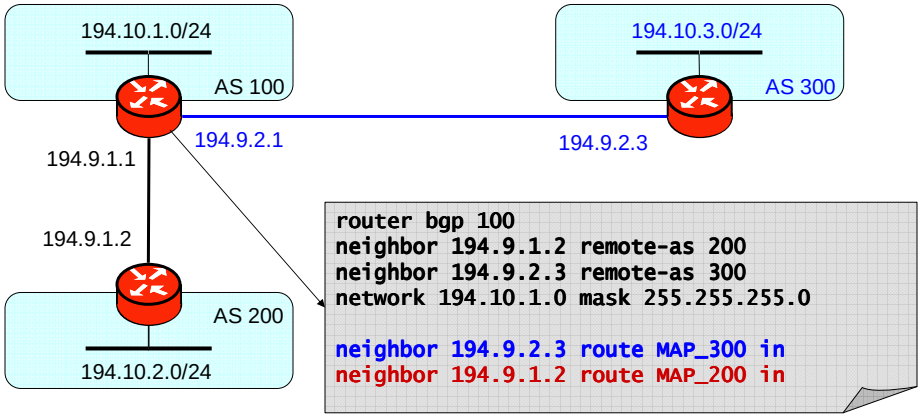
- Route-map *map-tag* [permit|deny] *instance-no*
- *First instance-condition*: set match
- *Next-instance-condition*:set match
- …
- Route-map SetMetric permit 10
- Match ip address 1
- Set metric 200

- Route-map SetMetric permit 20
- Set metric 300

- Access-list 1 permit 194.10.3.0 0.0.0.255

Politique de routage

- Filtrage par route map :
  - AS100 veut privilégier la route par défaut annoncée par AS300



Politique de routage (suite)

- Filtrage par route map :
  - AS100 veut privilégier la route par défaut annoncée par AS300