LE ROUTAGE IP

septembre 2006

Luc.Saccavini@inria.fr

Eléments sur le protocole IP

- Chaque «objet IP» est identifié par une/des adresse(s) 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**

En pratique la partie communicante de l'objet IP est désignée interface réseau (ou interface).

On associe souvent la notion de machine à celle d'adresse IP. Il faut bien noter que cette association n'est pas bijective car une machine peut avoir plusieurs interfaces réseau, et donc plusieurs adresses IP (cas des routeurs, de gros serveurs…). De plus, une interface réseau peut aussi avoir plusieurs adresses IP.

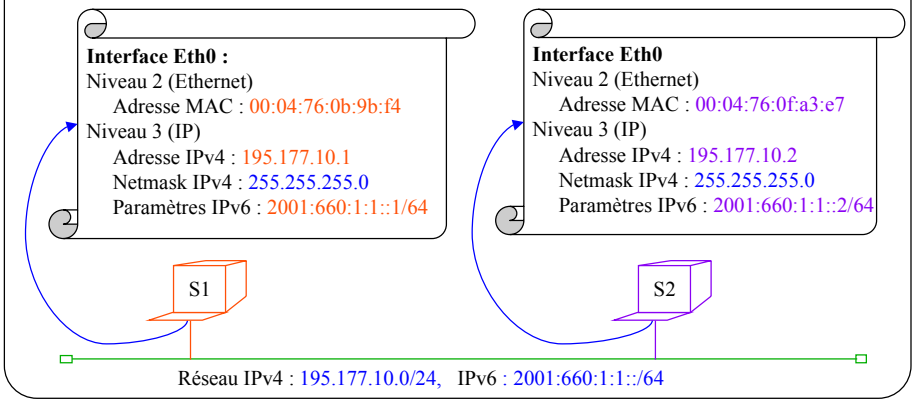
En IPv6, c'est d'ailleurs tours le cas, car une interface à au moins une adresse 'lien local' et une adresse 'globale'.

Eléments sur le protocole 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
- Il existe 2 versions du protocole IP
  - IPv4 : adressage 32bits (version actuelle)
  - IPv6 : adressage 128bits, auto-configuration…(nouvelle version)
- Le protocole IP est dit «best effort»
  - Chaque nœud du réseau fait de son mieux pour acheminer les paquets

Exemple 1 : IP sur Ethernet

❑ Paramètres des interfaces réseau eth0 de S1 et S2

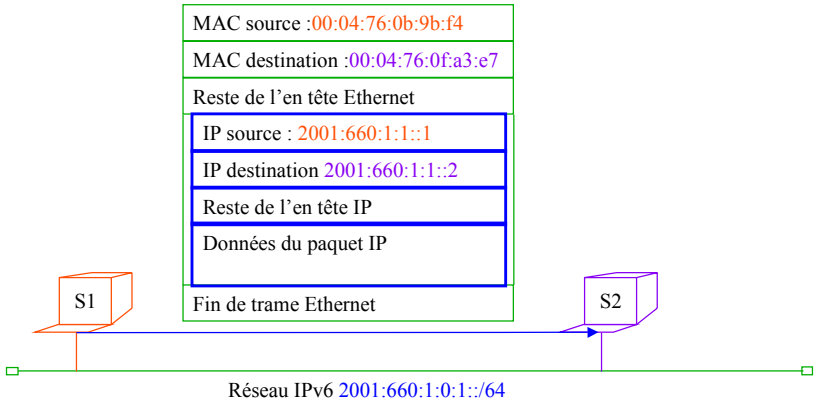


Sur une machine Linux, la configuration de l'interface eth0 se voit avec la commande ifconfig (ou avec la comande ' ip addr '), exemple pour la machine S1 :

```
S1>ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:76:0B:9B:F4
          inet addr:195.177.10.1  Bcast:195.177.10.255  Mask:255.255.255.0
          inet6 addr: 2001:660:1:1::1/64 Scope:Global
          inet6 addr: fe80::204:76ff:fe0b:9bf4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7442732 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5809921 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1301106660 (1240.8 Mb)  TX bytes:1767866154 (1685.9 Mb)
```

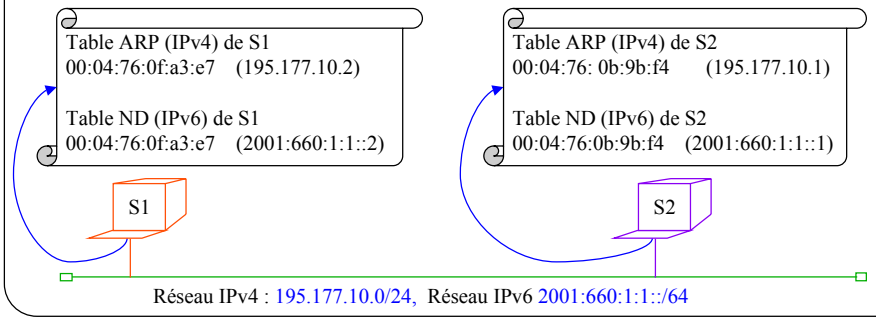
Exemple 1 : IP sur Ethernet

- Transmission d'un paquet IPv6 de S1 vers S2 dans une trame Ethernet



Exemple 1 : IP sur Ethernet

- Dans le cas d'Ethernet, un mécanisme spécifique : ARP (*Address Resolution Protocol*, RFC826) en IPv4 ou ND (*Neighbor Discovery*) en IPv6 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/ND de S1 et S2



Sur une machine Linux, la table arp (pour IPv4) peut se voir avec la commande arp, exemple pour la machine S1 (après le ping une nouvelle entrée est crée, elle correspondant à la machine s2) :

```
S1>arp -a
gw.domaine.fr (195.177.10.30) at 00:0E:D6:66:F3:80 [ether] on eth0

S1>ping -c 1 s2.domaine.fr
PING s2.domaine.fr (195.177.10.3) 56(84) bytes of data.
64 bytes from s2.domaine.fr (195.177.10.3): icmp_seq=1 ttl=64 time=3.35 ms
--- s2.domaine.fr ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.359/3.359/3.359/0.000 ms

S1>arp -a
gw.domaine.fr (195.177.10.30) at 00:0E:D6:66:F3:80 [ether] on eth0
s2.domaine.fr (195.177.10.3)  at 00:04:76:0F:A3:E7 [ether] on eth0
```

La même séquence en IPv6 avec la commande ndp :

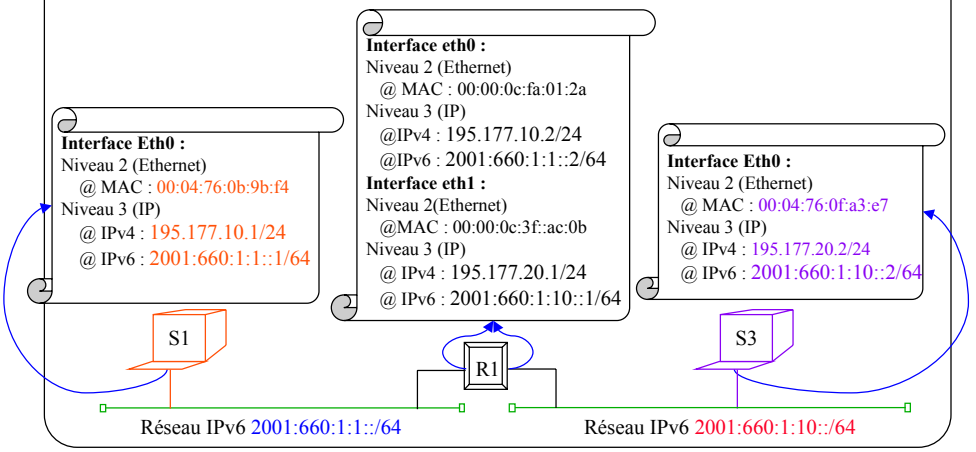
```
S1>ndp -an
Neighbor                        Linklayer Address  Netif Expire    St Flgs Prbs
2001:660:1:1::30                00:0e:d6:66:f3:80   eth0 unknown    S  R

S1>ping6 -c 1 s2.domaine.fr
PING s2(s2.domaine.fr) 56 data bytes
64 bytes from s2.domaine.fr: icmp_seq=1 ttl=64 time=0.209 ms

S1>ndp -an
Neighbor                        Linklayer Address  Netif Expire    St Flgs Prbs
2001:660:1:1::30                00:0e:d6:66:f3:80   eth0 unknown    S  R
2001:660:1:1::2                 00:04:76:0f:a3:e7   eth0 unknown    D
```

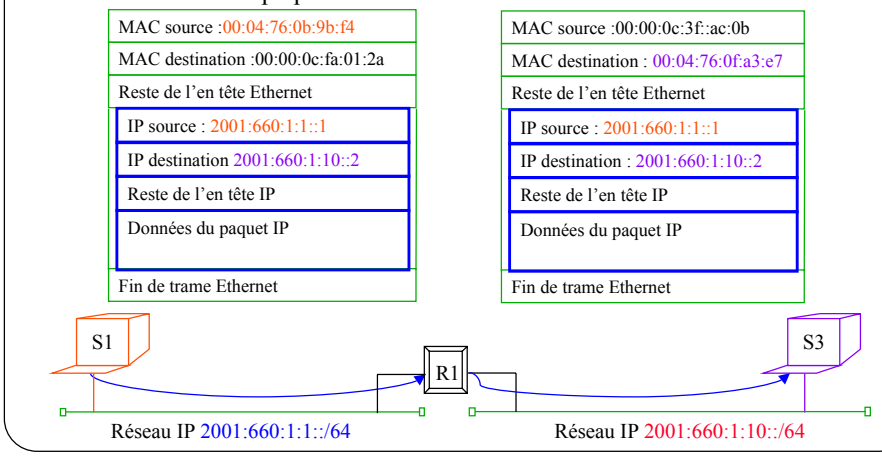
Exemple 2 : IP sur Ethernet (réseaux IP différents)

❑ Paramètres des interfaces réseau de S1, S3 et R1



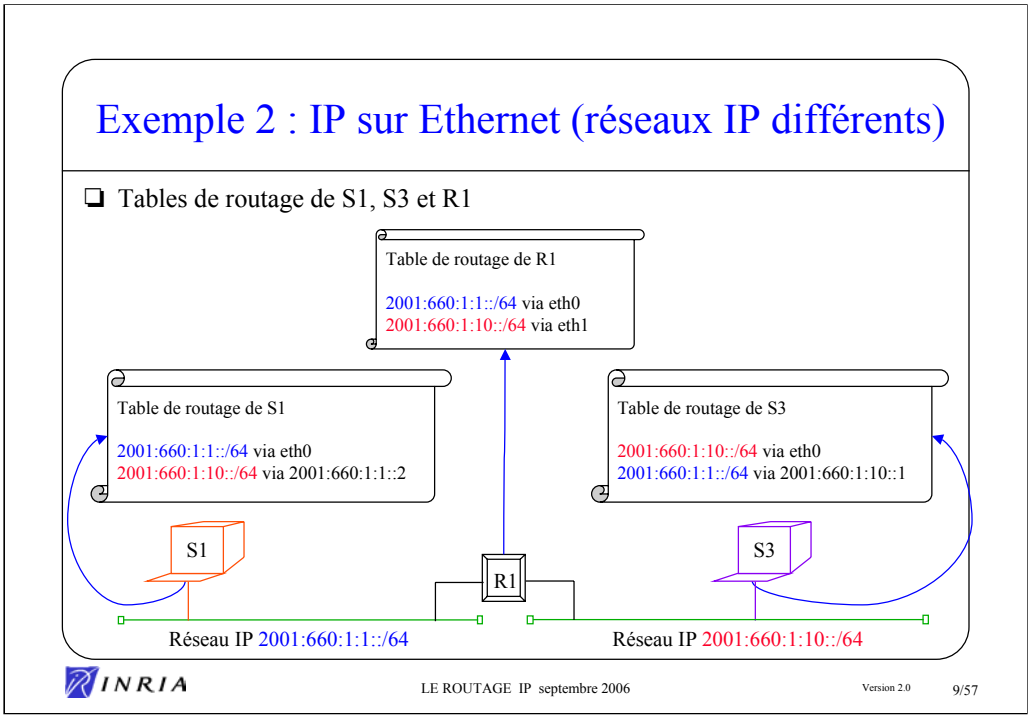
Exemple 2 : IP sur Ethernet (réseaux IP différents)

❑ Transmission d'un paquet IPv6 de S1 vers S3 via le routeur R1



Le paquet IP est bien sûr conservé à l'identique lors de la traversée du routeur R1.

Question : la phrase ci-dessus est-elle strictement vraie ? Pourquoi ?

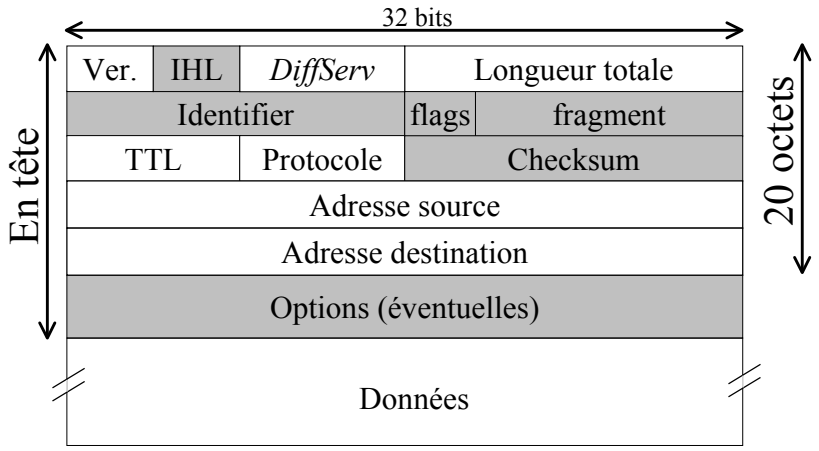


Sur une machine Linux, la table de routage peut se voir avec la commande netstat (ou avec la commande ‘ ip route ’), exemple pour la machine S1 (table de routage IPv4 puis IPv6):

```
S1>netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
195.177.10.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
195.177.20.0    195.177.10.2    255.255.255.0   UG        0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         195.177.10.2    0.0.0.0         UG        0 0          0 eth0
```

```
S1>netstat -rn -A inet6
Kernel IPv6 routing table
Destination                    Next Hop               Flags Metric Ref    Use Iface
::1/128                        ::                     U     0      20980    1 lo
2001:660:1:10::/64             2001:660:1:1::2        UG    1      0        0 eth0
2001:660:1:1::2/128            2001:660:1:1::2        UAC   0      4        1 eth0
2001:660:1:1::1/128            ::                     U     0      1956345  1 lo
2001:660:1:1::/64              ::                     UA    256    6242     0 eth0
fe80:: 204:76ff:fe0b:9bf4/128  ::                     U     0      1463     1 lo
fe80::/64                      ::                     U     256    0        0 eth0
ff02::1/128                    ff02::1                UC    0      1        0 eth0
ff00::/8                       ::                     U     256    0        0 eth0
::/0                           2001:660:1:1::2        UG    1024   935      0 eth0
```

Format du paquet IPv4



Ver. : numéro de version du protocole (4)

IHL : longueur de l'en-tête (en octets)

DiffServ : ancien champ TOS, classe de trafic

Longueur totale : longueur totale du datagramme (en octets, maximun 64ko)

Identifier :

Flags :

Fragment :

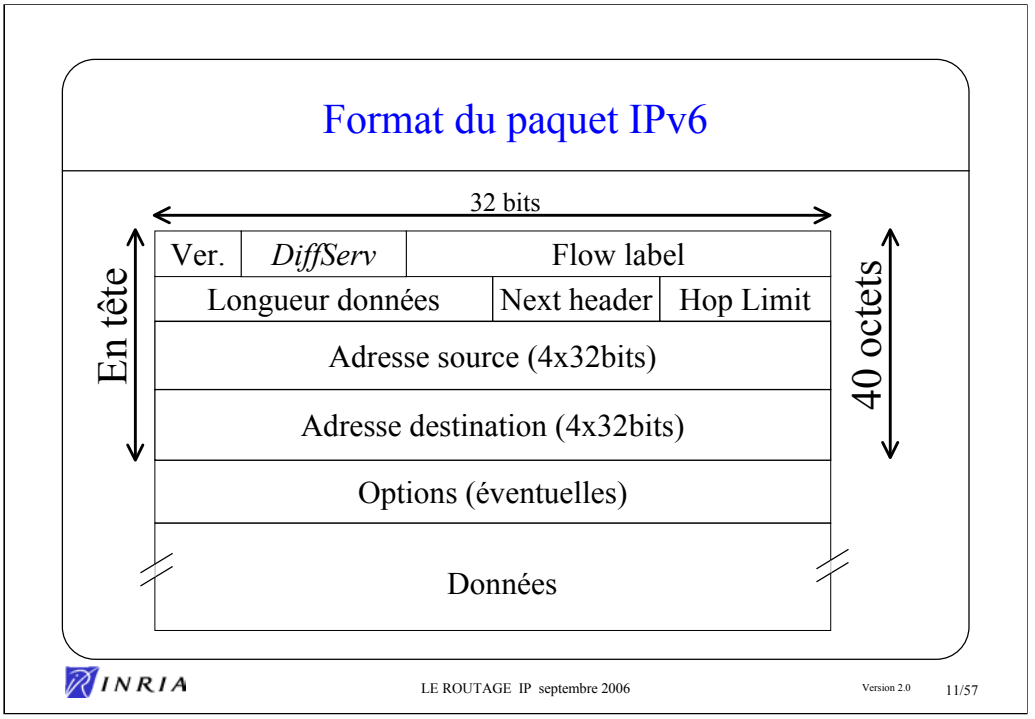
TTL : *Time To Live*, durée de vie du paquet (en nombre de routeurs traversés)

Protocol : TCP(6), UDP(17), ICMP(1), OSPF(89)

Checksum : clé de contrôle de l'en-tête

Adresse source : adresse source du paquet (4 octets)

Adresse destination : adresse destination du paquet (4 octets)



Ver. : numéro de version du protocole (6)

DiffServ : ancien champ TOS, classe de trafic

Flow label : identifiant de flux

Longueur données : longueur totale du datagramme (en octets), maximun 64ko sans option spécifique (la longueur des données du datagramme peut aller jusqu'à 4Go en utilisant l'option *jumbogram* de l'extension « proche-en-proche »

Next hedear : valeur de l'en tête suivant; peut être un protocole de niveau supérieur (TCP, UDP, ICMP…) ou la désignation de l'option suivante (« proche-en-proche », « routing », « fragment »….)

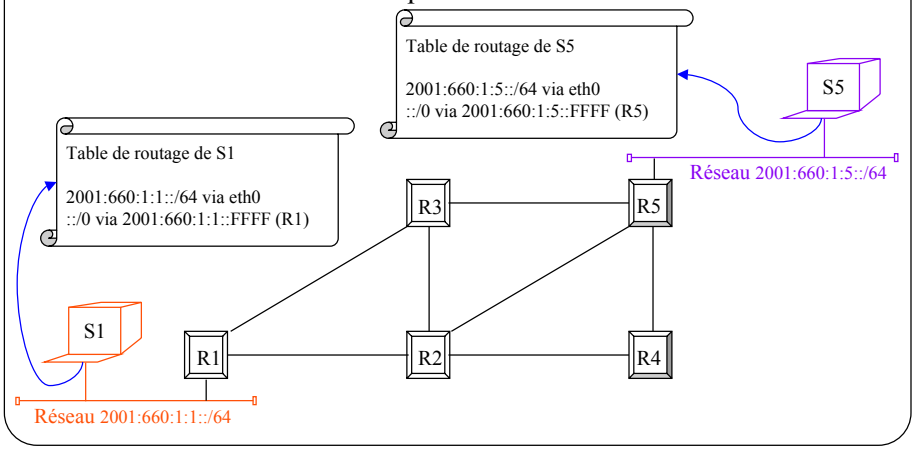
Hop Limit : durée de vie du paquet (en nombre de routeurs traversés)

Adresse source : adresse source du paquet (16 octets)

Adresse destination : adresse destination du paquet (16 octets)

Exemple 3: routage par défaut

❑ La station S1 veut communiquer avec la station S5



Exercice :

- lister l'ensemble des routes de S1 vers S2
- lister l'ensemble des routes de S2 vers S1

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)
- Routage statique
  - Les tables de routage sont positionnées manuellement
- Routage dynamique
  - Des protocoles propagent automatiquement les tables ou informations de routage entre les routeurs

Le routage IP 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

Le routage IP 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 dynamiques

- Il existe 2 grandes familles de protocoles de routage
  - 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

Rappel sommaire sur les types de protocoles de routage :

- distance vecteur : la distance est le nombre de routeurs pour joindre une destination, chaque routeur ne connaît que son voisinage et propage les routes qu'il connaît à ses voisins (ex. RIP).

- états des liens : chaque routeur connaît la topologie et l'état de l'ensemble des liens du réseau, puis en déduit les chemins optimaux. À chaque interaction les routeurs s'envoient toute leur table de routage (ex. OSPF).

Le protocole BGP peut être considéré comme à mi-chemin entre les deux types de protocoles précédents. En effet, l'échange de chemins d'AS permet à chaque routeur de reconstruire une grande partie de la topologie du réseau, ce qui est caractéristique des protocoles de type «état des liens», mais deux routeurs voisins n'échangent que les routes qu'ils connaissent, ce qui est caractéristique d'un protocole de type «distance-vecteur».

Références sur les autres protocoles de routage :

	IPv4	IPv6
RIP (Routing Information Protocol)	RFC 2453, 11/98 «RIPv2»	RFC2080, 01/97 «RIPng»
IGRP	voir manuel IOS de Cisco	
EIGRP	voir manuel IOS de Cisco	
OSPF (Open Shortest Path First)	RFC 2328, 04/98 «OSPv2»	RFC 2740, 12/99 «OSPv3»
IS-IS (Intermediate System to Intermediate System ISO/IEC 10589, (ou RFC1142, 02/90)		
EGP (Exterior Gateway Protoco)	RFC 904 04/84	--------
IDRP (Inter Domain Routing Protocol)	ISO/IEC IS10747 10/93	
BGP (Border Gateway Protocol)	RFC 4271, 01/06 «BGP4»	RFC 2545, 03/99 «BGP4+»

Le protocole RIP (v2 et ng) : présentation

Historique de RIP (Routing Internet Protocol)

- Étapes de la standardisation IETF
  - 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

La métrique M d'une route qui est sa distance qui sépare les deux réseaux cibles (origine et destination) exprimée comme le nombre de routeurs qui séparent ces deux réseaux.

La métrique associée à une route d'interface d'un routeur (pour un réseau directement connecté à cette interface) sera donc égale à 1.

La métrique maximale possible est de 15, au delà le réseau est considéré comme inaccessible.

L'administrateur a la possibilité de fixer une valeur métrique différente de la métrique par défaut (1) pour certains réseaux. Cette valeur doit être un entier strictement supérieur à 1. Dans ce cas, la diamètre maximum du réseaux exprimé en nombre de routeurs en sera réduit d'autant.

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.

Exemple 4 : fonctionnement de RIP

- Évolution de la table de routage de R1 : étape 1
  - Annonce de (2001:660:1:1::/64, d=0) à R3 et R2
  - Réception de R2 : (2001:660:1:2::/64, d=0)
  - Réception de R3 : (2001:660:1:3::/64, d=0)

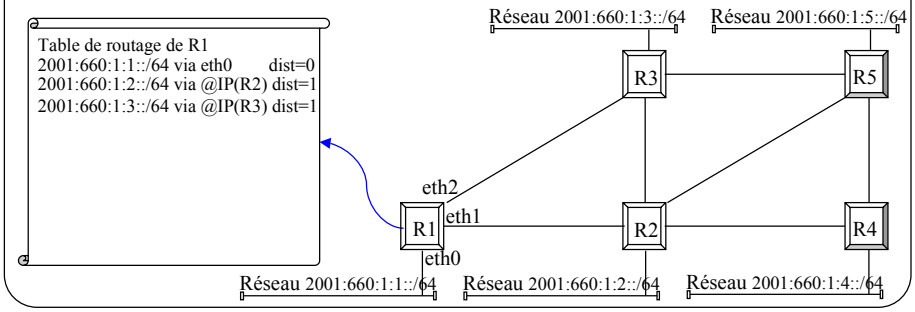


Table de routage de R2 : étape 1
2001:660:1:2::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:3::/64 via R3 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:4::/64 via R4 dist=1

Table de routage de R3 : étape 1
2001:660:1:3::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1

Table de routage de R4 : étape 1
2001:660:1:4::/64 via eth0
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1

Table de routage de R5 : étape 1
2001:660:1:5::/64 via eth0
2001:660:1:3::/64 via R3 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:4::/64 via R4 dist=1

Exemple 4 : fonctionnement de RIP

- Évolution de la table de routage de R1 : étape 2
  - Annonce de (2001:660:1:2::/64, d=1) à R3 et (2001:660:1:3::/64, d=1) à R2
  - Réception de R2 : (2001:660:1:3::/64, 2001:660:1:5::/64, 2001:660:1:4::/64, d=1)
  - Réception de R3 : (2001:660:1:2::/64, 2001:660:1:5::/64, d=1)

Table de routage de R1

2001:660:1:1::/64 via eth0 dist=0
2001:660:1:3::/64 via @IP(R3) dist=1
2001:660:1:2::/64 via @IP(R2) dist=1
2001:660:1:3::/64 via @IP(R2) dist=2 **X**
2001:660:1:5::/64 via @IP(R2) dist=2
2001:660:1:4::/64 via @IP(R2) dist=2
2001:660:1:2::/64 via @IP(R3) dist=2 **X**
2001:660:1:5::/64 via @IP(R3) dist=2 **X**
(**X** routes éliminées)

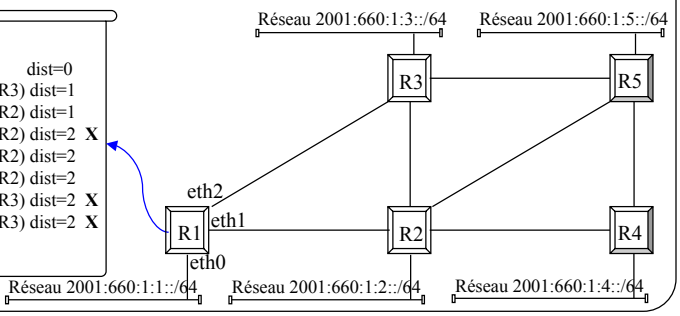


Table de routage de R2 : étape 2

2001:660:1:2::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:3::/64 via R3 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:4::/64 via R4 dist=1
2001:660:1:3::/64 via R1 dist=2 **X**
2001:660:1:1::/64 via R3 dist=2 **X**
2001:660:1:5::/64 via R3 dist=2 **X**
2001:660:1:5::/64 via R4 dist=2 **X**
2001:660:1:3::/64 via R5 dist=2 **X**
2001:660:1:4::/64 via R5 dist=2 **X**

Table de routage de R4 : étape 2

2001:660:1:4::/64 via eth0
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:1::/64 via R2 dist=2
2001:660:1:3::/64 via R2 dist=2
2001:660:1:5::/64 via R2 dist=2 **X**
2001:660:1:3::/64 via R5 dist=2 **X**
2001:660:1:2::/64 via R5 dist=2 **X**

Table de routage de R3 : étape 2

2001:660:1:3::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:2::/64 via R1 dist=2 **X**
2001:660:1:1::/64 via R2 dist=2 **X**
2001:660:1:5::/64 via R2 dist=2 **X**
2001:660:1:4::/64 via R2 dist=2
2001:660:1:2::/64 via R5 dist=2 **X**

Table de routage de R5 : étape 2

2001:660:1:5::/64 via eth0
2001:660:1:3::/64 via R3 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:4::/64 via R4 dist=1
2001:660:1:1::/64 via R3 dist=2
2001:660:1:2::/64 via R3 dist=2 **X**
2001:660:1:1::/64 via R2 dist=2 **X**
2001:660:1:3::/64 via R2 dist=2 **X**
2001:660:1:4::/64 via R2 dist=2 **X**
2001:660:1:2::/64 via R4 dist=2 **X**

Exemple 4 : fonctionnement de RIP

- Évolution de la table de routage de R1 : étape 3
  - Annonce de (2001:660:1:5::/64, 2001:660:1:4::/64, d=2) à R3
  - Réception de R2 : rien de nouveau
  - Réception de R3 : (2001:660:1:4::/64, d=2)

Table de routage de R1

2001:660:1:1::/64 via eth0 dist=0
2001:660:1:3::/64 via @IP(R3) dist=1
2001:660:1:2::/64 via @IP(R2) dist=1
2001:660:1:5::/64 via @IP(R2) dist=2
2001:660:1:4::/64 via @IP(R2) dist=2
2001:660:1:4::/64 via @IP(R3) dist=3 **X**
(**X** routes éliminées)

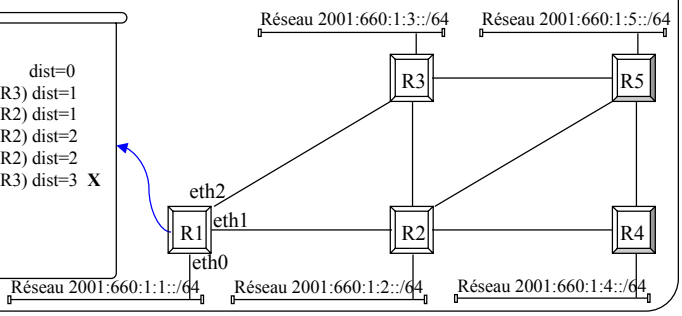


Table de routage de R2 : étape 3
2001:660:1:2::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:3::/64 via R3 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:4::/64 via R4 dist=1
2001:660:1:4::/64 via R3 dist=3 **X**
2001:660:1:1::/64 via R5 dist=3 **X**

Table de routage de R4 : étape 3
2001:660:1:4::/64 via eth0
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:1::/64 via R2 dist=2
2001:660:1:3::/64 via R2 dist=2

Table de routage de R3 : étape 3
2001:660:1:3::/64 via eth0
2001:660:1:1::/64 via R1 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:5::/64 via R5 dist=1
2001:660:1:4::/64 via R2 dist=2

Table de routage de R5 : étape 3
2001:660:1:5::/64 via eth0
2001:660:1:3::/64 via R3 dist=1
2001:660:1:2::/64 via R2 dist=1
2001:660:1:4::/64 via R4 dist=1
2001:660:1:1::/64 via R3 dist=2

Fonctionnement de RIP (compléments)

- 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-annoncées 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)

Les 3 minuteurs de RIP sont :

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).

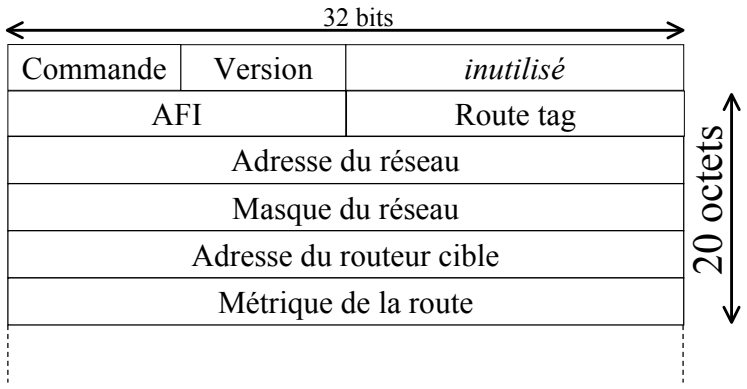
Si pendant ce temps une nouvelle route vers ce préfixe est apprise, elle remplace la route invalide.

route-flush : périodicité de nettoyage de la table des informations RIP. Les routes marquées comme invalides sont effacées.

Remarque :

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. (voir : Floyd, S., et V. Jacobson, The synchronisation of Periodic Routing Messages, ACM Sigcom '93 symposium, septembre 1993)

Format du paquet RIP



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)

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}.

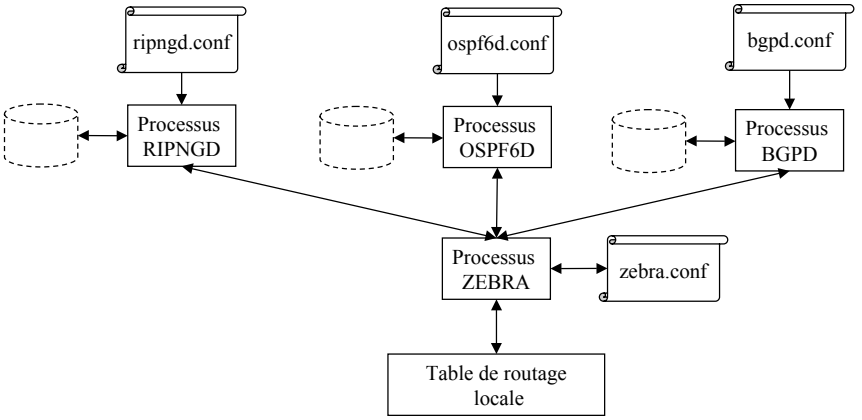
Les paquets de type réponse peuvent contenir jusqu'à 25 routes par paquet. S'il y a plus de 25 routes à envoyer, plusieurs paquets sont émis.

Le protocole RIP : exemple de mise en œuvre

Les routeurs IP

- Routeur IP = équipement capable de transférer des paquets IP d'un réseau à un autre
- Deux solutions possibles :
  - Équipement dédié unique
    - Matériel : Cisco, Foundry, Juniper, 3Com….
    - Logiciel associé : IOS, Foundry OS, JunOS, ….
    - Configuration par : telnet, snmp, http
  - Routeur construit à partir de
    - Matériel : Plateforme x86 sous Unix (BSD, Linux….)
    - Logiciel de routage : Quagga (ex. zebra) (www.quagga.net) ou Xorp (www.xorp.org)

Organisation logique d'un routeur zebra



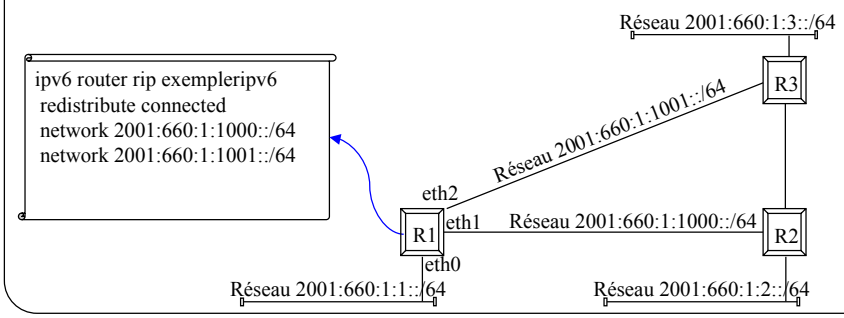
Les processus du logiciel Zebra sont configurables par telnet sur les ports :

- 2601 pour zebra
- 2602 pour ripd
- 2603 pour ripngd (IPv6)
- 2604 pour ospfd
- 2605 pour bgpd
- 2606 pour ospf6d (IPv6)

Les fichiers de configuration (ripd.conf, ospfd.conf, bgpd.conf, zebra.conf) sont installés dans /etc/zebra/

Exemple de configuration de RIP : IOS ou Zebra

❑ Configuration minimale d'un processus RIP



Fichier de configuration de R1 sur IOS/Cisco

```
ip subnet-zero
!
interface Ethernet0
 ipv6 enable
 ipv6 address 2001:660:1:1::1/64
 no ip directed-broadcast
!
interface Ethernet1
 ipv6 enable
 ipv6 address 2001:660:1:1000::1/64
 ipv6 rip exempleripv6 enable
 no ip directed-broadcast
!
interface Ethernet2
 ipv6 enable
 ipv6 address 2001:660:1:1001::1/64
 ipv6 rip exempleripv6 enable
 no ip directed-broadcast
!
ipv6 unicast-routing
ipv6 router rip exempleripv6
 version 2
 redistribute connected
 network 2001:660:1:1000::1/64
 network 2001:660:1:1001::1/64
 neighbor 2001:660:1:1000::2
 neighbor 2001:660:1:1001::2
!
ip classless
```

Fichier de configuration de R1 (zebra.conf, Zebra)

```
interface Ethernet0
 ipv6 enable
 ipv6 address 2001:660:1:1::1/64
 no ip directed-broadcast
!
interface Ethernet1
ipv6 enable
ipv6 address 2001:660:1:1000::1/64
 no ip directed-broadcast
!
interface Ethernet2
ipv6 enable
ipv6 address 2001:660:1:1001::1/64
 no ip directed-broadcast
!
```

Fichier de configuration de R1 (ripngd.conf, Zebra)

```
enable ipv6 forwarding
ipv6 router rip exempleripv6
 redistribute connected
 network 2001:660:1:1000::/64
 network 2001:660:1:1001::/64
 neighbor 2001:660:1:1000::2
 neighbor 2001:660:1:1001::2
 route 2001:660:1:1::/64
```

Aspects sécurité

- Lors de la configuration des routeurs on veut se prémunir
  - De routeurs intrus (identification de ses voisins)
  - D'informations fausses ou erronées (filtrage d'annonces)
    - Mauvaise route par défaut
    - Erreurs de paramétrage

RIP : Authentification MD5 (1)

- Configuration zebra ou Cisco (IPv4 seulement)

- key chain essai
- key 1
- key-string truc
- !
- interface eth0
- ip rip authentication mode md5
- ip rip authentication key-chain essai

- *truc* est le secret partagé entre les routeurs RIP

RIP : Authentification MD5 (2)

- Possibilité de gérer plusieurs clés
- Intervalle de validité d'une clé (émission et réception)

- key chain essai
- key 1
- key-string truc
- send-lifetime 00:00:00 Jan 01 2004 duration 3600000
- key 2
- key-string bidule
- accept-lifetime 10:00:00 Jan 01 2004 infinite

Filtrage (1)

Un exemple (zebra) de filtrage en entrée (in)

```
router rip
 distribute-list filtre in eth0
!
access-list filtre permit 10.0.0.0/8
access-list filtre deny any
```

Ici, on ne retiendra que les annonces de routes (acquises via l'interface eth0) dont le préfixe est inclus dans le préfixe 10.0.0.0/8

Filtrage (2)

Un exemple (zebra) de filtrage en sortie (out)

```
router rip
 distribute-list filtre out eth0
!
access-list filtre deny 192.168.0.0/16
access-list filtre permit any
```

Ici, on annoncera (sur l'interface eth0) toutes les routes sauf celles dont le préfixe est inclus dans le préfixe 192.168.0.0/16

Autres commandes utiles

router rip

neighbor 192.168.0.2

La directive neighbor permet de spécifier un routeur RIP voisin ne comprenant pas le multicast et donc ne répondant pas à l'adresse multicast 224.0.0.9. Un lien direct sera alors établi entre les 2 routeurs

router rip

passive-interface eth1

Tous les paquets RIP reçus sur l'interface eth1 sont traités normalement. En revanche, aucun paquet RIP unicast ou multicast n'est envoyé sur cette interface, excepté ceux à destination de voisins (directive neighbor)

Différences entre RIPv2 et RIPng

- RIPng est une adaptation de RIPv2 pour IPv6, il n'y a pas eu d'évolutions conceptuelles mais juste les modifications suivantes
  - Usage d'un nouveau numéro de port udp : 521 (au lieu de 520)
  - Modification du format du paquet d'annonce de routes
  - Suppression du mécanisme d'authentification spécifique, *l'authentification repose sur IPSec (pas d'authentification MD5 comme dans RIP)*
  - Limitation du nb. de routes annoncées par le MTU uniquement
  - Possibilité d'annoncer spécifiquement un *next hop*

Le protocole OSPF : présentation

Historique d'OSPF (Open Shortest Path First)

- Étapes de la standardisation IETF
  - OSPFv1 : RFC1131 (10/89) puis RFC1247 (07/91)
  - OSPFv2 : RFC 2328 (04/98), RFC3630 (06/03) [extensions TE]
  - OSPFv3 :
    - RFC 2740 (12/99) : adaptation pour IPv6
    - RFC 3101 (01/03) : aires NSSA
    - RFC 4552 (06/06) : confidentialité et authentification des échanges

Pourquoi OSPF ?

- OSPF a été conçu pour s'affranchir des limitations de RIP
  - Possibilité de gérer des domaines de diamètre > 16 routeurs
  - Amélioration du temps de convergence (changement de l'algorithme utilisé)
  - 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)

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 acquiers 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 routage 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

Calcul des meilleures routes : état des liens

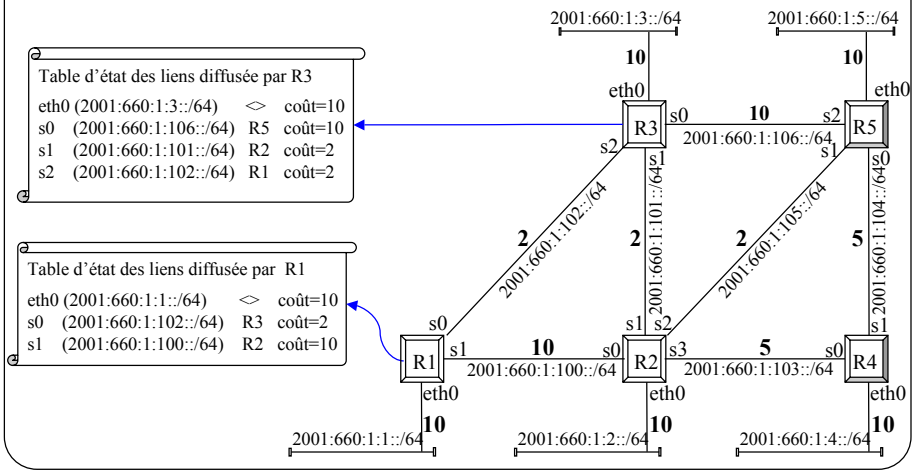


Table d'état des liens diffusée par R2

eth0 (2001:660:1:2::/64) <> coût=10
s0 (2001:660:1:100::/64) R1 coût=10
s1 (2001:660:1:101::/64) R3 coût=2
s2 (2001:660:1:105::/64) R5 coût=2
s3 (2001:660:1:103::/64) R4 coût=5

Table d'état des liens diffusée par R4

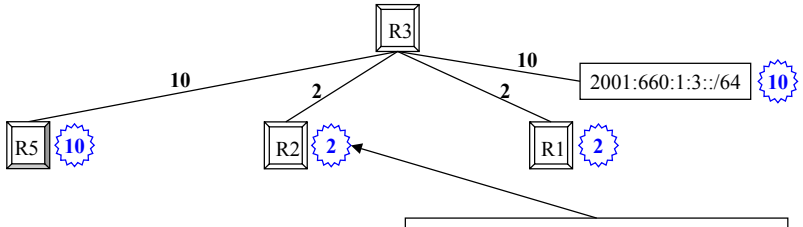
eth0 (2001:660:1:4::/64) <> coût=10
s0 (2001:660:1:103::/64) R2 coût=5
s1 (2001:660:1:104::/64) R5 coût=5

Table d'état des liens diffusée par R5

eth0 (2001:660:1:5::/64) <> coût=10
s0 (2001:660:1:104::/64) R4 coût=5
s1 (2001:660:1:105::/64) R2 coût=2
s2 (2001:660:1:106::/64) R3 coût=10

Calcul des meilleures routes : arbre OSPF

❑ Calcul de l'arbre des coûts minimums depuis R3 : étape 1



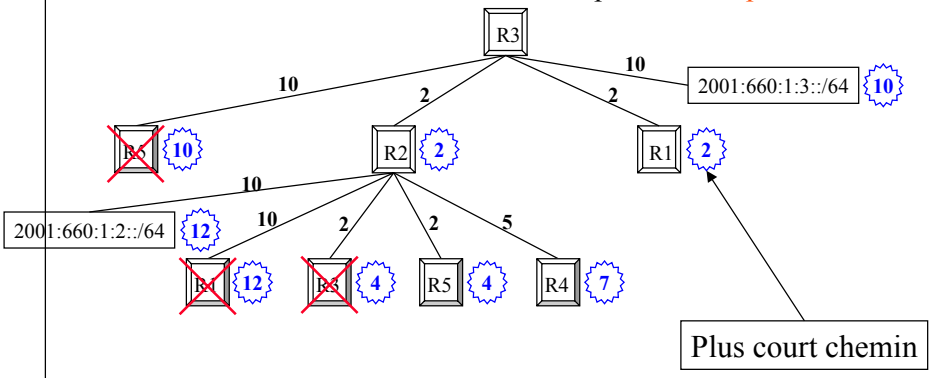
M Valeur de la métrique pour venir de R3

C'est le plus court chemin, il est donc examiné en premier

Pour simplifier le schéma, les réseaux d'interconnexion entre les routeurs ne sont pas figurés (valeur de métrique identique à celle des routeurs correspondants)

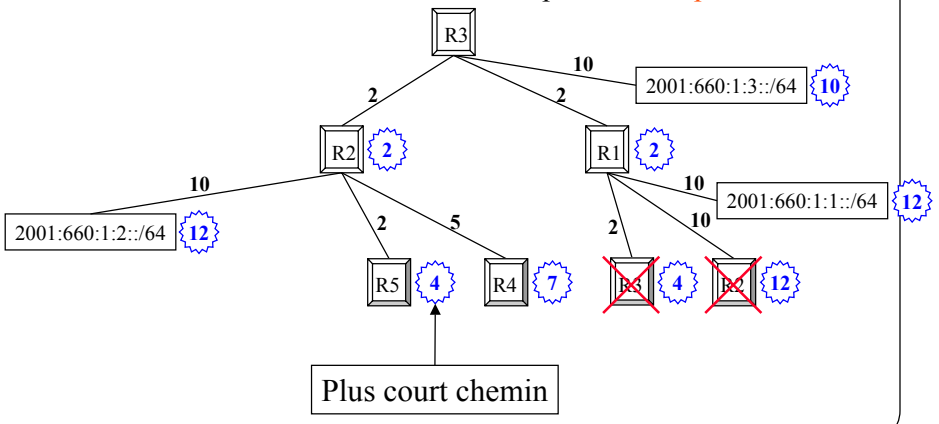
Calcul des meilleures routes : arbre OSPF

❑ Calcul de l'arbre des coûts minimums depuis R3 : étape 2



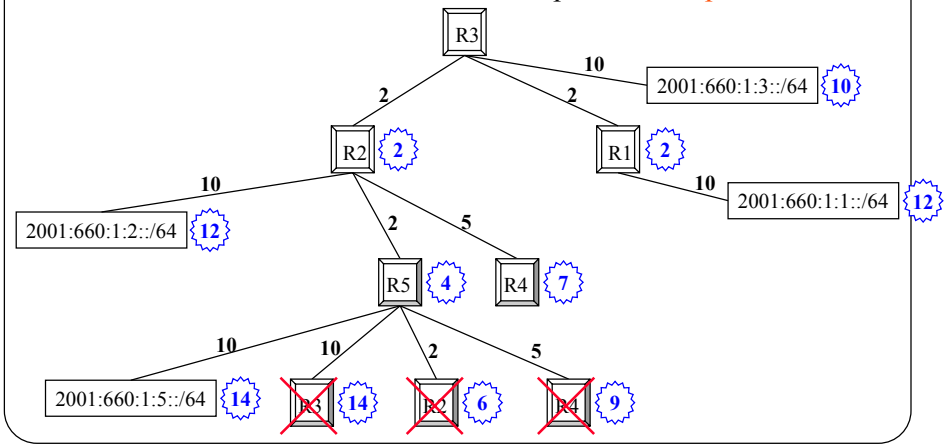
Calcul des meilleures routes : arbre OSPF

❑ Calcul de l'arbre des coûts minimums depuis R3 : étape 3



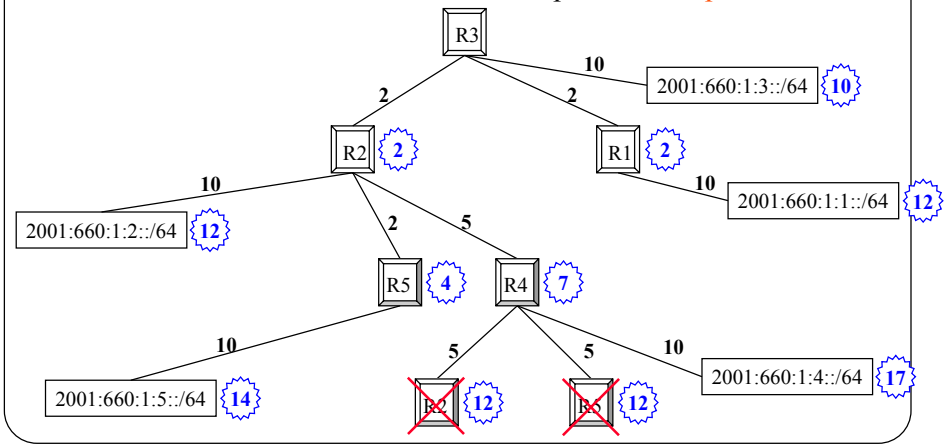
Calcul des meilleures routes : arbre OSPF

❑ Calcul de l'arbre des coûts minimums depuis R3 : étape 4



Calcul des meilleures routes : arbre OSPF

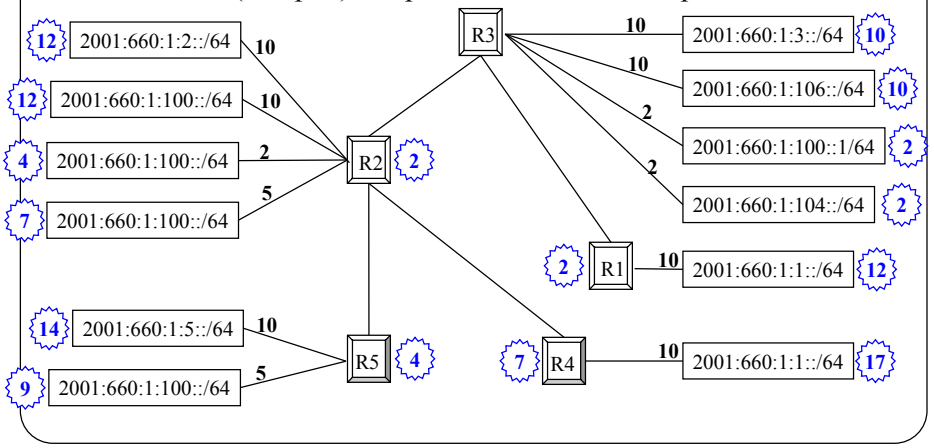
❑ Calcul de l'arbre des coûts minimums depuis R3 : étape 5



Exercice : tracer l'arbre des coût minimums ayant pour origine le routeur R1

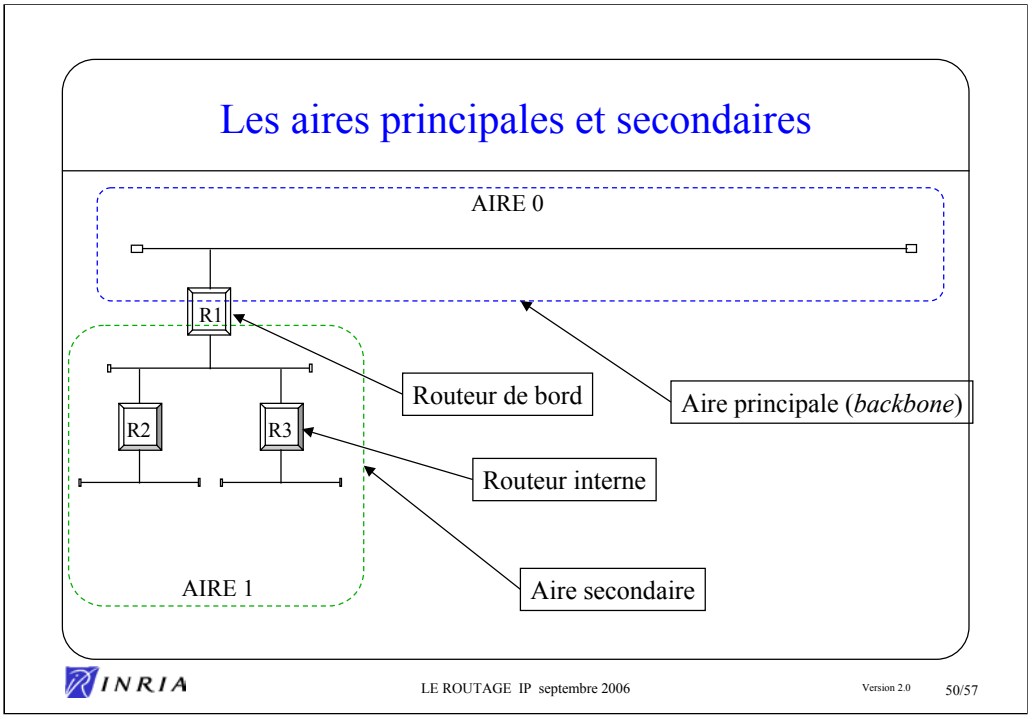
Calcul des meilleures routes : arbre OSPF

❑ D'où l'arbre (complet) des plus courts chemins depuis R3



Optimisation d'OSPF : les aires

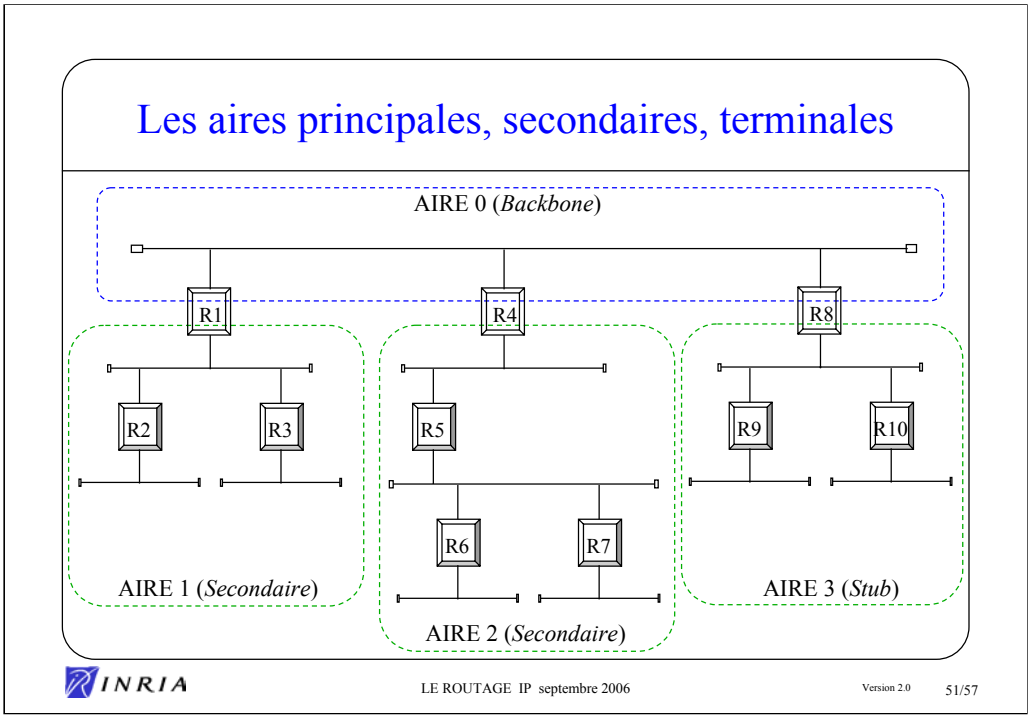
- Dans grand domaine, chaque changement provoque une diffusion de la table de l'état des liens de tous les routeurs, ce qui provoque
  - Une consommation de bande passante importante
  - Une charge CPU importante sur les routeurs
  - Alors que la portée d'une modification reste assez localisée
- D'où 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)



C'est le fait de lui attribuer le numéro 0 qui rend une aire principale

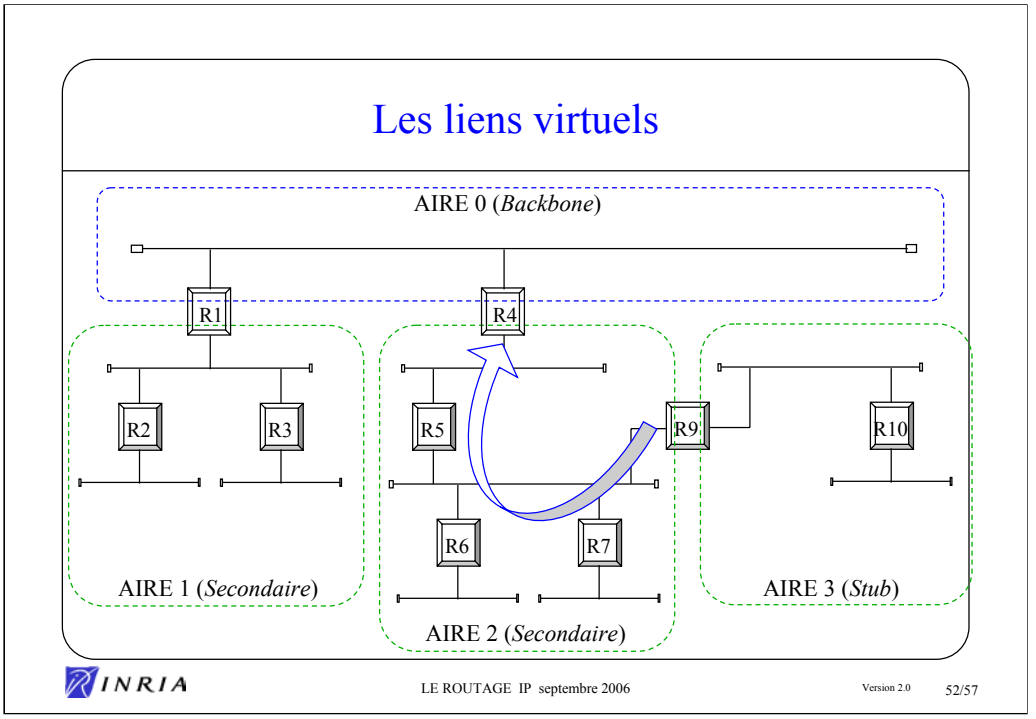
Il y a toujours une (et une seule) aire principale par domaine OSPF. Si on ne veut pas découper le domaine, tous les interfaces de tous les routeurs sont dans l'aire 0.

Les autres aires sont de type *Secondaire*, elles sont numérotables de 1 à 4294967295 (2**31-1).



On notera que ce sont les interfaces des routeurs (les liens) qui sont positionnés dans les différentes aires.

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 les routeurs en leur diffusant une route par défaut.

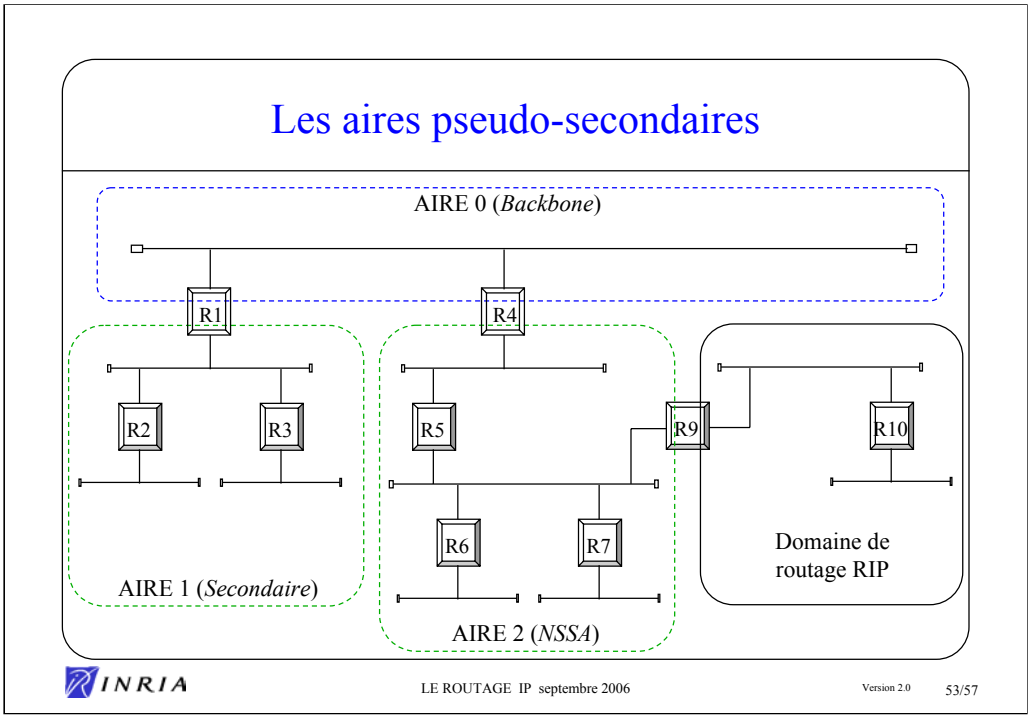


Les liens virtuels sont utilisés dans les deux cas suivants :

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 quant cette dernière n'est plus assurée (il s'agit bien sûr d'une solution de dépannage !).

On notera que les liens virtuels font dépendre 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.



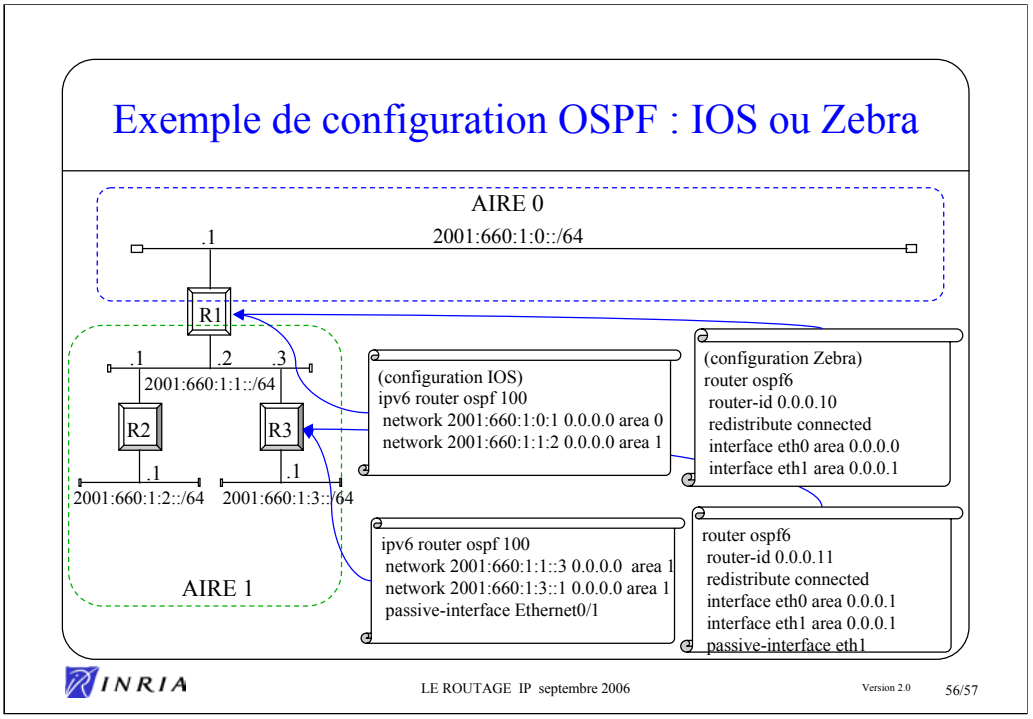
L'aire 2 est de type NSSA (Not So Stuby Area). C'est une aire terminale 'améliorée' pour permettre l'injection des routes que le routeur R9 va apprendre du domaine de routage RIP.

Fonctionnement d'OSPF (compléments)

- Fonctionne au dessus d'IP (protocole N° 89)
- Utilise les adresses multicast
  - 224.0.0.5 (pour tous les routeurs du lien)
  - 224.0.0.6 (pour le routeur désigné principal et de secours)
- Le nombre de routeurs n'est pas limité
- Valeur par défaut de la métrique : 10**8/(débit nominal du lien)
- Authentification possible par
  - Mot de passe simple
  - Condensat (MD5) du message (le mot de passe de circule pas)

La meilleure convergence d'OSPF par rapport à RIP est liée à l'algorithme utilisé. En effet, pour un nombre de liaisons L et un nombre de routeurs R, l'algorithme de Bellman-Ford (RIP) a une complexité qui est en O(LxR) alors que l'algorithme de Dijkstra a une complexité en O(Lxlog(L)).

Le protocole OSPF : exemple de mise en œuvre



C’est la directive *network* qui permet de positionner une interface dans une aire donnée.

Bibliographie (sommaire)

- OSPF : Anatomy of an Internet Routing Protocol, John T. Moy, Addison Wesley, février 1998, ISBN 0201634724
- OSPF Complete Implementation, John T. Moy, Addison Wesley, 2001, ISBN 0-201-30966-1
- IP Routing Protocols: RIP, OSPF, BGP, PNNI and Cisco Routing Protocols, Uyless Black, Prentrice Hall PTR, 2000, ISBN 0-13-014248-4
- RIP An Intra-Domain Routing Protocol, Gary Scott Malkin, Addison Wesley, décembre 1999
- IP Routing, Ravi Malhotra, O'Reilly, janvier 2002, ISBN 0-596-00275-0