OSPF (Open Shortest Path First)

Ecrivez un compte rendu des résultats de votre expérimentation, et répondez aux questions.

http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_ospf/netkit-labs_ospf.pdf

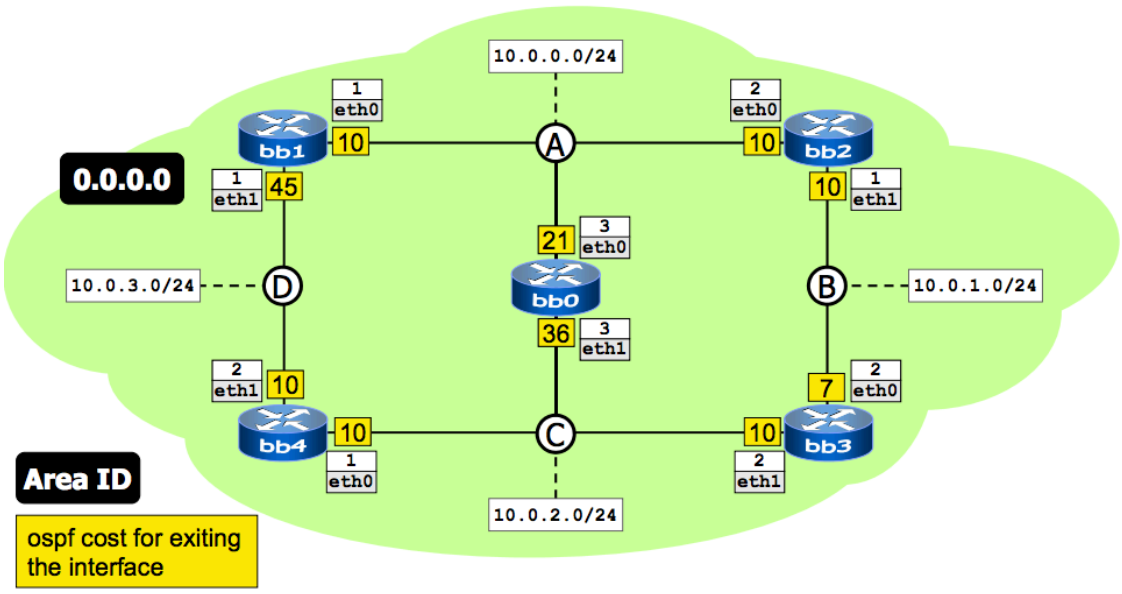
http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_ospf/netkit-labs_ospf.tar.gz

OSPF est un *interior gateway protocol* (tout comme RIP, IS-IS)
- version 2 --> specification: RFC 2328 authentication confidentiality : RFC 5709
- version 3 (with IPv6 support) --> spécification: RFC 5340 authentication confidentiality : RFC 4552

OSPF: *overview*

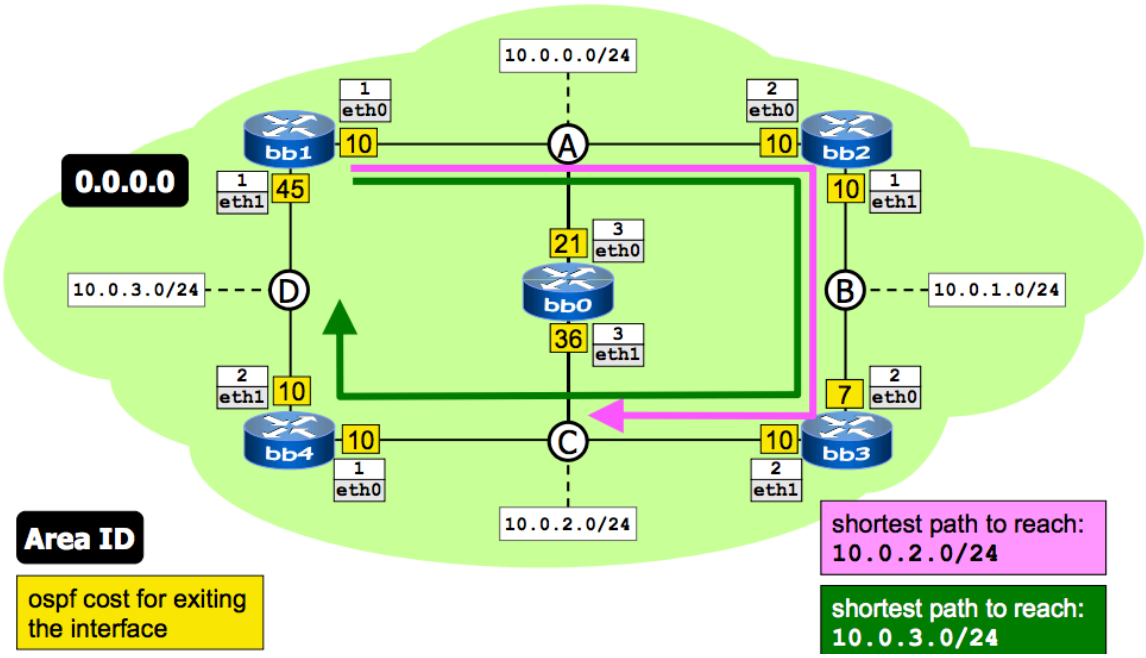
Chaque routeur envoie en "flooding" l'état local ("local state", c'est à dire ses interfaces utilisables, ses voisins joignables) à travers le réseau, utilisant le protocole **"link state advertisement (lsa)"**
- en se basant sur cette information reçue, chaque routeur construit et maintient à jour un **"link state database (lsdb)",** qui décrit la topologie du réseau
-- identique pour chaque routeur
-- chaque "entry" de la base (*lsdb*) représente un état local d'un routeur
- chaque router utilise le *lsdb* pour calculer l'arbre des plus courts chemins (*shortest path tree*) ayant lui-même comme racine
- chaque interface peut avoir un *cout* (qui peut être changé par l'administrateur du réseau)

OSPF lab*, single-area* : topologie du réseau



single (backbone) area (0.0.0.0)
- Chaque interface a un cout OSPF
    - default : 10
    - les couts ont été choisis pour forcer des plus courts chemins
- Pour changer le cout d'une interface, utilisez la commande suivante (exemple) :
interface eth1 ospf cost 45

Plus courts chemins (exemples)



Expérimentation

- exécutez un *traceroute* à partir de/à l'arrivée de différentes interfaces.
- exécutez un *traceroute –I* à partir de bb1 à 10.0.2.1
-- **Question Q1** : quel chemin le traceroute est supposé suivre ?
-- **Question Q2** : quel chemin les réponses ICMP (*ICMP replies*) sont supposées suivre ?

- exécutez un *traceroute –I* à partir de bb1 vers 10.0.3.2
-- **Question Q3.A :** quel chemin le traceroute est supposé suivre ?

- Observez l'interconnexion entre les routes OSPF et les réseaux directement connectés (en exécutant la commande *« show ip route »* dans "zebra")
- **Q3.B :** Essayez de changer les couts et observez les effets de ces changements

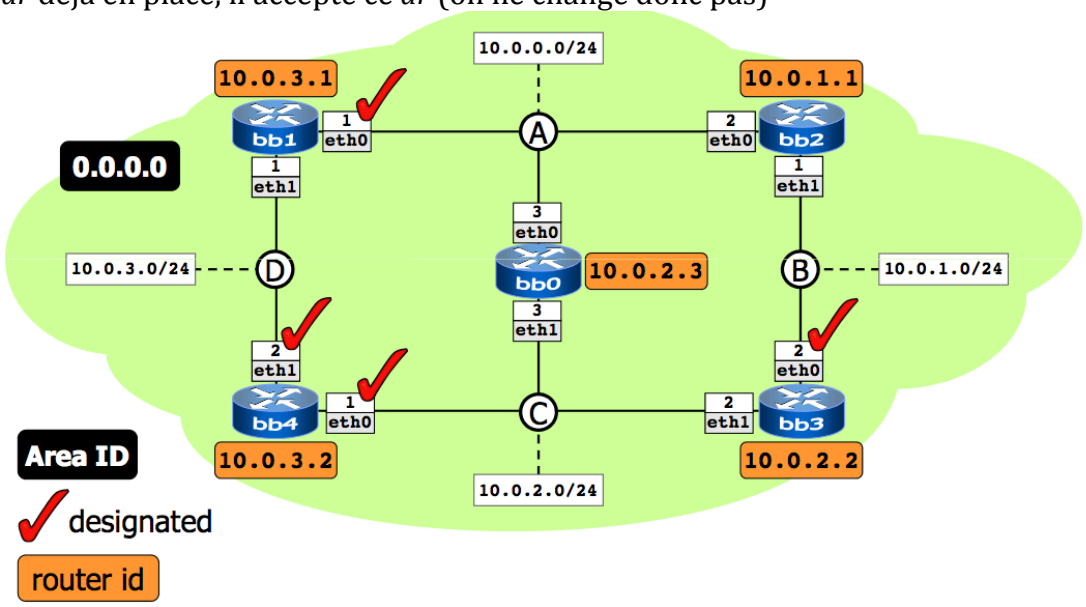
Maintenant, accédez au *ospfd cli* sur les différents routeurs, et utilisez les commandes suivantes :
- show ip ospf database
- show ip ospf neighbor
- show ip ospf route

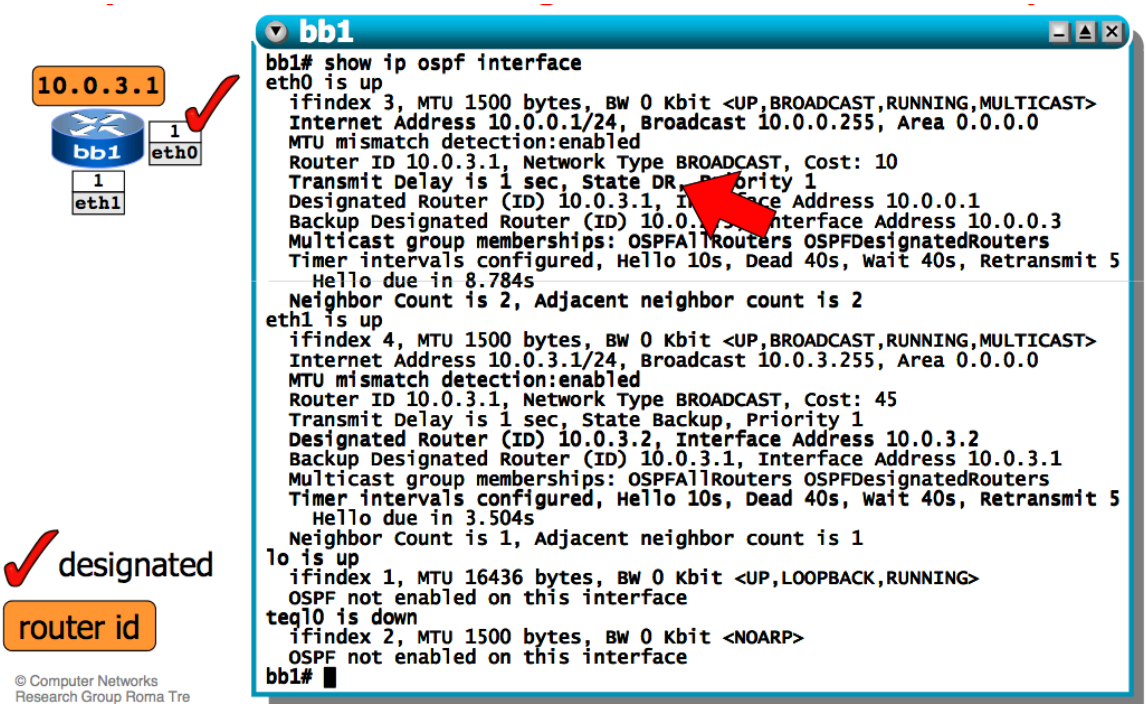
- **<u>Question Q4</u>** : Vérifiez que le *lsdb* est exactement le même pour chaque routeur.

Designated routers
(interface du routeur désignés pour chaque réseau)

- Pour chaque réseau, une des interfaces connectées à ce réseau est élue comme **« designated » (*dr*)**
- L'élection est basée sur des priorités, en utilisant les « hello packets »
-- le routeur (interface) qui envoie les hello packets avec la priorité plus élevée gagne l'élection
-- en cas d'égalité, *tie break* en utilisant le « router id » plus élevée (par default, le router id est l'adresse d'une de ses interfaces)
-- priorité dans le range $\in [0,255]$, par default priority = 1,
si priority=0 $\Rightarrow$ « never become a dr »
- un *dr* de backup (celui avec la seconde plus haute priorité) est aussi élu, pour sortir rapidement des situations de « *failure* » du *dr* principal

- le changement d'un *dr* équivaut à un changement du modèle de topologie OSPF (des nouveau *lsas* sont envoyés)
- pour cette raison, le *dr* est changé rarement
- Si un routeur avec une haute priorité se réveille/est ajouté au réseau, et trouve un *dr* déjà en place, il accepte ce *dr* (on ne change donc pas)





Vision du réseau en OSPF

- En échangeant les messages de *« link state update »*, chaque routeur apprend la topologie complète du réseau, c'est à dire :
-- les routeurs existants
-- les *subnets*
-- les adjacences (c'est à dire, les *interconnexions*) entre routeurs et réseaux

Neighborhood

- Les voisins d'un routeur (*router neighbors*) peuvent être montrés en utilisant la commande : **show ip ospf neighbor**

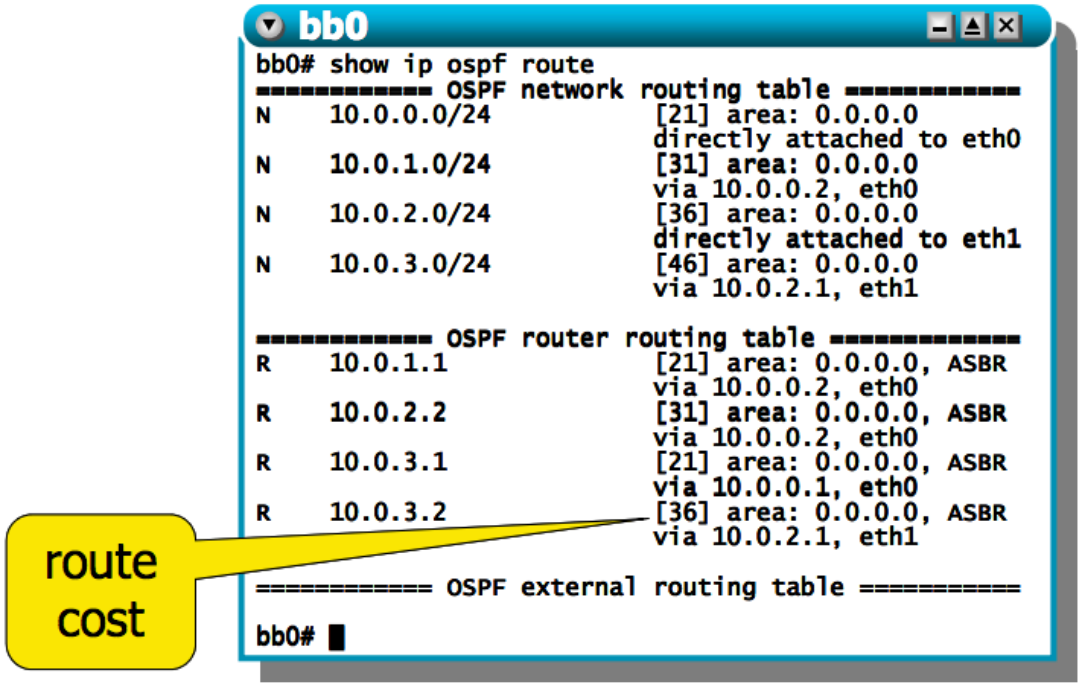
- Il faut noter que : les *lsas* sont échangés entre voisins seulement si en *Full state* (et donc, capables d'un échange bidirectionnel de l'information) ; pour atteindre l'état de *Full state*, les conditions suivantes doivent être réunies :
- les voisins ont été découverts (utilisant les « hello packets »)
- une communication bidirectionnelle est possible
- un *designated router* a été élu
- Une fois rejoint, chaque routeur synchronise immédiatement son *lsdb*

```
bb0
bb0# show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
10.0.3.1          1 Full/DR         30.462s 10.0.0.1    eth0:10.0.0.3            0     0     0
10.0.1.1          1 Full/DROther    30.462s 10.0.0.2    eth0:10.0.0.3            0     0     0
10.0.3.2          1 Full/DR         31.587s 10.0.2.1    eth1:10.0.2.3            0     0     0
10.0.2.2          1 Full/DROther    31.586s 10.0.2.2    eth1:10.0.2.3            0     0     0
bb0#
```

OSPF routing table

Le tableau de routage OSPF peut être montré en utilisant la commande :
show ip ospf route



Expérimentation

- **Q5** Exécutez les commandes **show ip ospf database** et **show ip ospf neighbor** sur les différents routeurs
- **Q6** capturez, à l'aide de tcpdump, les paquets OSPF échangés entre les routeurs

OSPF détecte rapidement les changements de topologie

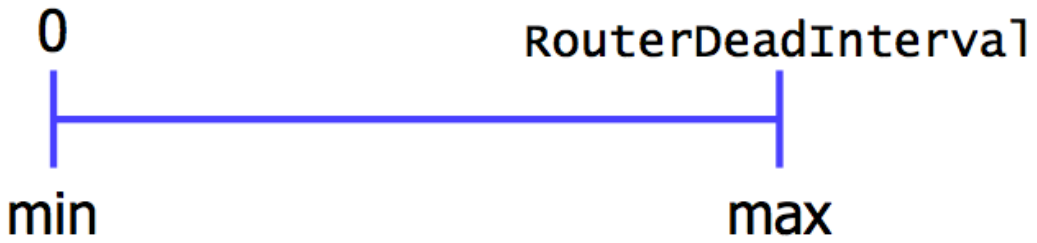
Premier cas (#1) : ***link fault***
- éliminer (*bring/shut down*) une interface réseau en utilisant la commande **ifconfig**
-- ce changement est immédiatement propagé par le routeur dans les paquets *lsa*
-- les tableaux de routage sont immédiatement mis à jour (**show ip ospf route**)
-- le *lsdb* est géré un peu différemment …

Si le *link fault* intéresse un *dr*, cette information est immédiatement répercutée sur le(s) lsdb(s)...
- ... et éventuellement annoncée à nouveau quand un nouveau *dr* est re-élu

- Sinon, OSPF attend l'expiration du *RouterDeadInterval timer* (default: 40s) avant d'enlever l'information d'adjacence du lsdb (**show ip ospf database network**)

A noter que : les réseaux connectés à un seul routeur, appelés *stub networks*, sont visibles seulement utilisant la commande **show ip ospf database network**

Donc, le temps total de réaction au *link failure* (estimation) est :

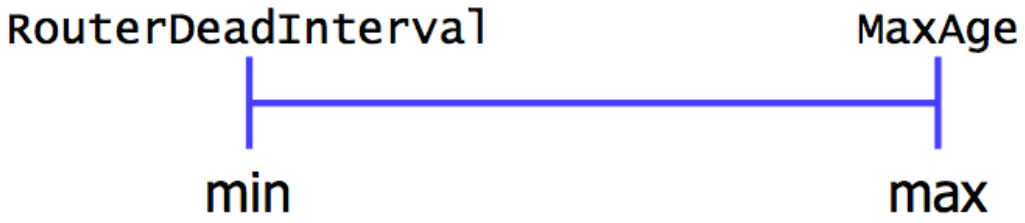


Deuxième cas (#2) : *router fault*
- il suffit d'éteindre un routeur (soit la machine virtuelle correspondante, soit en éliminant toutes ses interfaces en même temps)
- dans ce cas, le routeur, n'a évidemment pas la possibilité de propager/envoyer les *lsas …*
- … et donc ce changement de topologie ne peut pas se propager immédiatement
- les routeurs voisins peuvent d'apercevoir de ce changement (et donc mettre à jour les tableaux de routage) après l'expiration du *RouterDeadInterval timer*

Après que le changement est propagé …
- … l'information dans le lsdb concernant les réseaux pour lesquels le routeur « failed » n'était PAS *dr* est immédiatement prise des lsdbs des autres routeurs
--- le *dr* prend soin d'envoyer les lsas appropriés

- ... l'information dans le lsdb concernant les réseaux pour lesquels le roteur « failed » était bien le *dr* (en incluant aussi ceux dans lesquels un *dr* sera re-élu) est plus « compliquée »
--- OSPF attend l'expiration du *lsa* (cela arrive quand l'âge du lsa devient = MaxAge, 1 heure) avant d'entreprendre n'importe quelle action.

Donc, le temps total de réaction au *router failure* (estimation) est :



OSPF areas

- Il s'agit d'une abstraction qui permet de simplifier l'administration du réseau, et d'améliorer la « scalabilité » (passage à échelle) du protocole.
--- En effet, la topologie à l'intérieur d'une *area* est invisible à l'extérieur de cette area même

--- Les routeurs d'une area donnée ne voient pas les détails externes à la topologie de l'area
- Chaque *area* utilise une instance distincte de l'algorithme de *link state*
--- Tous les routeurs d'une même area construisent et maintiennes à jour le même *lsdb*
--- Chaque routeur maintient un *lsdb* distinct pour chaque *area* à laquelle il appartient

Une area OSPF est identifiée avec 32-bit, souvent en format *dotted decimal* (1.2.3.4)
--- Des interfaces différentes d'un même routeur peuvent être assignées à différentes areas

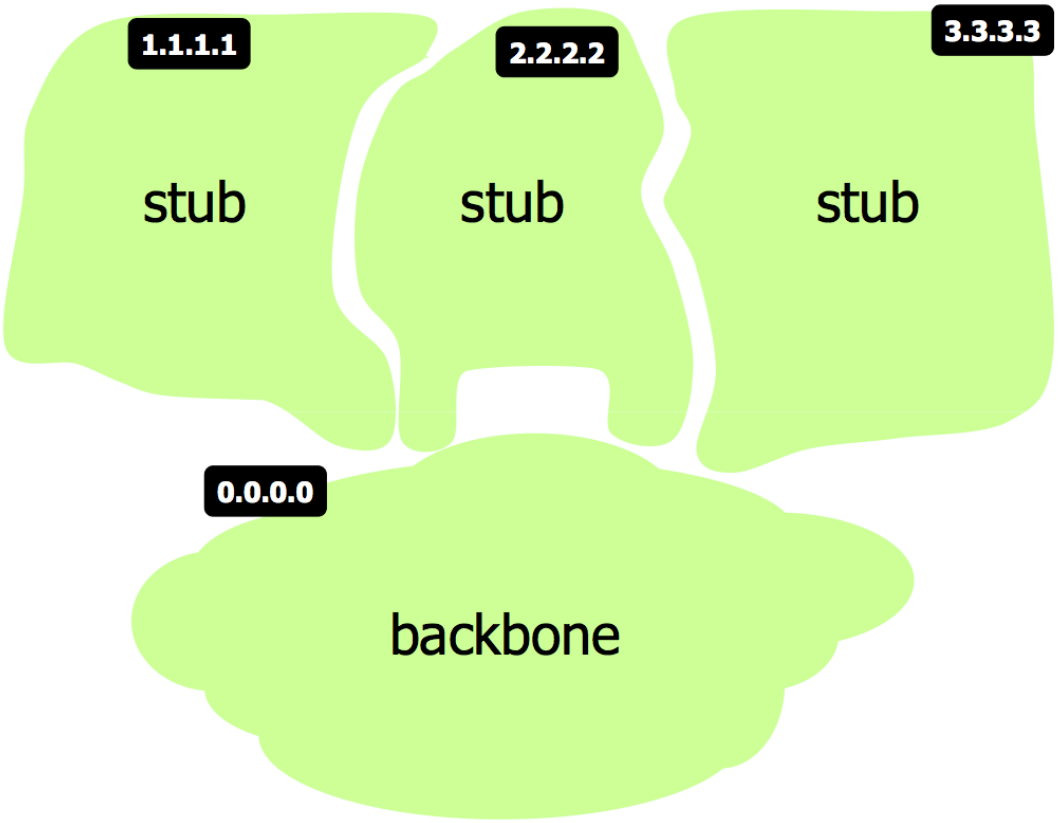
- Mais :
-- chaque interface
-- chaque réseau
-- chaque « router adjacency » …
... est associé à une et une seule *area*

Area types

backbone (0.0.0.0)
- doit être (virtuellement) connectée
- toutes les autres areas sont connectées à elle
- elle contient tous les *area border routers*

stub
- ne reçoit pas les *advertisements* es routeurs externes
- les routeurs internes se voient offrir une route par default
- ne peuvent pas contenir des *autonomous system boundary routers*
- le *backbone* ne peut pas être une *stub area*

transit
- utilisée pour faire passer le trafic d'une area à une autre, à travers des *virtual links*



Typologies de Routers

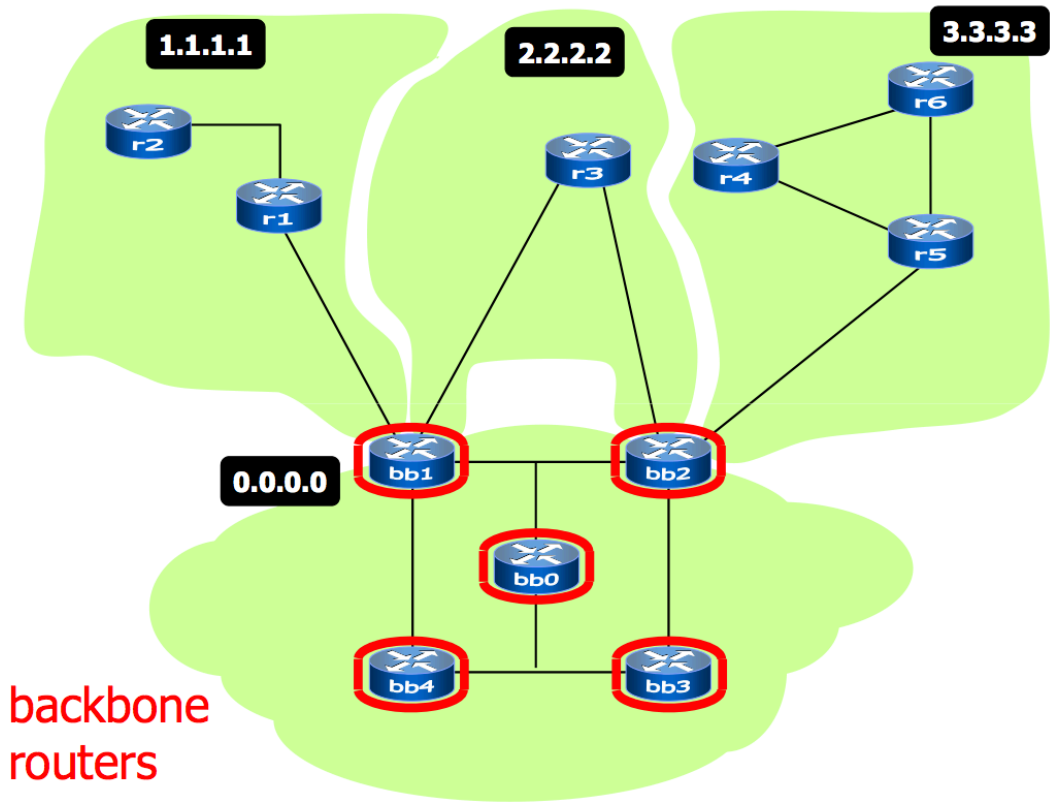
internal router
- toutes ses interfaces appartiennent à une même *area*

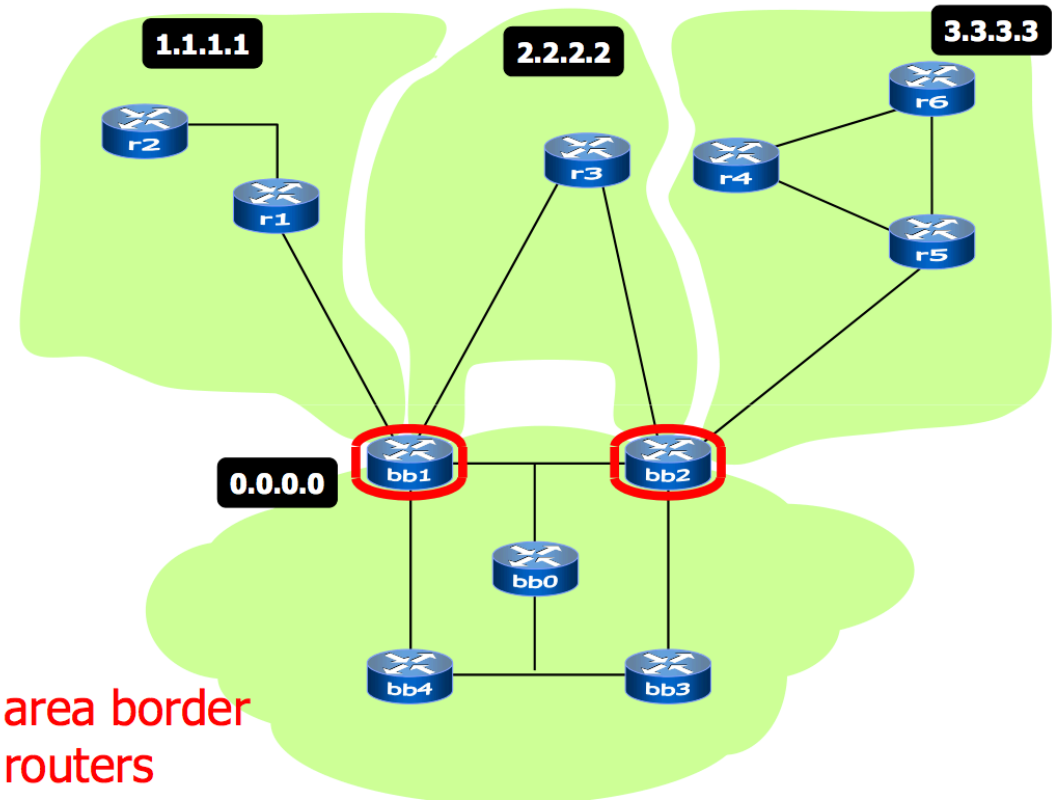
area border router (abr)
- connecte une ou plusieurs *areas* au *backbone*
- il maintients plusieurs *lsdbs*, un pour chaque *area*

backbone router
- a au moins une interface connectée au backbone
- un *abr* est toujours un b*ackbone router*

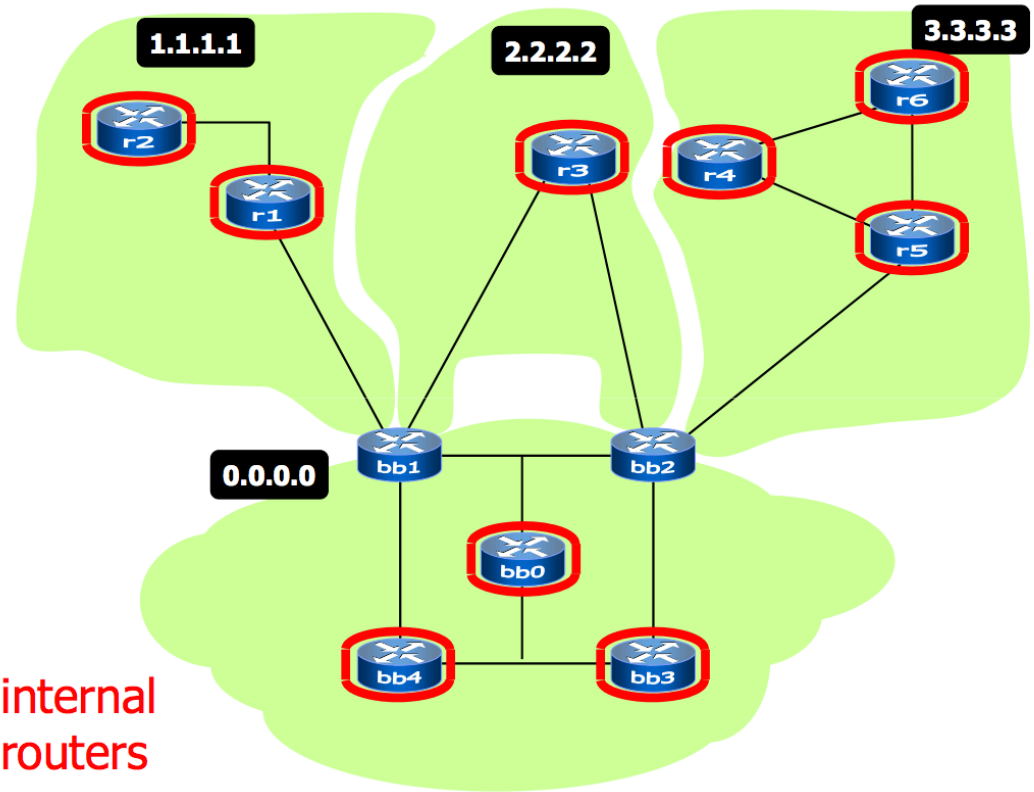
autonomous system boundary router (asbr)
- il fait l'importation et ensuite le *flooding* des informations de routage en provenance d'autres protocoles de routage (typiquement, BGP)

- à noter : un routeur peut appartenir à plusieurs des 4 catégories ci-dessus.

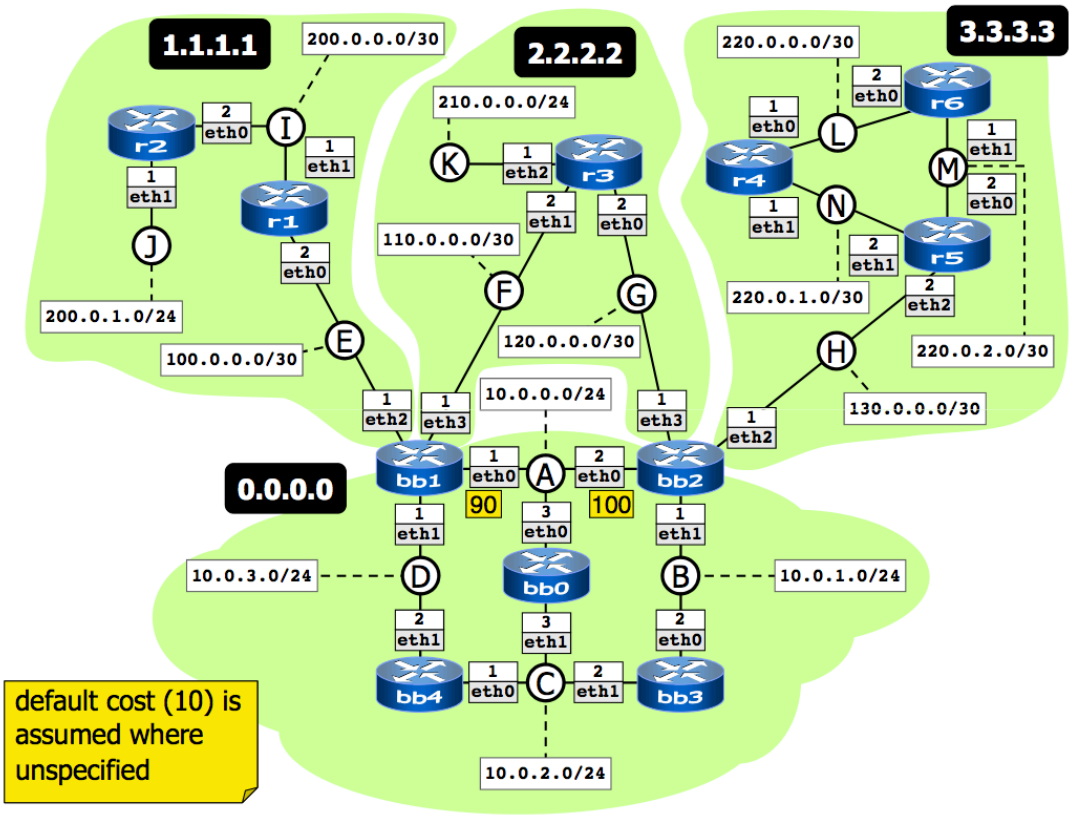




area border routers



internal routers



Area configuration

- L'information sur une « area » peut être trouvée dans 2 endroits :
-- en habilitant OSPF sur une interface du routeur
network 200.0.0.0/16 area 1.1.1.1
-- en spécifiant le type d'area (pas nécessaire pour le *backbone*)
area 1.1.1.1 stub

Expérimentation : *Areas*

- **Q7** : vérifiez que chaque routeur connaît la topologie détaillée seulement de sa propre « area » (en utilisant les commandes « show ip ospf neighbor » et « show ip ospf database network »)

- **Q8** : vérifiez que chaque routeur connaît ce qui se passe à l'extérieur de sa propre « area », utilisant la commande **show ip ospf database summary**
- **Q8.A** : en particulier, vérifiez les valeurs « Metric » , qui montrent la distance (en terme de cout) de la destination à partir de *l'advertising abr (area border router)*
- **Q8.B** : vérifiez que les routeurs dans les stub areas ont une route par default, tandis que les routeurs dans la backbone area n'en ont pas
- **Q8.C** : vérifiez, enfin, quelle métrique (« Metric ») est assignée à la default route

Expérimentation : *fault recovery*

- Expérimentez la capacité de « *fault recovery* » d'OSPF
- Quand plusieurs chemins (routes) ayant le même cout existent, OSPF garde en mémoire toutes ces routes.
- **Q9** : en particulier, vérifier cela dans le routeur **r3**, en regardant sa route par default.

- Dans ce cas, Zebra sélectionne le chemin parmi les multiples existants.
- **Q10** : maintenant, dans **bb1** éliminez (shut down) l'interface **eth3** en utilisant la commande **ifconfig**, attendez quelques seconds et vérifiez quel chemin suivent les paquets (traceroute)
- **Q11** : ensuite, toujours dans **bb1**, re-allumez l'interface **eth3** et vérifiez à nouveau comment le routage est affecté par ce changement.