Introduction au routage dynamique OSPF avec FRRouting

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

Le protocole OSPF (*Open Shortest Path First*) permet un routage dynamique efficace et évolutif dans les grands réseaux IP, en calculant automatiquement les meilleures routes en fonction de l'état des liens.

Ce support de travaux pratiques est une introduction au protocole de routage dynamique OSPF. Il détaille la mise en place d'une topologie en triangle utilisant des VLANs, ainsi que la configuration des routeurs avec la suite logicielle FRRouting pour activer et paramétrer OSPF dans une aire unique.

Les manipulations présentées expliquent comment préparer les systèmes, valider les communications entre routeurs et configurer les démons OSPFv2 et OSPFv3 étape par étape.

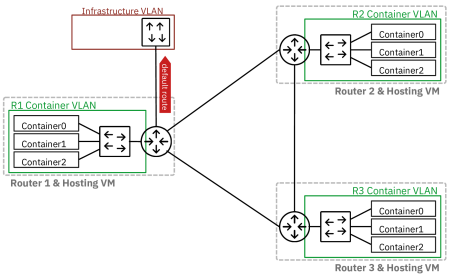


Table des matières

1. Copyright et Licence 1
2. Objectifs 2
3. Topologie réseau étudiée 2
4. Préparer les systèmes pour le routage IPv4 et IPv6 3
    4.1. Raccorder et lancer les routeurs virtuels 3
    4.2. Activer le routage sur les routeurs virtuels 5
    4.3. Appliquer une première configuration réseau 6
5. Installer le paquet frr et lancer les démons de routage OSPF 7
6. Valider les communications entre routeurs 10
7. Configurer les démons OSPFv2 et OSPFv3 12
8. Publier les routes par défaut via OSPF 20
9. Ajouter un réseau d'hébergement à chaque routeur 24
10. Adapter de la métrique de lien au débit 29
11. Sauvegarder les fichiers de configuration 31
12. Pour conclure... 33

1. Copyright et Licence

```
Copyright (c) 2000,2026 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included
in the section entitled "GNU Free Documentation License".
```

```
Copyright (c) 2000,2026 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon
les termes de la Licence de Documentation Libre GNU (GNU Free Documentation
License), version 1.3 ou toute version ultérieure publiée par la Free Software
Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture,
et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est
incluse dans la section intitulée « Licence de Documentation Libre GNU ».
```

Méta-information

Ce document est écrit avec **DocBook** XML sur un système **Debian GNU/Linux**. Il est disponible en version imprimable au format PDF : ospf.pdf.

2. Objectifs

Après avoir réalisé les manipulations présentées dans ce document, les étudiants seront capables de :

1. Configurer et activer le protocole de routage dynamique OSPF (versions 2 et 3) sur des routeurs virtuels Linux utilisant FRRouting.
2. Mettre en place une topologie réseau en triangle avec des VLANs et configurer le routage inter-VLAN.
3. Publier et redistribuer des routes par défaut via OSPF dans une aire unique.
4. Créer et configurer des réseaux d'hébergement de conteneurs attachés à chaque routeur, en utilisant des commutateurs virtuels et de l'adressage automatique.
5. Ajuster les métriques OSPF en fonction des débits des liens et optimiser le routage.

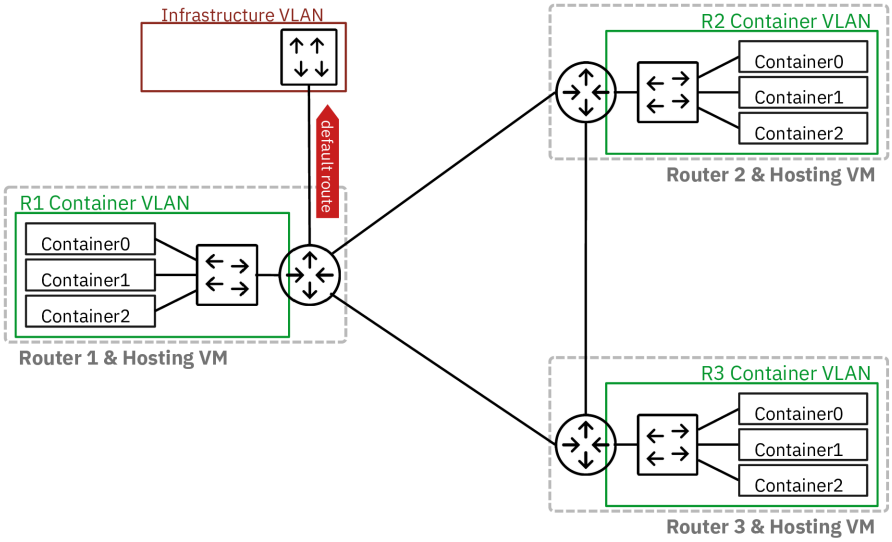
3. Topologie réseau étudiée

La topologie réseau étudiée peut être présentée sous deux formes distinctes : logique et physique.

Topologie logique

On retrouve un grand classique dans l'introduction aux protocoles de routage dynamiques : le triangle dont les trois côtés (liens) sont de type LAN.

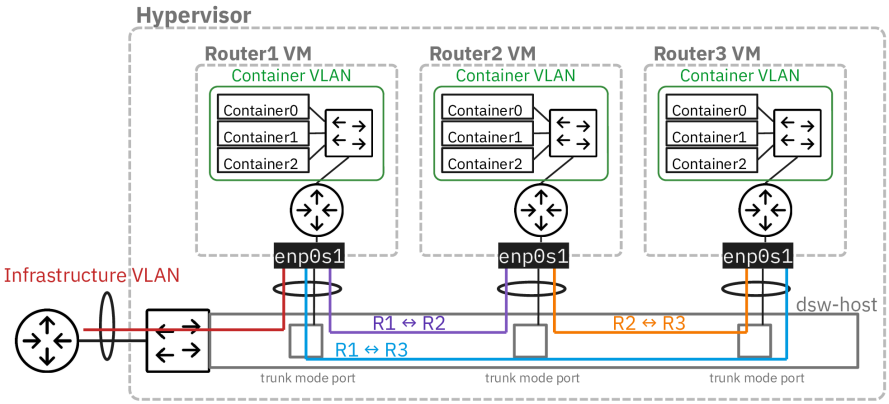
Si on se place sur un sommet du triangle, le côté opposé correspond à un réseau “inconnu” que l'on peut joindre via les routeurs voisins.



Topologie logique en triangle

Topologie physique

On s'appuie sur le support **Routage Inter-VLAN** pour constituer une topologie physique à base de réseaux locaux virtuels ou VLANs. On fait correspondre à chaque lien de la topologie logique en triangle un numéro de VLAN défini.



Topologie physique en étoile

Après avoir mis en œuvre la topologie physique en s'appuyant sur le support de la séance de travaux pratiques précédente : **Routage Inter-VLAN**, on implante les démons de routage OSPF sur les trois routeurs R1, R2 et R3.

Ce support se limite à l'étude du routage dynamique à l'intérieur d'une aire unique. La seule “frontière” de communication inter-aires visible est constituée par le lien vers l'Internet. Cette route par défaut sera redistribuée via OSPF par le routeur R1 aux autres routeurs. On verra alors un exemple de route externe dans les bases de données OSPF.

Voici le plan d'adressage de la maquette qui a été utilisée pour rédiger ce document.

Tableau 1. Plan d'adressage de la maquette « Introduction au routage dynamique OSPF »

Rôle	OSPF router-id	VLAN	Type	Adresses
R1	OSPFv2 : 1.0.0.4 OSPFv3 : 1.0.0.6	360	Passerelle	192.168.104.129/29 fe80:168::1
		440	R1 -> R2	10.44.0.1/29 fe80::1b8:1/64
		441	R1 -> R3	10.44.1.1/29 fe80::1b9:1/64
		10	Hébergement	10.10.0.1/24 fd14:ca46:3864:a::1/64
R2	OSPFv2 : 2.0.0.4 OSPFv3 : 2.0.0.6	440	R2 -> R1	10.44.0.2/29 fe80::1b8:2/64
		442	R2 -> R3	10.44.2.2/29 fe80::1ba:2/64
		20	Hébergement	10.20.0.1/24 fd14:ca46:3864:14::1/64
R3	OSPFv2 : 3.0.0.4 OSPFv3 : 3.0.0.6	441	R3 -> R1	10.44.1.3/29 fe80::1b9:3/64
		442	R3 -> R2	10.44.2.3/29 fe80::1ba:3/64
		30	Hébergement	10.30.0.1/24 fd14:ca46:3864:1e::1/64

4. Préparer les systèmes pour le routage IPv4 et IPv6

4.1. Raccorder et lancer les routeurs virtuels

Les trois machines virtuelles de routage doivent être raccordées au commutateur de distribution de l'hyperviseur.

Q1 Comment configurer les ports du commutateur avant le lancement des machines virtuelles ?

On utilise le script de procédure `switch-conf.py` qui applique les déclarations contenues dans un fichier YAML. Le code du script est accessible à partir du dépôt Git **startup-scripts**.

Voici une copie du fichier de configuration `switch.yaml` des trois ports de commutateur.

```
ovs:
  switches:
    - name: dsw-host
      ports:
        - name: tap5 # R1 switch port
          type: OVSPort
          vlan_mode: trunk
          trunks: [360, 440, 441]
        - name: tap6 # R2 switch port
          type: OVSPort
          vlan_mode: trunk
          trunks: [52, 440, 442] # VLAN 52 provides Internet temporary access
#         trunks: [440, 442]
        - name: tap7 # R3 switch port
          type: OVSPort
          vlan_mode: trunk
          trunks: [52, 441, 442] # VLAN 52 provides Internet temporary access
#         trunks: [441, 442]
```

On applique les paramètres définis ci-dessus.

```
$HOME/masters/scripts/switch-conf.py switch.yaml
```

Les numéros de ports de commutateur et de VLAN donnés dans les exemples ci-dessus doivent être adaptés selon le plan d'adressage spécifique à vos travaux pratiques.

Q2 **Comment afficher les entrées de la table CAM du commutateur de distribution sur l'hyperviseur ?**

Rechercher les options de la commande **ovs-appctl** à partir de la console de l'hyperviseur.

L'affichage de la table CAM du commutateur `dsw-host` de l'hyperviseur se fait à l'aide de la commande :

```
sudo ovs-appctl fdb/show dsw-host
```

Pour sélectionner un VLAN particulier, on ajoute un appel à **grep**. Voici des exemples d'affichage pour les VLANs de la maquette **une fois que les routeurs virtuels sont lancés et échangent entre eux**.

- Lien entre R1 et R2.

```
sudo ovs-appctl fdb/show dsw-host | grep 440
   10   440  b8:ad:ca:fe:00:05  274
   18   440  b8:ad:ca:fe:00:06  120
```

- Lien entre R1 et R3.

```
sudo ovs-appctl fdb/show dsw-host | grep 441
   12   441  b8:ad:ca:fe:00:07    1
   10   441  b8:ad:ca:fe:00:05    1
```

- Lien entre R2 et R3.

```
sudo ovs-appctl fdb/show dsw-host | grep 442
   18   442  b8:ad:ca:fe:00:06   51
   12   442  b8:ad:ca:fe:00:07   51
```

Q3 **Comment lancer les trois routeurs virtuels ?**

On utilise le script de procédure `lab-startup.py` qui applique les déclarations contenues dans un fichier YAML. Le code du script est accessible à partir du dépôt Git **startup-scripts**.

Voici une copie du fichier `frr-lab.yaml` de déclaration des trois routeurs virtuels.

```
kvm:
  vms:
    - vm_name: R1
      os: linux
      master_image: debian-testing-amd64.qcow2 # master image to be used
      force_copy: false # do not force copy the master image to the VM image
      memory: 1024
      tapnum: 5
    - vm_name: R2
      os: linux
      master_image: debian-testing-amd64.qcow2 # master image to be used
      force_copy: false # do not force copy the master image to the VM image
      memory: 1024
      tapnum: 6
    - vm_name: R3
      os: linux
      master_image: debian-testing-amd64.qcow2 # master image to be used
      force_copy: false # do not force copy the master image to the VM image
      memory: 1024
      tapnum: 7
```

Les numéros d'interfaces `tap` sont à changer suivant les attributions du plan d'adressage de travaux pratiques.

On lance les trois routeurs virtuels avec le script **lab-startup.py**.

```
$HOME/masters/scripts/lab-startup.py frr-lab.yaml
```

Q4 **Comment changer le nom d'hôte de chacun des trois routeurs virtuels ?**

Utiliser la connexion console avec la commande **telnet** et définir le nouveau nom d'hôte avec la commande **hostnamectl**.

On obtient la liste des numéros de ports d'accès console au lancement des routeurs virtuels ou à l'aide de la commande **lsof -i** par exemple. Ensuite, on ouvre la connexion sur le port ouvert et on change le nom d'hôte et on redémarre le système.

```
telnet localhost 2306
```

```
Trying ::1...
Connected to localhost.
Escape character is '^]'.

localhost login: etu
Mot de passe :
Linux localhost 6.11.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.11.4-1 (2024-10-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
etu@localhost:~$ sudo hostnamectl hostname R2
etu@localhost:~$ sudo reboot
```

4.2. Activer le routage sur les routeurs virtuels

Sans modification de la configuration par défaut, un système GNU/Linux n'assure pas la fonction de routage du trafic d'une interface réseau à une autre.

L'activation du routage correspond à un réglage de paramètres du sous-système réseau du noyau Linux. L'outil qui permet de consulter et modifier les réglages de paramètre sur le noyau est appelé sysctl.

Q5 **Comment activer le routage dans le sous-système réseau du noyau Linux ?**

Utiliser la commande **sysctl** pour effectuer des recherches et identifier les paramètres utiles.

Il est dorénavant recommandé de créer un fichier de configuration spécifique par fonction. C'est la raison pour laquelle on crée un nouveau fichier `/etc/sysctl.d/70-routing.conf`.

Voici un exemple de création de fichier qui active les clés relatives au routage IPv4 et IPv6.

```
cat << EOF | sudo tee /etc/sysctl.d/70-routing.conf
# The 70-routing.conf parameters are applied last, after the files in the
# /usr/lib/sysctl.d directory.
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians=1
EOF
```

Avertissement

Il ne faut pas oublier d'appliquer les nouvelles valeurs des paramètres de configuration.

```
sudo sysctl --system
```

On utilise la même commande pour vérifier que la fonction routage est bien activée pour les deux protocoles réseau.

```
sudo sysctl net.ipv4.ip_forward net.ipv6.conf.all.forwarding
```

```
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
```

4.3. Appliquer une première configuration réseau

Q6 Comment appliquer les configurations réseau IPv4 et IPv6 à partir de l'unique interface du routeur ?

Consulter la documentation de **Netplan** pour obtenir les informations sur la configuration des interfaces réseau à l'adresse **Netplan documentation**.

Il existe plusieurs possibilités pour configurer une interface réseau. Dans le contexte de ces manipulations, on utilise **Netplan** dans le but de séparer la partie déclarative du moteur de configuration.

C'est `systemd-networkd` qui joue le rôle de moteur de configuration sur les machines virtuelles utilisées avec ces manipulations.

La configuration par défaut de l'interface `enp0s1` doit être éditée et remplacée. Il faut configurer autant de sous-interfaces que de VLANs utilisés.

- L'interface principale correspond à l'interface “physique” de la machine. Elle est nommée `enp0s1` en fonction de l'ordre des adresses des composants raccordés au bus PCI.
- Une sous-interface doit être créée pour chaque VLAN désigné dans le plan d'adressage.

  Le routeur `R1` se distingue des deux autres par son raccordement au VLAN d'infrastructure qui lui donne directement accès à l'Internet.

  Les routeurs `R2` et `R3` doivent utiliser un accès temporaire à un VLAN sur lequel l'adressage automatique est actif. Le but de cet accès temporaire est de permettre l'installation des paquets nécessaires à la suite des manipulations.

Voici une copie du fichier `/etc/netplan/enp0s1.yaml` du routeur `R1`.

```
network:
    version: 2
    ethernets:
      enp0s1:
        dhcp4: false
        dhcp6: false
        accept-ra: false
        nameservers:
          addresses:
            - 172.16.0.2
            - 2001:db8:3::2

    vlans:
      # Infrastructure VLAN
      enp0s1.360:
        id: 360
        link: enp0s1
        addresses:
          - 192.168.104.130/29
          - 2001:db8:168::82/64
        routes:
          - to: default
            via: 192.168.104.129
          - to: "::/0"
            via: "fe80::168:1"
            on-link: true
      # R1 -> R2
      enp0s1.440:
        id: 440
        link: enp0s1
        addresses:
          - 10.44.0.1/29
          - fe80::1b8:1/64
      # R1 -> R3
      enp0s1.441:
        id: 441
        link: enp0s1
        addresses:
          - 10.44.1.1/29
          - fe80::1b9:1/64
```

Voici une copie du fichier `/etc/netplan/enp0s1.yaml` du routeur `R2` avec une sous-interface temporaire dans le VLAN `52`.

```
network:
    version: 2
    ethernets:
      enp0s1:
        dhcp4: false
        dhcp6: false
        accept-ra: false
        nameservers:
          addresses:
            - 172.16.0.2
            - 2001:db8:3::2

    vlans:
      # Temporary Internet access
      enp0s1.52:
        id: 52
        link: enp0s1
        dhcp4: true
        dhcp6: false
        accept-ra: true
      # R2 -> R1
      enp0s1.440:
        id: 440
        link: enp0s1
        addresses:
          - 10.44.0.2/29
          - fe80::1b8:2/64
      # R2 -> R3
      enp0s1.442:
        id: 442
        link: enp0s1
        addresses:
          - 10.44.2.2/29
          - fe80::1ba:2/64
```

Ce deuxième fichier de déclaration de la configuration réseau est pratiquement identique à celui du routeur R3. C'est pourquoi, on ne fournit pas de copie de ce dernier.

Une fois le fichier de configuration en place, il suffit de faire appel à la commande **netplan** pour appliquer les changements.

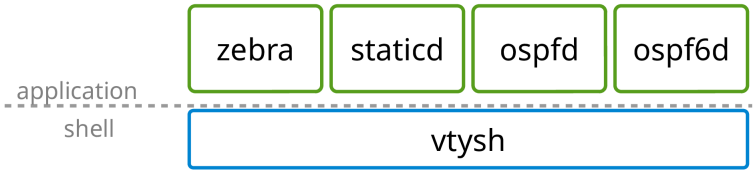
```
sudo netplan apply
```

Pour vérifier l'état courant de la configuration appliquée, on peut utiliser à nouveau la commande **netplan**.

```
sudo netplan status
```

5. Installer le paquet frr et lancer les démons de routage OSPF

La suite de démons de routage FRRouting couvre la totalité des protocoles de routage dynamiques. Le paquet FRR fournit autant de processus que de protocoles. Ici, on se concentre sur le protocole OSPF. Voici une représentation de l'organisation des démons actifs dans notre contexte.



Les manipulations de ce support s'appuient sur la documentation du projet : **FRRouting User Guide**

Note

Au moment de la rédaction de ce document, FRRouting ne permet pas d'utiliser simultanément les familles d'adresses IPv4 et IPv6 avec un seul démon. C'est pourquoi nous devons activer et configurer deux démons distincts : `ospfd` pour IPv4 et `ospf6d` pour IPv6.

Important

Les questions de cette section doivent être traitées sur les trois routeurs de la topologie étudiée.

Q7 Comment installer le paquet `frr` à partir du dépôt du site **FRRouting Project** ?

Pour installer le paquet FRR, on doit ajouter un nouveau dépôt au système.

On commence par ajouter la clé de signature des paquets à la configuration du gestionnaire.

```
sudo apt -y install curl gpg
```

```
curl -s https://deb.frrouting.org/frr/keys.asc | \
sudo gpg -o /usr/share/keyrings/frr-keyring.gpg --dearmor
```

On créé ensuite un nouveau fichier de liste de sources de paquets qui fait référence à cette clé de signature.

```
echo "deb [signed-by=/usr/share/keyrings/frr-keyring.gpg] \
https://deb.frrouting.org/frr trixie frr-stable" | \
sudo tee /etc/apt/sources.list.d/frr.list
```

Avant de lancer l'installation des paquets de la suite FRRouting, on doit mettre à jour le catalogue local.

```
sudo apt update
sudo apt -y install frr frr-pythontools
```

On peut afficher les informations sur le paquet `frr`.

```
apt show ^frr$
```

Sans configuration particulière, les services zebra et staticd sont lancés. Aucun protocole de routage dynamique n'est activé.

```
systemctl status frr
```

```
# frr.service - FRRouting
     Loaded: loaded (/usr/lib/systemd/system/frr.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-11-02 16:17:42 CET; 5min ago
 Invocation: a1d3f0e79647471bb1a17069f3f4c69a
       Docs: https://frrouting.readthedocs.io/en/latest/setup.html
    Process: 2098 ExecStart=/usr/lib/frr/frrinit.sh start (code=exited, status=0/SUCCESS)
   Main PID: 2107 (watchfrr)
     Status: "FRR Operational"
      Tasks: 8 (limit: 1032)
     Memory: 14.9M (peak: 28.2M)
        CPU: 416ms
     CGroup: /system.slice/frr.service
             ├─2107 /usr/lib/frr/watchfrr -d -F traditional zebra mgmtd staticd
             ├─2117 /usr/lib/frr/zebra -d -F traditional -A 127.0.0.1 -s 90000000
             ├─2122 /usr/lib/frr/mgmtd -d -F traditional -A 127.0.0.1
             └─2124 /usr/lib/frr/staticd -d -F traditional -A 127.0.0.1

nov. 02 16:17:42 R2 staticd[2124]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
nov. 02 16:17:42 R2 frrinit.sh[2144]: [2144|staticd] done
nov. 02 16:17:42 R2 zebra[2117]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
nov. 02 16:17:42 R2 frrinit.sh[2128]: [2128|zebra] done
nov. 02 16:17:42 R2 watchfrr[2107]: [QDG3Y-BY5TN] zebra state -> up : connect succeeded
nov. 02 16:17:42 R2 watchfrr[2107]: [QDG3Y-BY5TN] mgmtd state -> up : connect succeeded
nov. 02 16:17:42 R2 watchfrr[2107]: [QDG3Y-BY5TN] staticd state -> up : connect succeeded
nov. 02 16:17:42 R2 watchfrr[2107]: [KWE5Q-QNGFC] all daemons up, doing startup-complete notify
nov. 02 16:17:42 R2 frrinit.sh[2098]: Started watchfrr.
nov. 02 16:17:42 R2 systemd[1]: Started frr.service - FRRouting.
```

Q8 **Comment vérifier que la console unifiée du service `frr` est active ?**

Afficher le contenu du fichier `/etc/frr/vtysh.conf` et vérifier qu'il contient l'entrée `service integrated-vtysh-config`.

L'accès à cette console unifiée est important puisqu'il permet d'utiliser une console unique pour les trois démons qui sont utilisés dans la suite des manipulations : zebra, ospfd et ospf6d.

Voici un exemple pour un routeur de la maquette.

```
sudo grep "service integrated-vtysh-config" /etc/frr/vtysh.conf
```

```
service integrated-vtysh-config
```

Q9 **Comment ajouter l'utilisateur `etu` aux groupes `frr` et `frrvty` ?**

Utiliser la commande **adduser**.

Une fois que l'utilisateur appartient à ces groupes, il a un accès direct à la console de configuration des protocoles actifs.

Comme dans les autres travaux pratiques de la série, on utilise une boucle.

```
for grp in frr frrvty
do
    sudo adduser etu $grp
done
```

Il ne faut pas oublier de déconnecter/reconnecter l'utilisateur pour bénéficier de la nouvelle attribution de groupe.

On vérifie l'appartenance aux groupes avec la commande **groups**.

```
groups
```

```
etu adm sudo users frrvty frr
```

Q10 Comment activer les deux démons des protocoles OSPFv2 et OSPFv3 ?

Consulter le fichier de configuration : /etc/frr/daemons.

Il faut remplacer les clés no en yes pour les démons des deux versions du protocole OSPF.

```
sudo sed -i 's/ospfd=no/ospfd=yes/' /etc/frr/daemons
sudo sed -i 's/ospf6d=no/ospf6d=yes/' /etc/frr/daemons
sudo systemctl restart frr
```

On peut alors afficher l'état du service frr et vérifier que les nouveaux démons de routage dynamique OSPF sont bien activés.

```
systemctl status frr
```

```
# frr.service - FRRouting
     Loaded: loaded (/usr/lib/systemd/system/frr.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-11-02 17:40:32 CET; 6s ago
 Invocation: 84e33888211f45f297c9135cace76751
       Docs: https://frrouting.readthedocs.io/en/latest/setup.html
    Process: 2467 ExecStart=/usr/lib/frr/frrinit.sh start (code=exited, status=0/SUCCESS)
   Main PID: 2476 (watchfrr)
     Status: "FRR Operational"
      Tasks: 12 (limit: 1032)
     Memory: 23M (peak: 34.8M)
        CPU: 380ms
     CGroup: /system.slice/frr.service
             ├─2476 /usr/lib/frr/watchfrr -d -F traditional zebra mgmtd ospfd ospf6d staticd
             ├─2488 /usr/lib/frr/zebra -d -F traditional -A 127.0.0.1 -s 90000000
             ├─2493 /usr/lib/frr/mgmtd -d -F traditional -A 127.0.0.1
             ├─2495 /usr/lib/frr/ospfd -d -F traditional -A 127.0.0.1
             ├─2498 /usr/lib/frr/ospf6d -d -F traditional -A ::1
             └─2501 /usr/lib/frr/staticd -d -F traditional -A 127.0.0.1

nov. 02 17:40:32 R2 mgmtd[2493]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
nov. 02 17:40:32 R2 frrinit.sh[2504]: [2504|mgmtd] done
nov. 02 17:40:32 R2 watchfrr[2476]: [QDG3Y-BY5TN] zebra state -> up : connect succeeded
nov. 02 17:40:32 R2 watchfrr[2476]: [QDG3Y-BY5TN] mgmtd state -> up : connect succeeded
nov. 02 17:40:32 R2 watchfrr[2476]: [QDG3Y-BY5TN] ospfd state -> up : connect succeeded
nov. 02 17:40:32 R2 watchfrr[2476]: [QDG3Y-BY5TN] ospf6d state -> up : connect succeeded
nov. 02 17:40:32 R2 watchfrr[2476]: [QDG3Y-BY5TN] staticd state -> up : connect succeeded
nov. 02 17:40:32 R2 watchfrr[2476]: [KWE5Q-QNGFC] all daemons up, doing startup-complete notify
nov. 02 17:40:32 R2 frrinit.sh[2467]: Started watchfrr.
nov. 02 17:40:32 R2 systemd[1]: Started frr.service - FRRouting.
```

On peut aussi lister les démons actifs à partir de la console du service.

```
vtysh
```

```
Hello, this is FRRouting (version 10.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R2# sh daemons
 mgmtd zebra ospfd ospf6d watchfrr staticd
```

Q11 Comment désactiver les sous-interfaces d'accès temporaire à Internet pour les routeurs R2 et R3 ?

Il existe au moins deux possibilités.

- Utiliser directement la commande **ip** pour désactiver une sous-interface au niveau liaison.
- Éditer le fichier de déclaration et commenter tous les paramètre de la sous-interface temporaire.

Si on utilise la commande **ip**, l'interface sera bien désactivée mais elle apparaîtra à nouveau lors du redémarrage du routeur virtuel.

```
sudo ip link set enp0s1.52 down
```

Si on édite le fichier de déclaration en commentant les paramètres de la sous-interface, on s'assure qu'elle n'apparaîtra plus, même lors d'un redémarrage du routeur. Voici un exemple pour le routeur R3.

```
network:
    version: 2
    ethernets:
      enp0s1:
        dhcp4: false
        dhcp6: false
        accept-ra: false
        nameservers:
          addresses:
            - 172.16.0.2
            - 2001:db8:3::2

    vlans:
#      # Temporary Internet access
#      enp0s1.52:
#        id: 52
#        link: enp0s1
#        dhcp4: true
#        dhcp6: false
#        accept-ra: true
      # R3 -> R1
      enp0s1.440:
        id: 440
        link: enp0s1
        addresses:
          - 10.44.0.3/29
          - fe80::1b8:3/64
      # R3 -> R2
      enp0s1.442:
        id: 442
        link: enp0s1
        addresses:
          - 10.44.2.3/29
          - fe80::1ba:3/64
```

Une fois l'ensemble des opérations de cette section réalisées, chaque routeur dispose des outils pour mettre en œuvre la topologie physique et ensuite les protocoles de routage dynamique OSPF.

6. Valider les communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider plusieurs éléments :

- Le raccordement des routeurs aux ports de commutateurs désignés
- Les communications entre chaque routeur
- La visualisation des tables de routage pour les interfaces réseau configurées

Q12 Quelles sont les tests à effectuer pour vérifier l'état des différents "côtés" de la topologie triangle ?

Il faut afficher la table de routage de chaque routeur puis la table des voisins. Ainsi, on peut contrôler les correspondances entre les adresses de couche réseau et de couche liaison.

Dans le contexte de la maquette on obtient les résultats suivants pour le routeur R1. On se limite à l'affichage des entrées de la table de routage apprises par le noyau avec l'option `proto kernel`.

```
ip route ls proto kernel
```

```
10.44.0.0/29 dev enp0s1.440 scope link src 10.44.0.1
10.44.1.0/29 dev enp0s1.441 scope link src 10.44.1.1
192.168.104.128/29 dev enp0s1.360 scope link src 192.168.104.130
```

```
ip -6 route ls proto kernel
```

```
2001:db8:168::/64 dev enp0s1.360 metric 256 pref medium
fe80::/64 dev enp0s1 metric 256 pref medium
fe80::/64 dev enp0s1.441 metric 256 pref medium
fe80::/64 dev enp0s1.360 metric 256 pref medium
fe80::/64 dev enp0s1.440 metric 256 pref medium
```

Pour les entrées relatives au voisinage réseau, on se limite aussi à l'affichage des voisins directs des interfaces du routeur.

On repère dans la copie d'écran ci-dessous les adresses des deux autres routeurs : R2 et R3.

```
ip nei ls
```

```
10.44.0.2 dev enp0s1.440 lladdr b8:ad:ca:fe:00:06 REACHABLE❶
10.44.1.3 dev enp0s1.441 lladdr b8:ad:ca:fe:00:07 REACHABLE❷
fe80::1b9:3 dev enp0s1.441 lladdr b8:ad:ca:fe:00:07 STALE❸
fe80::baad:caff:fefe:6 dev enp0s1.440 lladdr b8:ad:ca:fe:00:06 router STALE❹
fe80:168::1 dev enp0s1.360 lladdr 80:6a:00:dc:67:53 router STALE
fe80::baad:caff:fefe:7 dev enp0s1.441 lladdr b8:ad:ca:fe:00:07 STALE❺
fe80::1b8:2 dev enp0s1.440 lladdr b8:ad:ca:fe:00:06 STALE❻
```

❶❻❹Entrées du routeur R2
❷❸❺Entrées du routeur R3

Important

Les tables de voisinage réseau sont peuplées à partir des échanges entre routeurs. Il est donc nécessaire de lancer les tests de communication ICMP en amont de l'affichage de ces tables.

Q13 **Quelles sont les opérations à effectuer pour valider les communications IPv4 et IPv6 entre chacun des routeurs ?**

Lancer les tests ICMP usuels entre chaque routeur sur chaque lien actif.

Exemple entre R1 et R2 ; toujours dans le contexte de la maquette.

- Requête IPv4 de R1 vers R2 sur le VLAN 440.

```
ping -qc2 10.44.0.2
```

```
PING 10.44.0.2 (10.44.0.2) 56(84) bytes of data.

--- 10.44.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.533/7.471/14.409/6.938 ms
```

Aucun paquet n'a été perdu. Le routeur R2 est bien joignable depuis R1.

- Requête IPv6 *multicast* depuis R1 sur le VLAN 480.

```
ping -qc2 ff02::1%enp0s1.440
```

```
PING ff02::1%enp0s1.440 (ff02::1%enp0s1.440) 56 data bytes

--- ff02::1%enp0s1.440 ping statistics ---
2 packets transmitted, 2 received, +1 duplicates, 0% packet loss, time 1031ms
rtt min/avg/max/mdev = 0.065/0.283/0.719/0.307 ms
```

L'opération est à répéter sur chaque lien entre deux routeurs reliés sur le même VLAN.

Q14 **Quelles sont les opérations à effectuer pour visualiser les tables de routage IPv4 et IPv6 existantes d'un routeur au niveau de la console unifiée vtysh ?**

Afficher les tables de routage à partir de la console vtysh avec les commandes du système Cisco™ IOS **show ip route** et **show ipv6 route**.

Dans le contexte de la maquette, on obtient les résultats suivants pour le routeur R2.

```
R2# sh ip route connected
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 10.44.0.0/29 is directly connected, enp0s1.440, 00:40:37
C>* 10.44.2.0/29 is directly connected, enp0s1.442, 00:40:37
```

```
R2# sh ipv6 route connected
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIPng, O - OSPFv3, I - IS-IS, B - BGP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C * fe80::/64 is directly connected, enp0s1, 00:41:41
C * fe80::/64 is directly connected, enp0s1.442, 00:41:41
C>* fe80::/64 is directly connected, enp0s1.440, 00:41:41
```

Comme dans le cas de la question précédente, l'affichage des tables de routage doit être fait sur les trois routeurs virtuels pour vérifier la cohérence de la topologie avant de passer au routage dynamique.

7. Configurer les démons OSPFv2 et OSPFv3

Dans cette section, on introduit les premières commandes de configuration du protocole de routage dynamique OSPF qui permettent d'activer le protocole puis d'ajouter des entrées de réseau dans la base de données de ce protocole.

Q15 Comment peut on contrôler que le protocole OSPF est actif ou non sur un routeur ?

Une fois la console vtysh ouverte, lancer les commandes de visualisation de l'état du protocole listées ci-dessous. Ces commandes peuvent être lancées sur chacun des trois routeurs.

show daemons
show ip ospf
show ipv6 ospf6

Dans les informations données dans la copie d'écran ci-dessous, il apparaît qu'aucune configuration du protocole de routage dynamique n'a été activée.

```
R1# sh daemons
 zebra ospfd ospf6d watchfrr staticd
```

```
R1# sh ip ospf
```

```
R1# sh ipv6 ospf
% OSPFv3 instance not found
```

Q16 Quelles sont les opérations à effectuer pour activer les protocoles de routage OSPFv2 et OSPFv3 ? Comment affecter manuellement l'identifiant du routeur ?

Avertissement

Les identifiants à utiliser lors de la séance de travaux pratiques sont donnés dans les tableaux des plans d'adressage.

La liste des commandes utiles en mode configuration dans la console vtysh est la suivante.

router ospf
router ospf6
ospf router-id X.X.X.X
ospf6 router-id X.X.X.X
log detail

Toujours à partir de la console vtysh, on accède au mode configuration à l'aide de la commande **conf t**. Voici un exemple de séquence sur le troisième routeur.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# ospf router-id 1.0.0.4
R1(config-router)# log detail
R1(config-router)# ^Z
```

```
R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# ospf6 router-id 1.0.0.6
R1(config-ospf6)# log detail
R1(config-ospf6)# ^Z
```

Une fois que chaque démon de routage OSPF possède un identifiant unique, on peut afficher les propriétés du protocole de routage dynamique même si aucun échange de route n'a encore eu lieu.

```
sh run ospfd
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
router ospf
 ospf router-id 1.0.0.4
 log-adjacency-changes detail
exit
!
end
```

```
sh run ospf6d
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
router ospf6
 ospf6 router-id 1.0.0.6
 log-adjacency-changes detail
exit
!
end
```

Le choix de codage des identifiants OSPF a pour but d'éviter une confusion avec les adresses des réseaux actifs sur chaque routeur.

Si on reprend les instructions de la question précédente, on obtient l'état de chacun des démons de protocole de routage dynamique.

```
sh ip ospf
```

```
OSPF Routing Process, Router ID: 1.0.0.4
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 0 millisec(s)
 Minimum hold time between consecutive SPFs 50 millisec(s)
 Maximum hold time between consecutive SPFs 5000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm has not been run
 SPF timer is inactive
 LSA minimum interval 5000 msecs
 LSA minimum arrival 1000 msecs
 Write Multiplier set to 20
 Refresh timer 10 secs
 Maximum multiple paths(ECMP) supported 256
 Administrative distance 110
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 0
 All adjacency changes are logged
```

```
sh ipv6 ospf
```

```
OSPFv3 Routing Process (0) with Router-ID 1.0.0.6
 Running 00:09:20
 LSA minimum arrival 1000 msecs
 Maximum-paths 256
 Administrative distance 110
 Initial SPF scheduling delay 0 millisec(s)
 Minimum hold time between consecutive SPFs 50 millsecond(s)
 Maximum hold time between consecutive SPFs 5000 millsecond(s)
 Hold time multiplier is currently 1
 SPF algorithm has not been run
 SPF timer is inactive
 Number of AS scoped LSAs is 0
 Number of areas in this router is 0
 Authentication Sequence number info
  Higher sequence no 0, Lower sequence no 0
 All adjacency changes are logged
```

L'affectation des identifiants des démons de routage OSPF doit être réalisée sur les trois routeurs de la topologie. Ces identifiants seront très utiles et importants dès qu'il faudra afficher les listes de routeurs voisins au sens du protocole OSPF.

Q17 Comment activer les protocoles de routage OSPFv2 et OSPFv3 pour les réseaux d'interconnexion de chaque routeur ?

Il faut activer le protocole de routage dynamique sur chaque interface de la topologie qui participe à la construction du triangle.

La liste des commandes utiles en mode console et en mode configuration dans vtysh est la suivante.

show ip route connected
show ip route ospf
show ipv6 route connected
show ipv6 route ospf
ip ospf area 0
ipv6 ospf6 area 0

Voici un exemple de séquence d'instructions pour le routeur R1.

On commence par lister les entrées marquées **C** ou ***connected*** des tables de routage IPv4 et IPv6 de façon à reconnaître les deux côtés de la topologie triangle connus du “sommet” R1.

```
sh ip route connected
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 10.44.0.0/29 is directly connected, enp0s1.440, 01:26:12
C>* 10.44.1.0/29 is directly connected, enp0s1.441, 01:26:12
C>* 192.168.104.128/29 is directly connected, enp0s1.360, 01:26:12
```

```
sh ipv6 route connected
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIPng, O - OSPFv3, I - IS-IS, B - BGP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 2001:db8:168::/64 is directly connected, enp0s1.360, 01:30:26
C * fe80::/64 is directly connected, enp0s1.360, 01:30:25
C * fe80::/64 is directly connected, enp0s1.440, 01:30:26
C * fe80::/64 is directly connected, enp0s1, 01:30:26
C>* fe80::/64 is directly connected, enp0s1.441, 01:30:26
```

À partir de ces affichages, on sait que l'on doit activer les protocoles OSPF pour les deux sous-interfaces : enp0s1.440 et enp0s1.441.

```
R1# conf t
R1(config)# int enp0s1.440
R1(config-if)# ip ospf area 0
R1(config-if)# ipv6 ospf6 area 0
R1(config-if)# int enp0s1.441
R1(config-if)# ip ospf area 0
R1(config-if)# ipv6 ospf6 area 0
R1(config-if)# ^Z
```

Ces opérations doivent être répétées sur les trois routeurs de la topologie.

Q18 Comment visualiser l'état des interfaces actives pour chaque processus de protocole de routage dynamique OSPFv2 ou OSPFv3 ?

Les interfaces sont dites actives pour les protocoles OSPFv2 ou OSPFv3 dès qu'elles ont été ajoutées aux processus de routage dynamique en précisant l'aire à laquelle elles appartiennent.

La liste des commandes utiles dans la console vtysh est la suivante.

show ip ospf interface
show ipv6 ospf6 interface

En prenant l'exemple du routeur R2, on obtient les résultats suivants.

```
R2# sh ip ospf interface enp0s1.440
```

```
enp0s1.440 is up❶
  ifindex 4, MTU 1500 bytes, BW 0 Mbit❷ <UP,LOWER_UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.44.0.2/29, Broadcast 10.44.0.7, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 2.0.0.4❸, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 1.0.0.4 Interface Address 10.44.0.1/29❹
  Backup Designated Router (ID) 2.0.0.4, Interface Address 10.44.0.2❺
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 5.086s
  Neighbor Count is 1, Adjacent neighbor count is 1❻
  Graceful Restart hello delay: 10s
```

La copie d'écran ci-dessus permet d'identifier les éléments suivants :

❶ L'indicateur **is up** confirme que l'interface est bien active pour le protocole de routage. Cela signifie que les messages du protocole OSPF sont transmis sur cette interface et que des échanges avec des routeurs OSPF voisins sur ce réseau peuvent avoir lieu.
❷ Le fait que la bande passante soit “vue” comme étant nulle montre que le calcul de coût de lien ne prend pas en compte ce facteur. Ce point sera repris dans le calcul des coûts à partir d'une référence définie dans la configuration des démons OSPF.
❸ On retrouve ici l'identifiant du routeur transmis vers les autres routeurs voisins.
❹ Cette ligne identifie le routeur OSPF R1 comme étant le routeur de référence sur ce réseau IPv4. Dans notre cas, la topologie triangle ne comprend qu'un seul routeur voisin OSPF par réseau, ce qui limite l'utilité d'un routeur de référence pour les calculs des meilleurs routes. Si on avait plusieurs routeurs présents sur un même réseau, le rôle de référent serait essentiel pour limiter les échanges de bases de données de routage lors des calculs de tables de topologie.
❺ Cette ligen identifie le routeur OSPF R2 comme étant le routeur de référence de secours sur ce réseau. Comme dans le cas précédent, l'utilisation d'une topologie triangle limite l'importance de ce rôle comme il n'y a que deux routeurs pour chaque côté de cette topologie.
❻ Le routeur OSPF a un routeur voisin sur ce réseau. Il s'agit du routeur R1.

On reprend la même démarche pour le protocole OSPFv3.

```
R2# sh ipv6 ospf6 interface enp0s1.440
```

```
enp0s1.440 is up❶, type BROADCAST
  Interface ID: 4
  Internet Address:
    inet : 10.44.0.2/29
    inet6: fe80::baad:caff:fefe:6/64
    inet6: fe80::1b8:2/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 10
  State BDR, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
   Hello 10(8.803), Dead 40, Retransmit 5
  DR: 1.0.0.6 BDR: 2.0.0.4❷
  Number of I/F scoped LSAs is 2
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
  Graceful Restart hello delay: 10s
  Authentication Trailer is disabled
```

Relativement, à l'affichage des informations sur l'association entre une interface réseau et le démon de protocole OSPFv2 pour IPv4, l'affichage pour OSPFv3 et IPv6 est plus succinct.

❶ Cette information est identique à celle du démon OSPFv2. L'interface est associée et active. Les messages du protocole OSPFv3 peuvent être échangés sur le réseau auquel cette interface appartient.
❷ Les identifiants des routeurs référence et de secours sont affichés sur une seule ligne.

Q19 Comment vérifier que l'identifiant de routeur a correctement été attribué ?

À partir des commandes proposées et de résultats des questions précédentes, rechercher l'information demandée.

Quelque soit la version du protocole OSPF, l'identifiant de routeur est toujours codé sous la forme d'une adresse IPv4.

Pour valider la conformité entre les identifiants définis dans le plan d'adressage des travaux pratiques et les valeurs effectivement utilisées par les deux démons de routage dynamique, on affiche le contenu des base de topologie du protocole.

Avec le démon OSPFv2, l'identification du routeur est immédiate.

```
R1# sh ip ospf database
```

```
       OSPF Router with ID (1.0.0.4)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.0.0.4        1.0.0.4          884 0x8000001f 0x3bee 2
2.0.0.4        2.0.0.4         1016 0x8000001d 0xab78 2
3.0.0.4        3.0.0.4          843 0x8000001d 0xd34a 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.44.0.1      1.0.0.4          994 0x80000018 0xe61c
10.44.1.1      1.0.0.4          984 0x80000018 0xe61a
10.44.2.3      3.0.0.4          843 0x80000018 0xbc3e
```

En revanche, le démon OSPFv3 ne donne pas d'information sur l'identifiant du processus en cours. L'affichage est cependant beaucoup plus exhaustif avec les informations par interface.

```
R1# sh ipv6 ospf6 database
```

```
       Area Scoped Link State Database (Area 0)

Type LSId            AdvRouter       Age   SeqNum                  Payload
Rtr  0.0.0.0         1.0.0.6        1012 8000001b              1.0.0.6/0.0.0.5
Rtr  0.0.0.0         1.0.0.6        1012 8000001b              1.0.0.6/0.0.0.3
Rtr  0.0.0.0         2.0.0.4        1010 80000019              1.0.0.6/0.0.0.5
Rtr  0.0.0.0         2.0.0.4        1010 80000019              3.0.0.6/0.0.0.4
Rtr  0.0.0.0         3.0.0.6        1010 80000019              1.0.0.6/0.0.0.3
Rtr  0.0.0.0         3.0.0.6        1010 80000019              3.0.0.6/0.0.0.4
Net  0.0.0.3         1.0.0.6        1088 80000017                     1.0.0.6
Net  0.0.0.3         1.0.0.6        1088 80000017                     3.0.0.6
Net  0.0.0.5         1.0.0.6        1012 80000017                     1.0.0.6
Net  0.0.0.5         1.0.0.6        1012 80000017                     2.0.0.4
Net  0.0.0.4         3.0.0.6        1010 80000017                     3.0.0.6
Net  0.0.0.4         3.0.0.6        1010 80000017                     2.0.0.4

       I/F Scoped Link State Database (I/F enp0s1.440 in Area 0)❶

Type LSId            AdvRouter       Age   SeqNum                  Payload
Lnk  0.0.0.5         1.0.0.6         303 8000001a                fe80::1b8:1
Lnk  0.0.0.4         2.0.0.4        1019 80000017                fe80::1b8:2

       I/F Scoped Link State Database (I/F enp0s1.441 in Area 0)❷

Type LSId            AdvRouter       Age   SeqNum                  Payload
Lnk  0.0.0.3         1.0.0.6         288 8000001a                fe80::1b9:1
Lnk  0.0.0.3         3.0.0.6        1090 80000017                fe80::1b9:3

       AS Scoped Link State Database

Type LSId            AdvRouter       Age   SeqNum                  Payload
```

❶ Sur le côté R1-R2 du triangle, on utilise le VLAN 440 et les identifiants de routeurs correspondent.
❷ Sur le côté R1-R3 du triangle, on utilise le VLAN 441 et les identifiants de routeurs correspondent.

Q20 Comment identifier le type de réseau d'une interface ?

À partir des résultats des questions précédentes, rechercher l'information demandée.

Comme on utilise des liens Ethernet dans ce contexte de travaux pratiques, le type le plus important est le réseau de diffusion ou *BROADCAST*.

```
R3# sh ip ospf interface enp0s1.442
```

```
enp0s1.442 is up
  ifindex 4, MTU 1500 bytes, BW 0 Mbit <UP,LOWER_UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.44.2.3/29, Broadcast 10.44.2.7, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 3.0.0.4, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 3.0.0.4 Interface Address 10.44.2.3/29
  Backup Designated Router (ID) 2.0.0.4, Interface Address 10.44.2.2
  Saved Network-LSA sequence number 0x80000019
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 1.361s
  Neighbor Count is 1, Adjacent neighbor count is 1
  Graceful Restart hello delay: 10s
```

```
R3# sh ipv6 ospf6 interface enp0s1.442
```

```
enp0s1.442 is up, type BROADCAST
  Interface ID: 4
  Internet Address:
    inet : 10.44.2.3/29
    inet6: fe80::baad:caff:fefe:7/64
    inet6: fe80::1ba:3/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 10
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
   Hello 10(9.055), Dead 40, Retransmit 5
  DR: 3.0.0.6 BDR: 2.0.0.4
  Number of I/F scoped LSAs is 2
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
  Graceful Restart hello delay: 10s
  Authentication Trailer is disabled
```

Q21 Comment obtenir la liste du ou des routeurs voisins pour chaque processus de protocole de routage dynamique OSPFv2 ou OSPFv3 ?

Dès qu'une interface est active, il y a émission de paquets HELLO et si un autre routeur avec une interface active envoie aussi des paquets HELLO dans le même VLAN, les deux routeurs cherchent à former une adjacence.

La liste des commandes utiles dans la console vtysh est la suivante.

show ip ospf neighbor
show ipv6 ospf6 neighbor

À partir du routeur R1, voici un exemple de liste de routeurs OSPF voisins dans laquelle on reconnaît les identifiants des routeurs R2 et R3.

```
R1# sh ip ospf neighbor
```

```
Neighbor ID     Pri State           Up Time         Dead Time Address         Interface                RXmtL RqstL DBsmL
2.0.0.4           1 Full/Backup     11h34m21s        37.712s 10.44.0.2       enp0s1.440:10.44.0.1         0     0     0
3.0.0.4           1 Full/Backup     11h35m30s        33.792s 10.44.1.3       enp0s1.441:10.44.1.1         0     0     0
```

```
R1# sh ipv6 ospf6 neighbor
```

```
Neighbor ID     Pri    DeadTime    State/IfState         Duration I/F[State]
2.0.0.4           1    00:00:37    Full/BDR             11:36:00 enp0s1.440[DR]
3.0.0.6           1    00:00:37    Full/BDR             11:37:15 enp0s1.441[DR]
```

Les deux listes de voisins donnent une information essentielle sur l'état de protocole de routage avec le mot clé Full. Cet état indique que deux routeurs adjacents ont entièrement synchronisé leurs bases de données d'état de liens, permettant ainsi un échange complet des informations de routage.

Q22 Comment identifier le rôle des différentes interfaces des routeurs pour chacun des liens du triangle de la topologie logique ?

Avertissement

La réponse à cette question suppose que les démons OSPF des trois routeurs de la topologie logique en triangle aient convergé. On doit repérer l'état Full pour les listes de routeurs voisins.

De plus, la réponse varie en fonction de l'ordre d'activation des démons OSPF des différents routeurs. En effet, un routeur peut être élu routeur désigné (DR) en l'absence de routeurs voisins. Cette élection n'est pas remise en cause tant qu'il n'y pas de changement d'état de lien.

À partir des résultats des questions précédentes sur les interfaces actives, il est possible de compléter le schéma de la topologie étudiée avec l'état des interfaces pour chacun des trois liens.

Sur un même réseau de diffusion, il est possible de trouver plusieurs routeurs OSPF. Établir une relation de voisinage et procéder aux échanges de bases de données topologiques entre chaque routeur revient à constituer un réseau de relations complètement maillé. À chaque nouveau calcul de topologie, ce réseau complètement maillé est inefficace. C'est la raison pour laquelle la notion de routeur référent ou ***Designated Router*** a été introduite. Lors d'un recalcul de topologie, tous les routeurs s'adressent au routeur référent qui correspond au cœur d'un réseau en topologie étoile.

Dans le contexte de la topologie triangle étudiée, il y a bien élection d'un routeur référent et d'un routeur référent de secours. Cependant, comme il n'y a que deux routeurs par domaine de diffusion ou VLAN, on ne peut pas caractériser l'utilité de cette élection.

Q23 **Quels sont les réseaux IPv4 et IPv6 présents dans la table de topologie du protocole OSPF ?**

On cherche a visualiser la liste des préfixes des réseaux connus des deux démons OSPF.

La liste des commandes utiles dans la console vtysh est la suivante.

show ip ospf route
show ipv6 ospf6 route

Une fois que les trois routeurs de la topologie ont convergé, chaque démon connaît les trois préfixes qui correspondent aux trois côtés du triangle. Un routeur correspond à un sommet du triangle et il doit apprendre le préfixe réseau du côté opposé via ses deux routeurs voisins.

Voici la vue depuis le routeur `R1`.

```
R1# sh ip ospf route
```

```
============ OSPF network routing table ============
N    10.44.0.0/29          [10] area: 0.0.0.0
                           directly attached to enp0s1.440
N    10.44.1.0/29          [10] area: 0.0.0.0
                           directly attached to enp0s1.441
N    10.44.2.0/29          [20] area: 0.0.0.0
                           via 10.44.0.2, enp0s1.440
                           via 10.44.1.3, enp0s1.441

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

Les valeurs notées entre crochets correspondent à la métrique du lien pour joindre le réseau noté à gauche.

```
R1# sh ipv6 ospf6 route
```

Cette dernière commande ne produit aucun résultat et c'est normal ! En effet, avec le protocole IPv6 les relations de voisinage entre routeurs (adjacences) se font avec les adresses IPv6 de lien local.

Formulé autrement, les trois côtés de la topologie triangle sont des réseaux de transit qui servent à acheminer le trafic entre routeurs voisins. Il est inutile de gaspiller un préfixe réseau IPv6 pour cette tâche.

Q24 **Comment sont calculées les coûts de liens pour joindre les réseaux ?**

Rechercher les informations sur les calculs de coûts dans les supports de cours sur le protocole OSPF.

Depuis la première spécification du protocole OSPF, le calcul de métrique peut se faire suivant deux méthodes.

- À partir de l'expression : 10^8 / Bande_Passante_du_lien
- À partir d'une valeur `Cost` définie manuellement

Note

La valeur du numérateur (10^8) correspond à un débit de 100Mbps. À l'époque de la rédaction du standard OSPFv2, ce débit a servi de référence. Aujourd'hui, cette valeur est complètement dépassée. C'est la raison pour laquelle on adapte le calcul de métrique en changeant le coefficient du numérateur. Voir la Section 10, « Adapter de la métrique de lien au débit ».

Dans notre cas, chaque interface de routeur a un coût de `10` et une valeur de “bande passante” de `0` par défaut. Utilisez la commande d'affichage des information OSPF pour revoir ces valeurs.

show ip ospf interface
show ipv6 ospf6 interface

Pour OSPFv2, les deux premiers réseaux de la table sont joignable via un unique lien avec un coût de `10`. Le troisième réseau est joignable via deux liens ; d'où la métrique de `20`.

Pour les préfixes IPv6, aucun préfixe n'est présent sachant que les relations de voisinage entre routeurs utilisent obligatoirement les adresses de lien local appartenant au préfixe `fe80::/10`.

Les préfixes des réseaux d'hébergement de conteneurs apparaitront dès que le protocole de routage aura été activé pour les interfaces SVI.

Q25 Comment visualiser les tables de routage depuis la console **vtysh** ?

L'affichage demandé illustre les mécanismes de choix entre différentes solutions pour une même destination. Cet affichage est à comparer avec celui demandé à la question suivante.

La liste des commandes utiles dans la console vtysh est la suivante.

show ip route
show ipv6 route

On reprend à nouveau l'exemple du routeur R1.

```
R1# sh ip route
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* 0.0.0.0/0 [0/0] via 192.168.104.129, enp0s1.360, 15:22:36
O   10.44.0.0/29 [110/10] is directly connected, enp0s1.440, weight 1, 13:49:19
C>* 10.44.0.0/29 is directly connected, enp0s1.440, 15:22:36
L>* 10.44.0.1/32 is directly connected, enp0s1.440, 15:22:36
O   10.44.1.0/29 [110/10] is directly connected, enp0s1.441, weight 1, 13:48:58
C>* 10.44.1.0/29 is directly connected, enp0s1.441, 15:22:36
L>* 10.44.1.1/32 is directly connected, enp0s1.441, 15:22:36
O>* 10.44.2.0/29 [110/20] via 10.44.0.2, enp0s1.440, weight 1, 12:30:55
  *                       via 10.44.1.3, enp0s1.441, weight 1, 12:30:55
C>* 192.168.104.128/29 is directly connected, enp0s1.360, 15:22:36
L>* 192.168.104.130/32 is directly connected, enp0s1.360, 15:22:36
```

```
R1# sh ipv6 route
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIPng, O - OSPFv3, I - IS-IS, B - BGP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* ::/0 [0/1024] via fe80:168::1, enp0s1.360 onlink, 15:23:47
K d 2001:db8:168::/64 [0/512] is directly connected, enp0s1.360, 15:23:44
C>* 2001:db8:168::/64 is directly connected, enp0s1.360, 15:23:46
L>* 2001:db8:168::82/128 is directly connected, enp0s1.360, 15:23:46
C * fe80::/64 is directly connected, enp0s1.360, 15:23:45
C * fe80::/64 is directly connected, enp0s1.440, 15:23:46
C * fe80::/64 is directly connected, enp0s1, 15:23:46
C>* fe80::/64 is directly connected, enp0s1.441, 15:23:46
```

- Les entrées marquées avec le caractère * correspondent aux routes retenues et mémorisées par le sous-système réseau du noyau. Les autres entrées sont placées en réserve au cas où la solution initialement retenue serait en défaut.

- L'entrée notée K correspond à une route apprise depuis le sous-système réseau du noyau.

- Les entrées notées C correspondent à des routes pour lesquelles il existe une interface sur le routeur. Les métriques de ses routes ont la valeur 0. Ce sont les routes les plus prioritaires.

- Les entrées notées L correspondent aux adresses locales du routeur dans un réseau connecté.

- Les entrées notées O correspondent aux routes apprises via le protocole OSPF. La métrique de ces routes se décompose en deux parties. La valeur figée à 110 définit le niveau de priorité du protocole OSPF (*Administrative Distance*) relativement aux autres protocoles de routage. Les valeurs notées après le / sont les métriques de liens calculées comme indiqué ci-dessus.

Q26 Comment visualiser les tables de routage au niveau système ?

Utiliser une commande usuelle de visualisation de la table de routage.

ip route ls
ip -6 route ls

Avec la commande **ip**, on voit apparaître les sources d'alimentation de la table de routage finale du système.

- kernel pour les entrées connues du sous-système réseau du noyau. Ce sont les entrées avec le caractère C dans la console **vtysh**.

- ospf pour les entrées apprises via le protocole de routage dynamique. Le réseau correspond au côté opposé au sommet du triangle est appris via OSPF puisque le sous-système réseau du noyau ne le connaît pas.

```
ip route ls
```

```
default via 192.168.104.129 dev enp0s1.360 proto static
10.44.0.0/29 dev enp0s1.440 proto kernel scope link src 10.44.0.1
10.44.1.0/29 dev enp0s1.441 proto kernel scope link src 10.44.1.1
10.44.2.0/29 nhid 30 proto ospf metric 20
        nexthop via 10.44.1.3 dev enp0s1.441 weight 1
        nexthop via 10.44.0.2 dev enp0s1.440 weight 1
192.168.104.128/29 dev enp0s1.360 proto kernel scope link src 192.168.104.130
```

La table de routage IPv6 ne fait apparaître aucune nouvelle entrée puisque les réseaux de conteneurs desservis par R2 et R3 n'ont pas encore été annoncés à ce stade de la configuration.

```
ip -6 route ls
```

```
2001:db8:168::/64 dev enp0s1.360 proto kernel metric 256 pref medium
2001:db8:168::/64 dev enp0s1.360 proto ra metric 512 expires 2591830sec pref high
fe80::/64 dev enp0s1 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.441 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.360 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.440 proto kernel metric 256 pref medium
default via fe80:168::1 dev enp0s1.360 proto static metric 1024 onlink pref medium
```

8. Publier les routes par défaut via OSPF

Dans la topologie logique étudiée, le routeur R1 dispose d'un lien montant vers l'Internet. On peut donc considérer que ce lien est la route par défaut vers tous les réseaux non connus de l'aire OSPF contenant les trois routeurs.

Il est possible de publier une route par défaut via le protocole OSPF depuis le routeur R1 vers les routeurs R2 et R3.

Pour rappel, revoir la question Q : Q19 et consulter les bases de données OSPFv2 et OSPFv3 avant la mise en place de la publication de route par défaut. On reconnaît les LSAs (***Link State Advertisement)***) de type 1 et 2 qui correspondent respectivement aux annonces de routeurs et de réseaux.

Q27 **Quelle est la condition préalable à respecter pour que le routeur R1 soit en mesure de publier une route par défaut via le protocole de routage OSPF ?**

À partir des tables de routage relevées dans la Section 7, « Configurer les démons OSPFv2 et OSPFv3 », repérer le routeur qui dispose d'un accès vers un réseau qui n'appartient pas à la topologie triangle.

ip route ls default
ip -6 route ls default
sh ip route kernel
sh ipv6 route kernel

Une route par défaut doit exister avant d'être injectée dans une aire OSPF. Dans notre cas, une route statique par défaut suffit à respecter la condition préalable.

Sur la maquette, on valide la présence des routes par défaut à l'aide la commande **ip** au niveau système.

```
ip route ls default
```

```
default via 192.168.104.129 dev enp0s1.360 proto static
```

```
ip -6 route ls default
```

```
default via fe80:168::1 dev enp0s1.360 proto static metric 1024 onlink pref medium
```

Au niveau de la console **vtysh**, ces même routes correspondent aux entrées marquées K pour ***kernel***.

```
R1# sh ip route kernel
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* 0.0.0.0/0 [0/0] via 192.168.104.129, enp0s1.360, 15:40:19
```

```
R1# sh ipv6 route kernel
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIPng, O - OSPFv3, I - IS-IS, B - BGP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* ::/0 [0/1024] via fe80:168::1, enp0s1.360 onlink, 15:41:24
K d 2001:db8:168::/64 [0/512] is directly connected, enp0s1.360, 15:41:2
```

Q28 Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPFv2 ?

Parmi toutes les méthodes de redistribution de routes disponibles avec le protocole OSPFv2, il en existe une dédiée à l'injection de route par défaut dans une aire normale. Consulter le guide **FRRouting User Guide**.

Rechercher le mot clé `redistribution` dans la section **OSPF route-map**.

L'instruction qui correspond à la redistribution de route par défaut à destination des autres routeurs de l'aire OSPF est la suivante.

default-information originate

On doit l'appliquer dans la section ***router ospf*** de la configuration du routeur R1.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# default-information originate
R1(config-router)# ^Z
```

Une fois cette instruction exécutée, le rôle du routeur R1 change. Il devient ***Autonomous System Boundary Router*** ou ASBR. Les bases de données sont complétées avec des LSAs de type 5.

```
R1# sh ip ospf database external
```

```
       OSPF Router with ID (1.0.0.4)

                AS External Link States

  LS age: 32
  Options: 0x2  : *|-|-|-|-|-|E|-
  LS Flags: 0xb
  LS Type: AS-external-LSA
  Link State ID: 0.0.0.0 (External Network Number)
  Advertising Router: 1.0.0.4
  LS Seq Number: 80000001
  Checksum: 0x269d
  Length: 36

  Network Mask: /0
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 10
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

On voit apparaître une nouvelle rubrique baptisée ***AS External Link States***. Ce nouveau rôle pour le routeur R1 apparaît aussi lorsque l'on affiche l'état de l'instance de routage OSPFv2.

```
R1# sh ip ospf
```

```
OSPF Routing Process, Router ID: 1.0.0.4
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 2m04s ago
Last SPF duration 68 usecs
SPF timer is inactive
LSA minimum interval 5000 msecs
LSA minimum arrival 1000 msecs
Write Multiplier set to 20
Refresh timer 10 secs
Maximum multiple paths(ECMP) supported 256
Administrative distance 110
This router is an ASBR (injecting external routing information)
Number of external LSA 1. Checksum Sum 0x0000269d
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1
All adjacency changes are logged
Area ID: 0.0.0.0 (Backbone)
  Number of interfaces in this area: Total: 2, Active: 2
  Number of fully adjacent neighbors in this area: 2
  Area has no authentication
  SPF algorithm executed 9 times
  Number of LSA 6
  Number of router LSA 3. Checksum Sum 0x0001a8b4
  Number of network LSA 3. Checksum Sum 0x0002727f
  Number of summary LSA 0. Checksum Sum 0x00000000
  Number of ASBR summary LSA 0. Checksum Sum 0x00000000
  Number of NSSA LSA 0. Checksum Sum 0x00000000
  Number of opaque link LSA 0. Checksum Sum 0x00000000
  Number of opaque area LSA 0. Checksum Sum 0x00000000
```

Le routeur R1, est maintenant à la frontière entre deux systèmes autonomes. Il est responsable de l'émission des LSAs de type 5 à destination des autres routeurs de l'aire.

Ici, R1 possède une route statique définie au niveau système vers l'Internet. Cette route statique est redistribuée R2 et R3. Cette route apparaît comme une entrée de type E2 dans les tables de routage de ces routeurs.

L'indicateur E2 correspond au type par défaut des routes apprises par le biais de la redistribution. La métrique est un point important à considérer avec les routes de type E2. Ces routes ne présentent que le coût du chemin allant du routeur ASBR vers le réseau de destination ; ce qui ne correspond pas au coût réel du chemin à l'intérieur de l'aire OSPF.

Q29 **Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPFv3 ?**

Reprendre la même démarche de la question précédente avec le protocole OSPFv3. Consulter le guide **FRRouting User Guide**.

Rechercher le mot clé `redistribution` dans la section **OSPF6 route-map**.

L'instruction est identique pour les deux versions du protocole OSPF.

default-information originate

On doit l'appliquer dans la section ***router ospf6*** de la configuration du routeur R1.

```
R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# default-information originate
R1(config-ospf6)# ^Z
```

Une fois cette instruction exécutée, le rôle du routeur R1 change. Il devient ***Autonomous System Boundary Router*** ou ASBR. Les bases de données sont complétées avec des LSAs de type 5.

```
R1# sh ipv6 ospf6 database as-external
```

```
        AS Scoped Link State Database

Type LSId            AdvRouter       Age   SeqNum                       Payload
ASE  0.0.0.1         1.0.0.6          42 80000001                          ::
```

On voit apparaître une nouvelle rubrique baptisée ***AS Scoped Link State Database***. Ce nouveau rôle pour le routeur R1 apparaît aussi lorsque l'on affiche l'état de l'instance de routage OSPFv3.

```
R1# sh ipv6 ospf6
```

```
OSPFv3 Routing Process (0) with Router-ID 1.0.0.6
Running 14:44:14
LSA minimum arrival 1000 msecs
Maximum-paths 256
Administrative distance 110
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millsecond(s)
Maximum hold time between consecutive SPFs 5000 millsecond(s)
Hold time multiplier is currently 1
SPF algorithm last executed 00:01:59 ago, reason R+, R-, A
Last SPF duration 0 sec 145 usec
SPF timer is inactive
Number of AS scoped LSAs is 1
Number of areas in this router is 1
Authentication Sequence number info
 Higher sequence no 0, Lower sequence no 0
All adjacency changes are logged

Area 0
    Number of Area scoped LSAs is 6
    Interface attached to this area: enp0s1.440 enp0s1.441
    SPF last executed 119.712612s ago
```

Q30 Comment la publication de route par défaut apparaît elle sur les autres routeurs de la topologie triangle ?

Relevez, dans la console **vtysh**, la métrique de la route par défaut sur les routeurs qui n'ont pas une connexion directe vers l'Internet.

Les instructions à utiliser pour traiter cette question entrent dans la liste suivante.

show ip route A.B.C.D/MM
show ipv6 route A:B:C::D/MM
show ip ospf route
show ipv6 ospf6 route

En prenant l'exemple du routeur R2, on retrouve les informations suivantes.

- Vue de la table de routage :

```
R2# sh ip route 0.0.0.0
```

```
Routing entry for 0.0.0.0/0
  Known via "ospf", distance 110, metric 10, best
  Last update 00:07:09 ago
  * 10.44.0.1, via enp0s1.440, weight 1
```

```
R2# sh ipv6 route ::
```

```
Routing entry for ::/0
  Known via "ospf6", distance 110, metric 10, best
  Last update 00:05:09 ago
  * fe80::1b8:1, via enp0s1.440, weight 1
```

- Vue de la base de topologie OSPF :

```
R2# sh ip ospf route
```

```
============ OSPF network routing table ============
N    10.44.0.0/29          [10] area: 0.0.0.0
                           directly attached to enp0s1.440
N    10.44.1.0/29          [20] area: 0.0.0.0
                           via 10.44.0.1, enp0s1.440
                           via 10.44.2.3, enp0s1.442
N    10.44.2.0/29          [10] area: 0.0.0.0
                           directly attached to enp0s1.442

============ OSPF router routing table =============
R    1.0.0.4               [10] area: 0.0.0.0, ASBR
                           via 10.44.0.1, enp0s1.440

============ OSPF external routing table ===========
N E2 0.0.0.0/0             [10/10] tag: 0
                           via 10.44.0.1, enp0s1.440
```

```
R2# sh ipv6 ospf6 route
```

```
*N E2 ::/0                          fe80::1b8:1              enp0s1.440 00:07:18
```

Avec les copies d'écran ci-dessus, on vérifie bien que les routes par défaut ont été apprises via le protocole OSPF.

Q31 **Comment assurer la traduction d'adresses source sur l'interface de sortie du routeur R1 vers l'Internet ?**

Reprendre la section **Activation de la traduction d'adresses** du support **Routage inter-VLAN et protocole PPPoE**.

Dans le cas de la maquette, c'est l'interface `enp0s1.360` qui assure l'interconnexion avec l'Internet. Voici une copie de l'instruction d'ajout de la règle de traduction des adresses sources et de sa sauvegarde pour IPv4 et IPv6.

- Installer le paquet `nftables`.

```
sudo apt -y install nftables
```

- Créer le fichier de règles `/etc/nftables.conf`.

```
cat << EOF | sudo tee /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

# Outbound interface
define RED_VLAN = enp0s1.360

table inet nat {
    chain postrouting {
        type nat hook postrouting priority 100;
        oifname $RED_VLAN counter masquerade
    }
}
EOF
```

- Appliquer la règle de traduction d'adresses sources en activant le service.

```
sudo systemctl enable --now nftables.service
```

- Afficher la liste des règles de filtrage actives.

```
sudo nft list ruleset
```

Pour terminer, on peut lancer des tests de communication ICMP IPv4 depuis les routeurs R2 ou R3.

```
etu@R2:~$ ping -qc2 9.9.9.9
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.

--- 9.9.9.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 28.604/28.676/28.748/0.072 ms
```

Important

À ce stade de la configuration, les tests IPv6 sont impossible à réaliser dans la mesure où toutes les adresses présentes sont des adresses de lien local.

---

9. Ajouter un réseau d'hébergement à chaque routeur

Dans cette section, on créé un réseau de conteneur attaché à chaque routeur de la topologie triangle. L'objectif est de peupler les tables de routage en ajoutant un lien OSPF au delà des routeurs.

Une fois ces réseaux en place, il est possible de réaliser des tests de connectivité entre conteneurs et d'optimiser les métriques.

Cette partie reprend le contenu de la section **Réseau d'hébergement de conteneurs** du support de travaux pratiques **Routage inter-VLAN et protocole PPPoE**.

Q32 **Quels sont les paquets à installer pour mettre en place un commutateur d'accès et le gestionnaire de conteneurs Incus ?**

Consulter la section **Réseau d'hébergement de conteneurs** du support **Routage inter-VLAN et protocole PPPoE**.

Note

Pour les routeurs R2 et R3, il faut rétablir l'accès temporaire à Internet pour installer les paquets demandés.

---

openvswitch-switch

Création d'un commutateur de raccordement des conteneurs et d'une interface virtuelle commutée

dnsmasq

Adressage automatique des conteneurs

incus

Gestion des conteneurs

```
sudo apt -y install openvswitch-switch dnsmasq incus
```

Q33 Quelles sont les modifications à apporter au fichier de déclaration YAML `/etc/netplan/enp0s1.yaml` pour créer le commutateur `asw-host` ?

Voici un extrait du fichier `/etc/netplan/enp0s1.yaml` avec les instructions de création du commutateur `asw-host`.

```
    openvswitch: {}

  bridges:
    asw-host:
      openvswitch: {}
```

Q34 Comment ajouter une nouvelle interface virtuelle commutée (*Switched Virtual Interface*) qui servira de passerelle par défaut pour tous les conteneurs hébergés ?

Rechercher dans la documentation Netplan des exemples de déclarations d'interfaces de type SVI appartenant à des VLANs.

Voici une copie complète du fichier `/etc/netplan/enp0s1.yaml` du routeur `R2` auquel on ajouté la déclaration d'une interface `vlan20` avec les adresses IPv4 et IPv6 conformes au contexte de la maquette utilisée pour la rédaction de ce document.

```
network:
    version: 2
    ethernets:
      enp0s1:
        dhcp4: false
        dhcp6: false
        accept-ra: false
        nameservers:
          addresses:
            - 172.16.0.2
            - 2001:db8:3::2

    openvswitch: {}

    bridges:
      asw-host:
        openvswitch: {}

    vlans:
#       # Temporary Internet access
#      enp0s1.52:
#        id: 52
#        link: enp0s1
#        dhcp4: true
#        dhcp6: false
#        accept-ra: true
      # R2 -> R1
      enp0s1.440:
        id: 440
        link: enp0s1
        addresses:
          - 10.44.0.2/29
          - fe80::1b8:2/64
      # R2 -> R3
      enp0s1.442:
        id: 442
        link: enp0s1
        addresses:
          - 10.44.2.2/29
          - fe80::1ba:2/64
      vlan20:
        id: 20
        link: asw-host
        addresses:
          - 10.20.0.1/24
          - fd14:ca46:3864:14::1/64
```

Avec ce fichier de déclaration l'interface d'accès temporaire est commentée et ne servira plus dans la suite des manipulations. À partir de la question suivante, tout le trafic des routeurs R2 et R3 transite par le routeur R1.

Q35 Comment publier le nouveau réseau d'hébergement via OSPF ?

Revoir la question Q : Q17 sur l'activation OSPF au niveau de chaque interface.

On active le protocole OSPF pour l'interface commutée virtuelle de chacun des trois routeurs. Voici un exemple pour le routeur R2.

```
R2# conf t
R2(config)# int vlan20
R2(config-if)# ip ospf area 0
R2(config-if)# ipv6 ospf6 area 0
R2(config-if)# ip ospf passive
R2(config-if)# ipv6 ospf6 passive
R2(config-if)# ^Z
```

Une fois cette opération réalisée, le routage dynamique est complet et le trafic des routeurs R2 et R3 est acheminé via R1. On peut lancer des tests ICMP.

```
ping -qc2 9.9.9.9
```

```
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.

--- 9.9.9.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 28.502/29.014/29.526/0.512 ms
```

```
ping -qc2 2620:fe::fe
```

```
PING 2620:fe::fe (2620:fe::fe) 56 data bytes

--- 2620:fe::fe ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 40.675/47.384/54.094/6.709 ms
```

Q36 Comment administrer et initialiser le gestionnaire de conteneurs **Incus** ?

Consulter la section **Réseau d'hébergement de conteneurs**.

Pour commencer, l'utilisateur normal doit appartenir aux deux groupes système `incus` et `incus-admin`. On reprend la démarche précédente.

```
for grp in incus incus-admin
do
    sudo adduser etu $grp
done
```

Comme l'affectation de groupe se joue à la connexion, il faut se déconnecter/reconnecter pour rendre l'attribution effective.

```
groups
```

```
etu adm sudo users frrvty frr incus-admin incus
```

Pour l'initialisation du gestionnaire de conteneur, on utilise un fichier de déclaration préparé en amont. Voici une copie du fichier `incus.yaml` pour le routeur R2.

```
config: {}
networks: []
storage_pools:
- config: {}
  description: ""
  name: default
  driver: dir
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: bridged
      parent: asw-host
      type: nic
      vlan: 20
    root:
      path: /
      pool: default
      type: disk
  name: default
projects: []
cluster: null
```

L'initialisation se fait à partir de ce fichier.

```
cat incus.yaml | incus admin init --preseed
```

On peut afficher le résultat de cette initialisation du gestionnaire de conteneurs avec l'instruction suivante.

```
incus profile show default
```

En cas d'erreur, il est aussi possible d'éditer le profil par défaut.

```
incus profile edit default
```

Important

Les opérations de cette question doivent être réalisées sur les trois routeurs en adaptant les numéros de VLAN de réseau d'hébergement.

Q37 Comment mettre en place l'adressage automatique IPv4 et IPv6 dans le réseau d'hébergement ?

Consulter la section **Adressage automatique dans le réseau d'hébergement** du document **Routage inter-VLAN et protocole PPPoE**

On créé un fichier de configuration pour le service dnsmasq en l'adaptant au contexte du plan d'adressage de chaque routeur. Voici un exemple pour le routeur R1 de la maquette.

```
cat << EOF | sudo tee /etc/dnsmasq.conf
# Specify Container VLAN interface
interface=vlan10

# Enable DHCPv4 on Container VLAN
dhcp-range=10.10.0.20,10.10.0.200,3h

# Enable IPv6 router advertisements
enable-ra

# Enable SLAAC
dhcp-range=::,constructor:vlan10,ra-names,slaac

# Optional: Specify DNS servers
dhcp-option=option:dns-server,172.16.0.2,9.9.9.9
dhcp-option=option6:dns-server,[2001:db8:3::2],[2620:fe::fe]

# Avoid DNS listen port conflict between dnsmasq and systemd-resolved
port=0
EOF
```

Une fois le fichier de configuration en place, on peut relancer le service et afficher son état.

```
sudo systemctl restart dnsmasq
systemctl status dnsmasq
```

Important

Une fois encore, les opérations de cette question doivent être réalisées sur les trois routeurs en adaptant le nom de l'interface et les adresses IPv4 du réseau d'hébergement.

Q38 Comment créer 3 conteneurs avec un adressage ?

Consulter la section **Configuration et lancement des conteneurs** du document **Routage inter-VLAN et protocole PPPoE**.

On lance la création des 3 conteneurs demandés avec une boucle.

```
for i in {0..2}; do incus launch images:debian/trixie c$i; done
```

```
incus ls
```

```
+------+---------+--------------------+-----------------------------------------------+-----------+-----------+
| NAME |  STATE  |        IPV4        |                     IPV6                      |   TYPE    | SNAPSHOTS |
+------+---------+--------------------+-----------------------------------------------+-----------+-----------+
| c0   | RUNNING | 10.20.0.33 (eth0)  | fd14:ca46:3864:14:216:3eff:fe2c:c5b9 (eth0) | CONTAINER | 0         |
+------+---------+--------------------+-----------------------------------------------+-----------+-----------+
| c1   | RUNNING | 10.20.0.87 (eth0)  | fd14:ca46:3864:14:216:3eff:fe76:f0d0 (eth0) | CONTAINER | 0         |
+------+---------+--------------------+-----------------------------------------------+-----------+-----------+
| c2   | RUNNING | 10.20.0.153 (eth0) | fd14:ca46:3864:14:216:3eff:fe2c:d169 (eth0) | CONTAINER | 0         |
+------+---------+--------------------+-----------------------------------------------+-----------+-----------+
```

Q39 Comment vérifier la publication des réseaux de conteneurs dans l'aire OSPF ?

On vérifie que l'interface commutée virtuelle (***Switched Virtual Interface***)est bien active pour chacune des deux versions du protocole de routage dynamique.

Voici un exemple de configuration pour le routeur R2.

```
R2# sh ip ospf interface vlan20
```

```
vlan20 is up
  ifindex 6, MTU 1500 bytes, BW 0 Mbit <UP,LOWER_UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.20.0.1/24, Broadcast 10.20.0.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 2.0.0.4, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 2.0.0.4 Interface Address 10.20.0.1/24
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    No Hellos (Passive interface)
  Neighbor Count is 0, Adjacent neighbor count is 0
  Graceful Restart hello delay: 10s
```

```
R2# sh ipv6 ospf6 interface vlan20
```

```
vlan20 is up, type BROADCAST
  Interface ID: 6
  Internet Address:
    inet : 10.20.0.1/24
    inet6: fd14:ca46:3864:14::1/64
    inet6: fe80::2865:4aff:fe35:fdd0/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 10
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
   No Hellos (Passive interface)
  DR: 2.0.0.4 BDR: 0.0.0.0
  Number of I/F scoped LSAs is 1
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
  Graceful Restart hello delay: 10s
  Authentication Trailer is disabled
```

Q40 Comment valider l'accès à l'Internet depuis les conteneurs ?

On reprend la liste des tests ICMP pour valider l'acheminement du trafic au niveau de la couche réseau et on lance une mise à jour du catalogue des paquets pour valider la résolution des noms.

À partir du routeur R2, on utilise les séquences suivantes :

- Tests ICMP au niveau réseau :

```
for i in {0..2}
do
    echo ">>>>>>>>>>>>>>>>> c$i"
    incus exec c$i -- ping -qc2 9.9.9.9
done
```

```
for i in {0..2}
do
    echo ">>>>>>>>>>>>>>>>> c$i"
    incus exec c$i -- ping -qc2 2620:fe::fe
done
```

- Tests au niveau application avec résolution des noms :

```
for i in {0..2}
do
    echo ">>>>>>>>>>>>>>>>> c$i"
    incus exec c$i -- apt update
    incus exec c$i -- apt -y full-upgrade
done
```

Q41 Comment vérifier que tous les réseaux de conteneurs sont bien visibles depuis les routeurs de la topologie triangle ?

Afficher les tables de routage IPv4 et IPv6 de l'un des trois routeurs et vérifier la présence des trois réseaux de conteneurs.

Voici le résultat obtenu depuis la console du routeur R3. On identifie les réseaux d'hébergement des routeurs R1 et R2 pour lesquels on trouve des entrées marquées O>*.

```
R3# sh ip route
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

O>* 0.0.0.0/0 [110/10] via 10.44.1.1, enp0s1.441, weight 1, 02:12:47
O>* 10.10.0.0/24 [110/20] via 10.44.1.1, enp0s1.441, weight 1, 02:12:48
O>* 10.20.0.0/24 [110/20] via 10.44.2.2, enp0s1.442, weight 1, 00:11:37
O   10.30.0.0/24 [110/10] is directly connected, vlan30, weight 1, 02:12:36
C>* 10.30.0.0/24 is directly connected, vlan30, 02:12:54
L>* 10.30.0.1/32 is directly connected, vlan30, 02:12:54
O>* 10.44.0.0/29 [110/20] via 10.44.1.1, enp0s1.441, weight 1, 00:11:37
  *                       via 10.44.2.2, enp0s1.442, weight 1, 00:11:37
O   10.44.1.0/29 [110/10] is directly connected, enp0s1.441, weight 1, 02:12:55
C>* 10.44.1.0/29 is directly connected, enp0s1.441, 02:12:55
L>* 10.44.1.3/32 is directly connected, enp0s1.441, 02:12:55
O   10.44.2.0/29 [110/10] is directly connected, enp0s1.442, weight 1, 01:18:48
C>* 10.44.2.0/29 is directly connected, enp0s1.442, 02:12:55
L>* 10.44.2.3/32 is directly connected, enp0s1.442, 02:12:55
```

```
R3# sh ipv6 route
```

```
Codes: K - kernel route, C - connected, L - local, S - static,
       R - RIPng, O - OSPFv3, I - IS-IS, B - BGP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric, t - Table-Direct,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

O>* ::/0 [110/10] via fe80::1b9:1, enp0s1.441, weight 1, 00:14:55
O>* fd14:ca46:3864:a::/64 [110/20] via fe80::1b9:1, enp0s1.441, weight 1, 02:21:04
O>* fd14:ca46:3864:14::/64 [110/20] via fe80::baad:caff:fefe:6, enp0s1.442, weight 1, 00:14:56
O   fd14:ca46:3864:1e::/64 [110/10] is directly connected, vlan30, weight 1, 02:14:55
C>* fd14:ca46:3864:1e::/64 is directly connected, vlan30, 02:15:27
L>* fd14:ca46:3864:1e::1/128 is directly connected, vlan30, 02:15:27
C * fe80::/64 is directly connected, vlan30, 02:15:28
C * fe80::/64 is directly connected, asw-host, 02:15:29
C * fe80::/64 is directly connected, enp0s1.52, 02:19:13
C * fe80::/64 is directly connected, enp0s1.441, 22:24:07
C * fe80::/64 is directly connected, enp0s1.442, 22:24:08
C>* fe80::/64 is directly connected, enp0s1, 22:24:08
```

Si on reprend, la même démarche sur chaque routeur, on doit trouver deux entrées OSPF relatives aux réseaux d'hébergement raccordés aux deux “sommets du triangle” dans les tables de routage.

10. Adapter de la métrique de lien au débit

Par défaut, le calcul de métrique du le protocole OSPF se fait à partir d'un coût fixe de 10. Dans cette section, on revient au calcul du quotient entre le débit binaire de référence et le débit binaire de l'interface.

À l'origine, le calcul de ce quotient utilise la formule 10^8 / bande passante du lien.

Cette règle a été établie à une époque où l'utilisation d'un lien à 100Mbps était considéré comme une situation futuriste. Aujourd'hui, les liens à 100Mbps sont dépassés.

> **Important**
>
> Pour que les calculs de métriques soient cohérents sur l'ensemble de la topologie triangle, il est essentiel d'appliquer les modifications de configuration sur les trois routeurs.

Q42 Quelle est l'instruction à utiliser pour que le calcul de métrique OSPF se fasse sur la base d'un débit de lien à 40Gbps ?

Rechercher le mot clé *reference* dans l'index des instructions de configuration. Consulter la documentation **FRRouting User Guide**.

C'est l'instruction **auto-cost reference-bandwidth** qui permet de fixer une nouvelle référence de coût de lien en Mbps.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# auto-cost reference-bandwidth ?
  (1-4294967)  The reference bandwidth in terms of Mbits per second
R1(config-router)# auto-cost reference-bandwidth 400000
R1(config-router)# ^Z
```

```
R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# auto-cost reference-bandwidth ?
  (1-4294967)  The reference bandwidth in terms of Mbits per second
R1(config-ospf6)# auto-cost reference-bandwidth 400000
R1(config-ospf6)# ^Z
```

En appliquant les mêmes instructions sur les trois routeurs de la topologie, on fixe le numérateur du quotient de calcul des métriques de routes.

Q43 Comment modifier le débit binaire d'un lien à 10Gbps ?

Note

Sur une machine physique avec des composants réseau matériels, le débit d'un lien est directement extrait des paramètres du composant de l'interface connectée au lien.

Dans le cas d'interfaces qui n'ont aucune réalité physique, ce débit peut être attribué arbitrairement par configuration. On doit rechercher dans les options de la console vtysh le moyen d'attribuer une indication de débit aux sous-interfaces de VLANs.

En affichant les paramètres des interfaces des routeurs virtuels, on constate que le paramètre speed est à zéro.

```
R1# sh interface vlan10
```

```
Interface vlan10 is up, line protocol is up
  Link ups:       1    last: 2024/11/03 14:26:00.66
  Link downs:     2    last: 2024/11/03 14:26:00.65
  vrf: default
  index 8 metric 0 mtu 1500 speed 0 txqlen 1000
  flags: <UP,LOWER_UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: da:35:df:7e:5d:42
  inet 10.10.0.1/24
  inet6 fd14:ca46:3864:a::1/64
  inet6 fe80::d835:dfff:fe7e:5d42/64
  Interface Type Other
  Interface Slave Type None
  protodown: off
```

Pour assurer un calcul correct de métrique, on doit fixer manuellement le débit binaire des interfaces dans la configuration frr.

Q44 Comment peut-on identifier le débit d'un lien dans la configuration OSPF ?

Visualiser les paramètres des interfaces réseau depuis la console vtysh. Les commandes suivantes peuvent être lancées sur chacun des trois routeurs.

show ip ospf interface
show ipv6 ospf6 interface

Voici le résultat obtenu sur le routeur R1.

```
R1# conf t
R3(config)# int vlan10
R3(config-if)# bandwidth 10000
R3(config-if)# ^Z
```

```
R1# sh ip ospf interface vlan10
```

```
vlan10 is up
  ifindex 8, MTU 1500 bytes, BW 10000 Mbit <UP,LOWER_UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.10.0.1/24, Broadcast 10.10.0.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 1.0.0.4, Network Type BROADCAST, Cost: 4
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.0.0.4 Interface Address 10.10.0.1/24
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    No Hellos (Passive interface)
  Neighbor Count is 0, Adjacent neighbor count is 0
  Graceful Restart hello delay: 10s
```

Q45 Quel est le coût d'accès au réseau de conteneurs attaché au routeur R3 depuis le routeur R2 ? Justifier la valeur de métrique obtenue.

À partir des informations de routage sur R2, faire la somme des métriques de chaque lien entre les deux extrémités en communication.

Il est possible d'obtenir les informations de calcul de métrique à partir de l'affichage d'une route particulière à la console vtysh.

```
R1# sh ip route 10.20.0.0/24
```

```
Routing entry for 10.20.0.0/24
  Known via "ospf", distance 110, metric 8, best
  Last update 00:21:32 ago
  * 10.44.0.2, via enp0s1.440, weight 1
```

Dans l'exemple ci-dessus, la métrique d'accès au réseau 10.30.0.0/24 correspond à la somme des métriques de deux liens à 10Gbps. La somme donne une métrique de 40/10 + 40/10 = 8.

On obtient un résultat identique avec la table de routage IPv6.

```
R1# sh ipv6 route fd14:ca46:3864:1e::/64
```

```
Routing entry for fd14:ca46:3864:1e::/64
  Known via "ospf6", distance 110, metric 8, best
  Last update 00:25:44 ago
  * fe80::1b9:3, via enp0s1.441, weight 1
```

11. Sauvegarder les fichiers de configuration

Les commandes utiles pour l'affichage et la sauvegarde sont les suivantes.

show run
copy run start

Une fois sorti de la console frr, la configuration unifiée des démons est stockée dans le fichier /etc/frr/frr.conf.

Voici une copie des fichiers de configuration des trois routeurs une fois toutes les manipulations réalisées.

- Routeur R1.

```
frr version 10.1.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.440
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface enp0s1.441
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface vlan10
 bandwidth 10000
 ip ospf area 0
 ip ospf passive
 ipv6 ospf6 area 0
 ipv6 ospf6 passive
exit
!
router ospf
 ospf router-id 1.0.0.4
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
 default-information originate
exit
!
router ospf6
 ospf6 router-id 1.0.0.6
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
 default-information originate
exit
!
end
```

- Routeur R2.

```
frr version 10.1.1
frr defaults traditional
hostname R2
log syslog informational
service integrated-vtysh-config
```

```
!
interface enp0s1.440
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface enp0s1.442
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface vlan20
 bandwidth 10000
 ip ospf area 0
 ip ospf passive
 ipv6 ospf6 area 0
 ipv6 ospf6 passive
exit
!
router ospf
 ospf router-id 2.0.0.4
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
exit
!
router ospf6
 ospf6 router-id 2.0.0.4
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
exit
!
end
```

- Routeur R3.

```
frr version 10.1.1
frr defaults traditional
hostname R3
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.441
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface enp0s1.442
 bandwidth 10000
 ip ospf area 0
 ipv6 ospf6 area 0
exit
!
interface vlan30
 bandwidth 10000
 ip ospf area 0
 ip ospf passive
 ipv6 ospf6 area 0
 ipv6 ospf6 passive
exit
!
router ospf
 ospf router-id 3.0.0.4
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
exit
!
router ospf6
 ospf6 router-id 3.0.0.6
 log-adjacency-changes detail
 auto-cost reference-bandwidth 40000
exit
!
end
```

12. Pour conclure...

Ce document fournit une introduction détaillée à la configuration du protocole de routage dynamique OSPF avec FRRouting sur des routeurs virtuels Linux. Il couvre la préparation des systèmes, l'installation et l'activation des démons OSPF, ainsi que la validation des communications entre routeurs.

Les manipulations proposées permettent de mettre en place une topologie réseau en triangle et de configurer pas à pas le routage OSPF pour IPv4 et IPv6. Ce guide pratique offre une base solide pour comprendre et implémenter le routage dynamique OSPF dans un environnement virtuel.