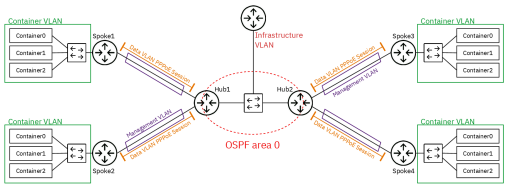
Synthèse sur l'interconnexion LAN/WAN

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé



L'objectif de ce dernier document de la série de travaux pratiques est de faire la synthèse sur l'interconnexion de réseaux locaux (LAN) et de réseaux étendus (WAN). Côté réseaux étendus, on retrouve les sessions PPPoE vers chaque site distant avec son réseau d'extrémité avec un l'hébergement de services représentés par les conteneurs Incus. Côté réseaux locaux, les routeurs *Hub* échangent leurs routes avec le protocole de routage dynamique OSPF.
Ces routeurs constituent ainsi un réseau de “collecte”. Que l'on soit dans le domaine LAN ou WAN, on fait un usage massif des VLANs.

Table des matières

1. Copyright et Licence

Méta-information

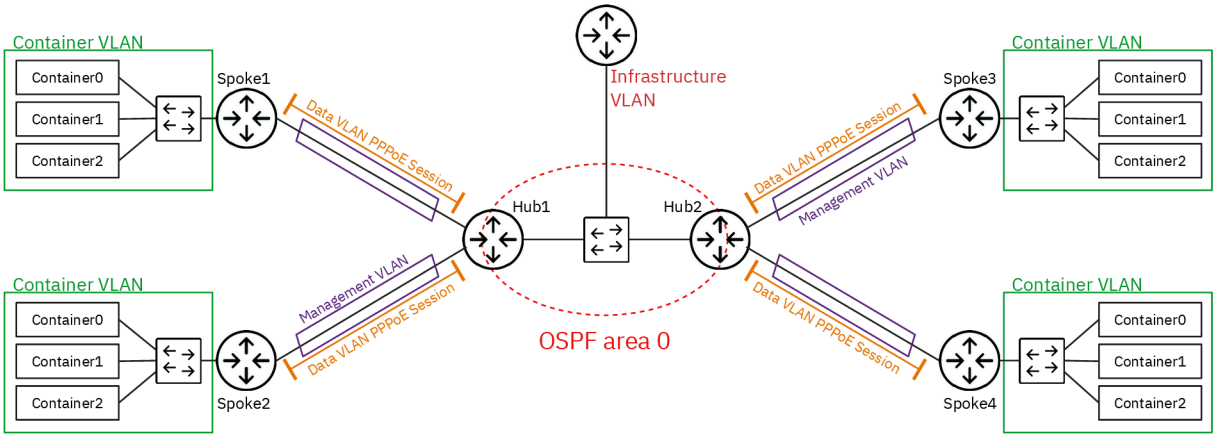
Ce document est écrit avec *DocBook* XML sur un système *Debian GNU/Linux*. Il est disponible en version imprimable au format PDF : `synthese.pdf`.

2. Topologie réseau & plan d'adressage

La topologie logique globale se présente comme une association de deux topologies triangulaires ***Hub & Spoke*** dans laquelle les routeurs ***Hub*** échangent leurs routes via le protocole de routage dynamique OSPF et fournissent l'accès Internet aux sites distants WAN.Elle couvre les aspects suivants :

- Topologie ***Hub & Spoke*** avec PPPoE pour l'interconnexion WAN
- Filtrage réseau avec netfilter/nftables
- Routage dynamique avec OSPF

L'objectif est de mettre en place une architecture réseau complète et sécurisée, combinant des techniques d'interconnexion WAN, de filtrage et de routage dynamique.



Routeur Hub

Le routeur ***Hub*** assure l'interconnexion entre les sites distants et le réseau de collecte. Il gère les sessions PPPoE avec les routeurs ***Spoke*** et annonce les routes vers les réseaux distants via le protocole OSPF aux autres routeurs du réseau de collecte.

> Note
>
> Les routeurs ***Hub*** partagent la même passerelle vers l'Internet.

Routeur Spoke

Le routeur ***Spoke*** assure l'interconnexion entre son réseau d'extrémité représenté par les conteneurs et le routeur ***Hub*** auprès duquel il s'authentifie pour ouvrir une session PPPoE qui lui permet de joindre tous les autres réseaux dont l'Internet.

Voici un exemple de plan d'adressage associé à la topologie ci-dessus. Il est utilisé pour la maquette de démonstration du fonctionnement de l'interconnexion réseau.

Tableau 1. Plan d'adressage de la maquette

Rôle	VLAN	Type	Liaison	Adresse/Authentification
Hub1	120	hosting	-> Internet	`172.20.120.2/23` `2001:678:3fc:78::2/64`
	470	collecte	`OSPFv2 id 1.0.0.4` `OSPFv3 id 1.0.0.6`	`172.20.70.1/29`
	471	mgmt lien local	-> Spoke1	`fe80:1d7::1/64`
	472	data point à point	-> Spoke1	`10.47.2.1:10.47.2.2`
	473	mgmt lien local	-> Spoke2	`fe80:1d9::1/64`
	474	data point à point	-> Spoke2	`10.47.4.1:10.47.4.2`
Spoke1	471	mgmt lien local	-> Hub1	`fe80:1d7::2/64`
	472	authentification	-> Hub1	`spoke1 / 0r4ng3.1`
	1	conteneurs	-	`10.0.1.1/24` `fda0:7a62:1::1/64`
Spoke2	473	mgmt lien local	-> Hub1	`fe80:1d9::2/64`
	474	authentification	-> Hub1	`spoke2 / 0r4ng3.2`
	2	conteneurs	-	`10.0.2.1/24` `fda0:7a62:2::1/64`
Hub2	120	hosting	-> Internet	`172.20.120.3/23` `2001:678:3fc:78::3/64`
	470	collecte	`OSPFv2 id 2.0.0.4` `OSPFv3 id 2.0.0.6`	`172.20.70.2/29`
	475	mgmt lien local	-> Spoke3	`fe80:1db::1/64`
	476	data point à point	-> Spoke3	`10.47.6.1:10.47.6.2`
	477	mgmt lien local	-> Spoke4	`fe80:1dd::1/64`
	478	data point à point	-> Spoke4	`10.47.8.1:10.47.8.2`
Spoke3	475	mgmt lien local	-> Hub2	`fe80:1db::2/64`
	476	authentification	-> Hub2	`spoke3 / 0r4ng3.3`
	3	conteneurs	-	`10.0.3.1/24` `fda0:7a62:3::1/64`
Spoke4	477	mgmt lien local	-> Hub2	`fe80:1dd::2/64`
	478	authentification	-> Hub2	`spoke4 / 0r4ng3.4`
	4	conteneurs	-	`10.0.4.1/24` `fda0:7a62:4::1/64`

3. Configurer les Routeurs Hub

Dans cette section, on s'intéresse aux rôles et fonctionnalités suivantes :

- Gérer les sessions PPPoE avec les *Spoke*
- Filtrer et traduire les adresses sources (NAT) pour le trafic sortant vers l'Internet
- Annoncer les routes vers les réseaux distants via OSPF

3.1. Configurer les serveurs PPPoE

Les tâches à réaliser sont :

- Activer le routage
- Activer la traduction des adresses sources vers l'Internet
- Installer le paquet pppoe
- Paramétrer le fichier `/etc/ppp/pppoe-server-options`.
- Créer les unités systemd pour le lancement des serveurs PPPoE

Voici un extrait des questions du support de travaux pratiques *Topologie Hub & Spoke avec le protocole PPPoE*.

Q1. 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. Par exemple :

```
sudo sysctl -a -r ".*forward.*"
```

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/10-routing.conf`.

```
cat << EOF | sudo tee /etc/sysctl.d/10-routing.conf
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians=1
EOF
```

Attention ! N'oubliez pas d'appliquer les nouvelles valeurs des paramètres de configuration.

```
sudo sysctl --system
```

Q2. Comment assurer la traduction d'adresses sources pour tous les flux réseaux sortants sur le réseau d'infrastructure (VLAN rouge) ?

Rechercher dans des exemples de configuration `nftables` avec la fonction `MASQUERADE`.

Voici un exemple de création du fichier `/etc/nftables.conf` avec le jeu d'instructions qui assure la traduction d'adresses sources pour IPv4 et IPv6.

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

flush ruleset

# Define variables
define RED_VLAN = enp0s1.360

table inet nat {
    chain postrouting {
        type nat hook postrouting priority 100;
        oifname $RED_VLAN counter packets 0 bytes 0 masquerade
    }
}
EOF
```

Avertissement

Il faut impérativement changer le nom d'interface en utilisant le numéro de VLAN attribué dans le plan d'adressage des travaux pratiques.

La création de ce fichier de règles n'est pas suffisante. Il faut appliquer les règles contenues dans le fichier.

```
sudo nft -f /etc/nftables.conf
```

Il faut aussi activer ce service pour assurer le chargement automatique des règles de filtrage au démarrage.

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

Q3. Quel paquet spécifique à la gestion du dialogue PPPoE à installer sur le routeur ***Hub*** ?

Rechercher dans le catalogue des paquets, la référence pppoe.

```
apt search ^pppoe
```

Le résultat de la commande apt show pppoe montre que c'est bien le paquet `pppoe` qui répond au besoin. On peut donc l'installer.

```
sudo apt -y install pppoe
```

Q4. Dans quel fichier sont stockés les paramètres d'identité et d'authentification utilisés par le protocole EAP pour la méthode CHAP ?

Consulter les pages de manuels du démon pppd à la section ***AUTHENTICATION***.

C'est le fichier `/etc/ppp/chap-secrets` qui contient les couples ***login/password*** utilisés lors de l'authentification.

Voici un exemple du contenu de ce fichier.

```
# Secrets for authentication using CHAP
# client     server secret   IP addresses
"spoke_site1" *  "0r4ng3_1"  *
"spoke_site2" *  "0r4ng3_2"  *
```

Q5. Dans quel fichier sont stockés les paramètres passés au démon pppd lors du lancement du serveur PPPoE ?

Consulter les pages de manuels de l'outil pppoe-server.

C'est le fichier `/etc/ppp/pppoe-server-options` qui contient la liste des paramètres utilisés lors du dialogue PPP.

Ce fichier contient tous les paramètres communs aux deux démons `pppd` qui sont lancés via pppoe-server. Voici comment créer le fichier.

```
cat << 'EOF' | sudo tee /etc/ppp/pppoe-server-options
# Gestion de session avec PAM
login
# Authentification EAP
require-eap
# Le Routeur Hub détient déjà une route par défaut
nodefaultroute
# Envoi de l'adresse de résolution DNS avec les adresses IPv4
ms-dns 172.16.0.2
# Ajout du protocole IPv6
+ipv6
# Informations détaillées dans la journalisation
debug
# Options préconisées par la documentation
noaccomp
default-asyncmap
nodeflate
nopcomp
novj
novjccomp
lcp-echo-interval 10
EOF
```

Q6. Comment créer les comptes utilisateurs locaux sur le routeur ***Hub*** sachant qu'ils ne sont autorisés ni à se connecter ni à avoir un répertoire personnel ?

Consulter les options de la commande adduser.

Voici un exemple dans le contexte de la maquette.

```
sudo adduser --gecos 'Spoke 1' --disabled-login --no-create-home spoke_site1
```

```
sudo adduser --gecos 'Spoke 2' --disabled-login --no-create-home spoke_site2
```

Q7. Quel paramètre supplémentaire doit être ajouté à la configuration de la commande pppoe-server pour distinguer les échanges entre les deux routeurs ***Spoke*** ?

Relativement au support *Routage inter-VLAN et protocole PPPoE*, il est essentiel de définir correctement les routes statiques vers les réseaux d'extrémité de chaque routeur ***Spoke***.

Consulter les options de la commande pppoe-server.

L'option `-u` permet de désigner une “unité“ qui sert à nommer l'interface. Par exemple, `-u 0` correspond à l'interface `ppp0`.

Q8. Comment créer deux nouvelles unités systemd responsables du lancement des processus pppoe-server ?

Consulter la page *systemd Services* et rechercher la procédure à suivre pour ajouter un service au lancement du système.

On commence par la création du fichier de service appelé : `/etc/systemd/system/pppoe-server.service` qui contient toutes les directives de lancement du processus pppoe-server avec les paramètres d'adressage du lien point à point.

Voici un exemple de création du fichier d'unité systemd pour le premier service.

```
cat << 'EOF' | sudo tee /etc/systemd/system/pppoe-server1.service
[Unit]
Description=PPPoE Server
After=systemd-networkd.service
Wants=systemd-networkd.service
BindsTo=sys-subsystem-net-devices-enp0s1.441.device
After=sys-subsystem-net-devices-enp0s1.441.device

[Service]
Type=forking
ExecCondition=/bin/sh -c '[ "$(systemctl show --property MainPID --value pppoe-server1.service)"
ExecStart=/usr/sbin/pppoe-server -I enp0s1.441 -C BRAS -L 10.44.1.1 -R 10.44.1.2 -N 1 -u 0
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF
```

Voici un exemple de création du fichier d'unité systemd pour le second service.

```
cat << 'EOF' | sudo tee /etc/systemd/system/pppoe-server2.service
[Unit]
Description=PPPoE Server
After=systemd-networkd.service
Wants=systemd-networkd.service
BindsTo=sys-subsystem-net-devices-enp0s1.443.device
After=sys-subsystem-net-devices-enp0s1.443.device

[Service]
Type=forking
ExecCondition=/bin/sh -c '[ "$(systemctl show --property MainPID --value pppoe-server2.service)"
ExecStart=/usr/sbin/pppoe-server -I enp0s1.443 -C BRAS -L 10.44.3.1 -R 10.44.3.2 -N 1 -u 1
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF
```

Q9. Comment activer les deux nouveaux services et contrôler leur état après lancement ?

Consulter la page *systemd Services* et rechercher la procédure à suivre pour activer et lancer un service.

On commence par la relecture de la liste des services disponibles par le gestionnaire systemd.

```
sudo systemctl daemon-reload
```

On active les nouveaux services.

```
for i in {1..2}; do sudo systemctl enable pppoe-server$i.service; done
```

On lance ce nouveau service.

```
for i in {1..2}; do sudo systemctl start pppoe-server$i.service; done
```

On vérifie que l'opération s'est déroulée correctement.

```
for i in {1..2}; do systemctl status pppoe-server$i.service; done
```

En l'état actuel de la configuration, aucune session PPP n'a encore été établie. Il faut maintenant passer à la configuration réseau du routeur ***Spoke*** pour avancer dans l'utilisation du protocole PPP.

3.2. Installer et configurer FRR

Les tâches à réaliser sont :

- Installer et préparer les services FRRouting

- Activer les démons des protocoles OSPFv2 pour IPv4 et OSPFv3 pour IPv6

Voici un extrait des questions du support de travaux pratiques *Introduction au routage dynamique OSPF avec FRRouting*.

Q10. 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 bookworm 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.
```

Q11. 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
```

Q12. 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
```

Q13. 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
```

Une fois la partie configuration initiale traitée, passons à la maquette de la synthèse.

Q14. Quel est l'état courant du routage à ce stade de la configuration ?

Afficher les tables de routage des routeurs *Hub* avec les liens PPP actifs. Les échanges via le protocole OSPF ne sont pas encore configurés.

Dans le contexte de la maquette, l'affichage des deux tables de routage du routeur *Hub numéro 1* montre des entrées codées avec les clés suivantes.

```
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 172.20.120.1, enp0s1.120, 00:00:36❶
L>* 10.47.2.1/32 is directly connected, ppp0, 00:00:36❷
C>* 10.47.2.2/32 is directly connected, ppp0, 00:00:36
L>* 10.47.4.1/32 is directly connected, ppp1, 00:00:36❸
C>* 10.47.4.2/32 is directly connected, ppp1, 00:00:36
C>* 172.20.70.0/29 is directly connected, enp0s1.470, 00:00:36❹
L>* 172.20.70.1/32 is directly connected, enp0s1.470, 00:00:36
C>* 172.20.120.0/23 is directly connected, enp0s1.120, 00:00:36❺
L>* 172.20.120.2/32 is directly connected, enp0s1.120, 00:00:36
```

```
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 fe80:78::1, enp0s1.120 onlink, 00:31:49❶
K>* 2001:678:3fc:78::/64 [0/512] is directly connected, enp0s1.120, 00:31:49
L>* 2001:678:3fc:78:baad:caff:fefe:5/128 is directly connected, enp0s1.120, 00:31:49
C * fe80::/64 is directly connected, enp0s1.120, 00:31:49
C * fe80::/64 is directly connected, enp0s1.474, 00:31:49
C * fe80::/64 is directly connected, enp0s1.471, 00:31:49
C * fe80::/64 is directly connected, enp0s1.472, 00:31:49
C * fe80::/64 is directly connected, enp0s1.473, 00:31:49
C * fe80::/64 is directly connected, enp0s1.470, 00:31:49
C>* fe80::/64 is directly connected, enp0s1, 00:31:49
C>* fe80::7188:100d:7562:f6d/128 is directly connected, ppp0, 00:31:49❷
C>* fe80::c00f:a3d0:f2c5:b154/128 is directly connected, ppp1, 00:31:49❸
C>* fe80:1d6::/64 is directly connected, enp0s1.470, 00:31:49❹
C>* fe80:1d7::/64 is directly connected, enp0s1.471, 00:31:49
C>* fe80:1d9::/64 is directly connected, enp0s1.473, 00:31:49
```

❶❶ La route par défaut est marquée `K>*` (***kernel***) parce qu'elle est importée de la configuration système. Elle n'est pas gérée par un service FRR.

❷❸❺ Avec IPv4, toutes les entrées des interfaces actives sont marquées `C>*` (***connected***). Le service FRR a importé les entrées du système et peut alimenter les démons de protocole de routage dynamique avec ces informations.

❹❹ Le nouveau réseau de collecte qui va servir aux échanges OSPF doit être présent dès maintenant avant d'activer le protocole de routage dynamique sur les interfaces.

❷❸❹ Avec IPv6, les réseaux d'interconnexion ou de transit n'utilisent que des adresses de lien local.

3.3. Configurer les démons OSPFv2 et OSPFv3

Les tâches à réaliser sont :

- Configurer les identifiants de routeur
- Activer OSPF sur les interfaces du réseau de collecte

Voici un jeu de questions extraites du support de travaux pratiques *Introduction au routage dynamique OSPF avec FRRouting*

Q15. 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. Voir Section 2, « Topologie réseau & plan 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.

Q16. 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>* 2001:678:3fc: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.

Q17. 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.

Q18. 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.

Q19. 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
```

Q20. 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
2.0.0.4           1 Full/Backup     11h34m21s         37.712s 10.44.0.2       enp0s1.440:10.44.0.
3.0.0.4           1 Full/Backup     11h35m30s         33.792s 10.44.1.3       enp0s1.441:10.44.1.
```

```
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.

Pour achever cette partie de la synthèse, il reste à étudier la redistribution des routes connectées dans les démons de routage OSPFv2 et OSPFv3.

Q21. Quel est l'état des voisins OSPF à ce stade de la configuration ?

Afficher la liste des voisins OSPFv2 OSPFv3 sur l'un des deux routeurs ***Hub***.

En se plaçant sur le second routeur ***Hub*** de la maquette, on obtient les informations suivantes.

```
sh ip ospf neighbor
```

```
Neighbor ID     Pri State     Up Time    Dead Time  Address      Interface              RXmtL Rqs
1.0.0.4           1 Full/DR   23.841s      36.142s  172.20.70.1  enp0s1.470:172.20.70.2     0
```

```
sh ipv6 ospf6 neighbor
```

```
Neighbor ID     Pri    DeadTime    State/IfState         Duration I/F[State]
1.0.0.6           1    00:00:30     Full/DR              00:00:29 enp0s1.470[BDR]
```

> Note
>
> En l'état actuel des échanges OSPF, l'affichage des tables de routage ne montre aucune route apprise via routage dynamique. Il est donc nécessaire d'importer les routes des liaisons PPPoE dans la configuration des démons OSPF.

3.4. Redistribuer les routes connectées dans OSPF

Les tâches à réaliser sont :

- Configurer la redistribution des routes connectées dans OSPF
- Valider la présence des routes IPv4 de transit vers les routeurs ***Spoke***

Q22. Quelle est l'instruction qui permet de redistribuer les routes connectées dans les démons OSPF ?

Rechercher les options de l'instruction redistribute dans la documentation FRR à l'adresse *FRRouting User Guide*.

L'instruction redistribute est générique et s'emploie dans la configuration de tous les protocoles de routage. Voici un extrait de configuration pour chacun des deux démons OSPF.

```
sh run ospfd
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname hub1
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.470
 ip ospf area 0
exit
!
router ospf
 ospf router-id 1.0.0.4
 log-adjacency-changes detail
 redistribute connected
exit
!
end
```

```
sh run ospf6d
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname hub1
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.470
 ipv6 ospf6 area 0
exit
!
router ospf6
 ospf6 router-id 1.0.0.6
 log-adjacency-changes detail
 redistribute connected
exit
!
end
```

Q23. Pourquoi choisir la redistribution des routes connectées ?

Il existe de nombreuses solutions pour intégrer des réseaux dans OSPF. Voici deux exemples :

- Activer OSPF sur les interfaces PPP
- Ajouter les adresses des réseaux de transit vers les routeurs ***Spoke***.

Cela s'explique par les principes de la topologie ***Hub & Spoke***. Un lien WAN sur lequel on établit une session PPPoE n'a pas de caractère permanent. Il faut donc trouver une solution qui permette au protocole de routage dynamique OSPF de fonctionner de façon optimale avec un nombre de routeurs ***Spoke*** variable.

Avec la maquette des travaux pratiques, chaque routeur ***Hub*** est relié à deux routeurs ***Spoke***. Dans un contexte plus réaliste, le nombre de routeurs ***Spoke*** peut varier en fonction de l'ouverture ou de la fermeture de sites distants, et le fonctionnement du réseau de collecte ne doit pas être impacté par ces variations.

C'est la raison pour laquelle les variations du nombre de routes connectées sont gérées au niveau système avec les services PPP. Les changements sont automatiquement répercutés dans la base d'information FRR puis publiés par le protocole de routage dynamique OSPF. La redistribution des routes connectées répond parfaitement à ce besoin.

Q24. Comment vérifier que la redistribution de route est effective ?

Rechercher les entrées OSPF dans les tables de routage des deux routeurs ***Hub***.

Voici ce que l'on observe sur le premier routeur de la maquette.

```
sh ip route ospf
```

```
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>* 10.47.6.2/32 [110/20] via 172.20.70.2, enp0s1.470, weight 1, 00:46:51
O>* 10.47.8.2/32 [110/20] via 172.20.70.2, enp0s1.470, weight 1, 00:46:51
O   172.20.70.0/29 [110/10] is directly connected, enp0s1.470, weight 1, 03:43:32
O   172.20.120.0/23 [110/20] via 172.20.70.2, enp0s1.470, weight 1, 00:46:51
```

Dans le cas du protocole IPv6, la redistribution n'est pas visible puisque les réseaux de transit utilisent uniquement des adresses de lien local pour former les adjacences entre routeurs. La commande ci-dessous ne produit aucune sortie. Il faut attendre la mise en place des réseaux

d'hébergement au delà des routeurs ***Spoke*** pour voir apparaître de nouvelles entrées dans la table de routage.

```
sh ipv6 route ospf
```

4. Configurer les routeurs Spoke

Les quatre routeurs ***Spoke*** de la topologie jouent toujours le rôle de routeur d'extrémité.

Pour les manipulations de configuration de ce rôle, on s'appuie sur le support de travaux pratiques : *Routage inter-VLAN et protocole PPPoE*.

Dans cette section, on s'intéresse aux rôles et fonctionnalités suivantes :

- Gérer les sessions PPP avec les routeurs ***Hub***
- Valider les communications entre routeurs ***Spoke***

4.1. Configurer les clients PPP

Les tâches à réaliser sont :

- Activer le routage
- Configurer le protocole PPP

Pour l'activation du routage, on retrouve les mêmes opérations que sur les routeurs ***Hub***.

Q25. 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. Par exemple :

```
sudo sysctl -a -r ".*forward.*"
```

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/10-routing.conf`.

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

```
cat << EOF | sudo tee /etc/sysctl.d/10-routing.conf
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians=1
EOF
```

Voici un exemple des résultats obtenus après application des nouveaux paramètres.

```
sudo sysctl --system
```

La configuration des “clients” PPP diffère de celle des routeurs ***Hub***.

Q26. Quel paquet fournit le démon de gestion des sessions du protocole PPP sur le routeur ***Spoke*** ?

Rechercher dans le catalogue des paquets, la référence ppp.

```
apt search ^ppp
```

```
ppp/testing 2.5.0-1+2 amd64
  protocole point à point (PPP) - démon

ppp-dev/testing 2.5.0-1+2 all
  protocole point à point (PPP) – fichiers de développement

ppp-gatekeeper/testing 0.1.0-201406111015-1.1 all
  PPP manager for handling balanced, redundant and failover links

pppoe/testing 4.0-1 amd64
  Pilote PPP sur Ethernet

pppoeconf/testing 1.21+nmu3 all
  configures PPPoE/ADSL connections

wmppp.app/testing 1.3.2-2 amd64
  contrôle de connexion et surveillance de la charge réseau avec aspect NeXTStep
```

Le résultat de la commande apt show ppp montre que c'est bien ce paquet qui répond au besoin.

```
sudo apt -y install ppp
```

Q27. Comment utiliser l'encapsulation des trames PPP dans Ethernet à partir du démon pppd fourni avec le paquet ppp ?

Rechercher dans le répertoire de documentation du paquet ppp.

Dans le répertoire `/usr/share/doc/ppp/`, on trouve le fichier `README.pppoe` qui indique que l'appel au module `rp-pppoe.so` permet d'encapsuler des trames PPP sur un réseau local Ethernet.

Toujours à partir du même répertoire, on trouve dans la liste des fichiers d'exemples de configuration un modèle adapté à notre contexte : `peers-pppoe`.

Q28. Dans quel fichier sont stockés les paramètres d'identité et d'authentification utilisés par le protocole CHAP ?

Consulter les pages de manuels du démon pppd à la section *AUTHENTICATION*.

C'est le fichier `/etc/ppp/chap-secrets` qui contient les couples *login/password* utilisés lors de l'authentification.

Voici un exemple du contenu de ce fichier. Le nom du client ainsi que son mot de passe secret doivent être identiques à chaque extrémité de la session PPP.

```
# Secrets for authentication using CHAP
# client server secret   IP addresses
"spoke_site0"   *  "5p0k3"   *
```

Q29. Quelles sont les options de configuration du démon pppd à placer dans le fichier `/etc/ppp/peers/pppoe-provider` pour assurer l'établissement de la session PPP entre les routeurs ?

Utiliser le fichier exemple PPPoE fourni avec la documentation du paquet ppp.

Voici comment créer un fichier `/etc/ppp/peers/pppoe-provider` avec les options correspondant au contexte de la maquette du routeur vert.

```
cat << 'EOF' | sudo tee /etc/ppp/peers/pppoe-provider
# Le nom d'utilisateur désigne l'entrée du fichier /etc/ppp/chap-secrets
user spoke_site0

# Chargement du module PPPoE avec les détails dans la journalisation
plugin rp-pppoe.so rp_pppoe_ac BRAS rp_pppoe_verbose 1

# Interface (VLAN) utilisé pour l'établissement de la session PPP
enp0s1.441

# Les adresses sont attribuées par le "serveur" PPPoE
noipdefault
# L'adresse de résolution DNS est aussi fournie par le serveur PPPoE
usepeerdns
# La session PPP devient la route par défaut du routeur Spoke
defaultroute

# Demande de réouverture de session automatique en cas de rupture
persist

# Le routeur Spoke n'exige pas que le routeur Hub s'authentifie
noauth

# Messages d'informations détaillés dans la journalisation
debug

# Utilisation du protocole IPv6
+ipv6

# Options préconisées par la documentation
noaccomp
default-asyncmap
nodeflate
nopcomp
novj
novjccomp
lcp-echo-interval 10
EOF
```

Q30. Comment lancer le démon pppd pour qu'il prenne en compte les paramètres définis dans le fichier complété à la question précédente ?

Consulter les pages de manuels du démon pppd.

C'est l'outil pon qui permet de désigner le fichier de configuration à utiliser. Voici une copie d'écran du lancement du démon pppd.

```
sudo pon pppoe-provider
```

Cette commande individuelle est à utiliser pour faire un tout premier test. Pour rendre la configuration persistante au redémarrage nous avons besoin de créer un service systemd. Il ne faut donc pas oublier d'arrêter le processus avant de passer à la question suivante. Le paquet fournit un outil dédié : poff.

```
sudo poff -a pppoe-provider
```

Q31. Quels sont les noms des deux sous-couches du protocole PPP qui apparaissent dans les journaux systèmes ? Quels sont les rôles respectifs de ces deux sous-couches ?

Consulter la page *Point-to-Point Protocol*.

La consultation des journaux système lors du dialogue PPP fait apparaître tous les détails. Voir la Section 4.1, « Configurer les clients PPP » [26].

Q32. Quels sont les en-têtes du dialogue qui identifient les requêtes (émises|reçues), les rejets et les acquittements ?

Consulter les journaux système contenant les traces d'une connexion PPP.

La copie d'écran donnée ci-dessus fait apparaître les directives `Conf*` pour chaque paramètre négocié.

- `ConfReq` indique une requête.
- `ConfAck` indique un acquittement.
- `ConfNak` indique un rejet.

Q33. Comment assurer une ouverture automatique de la session PPP à chaque réinitialisation système ?

Consulter la page *systemd Services* et rechercher la procédure à suivre pour ajouter un service au lancement du système.

On commence par la création du fichier de service appelé : `/etc/systemd/system/ppp.service` qui contient les appels aux outils pon et poff.

Voici l'instruction de création du fichier de service.

```
cat << EOF | sudo tee /etc/systemd/system/ppp.service
[Unit]
Description=PPPoE Client Connection
After=network.target
Wants=network.target
BindsTo=sys-subsystem-net-devices-enp0s1.441.device
After=sys-subsystem-net-devices-enp0s1.441.device

[Service]
Type=forking
ExecStart=/usr/bin/pon pppoe-provider
ExecStop=/usr/bin/poff pppoe-provider
Restart=on-failure
RestartSec=20

[Install]
WantedBy=multi-user.target
EOF
```

Q34. Comment activer le nouveau service et contrôler son état après lancement ?

Consulter la page *systemd Services* et rechercher la procédure à suivre pour activer et lancer un service.

On commence par la relecture de la liste des services disponibles par le gestionnaire systemd.

```
sudo systemctl daemon-reload
```

On active le nouveau service.

```
sudo systemctl enable ppp.service
```

On lance ce nouveau service.

```
sudo systemctl start ppp.service
```

On vérifie que l'opération s'est déroulée correctement.

```
systemctl status ppp.service
```

```
# ppp.service - PPPoE Client Connection
     Loaded: loaded (/etc/systemd/system/ppp.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-22 08:21:32 CEST; 13min ago
 Invocation: 69386a40f7574821b3986ed6c6c242f7
   Main PID: 496 (pppd)
      Tasks: 1 (limit: 1086)
     Memory: 2.8M (peak: 4.9M)
        CPU: 56ms
     CGroup: /system.slice/ppp.service
             └─496 /usr/sbin/pppd call pppoe-provider

sept. 22 08:21:37 spoke pppd[496]: rcvd [IPCP ConfAck id=0x2 <addr 10.4.41.2> <ms-dns1 172.16.0.2
sept. 22 08:21:37 spoke pppd[496]: Script /etc/ppp/ip-pre-up started (pid 510)
sept. 22 08:21:37 spoke pppd[496]: Script /etc/ppp/ip-pre-up finished (pid 510), status = 0x0
sept. 22 08:21:37 spoke pppd[496]: local  IP address 10.4.41.2
sept. 22 08:21:37 spoke pppd[496]: remote IP address 10.4.41.1
sept. 22 08:21:37 spoke pppd[496]: primary   DNS address 172.16.0.2
sept. 22 08:21:37 spoke pppd[496]: secondary DNS address 172.16.0.2
sept. 22 08:21:37 spoke pppd[496]: Script /etc/ppp/ip-up started (pid 514)
sept. 22 08:21:37 spoke pppd[496]: Script /etc/ppp/ipv6-up finished (pid 509), status = 0x0
sept. 22 08:21:37 spoke pppd[496]: Script /etc/ppp/ip-up finished (pid 514), status = 0x0
```

Q35. Comment utiliser la session PPP (le VLAN orange) comme lien unique de raccordement réseau du routeur ***Spoke*** ?

Maintenant que le fonctionnement de la session PPP est validé, nous n'avons plus besoin du raccordement temporaire sur le routeur ***Spoke***. Il faut donc commenter les entrées du fichier `/etc/netplan/enp0s1.yaml` qui ne sont plus utiles et attribuer l'adresse de résolution DNS de secours.

Une fois ces opérations effectuées, on peut redémarre le routeur ***Spoke*** pour se placer en situation de raccordement distant.

Pour commencer, on commente les entrées inutile du fichier `/etc/netplan/enp0s1.yaml`.

```
cat /etc/netplan/enp0s1.yaml
```

```
network:
  version: 2
  ethernets:
    enp0s1:
      dhcp4: false
      dhcp6: false
      accept-ra: false
#     nameservers:
#       addresses:
#          - 172.16.0.2
#          - 2001:678:3fc:3::2

  vlans:
    enp0s1.440: # VLAN violet
      id: 440
      link: enp0s1
      addresses:
        - fe80:1b8::2/64
    enp0s1.441: # VLAN orange
      id: 441
      link: enp0s1
      addresses: []
#   enp0s1.52: # VLAN accès temporaire
#     id: 52
#     link: enp0s1
#     dhcp4: true
#     dhcp6: false
#     accept-ra: true
```

On peut appliquer directement les modifications à l'aide de la commande netplan.

```
sudo netplan apply
```

```
sudo netplan status
```

Attention

L'affectation de l'adresse IPv4 ou IPv6 de résolution DNS pose problème. En effet, si le démon pppd propose bien deux adresses via l'option `usepeerdns`, ces propositions ne sont pas prises en charge par le service systemd-resolved.

On contourne cette difficulté en affectant une adresse IPv4 directement au service systemd-resolved.

On édite le fichier `/etc/systemd/resolved.conf` pour affecter directement l'adresse de résolution DNS. Voici une copie des lignes utiles du fichier modifié. Toutes les autres lignes sont commentées.

```
grep -Ev '(^#|^$)'  /etc/systemd/resolved.conf
[Resolve]
DNS=172.16.0.2
```

Il ne faut pas oublier de relancer le service pour prendre en compte les modifications du fichier.

```
sudo systemctl restart systemd-resolved
```

Le routeur ***Spoke*** est maintenant prêt à être redémarré pour utiliser le lien de raccordement distant comme seul canal d'accès aux autres réseaux.

```
sudo reboot
```

Pour visualiser les détails du fonctionnement du protocole point à point, voir la section « Traces d'une ouverture de session PPPoE » du document « *Routage inter-VLAN et protocole PPPoE* ».

4.2. Valider les communications

Après avoir vérifié que les quatre sessions PPP de la maquette sont actives, on peut lancer une série de tests ICMP suivant deux axes.

1. Accéder à l'Internet depuis chaque routeur ***Spoke*** avec IPv4.

   L'accès via IPv6 n'est pas possible sachant qu'aucune adresse de type ULA ou GUA n'est présente sur les routeurs ***Spoke***.

   ```
   etu@spoke4:~$ 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 = 28.778/29.342/29.907/0.564 ms
   ```

2. Accéder aux autres routeurs ***Spoke***.

   Le script ci-dessous peut être exécuté sur n'importe quel routeur ***Spoke*** et permet de vérifier qu'aucun paquet n'est perdu.

   ```
   for addr in "10.47.2.2" "10.47.4.2" "10.47.6.2" "10.47.8.2"
   do
       ping -qc2 $addr
   done
   ```

5. Ajouter les réseaux de services distants

Dans cette section, on s'intéresse aux rôles et fonctionnalités suivantes :

- Configurer un commutateur virtuel avec une interface commutée
- Router les réseaux d'hébergement via OSPF

5.1. Ajouter un commutateur virtuel

Voici la liste des questions traitées dans le document *Routage inter-VLAN et protocole PPPoE*

Q36. Quel est le paquet à installer pour ajouter un commutateur virtuel au routeur ***Spoke*** ?

Rechercher le mot clé openvswitch dans la liste des paquets.

Voici un exemple de recherche.

```
apt search ^openvswitch
```

C'est le paquet `openvswitch-switch` qui nous intéresse. On l'installe.

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

Q37. Comment déclarer un commutateur à l'aide de l'outil netplan.io ?

Consulter la documentation de ***Netplan*** pour obtenir les informations sur la configuration des commutateurs virtuels `openvswitch` à l'adresse *Netplan documentation*.

On peut aussi rechercher les informations dans les fichiers exemples fournis avec le paquet netplan.io.

Voici un exemple de recherche.

```
find /usr/share/doc/netplan* -type f -iname "openvswitch*"
/usr/share/doc/netplan/examples/openvswitch.yaml
```

Q38. Quelles sont les modifications à apporter au fichier de déclaration YAML `/etc/netplan/enp0s1.yaml` pour créer le commutateur `asw-host` ?

Voici une copie du fichier `/etc/netplan/enp0s1.yaml` qui contient les instructions de création du commutateur `asw-host` seul.

```
network:
  version: 2
  ethernets:
    enp0s1:
      dhcp4: false
      dhcp6: false
      accept-ra: false

    openvswitch: {}

  bridges:
    asw-host:
      openvswitch: {}

  vlans:
    enp0s1.440: # VLAN violet
      id: 440
      link: enp0s1
      addresses:
        - fe80:1b8::2/64
    enp0s1.441: # VLAN orange
      id: 441
      link: enp0s1
      addresses: []
```

On applique les nouvelles déclarations.

```
sudo netplan apply
```

On vérifie que le nouveau commutateur a bien été créé dans la base `Open vSwitch`.

```
sudo ovs-vsctl show
```

```
e288cc30-e290-44ae-8ed1-5e2a8d184033
    Bridge asw-host
        fail_mode: standalone
        Port asw-host
            Interface asw-host
                type: internal
    ovs_version: "3.4.0"
```

Q39. Comment ajouter une nouvelle interface virtuelle commutée (***Switched Virtual Interface***) qui servira de passerelle par défaut pour tous les hôtes du réseau d'hébergement du site distant ?

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

Voici une nouvelle copie du fichier `/etc/netplan/enp0s1.yaml` auquel on ajouté la déclaration d'une interface `vlan40` 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

  openvswitch: {}

  bridges:
    asw-host:
      openvswitch: {}

  vlans:
    enp0s1.440: # VLAN violet
      id: 440
      link: enp0s1
      addresses:
        - fe80:1b8::2/64
    enp0s1.441: # VLAN orange
      id: 441
      link: enp0s1
      addresses: []
    vlan40:     # VLAN vert
      id: 40
      link: asw-host
      addresses:
        - 203.0.113.1/24
        - fda0:7a62:28::1/64
        - fe80:28::1/64
```

5.2. Router les réseaux d'hébergement depuis les Hubs

La configuration mise en place dans cette section conditionne le routage des réseaux d'hébergement des services des sites distants à l'ouverture de session PPP.

Q40. Comment créer les routes statiques vers les réseaux d'hébergement sur le routeur ***Hub*** ?

Créer un script exécutable par protocole réseau dans lequel on utilise les noms d'interfaces PPP dérivés des options `-u` utilisées dans les paramètres de configuration des deux serveurs PPPoE.

Pour IPv4, c'est le répertoire est `/etc/ppp/ip-up.d/` qui doit contenir le script exécutable `staticroute`.

```
cat << 'EOF' | sudo tee /etc/ppp/ip-up.d/staticroute
#!/bin/bash

if [ -z "${CONNECT_TIME}" ]; then
    case "${PPP_IFACE}" in
        "ppp0")
            ip route add 10.0.10.0/24 dev ${PPP_IFACE}
            ;;
        "ppp1")
            ip route add 10.0.20.0/24 dev ${PPP_IFACE}
            ;;
    esac
fi
EOF
```

```
sudo chmod +x /etc/ppp/ip-up.d/staticroute
```

Pour IPv6, c'est le répertoire est `/etc/ppp/ipv6-up.d/` qui doit contenir le script exécutable appelé `staticroute`.

```
cat << 'EOF' | sudo tee /etc/ppp/ipv6-up.d/staticroute
#!/bin/bash

if [ -z "${CONNECT_TIME}" ]; then
    case "${PPP_IFACE}" in
        "ppp0")
            ip -6 route add fda0:7a62:a::/64 dev ${PPP_IFACE}
            ;;
        "ppp1")
            ip -6 route add fda0:7a62:14::/64 dev ${PPP_IFACE}
            ;;
    esac
fi
EOF
```

```
sudo chmod +x /etc/ppp/ipv6-up.d/staticroute
```

Une fois de plus, il faut relancer les sessions PPP pour observer les résultats et lancer les tests ICMP.

Auparavant, on peut afficher les tables de routages IPv4 et IPv6 du routeur ***Hub*** pour vérifier la présence des nouvelles routes statiques.

- En direction du premier routeur ***Spoke***.

```
ip route ls dev ppp0
```

```
10.0.10.0/24 scope link
10.44.1.2 proto kernel scope link src 10.44.1.1
```

```
ip -6 route ls dev ppp0
```

```
fda0:7a62:a::/64 metric 1024 pref medium
fe80::2c4a:3bb9:eead:8fa5 proto kernel metric 256 pref medium
fe80::cca6:346e:80d0:20cf proto kernel metric 256 pref medium
```

- En direction du second routeur ***Spoke***.

```
ip route ls dev ppp1
```

```
10.0.20.0/24 scope link
10.44.3.2 proto kernel scope link src 10.44.3.1
```

```
ip -6 route ls dev ppp1
```

```
fda0:7a62:14::/64 metric 1024 pref medium
fe80::b4d6:f38c:a930:c8f5 proto kernel metric 256 pref medium
fe80::f0fe:10bd:88f0:cb26 proto kernel metric 256 pref medium
```

Q41. Comment l'ajout de route vers les réseaux d'hébergement est-il répercuté par FRR ?

Afficher les tables de routage des routeurs ***Hub*** après avoir relancé les ouvertures de sessions PPP sur les quatre routeurs ***Spoke***.

Après redémarrage des services PPP sur les routeurs ***Spoke***, les routes vers les réseaux d'hébergement sont marquées K>*. Ce sont des entrées appartenant à la catégorie ***kernel***.

Voici deux extraits des tables de routage du routeur ***Hub numéro 1***.

```
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 172.20.120.1, enp0s1.120, 07:19:49
K>* 10.0.1.0/24 [0/0] is directly connected, ppp0, 00:19:42
K>* 10.0.2.0/24 [0/0] is directly connected, ppp1, 00:18:21
```

```
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 fe80:78::1, enp0s1.120 onlink, 07:21:03
K>* 2001:678:3fc:78::/64 [0/512] is directly connected, enp0s1.120, 07:21:03
K>* fda0:7a62:1::/64 [0/1024] is directly connected, ppp0, 00:20:56
K>* fda0:7a62:2::/64 [0/1024] is directly connected, ppp1, 00:19:35
```

Q42. Comment redistribuer ces nouvelles routes dans le protocole OSPF ?

Reprendre la section sur la redistribution des routes connectées en l'adaptant à la catégorie ***kernel***.

On ajoute une nouvelle instruction redistribute dans les configurations des démons OSPF de chacun des routeurs ***Hub***.

Voici deux extraits de configuration pour le routeur ***Hub numéro 2***.

```
sh run ospfd
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname hub2
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.470
 ip ospf area 0
exit
!
router ospf
 ospf router-id 2.0.0.4
 log-adjacency-changes detail
 redistribute kernel
 redistribute connected
exit
!
end
```

```
sh run ospf6d
```

```
Building configuration...

Current configuration:
!
frr version 10.1.1
frr defaults traditional
hostname hub2
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.470
 ipv6 ospf6 area 0
exit
!
router ospf6
 ospf6 router-id 2.0.0.6
 log-adjacency-changes
 redistribute kernel
 redistribute connected
exit
!
end
```

Q43. Comment vérifier que la redistribution de route est effective ?

En affichant les entrées OSPF des tables de routage d'un routeur ***Hub***, on doit voir apparaître les réseaux d'hébergement accessibles via le ***Hub*** opposé.

Voici deux exemples sur le routeurs ***Hub numéro 2***.

```
sh ip route ospf
```

```
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>* 10.0.1.0/24 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 00:27:03
O>* 10.0.2.0/24 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 00:27:03
O>* 10.47.2.2/32 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 00:34:51
O>* 10.47.4.2/32 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 00:33:30
O   172.20.70.0/29 [110/10] is directly connected, enp0s1.470, weight 1, 06:22:48
O   172.20.120.0/23 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 03:28:25
```

```
sh ipv6 route ospf6
```

```
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   2001:678:3fc:78::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 00:29:02
O>* fda0:7a62:1::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 00:29:02
O>* fda0:7a62:2::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 00:29:02
```

5.3. Installer et configurer Incus

Dans cette section, on s'intéresse aux rôles et fonctionnalités suivantes :

- Installer le gestionnaire de conteneurs système Incus
- Appliquer une configuration initiale
- Configurer l'adressage automatique pour les conteneurs

Pour la partie installation, on reprend les questions du document *Routage inter-VLAN et protocole PPPoE*.

Q44. Comment installer le gestionnaire de conteneurs Incus ?

Lancer une recherche dans la liste des paquets Debian.

Le paquet s'appelle tout simplement incus.

```
apt search ^incus
```

```
sudo apt -y install incus
```

Q45. Comment faire pour que l'utilisateur normal `etu` devienne administrateur et gestionnaire des conteneurs ?

Rechercher le nom du groupe système correspondant à l'utilisation des outils Incus.

Il faut que l'utilisateur normal appartienne au groupes systèmes `incus` et `incus-admin` pour qu'il ait tous les droits sur la gestion des conteneurs.

```
grep incus /etc/group
incus:x:990:
incus-admin:x:989:
```

```
sudo adduser etu incus
sudo adduser etu incus-admin
```

Avertissement

Attention ! Il faut se déconnecter/reconnecter pour bénéficier de la nouvelle attribution de groupe. On peut utiliser les commandes groups ou id pour vérifier le résultat.

```
groups
etu adm sudo users incus-admin incus
```

Pour la partie configuration initiale, on reprend l'utilisation d'un fichier YAML proposée dans le document *Introduction au routage dynamique OSPF avec FRRouting*.

Q46. 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.

Q47. 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,[2001:678:3fc: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.

Q48. 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    |
+------+---------+--------------------+----------------------------------------------+-----------+
| c0   | RUNNING | 10.20.0.33 (eth0)  | fd14:ca46:3864:14:216:3eff:fe2c:c5b9 (eth0) | CONTAINER |
+------+---------+--------------------+----------------------------------------------+-----------+
| c1   | RUNNING | 10.20.0.87 (eth0)  | fd14:ca46:3864:14:216:3eff:fe76:f0d0 (eth0) | CONTAINER |
+------+---------+--------------------+----------------------------------------------+-----------+
| c2   | RUNNING | 10.20.0.153 (eth0) | fd14:ca46:3864:14:216:3eff:fe2c:d169 (eth0) | CONTAINER |
+------+---------+--------------------+----------------------------------------------+-----------+
```

6. Bilan et conclusion

Arrivé à cette étape, la totalité des outils nécessaires à l'interconnexion des réseaux LAN et WAN est en place et il est temps de dresser le bilan avec les ultimes tests de communication entre les services des réseaux distants.

6.1. Tester toutes les communications

Après avoir relevé les adresses attribuées aux conteneurs des quatre routeurs ***Spoke***, on peut lancer une boucle de tests ICMP pour qualifier les communications.

Voici un exemple réalisé dans le contexte de la maquette utilisée pour rédiger ce document.

La commande suivante permet de liste les adresses attribuées au conteneurs sur chaque réseau d'extrémité.

```
incus ls -c 46 -f csv
```

Les résultats au format CSV peuvent être accumulés dans un fichier d'adresses qui sert pour les tests ICMP. Voici un extrait du fichier des adresses de conteneurs :

```
10.0.1.43 (eth0),fda0:7a62:1:0:216:3eff:fec9:2051 (eth0)
10.0.1.57 (eth0),fda0:7a62:1:0:216:3eff:fe4e:167c (eth0)
10.0.1.174 (eth0),fda0:7a62:1:0:216:3eff:fe18:3562 (eth0)
```

Pour qualifier l'accord de niveau de service ou ***Service Level Agreement*** (SLA), on peut utiliser un code comme celui du script ci-dessous qui teste la connectivité à partir de toutes les adresses fournies.

```
#!/bin/bash

# Test if input file exists
if [[ ! -f "$1" ]]; then
    echo "Error: missing or non-existent input file"
    exit 1
fi

file=$1
echo "-------------------------------------------------------------"
echo "CSV input file must be generated from the following command:"
echo "   incus ls -c 46 -f csv"
echo "-------------------------------------------------------------"

# Declare IPv4 and IPv6 array addresses
declare -a ipv4_addresses
declare -a ipv6_addresses

# Read CSV file content
mapfile -t lines < "$file"

# Extract IP addresses from each line
for line in "${lines[@]}"; do
     IFS=',' read -r ipv4 ipv6 <<< "$line"
     ipv4_addresses+=("${ipv4%% *}")
     ipv6_addresses+=("${ipv6%% *}")
done

echo "Launch pings..."
fail=false
for address in "${ipv4_addresses[@]}" "${ipv6_addresses[@]}"
do
     # Capture ping output
     ping_output=$(ping -qc2 "$address" | tr '\n' ' ')
     # Get packet loss number
     packets_lost=$(echo "$ping_output" |\
        grep -Eo '[[:digit:]]+\% packet loss' |\
        grep -Eo '[[:digit:]]+')

     if [[ $packets_lost != 0 ]]; then
        echo "Test failed : $packets_lost% lost packets for $address"
        fail=true
     else
        echo -n "."
     fi
done

if [[ fail==false ]]; then
        echo "Success!"
fi

exit 0
```

Voici un exemple du résultat d'exécution du script sur la maquette.

```
bash icmp-tests.sh addresses.csv
-------------------------------------------------------------
CSV input file must be generated from the following command:
   incus ls -c 46 -f csv
-------------------------------------------------------------
Launch pings...
.......................Success!
```

6.2. Afficher les tables de routage complètes

Pour faire le bilan sur le routage dynamique avec les protocoles OSPFv2 et OSPFv3, on affiche les tables de routage dans lesquelles tous les réseaux doivent figurer.

En se plaçant sur le routeur *Hub numéro 2*, on obtient les entrées suivantes :

```
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 172.20.120.1, enp0s1.120, 20:50:24
O>* 10.0.1.0/24 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 14:53:17
O>* 10.0.2.0/24 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 14:53:17
K>* 10.0.3.0/24 [0/0] is directly connected, ppp0, 14:48:53
K>* 10.0.4.0/24 [0/0] is directly connected, ppp1, 14:48:25
O>* 10.47.2.2/32 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 15:01:05
O>* 10.47.4.2/32 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 14:59:44
L>* 10.47.6.1/32 is directly connected, ppp0, 14:48:53
C>* 10.47.6.2/32 is directly connected, ppp0, 14:48:53
L>* 10.47.8.1/32 is directly connected, ppp1, 14:48:25
C>* 10.47.8.2/32 is directly connected, ppp1, 14:48:25
O   172.20.70.0/29 [110/10] is directly connected, enp0s1.470, weight 1, 20:49:02
C>* 172.20.70.0/29 is directly connected, enp0s1.470, 20:50:24
L>* 172.20.70.2/32 is directly connected, enp0s1.470, 20:50:24
O   172.20.120.0/23 [110/20] via 172.20.70.1, enp0s1.470, weight 1, 17:54:39
C>* 172.20.120.0/23 is directly connected, enp0s1.120, 20:50:24
L>* 172.20.120.3/32 is directly connected, enp0s1.120, 20:50:24
```

Les entrées marquées `O>*` de la table de routage ci-dessus correspondent aux routes apprises via le protocole OSPF qui ont été retenues par le sous-système réseau du noyau du routeur. Vu du routeur *Hub numéro 2*, on identifie quatre routes de ce type : deux pour les réseaux d'hébergement et deux pour les réseaux de transit. Ces quatre entrées correspondent aux routeurs *Spoke* raccordés au *Hub numéro 1*.

```
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 fe80:78::1, enp0s1.120 onlink, 20:53:20
O   2001:678:3fc:78::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 14:56:06
K>* 2001:678:3fc:78::/64 [0/512] is directly connected, enp0s1.120, 20:53:20
L>* 2001:678:3fc:78:baad:caff:fefe:8/128 is directly connected, enp0s1.120, 20:53:20
O>* fda0:7a62:1::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 14:56:06
O>* fda0:7a62:2::/64 [110/20] via fe80::baad:caff:fefe:5, enp0s1.470, weight 1, 14:56:06
K>* fda0:7a62:3::/64 [0/1024] is directly connected, ppp0, 14:51:49
K>* fda0:7a62:4::/64 [0/1024] is directly connected, ppp1, 14:51:21
C * fe80::/64 is directly connected, enp0s1.477, 20:53:20
C * fe80::/64 is directly connected, enp0s1.476, 20:53:20
C * fe80::/64 is directly connected, enp0s1.475, 20:53:20
C * fe80::/64 is directly connected, enp0s1.470, 20:53:20
C * fe80::/64 is directly connected, enp0s1.120, 20:53:20
C * fe80::/64 is directly connected, enp0s1.478, 20:53:20
C>* fe80::/64 is directly connected, enp0s1, 20:53:20
C>* fe80::4407:6511:26c1:58d1/128 is directly connected, ppp1, 14:51:21
C>* fe80::551a:aa5f:6325:8f85/128 is directly connected, ppp0, 14:51:49
C>* fe80:1d6::/64 is directly connected, enp0s1.470, 20:53:20
C>* fe80:1db::/64 is directly connected, enp0s1.475, 20:53:20
C>* fe80:1dd::/64 is directly connected, enp0s1.477, 20:53:20
```

Comme on l'a déjà vu, toutes les entrées marquées `C>*` correspondent aux interfaces actives ou connectées. On peut afficher le résumé de l'état des interfaces du routeur pour voir la correspondance.

```
 sh int brief
```

```
Interface       Status  VRF             Addresses
---------       ------  ---             ---------
enp0s1          up      default         fe80::baad:caff:fefe:8/64
enp0s1.120      up      default         172.20.120.3/23
                                        + 2001:678:3fc:78:baad:caff:fefe:8/64
enp0s1.470      up      default         172.20.70.2/29
                                        + fe80:1d6::2/64
enp0s1.475      up      default         + fe80:1db::1/64
enp0s1.476      up      default         fe80::baad:caff:fefe:8/64
enp0s1.477      up      default         + fe80:1dd::1/64
enp0s1.478      up      default         fe80::baad:caff:fefe:8/64
lo              up      default
ppp0            up      default         10.47.6.1/32
                                        fe80::1cd0:13ee:bdd1:22bb/128
ppp1            up      default         10.47.8.1/32
                                        fe80::7993:fc0f:f5c8:d0c4/128
```

6.3. Pour conclure...

Maintenant que l'on a validé l'ensemble des échanges entre les conteneurs situés aux extrémités de la topologie d'interconnexion de réseaux, on peut passer à la conclusion.

Ce dernier document de la série des travaux pratiques fournit une synthèse détaillée de tous les modes d'interconnexion de réseaux locaux (LAN) et étendus (WAN) en utilisant des protocoles comme PPPoE et OSPF. Il couvre les principes de configuration des routeurs d'une architecture *Hub & Spoke*, l’installation et la configuration de la solution FRRouting avec la mise en place du routage dynamique OSPF pour IPv4 et IPv6.

On peut considérer que nous sommes arrivés à la limite de l'étude des interconnexions de réseaux sans automatisation. Pour aller plus loin, l'automatisation ainsi que l'intégration et le déploiement continu permettraient de déployer et gérer ces configurations à grande échelle, réduisant ainsi les erreurs manuelles et améliorant l'efficacité opérationnelle des réseaux d’entreprise.