Routage inter-VLAN et protocole PPPoE dans un contexte cloud

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

L'évolution des réseaux étendus (WAN vers la fibre optique a entrainé un changement radical au niveau de la couche liaison. Le format de trame historique HDLC est remplacé par Ethernet qui devient universel. Le hic, c'est que par définition, Ethernet est un réseau de diffusion. C'est là que le protocole PPPoE intervient. Il permet de passer d'un réseau de diffusion à un fonctionnement point à point caractéristique des réseaux étendus.

Les manipulations présentées dans ces travaux pratiques illustrent l'interconnexion entre réseaux locaux et réseaux étendus dans un contexte de type Cloud IAAS (Infrastructure As A Service).

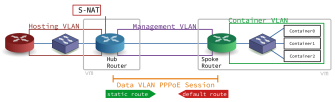


Table des matières

1. Copyright et Licence

1.1. Méta-information

Ce document est écrit avec DocBook XML sur un système Debian GNU/Linux. Il est disponible en version imprimable au format PDF : `interco.pppoe-cloud.qa.pdf`.

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite `$` ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite `#` nécessite les privilèges du super-utilisateur.

2. Interface Ethernet & protocole PPP

Avec la généralisation de l'utilisation de la fibre optique dans les réseaux étendus, le format de trame historique HDLC est progressivement abandonné. Il faut dire que ce format de trame date du développement des liaisons séries asynchrones. Aujourd'hui, les liaisons sur fibres optiques sont Full-Duplex et on ne se préoccupe plus de synchronisation au niveau de la couche liaison de données. Le format de trame Ethernet devient ainsi une référence universelle.

Le protocole PPP offre depuis l'origine une configuration indépendante de la technologie du réseau étendu.

L'association entre trame Ethernet et PPP se fait grâce à un autre protocole baptisé PPPoE. Ce dernier permet d'encapsuler des trames PPP dans des trames Ethernet. Il est décrit à la page Point-to-point protocol over Ethernet qui permet de traiter les questions ci-après.

Q1. Quelle est la raison de l'ajout d'un nouveau protocole entre Ethernet et PPP ?

Consulter la page Point-to-point protocol over Ethernet.

Le protocole PPP a été conçu pour fonctionner sur des liaisons point-à-point alors qu'un réseau local Ethernet est par définition un réseau de diffusion.

Sur un réseau de diffusion, le canal de transmission est partagé entre tous les hôtes qui accèdent au canal. Il a donc été nécessaire d'introduire un mécanisme de découverte des deux extrémités en communication avant de lancer les opérations du protocole PPP.

Q2. Donner la liste des messages de découverte et de session PPPoE en précisant qui est l'initiative de cette découverte.

Consulter la page Point-to-point protocol over Ethernet.

- Client to server: Initiation (PADI)
- Server to client: Offer (PADO)
- Client to server: request (PADR)
- Server to client: session-confirmation (PADS)
- Either end to other end: termination (PADT)

Q3. Quels sont les autres mécanismes de découverte de voisins connus dans un réseau local Ethernet ?

Voici la liste des «grands classiques».

- Address Resolution Protocol (ARP).

  Quelle est l'adresse MAC d'un hôte dont on connaît l'adresse IPv4 ?

- Neighbor Discovery Protocol (NDP).

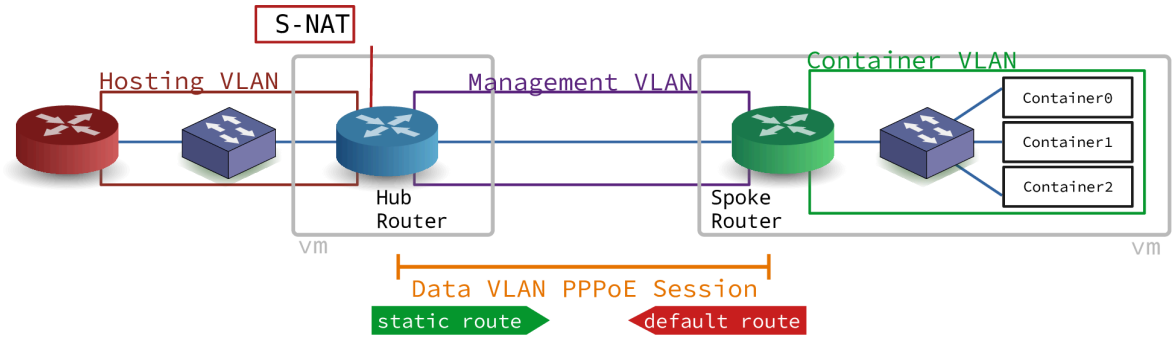
  Ce protocole est associé à IPv6. Il définit 5 messages ICMPv6 qui couvrent les mêmes opérations que celles réalisées par le protcole ARP sans avoir recours à la diffusion et qui ajoutent de nouvelles fonctions.

- Multicast DNS (mDNS) ou Bonjour.

  Ce protocole entre dans la famille zeroconf qui a pour but d'annoncer et de fournir des éléments de configuration aux hôtes du réseau sans faire appel à une infrastructure de services de la couche application tels que DNS et DHCP.

3. Topologies logiques et virtuelles

La représentation de la topologie logique ci-dessous montre que le routeur de couleur bleue assure l'interconnexion entre un réseau d'infrastructure opérateur appelé Hosting VLAN et un réseau étendu qui dessert un site distant. Ce site distant est représenté par le routeur de couleur verte. Les services hébergés sur le site distant appartiennent au réseau appelé Container VLAN. Sur le réseau étendu on distingue deux autres VLANs : le VLAN violet appelé Management VLAN est utilisé pour la supervision et le VLAN orange Data VLAN est utilisé pour l'acheminement des données du site distant. Ce dernier réseau à la particularité d'utiliser une session PPPoE entre les routeurs bleu et vert. Les deux rectangles en gris “matérialisent“ les machines virtuelles qui sont utilisées pour les manipulations.

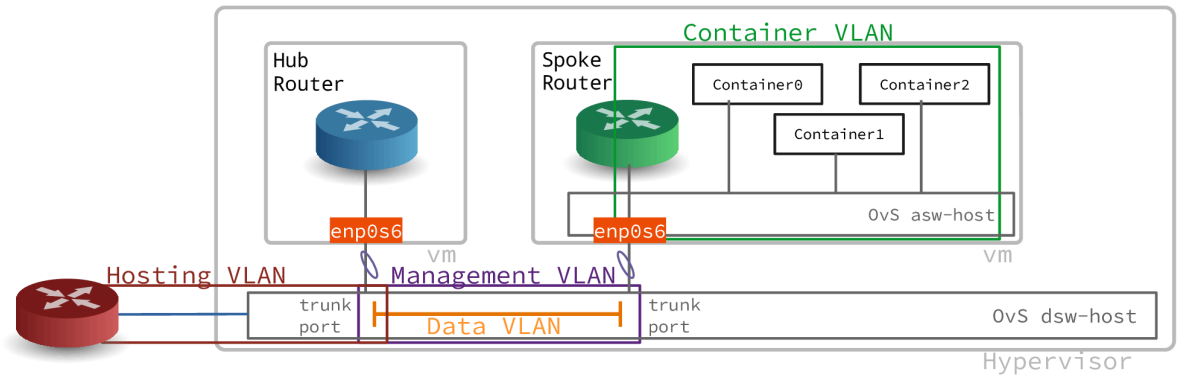


Topologie logique

La représentation de la topologie vue sous l'angle de l'hébergement sur un système hôte hyperviseur montre que le VLAN de couleur verte appelé Container VLAN n'est visible qu'à l'intérieur de la machine virtuelle qui représente le site distant. Ce VLAN est isolé et ses préfixes réseau IPv4 et IPv6 doivent être routés. C'est la raison pour laquelle la machine virtuelle du site distant dispose de son propre commutateur : `asw-host`.

Tous les autres VLANs sont présents sur le commutateur virtuel de couche distribution appelé `dsw-host`. Ce commutateur appartient au système hôte. Il assure le raccordement entre les réseaux physiques et virtualisés. Chacun des routeurs bleu et vert est raccordé avec un lien unique (port en mode trunk) sur lequel le trafic des VLANs doit transiter.

Côté conteneurs, le raccordement au commutateur `asw-host` sera assuré automatiquement par le gestionnaire LXD.



Topologie hébergée

4. Plan d'adressage

Attention ! Les adresses de passerelle côté opérateur (Hosting VLAN de couleur rouge) sont déjà implantées dans l'infrastructure de travaux pratiques tandis que toutes les autres adresses de passerelle sont à implanter sur les routeurs bleu et vert.

Tableau 1. Affectation des numéros de VLANs, des adresses de passerelle et des authentifiants - Groupe 1

Planète	VLAN	Numéro	Type	Adresse
Christophsis	Rouge	-	Passerelle	`192.168.37.9/29` `2001:678:3fc:133::1/64`
	Violet	400	Adresse	`fe80:190::1`
				`fe80:190::2`
	Orange	401	Point à point	`10.4.1.1:10.4.1.2`
			Authentifiants	`green_s1 / 5p0k3.1`
	Vert	10	Passerelle	`203.0.113.1/24` `fda0:7a62:a::1/64`
Corellia	Rouge	-	Passerelle	`172.22.9.33/29` `2001:678:3fc:151::1/64`
	Violet	402	Adresse	`fe80:192::1`
				`fe80:192::2`
	Orange	403	Point à point	`10.4.3.1:10.4.3.2`
			Authentifiants	`green_s2 / 5p0k3.2`
	Vert	11	Passerelle	`203.0.113.1/24` `fda0:7a62:b::1/64`
Delaya	Rouge	-	Passerelle	`10.31.0.193/26` `2001:678:3fc:136::1/64`
	Violet	404	Adresse	`fe80:194::1`
				`fe80:194::2`
	Orange	405	Point à point	`10.4.5.1:10.4.5.2`
			Authentifiants	`green_s3 / 5p0k3.3`
	Vert	12	Passerelle	`203.0.113.1/24` `fda0:7a62:c::1/64`
Kashyyyk	Rouge	-	Passerelle	`10.30.5.129/26` `2001:678:3fc:131::1/64`
	Violet	406	Adresse	`fe80:196::1`
				`fe80:196::2`
	Orange	407	Point à point	`10.4.7.1:10.4.7.2`
			Authentifiants	`green_s4 / 5p0k3.4`
	Vert	13	Passerelle	`203.0.113.1/24`

Planète	VLAN	Numéro	Type	Adresse
				`fda0:7a62:d::1/64`
Korriban	Rouge	-	Passerelle	`10.9.10.129/26` `2001:678:3fc:14f::1/64`
	Violet	408	Adresse	`fe80:198::1`
				`fe80:198::2`
	Orange	409	Point à point	`10.4.9.1:10.4.9.2`
			Authentifiants	`green_s5 / 5p0k3.5`
	Vert	14	Passerelle	`203.0.113.1/24` `fda0:7a62:e::1/64`
Kessel	Rouge	-	Passerelle	`10.30.6.65/28` `2001:678:3fc:132::1/64`
	Violet	410	Adresse	`fe80:19a::1`
				`fe80:19a::2`
	Orange	411	Point à point	`10.4.11.1:10.4.11.2`
			Authentifiants	`green_s6 / 5p0k3.6`
	Vert	15	Passerelle	`203.0.113.1/24` `fda0:7a62:f::1/64`
Mygeeto	Rouge	-	Passerelle	`10.9.15.17/29` `2001:678:3fc:150::1/64`
	Violet	412	Adresse	`fe80:19c::1`
				`fe80:19c::2`
	Orange	413	Point à point	`10.4.13.1:10.4.13.2`
			Authentifiants	`green_s7 / 5p0k3.7`
	Vert	16	Passerelle	`203.0.113.1/24` `fda0:7a62:10::1/64`
Nelvaan	Rouge	-	Passerelle	`10.31.1.145/28` `2001:678:3fc:137::1/64`
	Violet	414	Adresse	`fe80:19e::1`
				`fe80:19e::2`
	Orange	415	Point à point	`10.4.15.1:10.4.15.2`
			Authentifiants	`green_s8 / 5p0k3.8`
	Vert	17	Passerelle	`203.0.113.1/24` `fda0:7a62:11::1/64`
Rattatak	Rouge	-	Passerelle	`192.168.10.81/28` `2001:678:3fc:145::1/64`

Planète	VLAN	Numéro	Type	Adresse
	Violet	416	Adresse	`fe80:1a0::1`
				`fe80:1a0::2`
	Orange	417	Point à point	`10.4.17.1:10.4.17.2`
			Authentifiants	`green_s9 / 5p0k3.9`
	Vert	18	Passerelle	`203.0.113.1/24` `fda0:7a62:12::1/64`
Saleucami	Rouge	-	Passerelle	`192.168.12.17/29` `2001:678:3fc:138::1/64`
	Violet	418	Adresse	`fe80:1a2::1`
				`fe80:1a2::2`
	Orange	419	Point à point	`10.4.19.1:10.4.19.2`
			Authentifiants	`green_s10 / 5p0k3.10`
	Vert	19	Passerelle	`203.0.113.1/24` `fda0:7a62:13::1/64`
Taris	Rouge	-	Passerelle	`10.8.11.9/29` `2001:678:3fc:14b::1/64`
	Violet	420	Adresse	`fe80:1a4::1`
				`fe80:1a4::2`
	Orange	421	Point à point	`10.4.21.1:10.4.21.2`
			Authentifiants	`green_s11 / 5p0k3.11`
	Vert	20	Passerelle	`203.0.113.1/24` `fda0:7a62:14::1/64`
Teth	Rouge	-	Passerelle	`10.0.10.33/27` `2001:678:3fc:13c::1/64`
	Violet	422	Adresse	`fe80:1a6::1`
				`fe80:1a6::2`
	Orange	423	Point à point	`10.4.23.1:10.4.23.2`
			Authentifiants	`green_s12 / 5p0k3.12`
	Vert	21	Passerelle	`203.0.113.1/24` `fda0:7a62:15::1/64`
Utapau	Rouge	-	Passerelle	`172.21.12.17/29` `2001:678:3fc:14c::1/64`
	Violet	424	Adresse	`fe80:1a8::1`
				`fe80:1a8::2`
	Orange	425	Point à point	`10.4.25.1:10.4.25.2`

Planète	VLAN	Numéro	Type	Adresse
			Authentifiants	`green_s13 / 5p0k3.13`
	Vert	22	Passerelle	`203.0.113.1/24` `fda0:7a62:16::1/64`
Yavin	Rouge	-	Passerelle	`172.19.9.65/26` `2001:678:3fc:13b::1/64`
	Violet	426	Adresse	`fe80:1aa::1`
				`fe80:1aa::2`
	Orange	427	Point à point	`10.4.27.1:10.4.27.2`
			Authentifiants	`green_s14 / 5p0k3.14`
	Vert	23	Passerelle	`203.0.113.1/24` `fda0:7a62:17::1/64`

Tableau 2. Affectation des numéros de VLANs, des adresses de passerelle et des authentifiants - Groupe 2

Planète	VLAN	Numéro	Type	Adresse
Alderaan	Rouge	-	Passerelle	`172.17.64.129/25` `2001:678:3fc:64::1/64`
	Violet	220	Adresse	`fe80:dc::1`
				`fe80:dc::2`
	Orange	221	Point à point	`10.2.21.1:10.2.21.2`
			Authentifiants	`green_s20 / 5p0k3.20`
	Vert	24	Passerelle	`203.0.113.1/24` `fda0:7a62:18::1/64`
Bespin	Rouge	-	Passerelle	`172.20.135.65/28` `2001:678:3fc:87::1/64`
	Violet	222	Adresse	`fe80:de::1`
				`fe80:de::2`
	Orange	223	Point à point	`10.2.23.1:10.2.23.2`
			Authentifiants	`green_s21 / 5p0k3.21`
	Vert	25	Passerelle	`203.0.113.1/24` `fda0:7a62:19::1/64`
Centares	Rouge	-	Passerelle	`172.18.4.1/22` `2001:678:3fc:65::1/64`
	Violet	224	Adresse	`fe80:e0::1`
				`fe80:e0::2`
	Orange	225	Point à point	`10.2.25.1:10.2.25.2`
			Authentifiants	`green_s22 / 5p0k3.22`

Planète	VLAN	Numéro	Type	Adresse
	Vert	26	Passerelle	`203.0.113.1/24` `fda0:7a62:1a::1/64`
Coruscant	Rouge	-	Passerelle	`172.20.136.81/28` `2001:678:3fc:88::1/64`
	Violet	226	Adresse	`fe80:e2::1`
				`fe80:e2::2`
	Orange	227	Point à point	`10.2.27.1:10.2.27.2`
			Authentifiants	`green_s23 / 5p0k3.23`
	Vert	27	Passerelle	`203.0.113.1/24` `fda0:7a62:1b::1/64`
Dagobah	Rouge	-	Passerelle	`10.3.2.1/23` `2001:678:3fc:66::1/64`
	Violet	228	Adresse	`fe80:e4::1`
				`fe80:e4::2`
	Orange	229	Point à point	`10.2.29.1:10.2.29.2`
			Authentifiants	`green_s24 / 5p0k3.24`
	Vert	28	Passerelle	`203.0.113.1/24` `fda0:7a62:1c::1/64`
Endor	Rouge	-	Passerelle	`172.24.132.17/28` `2001:678:3fc:84::1/64`
	Violet	230	Adresse	`fe80:e6::1`
				`fe80:e6::2`
	Orange	231	Point à point	`10.2.31.1:10.2.31.2`
			Authentifiants	`green_s25 / 5p0k3.25`
	Vert	29	Passerelle	`203.0.113.1/24` `fda0:7a62:1d::1/64`
Felucia	Rouge	-	Passerelle	`10.6.8.1/23` `2001:678:3fc:69::1/64`
	Violet	232	Adresse	`fe80:e8::1`
				`fe80:e8::2`
	Orange	233	Point à point	`10.2.33.1:10.2.33.2`
			Authentifiants	`green_s26 / 5p0k3.26`
	Vert	30	Passerelle	`203.0.113.1/24` `fda0:7a62:1e::1/64`
Geonosis	Rouge	-	Passerelle	`172.20.131.33/29` `2001:678:3fc:83::1/64`

Planète	VLAN	Numéro	Type	Adresse
	Violet	234	Adresse	`fe80:ea::1`
				`fe80:ea::2`
	Orange	235	Point à point	`10.2.35.1:10.2.35.2`
			Authentifiants	`green_s27 / 5p0k3.27`
	Vert	31	Passerelle	`203.0.113.1/24` `fda0:7a62:1f::1/64`
Hoth	Rouge	-	Passerelle	`10.7.10.1/23` `2001:678:3fc:6a::1/64`
	Violet	236	Adresse	`fe80:ec::1`
				`fe80:ec::2`
	Orange	237	Point à point	`10.2.37.1:10.2.37.2`
			Authentifiants	`green_s28 / 5p0k3.28`
	Vert	32	Passerelle	`203.0.113.1/24` `fda0:7a62:20::1/64`
Jakku	Rouge	-	Passerelle	`172.20.130.25/29` `2001:678:3fc:82::1/64`
	Violet	238	Adresse	`fe80:ee::1`
				`fe80:ee::2`
	Orange	239	Point à point	`10.2.39.1:10.2.39.2`
			Authentifiants	`green_s29 / 5p0k3.29`
	Vert	33	Passerelle	`203.0.113.1/24` `fda0:7a62:21::1/64`
Kamino	Rouge	-	Passerelle	`192.168.107.1/25` `2001:678:3fc:6b::1/64`
	Violet	240	Adresse	`fe80:f0::1`
				`fe80:f0::2`
	Orange	241	Point à point	`10.2.41.1:10.2.41.2`
			Authentifiants	`green_s30 / 5p0k3.30`
	Vert	34	Passerelle	`203.0.113.1/24` `fda0:7a62:22::1/64`
Mustafar	Rouge	-	Passerelle	`192.168.110.129/25` `2001:678:3fc:6e::1/64`
	Violet	242	Adresse	`fe80:f2::1`
				`fe80:f2::2`
	Orange	243	Point à point	`10.2.43.1:10.2.43.2`

Planète	VLAN	Numéro	Type	Adresse
			Authentifiants	`green_s31 / 5p0k3.31`
	Vert	35	Passerelle	`203.0.113.1/24` `fda0:7a62:25::1/64`
Naboo	Rouge	-	Passerelle	`192.168.122.1/28` `2001:678:3fc:7a::1/64`
	Violet	244	Adresse	`fe80:f4::1`
				`fe80:f4::2`
	Orange	245	Point à point	`10.2.45.1:10.2.45.2`
			Authentifiants	`green_s32 / 5p0k3.32`
	Vert	36	Passerelle	`203.0.113.1/24` `fda0:7a62:26::1/64`
Tatooine	Rouge	-	Passerelle	`172.19.115.193/26` `2001:678:3fc:73::1/64`
	Violet	246	Adresse	`fe80:f6::1`
				`fe80:f6::2`
	Orange	247	Point à point	`10.2.47.1:10.2.47.2`
			Authentifiants	`green_s33 / 5p0k3.33`
	Vert	37	Passerelle	`203.0.113.1/24` `fda0:7a62:27::1/64`

Voici le plan d'adressage utilisé pour la maquette qui sert à la rédaction de ce support de travaux pratiques.

Tableau 3. Affectation des numéros de VLANs, des adresses de passerelle et des authentifiants

Planète	VLAN	Numéro	Type	Adresse
Maquette	Rouge	300	Passerelle	`10.141.0.161/27` `2001:678:3fc:12c::1/64`
	Violet	430	Adresse	`fe80:1ae::1`
				`fe80:1ae::2`
	Orange	431	Point à point	`10.4.31.1:10.4.31.2`
			Authentifiants	`etu / 5p0k3`
	Vert	40	Passerelle	`203.0.113.1/24` `fda0:7a62:28::1/64`

5. Raccordement au commutateur de distribution

Dans cette section, on étudie le raccordement des deux machines virtuelles au commutateur de distribution sur le système hôte.

Q4. Comment contrôler la configuration des ports du commutateur de distribution sur le système hôte ?

Le commutateur virtuel implanté sur le système hôte est géré par Open vSwitch. On fait donc appel à la commande ovs-vsctl pour afficher la configuration des ports. Le mot clé dans le cas de cette question est `vlan_mode`.

- Pour le port de raccordement du routeur, on obtient :

```
$ sudo ovs-vsctl list port tap100 | grep vlan_mode
vlan_mode           : trunk
```

- Pour le port de raccordement du serveur de conteneur, on obtient :

```
$ sudo ovs-vsctl list port tap1 | grep vlan_mode
vlan_mode           : access
```

Q5. Comment contrôler le numéro de VLAN attribué au port en mode accès du commutateur de distribution sur le système hôte ?

On reprend la même commande que dans la question précédente avec le mot clé `tag`.

```
$ sudo ovs-vsctl list port tap1 | grep tag
tag                 : 430
```

Q6. Comment affecter le numéro de VLAN attribué au port en mode accès du commutateur de distribution sur le système hôte ?

On reprend à nouveau la même commande avec l'option `set`.

```
$ sudo ovs-vsctl set port tap1 tag=430
```

Les valeurs données dans l'exemple ci-dessus sont à changer suivant les attributions de la Section 4, « Plan d'adressage ».

Q7. Comment s'assurer que le port du commutateur est bien configuré à chaque nouveau lancement de machine virtuelle ?

On place les commandes de configuration dans une section dédiée du script de lancement. Voici deux exemples de script de lancement :

```
#!/bin/bash
RAM=1024

echo "Lancement routeur Bleu ou Vert" !! À changer
CORDON_TRUNK=100 !! À changer

sudo ovs-vsctl set port tap$CORDON_TRUNK vlan_mode=trunk

$HOME/vm/scripts/ovs-startup.sh routeur.qcow2 $RAM $CORDON_TRUNK
```

Les numéros de port et de VLAN donnés dans les exemples ci-dessus sont à changer suivant le contexte.

6. Routeur Hub (bleu)

Dans cette section, on étudie la machine virtuelle qui joue le rôle de routeur entre le réseau opérateur et le réseau étendu qui dessert le site distant.

6.1. Configuration des interfaces du routeur

Une fois la machine virtuelle routeur lancée, les premières étapes consistent à lui attribuer un nouveau nom et à configurer les interfaces réseau pour joindre les hôtes voisins.

Q8. Comment changer le nom de la machine virtuelle ?

Il faut éditer les deux fichiers `/etc/hosts` et `/etc/hostname` en remplaçant le nom de l'image maître vm0 par le nom voulu. Il est ensuite nécessaire de redémarrer pour que le nouveau nom soit pris en compte par tous les outils du système.

```
etu@vm0:~$ sudo sed -i 's/vm0/bleu/g' /etc/hosts
etu@vm0:~$ sudo sed -i 's/vm0/bleu/g' /etc/hostname
sudo: impossible de résoudre l'hôte vm0: Échec temporaire dans la résolution du nom
etu@vm0:~$ sudo reboot
```

Q9. Comment appliquer les configurations réseau IPv4 et IPv6 à partir de l'unique interface du routeur ?

Consulter les pages de manuels du fichier de configuration système à l'aide de la commande man interfaces.

Il existe plusieurs possibilités pour configurer une interface réseau. Dans le contexte de ces manipulations, on utilise le fichier de configuration fourni par la distribution Debian GNU/Linux : `/etc/network/interfaces`.

La configuration de base fournie avec l'image maître suppose que l'interface obtienne un bail DHCP pour la partie IPv4 et une configuration automatique via SLAAC pour la partie IPv6. Cette configuration par défaut doit être éditée et remplacée. Il faut configurer trois interfaces.

Une interface doit être créée pour chacun des différents réseaux avec le numéro de VLAN désigné dans le tableau de la Section 4, « Plan d'adressage ».

- L'interface principale doit être placée en mode manuel (manual). Elle doit être activée/désactivée au niveau de la couche liaison.
- Une interface doit être créée pour le VLAN rouge. Cette interface doit désigner les passerelles IPv4 et IPv6 de façon à joindre l'Internet.
- Une interface doit être créée pour le VLAN violet avec une adresse de lien local IPv6 pour la supervision.
- Une interface doit être créée pour le VLAN orange avec les adresses IPv4 et IPv6 de passerelle pour le réseau étendu.

Voici une copie du fichier `/etc/network/interfaces` de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- VLAN ROUGE ----------
auto enp0s6.300
iface enp0s6.300 inet static
 address 10.141.0.162/27
 gateway 10.141.0.161
 dns-nameserver 9.9.9.9

iface enp0s6.300 inet6 static
 address 2001:678:3fc:12c::2/64
 gateway 2001:678:3fc:12c::1

# ---------- VLAN VIOLET ----------
auto enp0s6.430
iface enp0s6.430 inet6 static
 address fe80:1ae::1/64

# ---------- VLAN ORANGE ----------
auto enp0s6.431
iface enp0s6.431 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down
```

Une fois le fichier de configuration en place, il est préférable de redémarrer la machine virtuelle de façon à vérifier que la configuration des interfaces est bien appliquée après chaque réinitialisation.

Q10. Quels sont les tests de connectivité réalisables après application de la nouvelle configuration des interfaces réseau ?

Relever l'état des trois interfaces et procédez aux tests en respectant les couches de la modélisation.

La commande ip addr ls permet de relever l'état de la configuration pour chaque interface.

```
$ ip addr ls | grep state
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2: enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
3: enp0s6.300@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
4: enp0s6.430@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
6: enp0s6.431@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
```

Sans la confirmation que la configuration du routeur vert est prête, c'est du côté hébergement et accès Internet qu'il faut orienter les tests. Classiquemt, on cherche à joindre la passerelle en premier puis l'Internet ensuite via des requêtes ICMP. Enfin, on effectue un test de couche application avec une requête DNS.

```
$ ping -q -c2 10.141.0.161
PING 10.141.0.161 (10.141.0.161) 56(84) bytes of data.

--- 10.141.0.161 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.990/1.188/1.387/0.198 ms
$ ping -q -c2 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 = 12.218/12.227/12.237/0.009 ms
```

```
$ ping -q -c2 2001:678:3fc:12c::1
PING 2001:678:3fc:12c::1(2001:678:3fc:12c::1) 56 data bytes

--- 2001:678:3fc:12c::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.023/1.164/1.306/0.141 ms
$ ping -q -c2 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 1002ms
rtt min/avg/max/mdev = 39.999/40.391/40.784/0.392 ms
```

```
$ host quad9.net
quad9.net has address 216.21.3.77
quad9.net has IPv6 address 2620:0:871:9000::77
quad9.net mail is handled by 20 mx2.quad9.net.
quad9.net mail is handled by 100 keriomail.pch.net.
quad9.net mail is handled by 5 mx1.quad9.net.
```

6.2. Activation de la fonction routage

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. Son fichier de configuration principal est `/etc/sysctl.conf`.

Q11. Comment activer le routage dans le sous-système réseau du noyau Linuxe ?

Utiliser la commande sysctl pour effectuer des recherches et identifier les paramètres utiles. Par exemple : `$ sudo sysctl -a -r ".*forward.*"`.

Le fichier `/etc/sysctl.conf` contient des commentaires qui guident facilement vers les bons paramètres.

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

Voici un extrait du fichier `/etc/sysctl.conf` du routeur de la maquette après édition.

```
$ egrep -v '(^#|^$)' /etc/sysctl.conf
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians = 1
```

Voici une copie d'écran de l'application des nouveaux paramètres.

```
$ sudo sysctl --system
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
```

Q12. Quelles sont les conditions à réunir pour tester le fonctionnement du routage ?

Rechercher comment utiliser l'analyseur réseau tshark pour caractériser l'acheminement du trafic d'un réseau à l'autre.

Le plan d'adressage prévoit d'utiliser des préfixes ayant une portée locale pour les réseaux de conteneurs. Il n'est donc pas possible de passer par une requête ICMP pour caractériser l'accès aux réseaux distants. En effet, l'adresse source n'est pas reconnue par l'hôte distant et les routeurs de l'Internet ne disposent d'aucune solution pour joindre le réseau des conteneurs.

Voici un extrait de capture qui montre que le serveur de conteneur cherche à joindre un hôte sur l'Internet sans succès. Cette capture étant réalisée sur l'interface réseau côté hébergement, elle montre que le trafic est bien echeminé d'un réseau à l'autre.

```
$ tshark -i enp0s6.300
Capturing on 'enp0s6.300'
    1 0.000000000    192.0.2.2 → 9.9.9.9      DNS 81 Standard query 0xbdab A 1.debian.pool.ntp.org
    2 0.000056361    192.0.2.2 → 9.9.9.9      DNS 81 Standard query 0xab92 AAAA 1.debian.pool.ntp.org
```

6.3. Activation de la traduction d'adresses

Le résultat de la question ci-dessus montre que les hôtes situés dans le réseau des conteneurs ne peuvent pas joindre l'Internet puisque les préfixes réseau utilisés ont une portée limitée.

Q13. Quels sont les paquets qui fournissent les outils de gestion de la traduction d'adresses ?

Rechercher les paquets relatifs au filtrage et à la gestion des règles de pare-feux.

Sur les systèmes GNU/Linux, le système de pare-feux comprend une partie "espace utilisateur" appelée iptables et une partie "noyau" appelée netfilter.

C'est la partie "espace utilisateur" qui nous intéresse ici.

```
$ aptitude search iptables
p   arno-iptables-firewall          - single- and multi-homed firewall script wi
p   golang-github-coreos-go-iptable - Go bindings for iptables utility
i   iptables                        - administration tools for packet filtering
p   iptables-converter              - convert iptables-commands from a file to i
p   iptables-converter-doc          - convert iptables-commands from a file to i
p   iptables-netflow-dkms           - iptables target which generates netflows
p   iptables-persistent             - boot-time loader for netfilter rules, ipta
p   libiptables-chainmgr-perl       - Perl extension for manipulating iptables p
p   libiptables-parse-perl          - Perl extension for parsing iptables firewa
p   python-iptables-doc             - documentation for the python-iptables libr
p   python3-iptables                - Python bindings for iptables (Python 3 int
```

On voit que le paquet iptables est déjà installé et qu'il ne manque que la gestion de la sauvegarde des règles de filtrage et traduction d'adresses.

```
$ sudo apt install iptables-persistent
```

Q14. Quelles sont les règles à appliquer pour assurer une traduction des adresses sources en sortie sur le réseau hébergement ?

Rechercher dans les pages de manuel de la commande iptables.

C'est la cible `MASQUERADE` qui nous intéresse. Voici un exemple de règles de traduction des adresses sources pour la maquette.

```
$ sudo iptables -t nat -A POSTROUTING -o enp0s6.300 -j MASQUERADE
$ sudo sh -c "iptables-save >/etc/iptables/rules.v4"
```

```
$ sudo ip6tables -t nat -A POSTROUTING -o enp0s6.300 -j MASQUERADE
$ sudo sh -c "ip6tables-save >/etc/iptables/rules.v6"
```

Q15. Comment caractériser le fonctionnement de la traduction d'adresses sources ?

Rechercher dans les pages de manuel de la commande iptables les options d'affichage du décompte du trafic traité.

Voici un exemple d'affichage pour le trafic IPv4 uniquement.

```
$ sudo iptables -vnL -t nat
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    8   598 MASQUERADE  all  --  *      enp0s6.300  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

6.4. Activation du protocole PPPoE côté réseau étendu

Pour acheminer le trafic depuis et vers le site distant, il est nécessaire de passer par une authentification. Cette fonction est assurée à l'aide du protocole PPPoE.

Le protocole PPP n'a pas été conçu suivant le modèle Client/Serveur. Il suppose que deux processus pairs échangent des informations. Dans les questions qui suivent, le routeur bleu doit exiger que le routeur vert s'authentifie auprès de lui avant de délivrer les adresses de couche réseau.

Q16. Quel paquet spécifique à la gestion du dialogue PPPoE à installer sur le routeur Hub ?

Rechercher dans le catalogue des paquets, la référence pppoe.

```
$ aptitude search pppoe
i   pppoe      - Pilote PPP sur Ethernet
p   pppoeconf - configures PPPoE/ADSL connections
```

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

Q17. Quel est le rôle de l'outil contenu dans le paquet demandé à la question précédente relativement au démon pppd fourni avec le paquet ppp ?

Rechercher dans les pages de manuels de l'outil demandé à la question précédente.

L'outil pppoe-server gère directement l'encapsulation des trames PPP dans les trames Ethernet. Il communique ensuite les paramètres utiles au démon pppd qui fonctionne de façon totalement transparente vis-à-vis de la technologie du réseau sous-jacent.

Q18. 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 des informations du type suivant.

```
pppd[3262]: sent [LCP ConfReq id=0x1 <mru 1492> <magic 0x46010ac>]
kernel: [  895.700115] NET: Registered protocol family 24
pppd[3262]: rcvd [LCP ConfReq id=0x1 <magic 0xcab9fecc>] ❶
pppd[3262]: sent [LCP ConfAck id=0x1 <magic 0xcab9fecc>]
pppd[3262]: sent [LCP ConfReq id=0x1 <mru 1492> <magic 0x46010ac>]
pppd[3262]: rcvd [LCP ConfAck id=0x1 <mru 1492> <magic 0x46010ac>]
pppd[3262]: sent [LCP EchoReq id=0x0 magic=0x46010ac]
pppd[3262]: peer from calling number 52:54:00:12:34:05 authorized
pppd[3262]: sent [IPCP ConfReq id=0x1 <addr 0.0.0.0>] ❷
pppd[3262]: rcvd [LCP EchoReq id=0x0 magic=0xcab9fecc]
pppd[3262]: sent [LCP EchoRep id=0x0 magic=0x46010ac]
pppd[3262]: rcvd [IPCP ConfReq id=0x1 <addr 10.0.0.1>]
pppd[3262]: sent [IPCP ConfAck id=0x1 <addr 10.0.0.1>]
pppd[3262]: rcvd [LCP EchoRep id=0x0 magic=0xcab9fecc]
pppd[3262]: rcvd [IPCP ConfNak id=0x1 <addr 10.67.15.1>]
pppd[3262]: sent [IPCP ConfReq id=0x2 <addr 10.67.15.1>]
pppd[3262]: rcvd [IPCP ConfAck id=0x2 <addr 10.67.15.1>]
pppd[3262]: local  IP address 10.67.15.1
pppd[3262]: remote IP address 10.0.0.1
```

❶ La sous-couche Link Control Protocol (LCP) assure la configuration automatique des interfaces à chaque extrémité. Les paramètres négociés entre les deux hôtes en communication sont multiples : l'adaptation de la taille de datagramme, les caractères d'échappement, les numéros magiques et la sélection des options d'authentification.

❷ La sous-couche Network Control Protocol (NCP) assure l'encapsulation de multiples protocoles de la couche réseau. Dans l'exemple donné, c'est le protocole IPv4 qui est utilisé ; d'où l'acronyme IPCP.

Q19. 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.

Q20. 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 exmple du contenu de ce fichier.

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

Q21. 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.

Q22. Quelles sont les options du protocole PPP qui doivent être implantées dans le fichier demandé à la question précédente ?

Consulter les pages de manuels du démon pppd et rechercher les paramètres correspondant à la liste suivante.

- Afficher en détail toutes les étapes d'établissement de session dans les journaux système.

- Référencer l'identifiant du compte utilisateur à utiliser lors de l'authentification du routeur vert. Cette option implique que le compte utilisateur existe sur le système et qu'il soit présent dans le fichier `/etc/ppp/chap-secrets`.
- Imposer au routeur vert une authentification via le protocole CHAP (Challenge Handshake Authentication Protocol).
- Préserver la route par défaut, et donc l'accès Internet, du routeur bleu.
- Publier l'adresse IP du serveur DNS à utiliser pour la résolution des noms de domaines.
- Activer l'utilisation des protocoles IPv6CP et IPv6.

Voici une copie du fichier `/etc/ppp/pppoe-server-options` qui contient la liste des paramètres demandés.

```
debug
login
require-chap
nodefaultroute
ms-dns 172.16.0.2
+ipv6
```

Q23. Comment créer le compte utilisateur local sur le routeur bleu sachant qu'il n'est autorisé ni à se connecter ni à avoir un répertoire personnel ?

Consulter les options de la commande adduser.

Voici un exemple de commande adduser.

```
$ sudo adduser --disabled-login --no-create-home green
```

Q24. Quels sont les paramètres à donner au lancement de l'outil pppoe-server pour qu'il délivre les adresses au routeur vert après authentification de celui-ci ?

Consulter les options de la commande pppoe-server.

Voici un exemple de commande pppoe-server.

```
$ sudo pppoe-server -I enp0s6.431 -C BRAS -L 10.4.31.1 -R 10.4.31.2 -N 1
```

Q25. Quels sont les résultats obtenus une fois que la session PPP est établie et que les adresses de couche réseau ont été délivrées ?

Consulter les journaux système, la liste des procsessus, l'état des interfaces réseau et de la table de routage.

Attention ! Les résultats ne sont pertinents que si le dialogue avec le routeur vert est effectif.

- Consultation des journaux système.

```
pppoe-server[2963]: Session 1 created for client b0:ad:ca:fe:00:65 (10.4.31.2) on enp0s6.431 using Service-Na
pppd[2963]: pppd 2.4.7 started by etu, uid 0
pppd[2963]: using channel 20
pppd[2963]: Using interface ppp0
pppd[2963]: Connect: ppp0 <--> /dev/pts/0
pppd[2963]: sent [LCP ConfReq id=0x1 <mru 1492> <auth chap MD5> <magic 0xc4220b37>]
pppd[2963]: rcvd [LCP ConfAck id=0x1 <mru 1492> <auth chap MD5> <magic 0xc4220b37>]
pppd[2963]: rcvd [LCP ConfReq id=0x1 <mru 1492> <magic 0xcca979aa>]
pppd[2963]: sent [LCP ConfAck id=0x1 <mru 1492> <magic 0xcca979aa>]
pppd[2963]: sent [LCP EchoReq id=0x0 magic=0xc4220b37]
pppd[2963]: sent [CHAP Challenge id=0x67 <439a0c7ee25d6eead987864a45fd1331>, name = "bleu"]
pppd[2963]: rcvd [LCP EchoReq id=0x0 magic=0xcca979aa]
pppd[2963]: sent [LCP EchoRep id=0x0 magic=0xc4220b37]
pppd[2963]: rcvd [LCP EchoRep id=0x0 magic=0xcca979aa]
pppd[2963]: rcvd [CHAP Response id=0x67 <dbe0795771fa077ec9010fb8efe13cee>, name = "green"]
pppd[2963]: sent [CHAP Success id=0x67 "Access granted"]
pppd[2963]: Initializing PAM (2) for user green
pppd[2963]: ---> PAM INIT Result = 0
pppd[2963]: Attempting PAM account checks
pppd[2963]: PAM Account OK for green
pppd[2963]: PAM Session opened for user green
pppd[2963]: user green logged in on tty  intf ppp0
pppd[2963]: local  LL address fe80::6c07:edbf:a0b4:f114
pppd[2963]: remote LL address fe80::89f5:1241:ad65:3b22
pppd[2963]: local  IP address 10.4.31.1
pppd[2963]: remote IP address 10.4.31.2
```

- Liste des processus.

```
 pppoe-server -I enp0s6.431 -C BRAS -L 10.4.31.1 -R 10.4.31.2 -N 1
 \_ pppd pty /usr/sbin/pppoe -n -I enp0s6.431 -e 1:b0:ad:ca:fe:00:65 -S '' \
    file /etc/ppp/pppoe-server-options 10.4.31.1:10.4.31.2 nodetach noaccomp \
    nopcomp default-asyncmap mru 1492 mtu 1492
  \_ sh -c /usr/sbin/pppoe -n -I enp0s6.431 -e 1:b0:ad:ca:fe:00:65 -S ''
   \_ /usr/sbin/pppoe -n -I enp0s6.431 -e 1:b0:ad:ca:fe:00:65 -S
```

- État des interfaces.

```
$ $ ip addr ls dev enp0s6.431
6: enp0s6.431@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1
    link/ether b0:ad:ca:fe:00:64 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::b2ad:caff:fefe:64/64 scope link
       valid_lft forever preferred_lft forever

$ ip addr ls dev ppp0
26: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1492 qdisc pfifo_fast state UNKNOWN group default ql
    link/ppp
    inet 10.4.31.1 peer 10.4.31.2/32 scope global ppp0
       valid_lft forever preferred_lft forever
    inet6 fe80::6c07:edbf:a0b4:f114/10 scope link
       valid_lft forever preferred_lft forever
```

- Table de routage.

```
$ ip route ls dev ppp0
10.4.31.2 proto kernel scope link src 10.4.31.1
```

Q26. Quelles sont les modifications à apporter au fichier de configuration système des interfaces réseau pour que l'ouverture de session PPP soit disponible après chaque réinitialisation ?

Consulter les pages de manuel du fichier `/etc/network/interfaces` : `man interfaces`.

Voici une copie du fichier dans le contexte de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- VLAN ROUGE ----------
auto enp0s6.300
iface enp0s6.300 inet static
 address 10.141.0.162/27
 gateway 10.141.0.161
 dns-nameserver 9.9.9.9

iface enp0s6.300 inet6 static
 address 2001:678:3fc:12c::2/64
 gateway 2001:678:3fc:12c::1

# ---------- VLAN VIOLET ----------
auto enp0s6.430
iface enp0s6.430 inet6 static
 address fe80:1ae::1/64

# ---------- VLAN ORANGE ----------
auto enp0s6.431
iface enp0s6.431 inet manual
 up ip link set dev $IFACE up
 up pppoe-server -I $IFACE -C BRAS -L 10.4.31.1 -R 10.4.31.2 -N 1
 down killall pppoe-server
 down ip link set dev $IFACE down
```

6.5. Ajout des routes statiques vers le réseau des conteneurs

Pour joindre le réseau des conteneurs situé au delà du routeur vert, il est nécessaire d'ajouter une route statique pour chaque protocole de la couche réseau IPv4 et IPv6. Le choix du routage statique est justifié par le fait que l'on adresse un site distant d'extrémité via un lien unique.

Attention ! Les tests de connectivité vers le réseau des conteneurs supposent que ces conteneurs soient actifs et correctement configurés. Voici un exemple d'information sur l'état des conteneurs sur le site distant.

```
etu@vert:~$ lxc ls
+------------+---------+---------------------+----------------------------------------------+-----------+-----------+
|    NAME    |  STATE  |        IPV4         |                     IPV6                     |   TYPE    | SNAPSHOTS |
+------------+---------+---------------------+----------------------------------------------+-----------+-----------+
| container0 | RUNNING | 203.0.113.10 (eth0) | fda0:7a62:28:0:216:3eff:feda:e1a (eth0)  | CONTAINER | 0         |
+------------+---------+---------------------+----------------------------------------------+-----------+-----------+
| container1 | RUNNING | 203.0.113.11 (eth0) | fda0:7a62:28:0:216:3eff:fec4:d325 (eth0) | CONTAINER | 0         |
+------------+---------+---------------------+----------------------------------------------+-----------+-----------+
| container2 | RUNNING | 203.0.113.12 (eth0) | fda0:7a62:28:0:216:3eff:fe66:86fb (eth0) | CONTAINER | 0         |
+------------+---------+---------------------+----------------------------------------------+-----------+-----------+
```

On peut aussi s'assurer que les tables de routage du routeur vert désignent bien le routeur bleu comme passerelle vers tous les autres réseaux.

```
etu@bleu:~$ ssh fe80:1ae::2%enp0s6.430 "ip route ls default"
default dev ppp0 scope link
etu@bleu:~$ ssh fe80:1ae::2%enp0s6.430 "ip -6 route ls default"
default dev ppp0 metric 1024 pref medium
```

Q27. Comment ajouter manuellement les routes IPv4 et IPv6 vers le réseau desservi par le routeur vert ?

Consulter les pages de manuel sur le routage avec la commande : `man ip-route`.

Sachant que le site distant est raccordé via une liaison point à point unique, on choisit de désigner la destination par l'interface de la liaison.

```
$ sudo ip route add 203.0.113.0/24 dev ppp0
$ sudo ip -6 route add fda0:7a62:28::/64 dev ppp0
```

Q28. Quels sont les tests de connectivité qui permettent valider la communication à destination des conteneurs du réseau distant ?

Collecter les adresses IPv4 et IPv6 des conteneurs avant de lancer des requêtes ICMP.

Voici un exemple de test pour IPv4.

```
$ for num in {10..12}; do ping -q -c2 203.0.113.$num; done
```

Voici un exemple plus “compliqué“ du fait de l'adressage automatique pour IPv6.

```
$ for addr in fda0:7a62:28:0:216:3eff:feda:e1a \
fda0:7a62:28:0:216:3eff:fec4:d325 \
fda0:7a62:28:0:216:3eff:fe66:86fb; \
do ping -q -c2 $addr; done
```

Q29. Comment appliquer ces routes statiques dans la configuration système pour qu'elles soient activées à chaque établissement de session PPP ?

Il faut parcourir l'arborescence du répertoire `/etc/ppp/` pour repérer les scripts exécutés lors de l'ouverture de session. Créer un script pour chaque protocole de couche réseau qui ajoute la route statique voulue.

- Pour IPv4, le répertoire est `/etc/ppp/ip-up.d/`. Voici une copie du script exécutable `staticroute`.

```
#!/bin/sh

if [ -z "${CONNECT_TIME}" ]; then
 ip route add 203.0.113.0/24 dev ${PPP_IFACE}
fi
```

- Pour IPv6, le répertoire est `/etc/ppp/ipv6-up.d/`. Voici une copie du script exécutable `staticroute`.

```
#!/bin/sh

if [ -z "${CONNECT_TIME}" ]; then
 ip -6 route add fda0:7a62:28::/64 dev ${PPP_IFACE}
fi
```

7. Routeur Spoke (vert)

7.1. Configuration des interfaces du routeur

Une fois la machine virtuelle serveur de conteneurs lancée, les premières étapes consistent à lui attribuer un nouveau nom et à configurer les interfaces réseau pour joindre le routeur voisin et l'Internet.

Q30. Comment changer le nom de la machine virtuelle ?

Il faut éditer les deux fichiers `/etc/hosts` et `/etc/hostname` en remplaçant le nom de l'image maître vm0 par le nom voulu. Il est ensuite nécessaire de redémarrer pour que le nouveau nom soit pris en compte par tous les outils du système.

```
etu@vm0:~$ sudo sed -i 's/vm0/vert/g' /etc/hosts
etu@vm0:~$ sudo sed -i 's/vm0/vert/g' /etc/hostname
sudo: impossible de résoudre l'hôte vm0: Échec temporaire dans la résolution du nom
etu@vm0:~$ sudo reboot
```

Q31. Comment appliquer la configuration réseau IPv4 et IPv6 de l'interface du serveur ?

Consulter les pages de manuels du fichier de configuration système à l'aide de la commande man interfaces.

Il existe plusieurs possibilités pour configurer une interface réseau. Dans le contexte de ces manipulations, on utilise le fichier de configuration fourni par la distribution Debian GNU/Linux : `/etc/network/interfaces`.

La configuration de base fournie avec l'image maître suppose que l'interface obtienne un bail DHCP pour la partie IPv4 et une configuration automatique via SLAAC pour la partie IPv6.

La configuration par défaut doit être éditée et remplacée par une configuration manuelle pour l'interface `enp0s6` et pour le VLAN orange. Pour la supervision dans le VLAN violet, l'adresse IPv6 de lien locale est fournie dans le tableau du plan d'adressage.

En attendant que la configuration du routeur bleu soit prête, on ajoute temporairement une interface `enp0s6.217` avec une configuration automatique. Ainsi, il est possible d'installer et de configurer des services en parallèle. Cette interface doit être désactivée dès que tous les outils sont en place.

Voici une copie du fichier `/etc/network/interfaces` de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- VLAN VIOLET ----------
auto enp0s6.430
iface enp0s6.430 inet6 static
 address fe80:1ae::2/64

# ---------- VLAN ORANGE ----------
auto enp0s6.431
iface enp0s6.431 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- TEMPORAIRE -----------
auto enp0s6.217
iface enp0s6.217 inet dhcp
```

7.2. Activation de la fonction routage

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. Son fichier de configuration principal est `/etc/sysctl.conf`.

Q32. Comment activer le routage dans le sous-système réseau du noyau Linuxe ?

Utiliser la commande sysctl pour effectuer des recherches et identifier les paramètres utiles. Par exemple : `$ sudo sysctl -a -r ".*forward.*"`.

Le fichier `/etc/sysctl.conf` contient des commentaires qui guident facilement vers les bons paramètres.

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

Voici un extrait du fichier `/etc/sysctl.conf` du routeur de la maquette après édition.

```
$ egrep -v '(^#|^$)' /etc/sysctl.conf
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians = 1
```

Voici une copie d'écran de l'application des nouveaux paramètres.

```
$ sudo sysctl --system
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
```

7.3. Activation du protocole PPP dans le VLAN orange

Le routeur vert utilise un démon pppd sur le VLAN `Data` pour établir une session PPP avec le routeur bleu. À la différence de ce dernier, il n'est pas à l'initiative du dialogue PPPoE mais il doit être capable de gérer l'encapsulation des trames PPP sur un réseau local Ethernet.

Q33. Quel paquet fournit le démon de gestion des sessions du protocole PPP sur le routeur vert ?

Rechercher dans le catalogue des paquets, la référence ppp.

```
$ aptitude search ^ppp
i   ppp                - Point-to-Point Protocol (PPP) - daemon
p   ppp-dev            - Point-to-Point Protocol (PPP) - development fil
p   ppp-gatekeeper     - PPP manager for handling balanced, redundant an
p   pppoe              - PPP over Ethernet driver
p   pppoeconf          - configures PPPoE/ADSL connections
```

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

Q34. 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`.

Q35. 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
"green"  *  "5p0k3"   *
```

Q36. 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 une copie du fichier `/etc/ppp/peers/pppoe-provider` avec les options correspondant au contexte de la maquette du routeur vert.

```
# There should be a matching entry with the password in /etc/ppp/chap-secrets.
user "green"

# Load the PPPoE plugin.
plugin rp-pppoe.so

# Ethernet interface to which the modem is connected.
enp0s6.431

# Assumes that your IP address is allocated dynamically by the ISP.
noipdefault
# Try to get the name server addresses from the ISP.
usepeerdns
# Use this connection as the default route.
defaultroute

# Makes pppd "dial again" when the connection is lost.
persist

# Do not ask the remote to authenticate.
noauth

debug
+ipv6
```

Q37. 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'option `file` qui permet de désigner le fichier de configuration à utiliser. Voici une copie d'écran du lancement de pppd.

```
$ sudo pppd file /etc/ppp/peers/pppoe-provider
```

Q38. 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 des informations suivantes.

```
pppd[2272]: Plugin rp-pppoe.so loaded.
pppd[2273]: pppd 2.4.7 started by etu, uid 0
pppd[2273]: Send PPPOE Discovery V1T1 PADI session 0x0 length 4
pppd[2273]:  dst ff:ff:ff:ff:ff:ff  src b0:ad:ca:fe:00:65
pppd[2273]:  [service-name]
pppd[2273]: Recv PPPOE Discovery V1T1 PADO session 0x0 length 36
pppd[2273]:  dst b0:ad:ca:fe:00:65  src b0:ad:ca:fe:00:64
pppd[2273]:  [AC-name BRAS] [service-name] [AC-cookie  f9 fc 4d 12 f1 13 03 f3 bd e8 34 7b 83 d5 2e bf 5b 0b 00
pppd[2273]: Send PPPOE Discovery V1T1 PADR session 0x0 length 28
pppd[2273]:  dst b0:ad:ca:fe:00:64  src b0:ad:ca:fe:00:65
pppd[2273]:  [service-name] [AC-cookie  f9 fc 4d 12 f1 13 03 f3 bd e8 34 7b 83 d5 2e bf 5b 0b 00 00]
pppd[2273]: Recv PPPOE Discovery V1T1 PADS session 0x1 length 4
pppd[2273]:  dst b0:ad:ca:fe:00:65  src b0:ad:ca:fe:00:64
pppd[2273]:  [service-name]
pppd[2273]: PADS: Service-Name: ''
pppd[2273]: PPP session is 1
pppd[2273]: Connected to b0:ad:ca:fe:00:64 via interface enp0s6.431
pppd[2273]: using channel 19
pppd[2273]: Using interface ppp0
pppd[2273]: Connect: ppp0 <--> enp0s6.431
pppd[2273]: sent [LCP ConfReq id=0x1 <mru 1492> <magic 0xc8ff53a8>]
pppd[2273]: rcvd [LCP ConfReq id=0x1 <mru 1492> <auth chap MD5> <magic 0x11d42bb4>]
pppd[2273]: sent [LCP ConfAck id=0x1 <mru 1492> <auth chap MD5> <magic 0x11d42bb4>]
pppd[2273]: sent [LCP ConfReq id=0x1 <mru 1492> <magic 0xc8ff53a8>]
pppd[2273]: rcvd [LCP ConfAck id=0x1 <mru 1492> <magic 0xc8ff53a8>]
pppd[2273]: sent [LCP EchoReq id=0x0 magic=0xc8ff53a8]
pppd[2273]: rcvd [LCP EchoReq id=0x0 magic=0x11d42bb4]
pppd[2273]: sent [LCP EchoRep id=0x0 magic=0xc8ff53a8]
pppd[2273]: rcvd [CHAP Challenge id=0xde <208dcd62a1589d928f94e384cdb43910eb9b03bf9b0ac0>, name = "bleu"]
pppd[2273]: sent [CHAP Response id=0xde <32bb85b4fa53e24b73a4137db04278e6>, name = "green"]
pppd[2273]: rcvd [LCP EchoRep id=0x0 magic=0x11d42bb4]
pppd[2273]: rcvd [CHAP Success id=0xde "Access granted"]
pppd[2273]: CHAP authentication succeeded: Access granted
pppd[2273]: CHAP authentication succeeded
pppd[2273]: peer from calling number B0:AD:CA:FE:00:64 authorized
pppd[2273]: sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
pppd[2273]: sent [IPV6CP ConfReq id=0x1 <addr fe80::adfa:a20d:dbba:3553>]
pppd[2273]: rcvd [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
pppd[2273]: sent [CCP ConfReq id=0x1]
pppd[2273]: sent [CCP ConfRej id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
pppd[2273]: rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.4.31.1>]
pppd[2273]: sent [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
pppd[2273]: rcvd [IPV6CP ConfReq id=0x1 <addr fe80::f558:469f:1034:fd29>]
pppd[2273]: sent [IPV6CP ConfAck id=0x1 <addr fe80::f558:469f:1034:fd29>]
pppd[2273]: rcvd [IPCP ConfNak id=0x1 <addr 10.4.31.2> <ms-dns1 172.16.0.2> <ms-dns2 172.16.0.2>]
pppd[2273]: sent [IPCP ConfReq id=0x2 <addr 10.4.31.2> <ms-dns1 172.16.0.2> <ms-dns2 172.16.0.2>]
pppd[2273]: rcvd [IPV6CP ConfAck id=0x1 <addr fe80::adfa:a20d:dbba:3553>]
pppd[2273]: local  LL address fe80::adfa:a20d:dbba:3553
pppd[2273]: remote LL address fe80::f558:469f:1034:fd29
pppd[2273]: Script /etc/ppp/ipv6-up started (pid 2280)
pppd[2273]: rcvd [CCP ConfAck id=0x1]
pppd[2273]: rcvd [CCP ConfReq id=0x2]
pppd[2273]: sent [CCP ConfAck id=0x2]
pppd[2273]: rcvd [IPCP ConfReq id=0x2 <addr 10.4.31.1>]
pppd[2273]: sent [IPCP ConfAck id=0x2 <addr 10.4.31.1>]
pppd[2273]: rcvd [IPCP ConfAck id=0x2 <addr 10.4.31.2> <ms-dns1 172.16.0.2> <ms-dns2 172.16.0.2>]
pppd[2273]: not replacing default route to enp0s6.217 [172.16.99.1]
pppd[2273]: local  IP address 10.4.31.2
pppd[2273]: remote IP address 10.4.31.1
pppd[2273]: primary   DNS address 172.16.0.2
pppd[2273]: secondary DNS address 172.16.0.2
pppd[2273]: Script /etc/ppp/ip-up started (pid 2282)
pppd[2273]: Script /etc/ppp/ipv6-up finished (pid 2280), status = 0x0
pppd[2273]: Script /etc/ppp/ip-up finished (pid 2282), status = 0x0
```

Q39. 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.

Q40. Quelles sont les modifications à apporter au fichier système de configuration des interfaces réseau pour ouvrir la session PPP à chaque réinitialisation système ?

Consulter les pages de manuel du fichier `/etc/network/interfaces` : `man interfaces`.

Voici une copie du fichier modifié dans le contexte de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- VLAN VIOLET ----------
auto enp0s6.430
iface enp0s6.430 inet6 static
 address fe80:1ae::2/64

# ---------- VLAN ORANGE ----------
auto enp0s6.431
iface enp0s6.431 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- PPPoE ---------------
auto pppoe-provider
iface pppoe-provider inet ppp
        pre-up ifup enp0s6.431
        provider pppoe-provider

# ---------- TEMPORAIRE -----------
#auto enp0s6.217
#iface enp0s6.217 inet dhcp
```

7.4. Activation du commutateur virtuel asw-host

Dans le scénario étudié, les services sont hébergés dans un réseau de conteneurs propre au routeur vert. La mise en œuvre de cette configuration passe par l'installation d'un commutateur virtuel appelé `asw-host`. On utilise Open vSwitch pour configurer ce commutateur.

Q41. Quel est le paquet à installer pour pouvoir ajouter un commutateur virtuel au routeur vert ?

Rechercher le mot clé openvswitch dans la liste des paquets.

Voici un exemple de recherche.

```
$ sudo aptitude search ^openvswitch
p   openvswitch-common           - Open vSwitch common components
p   openvswitch-dbg              - Debug symbols for Open vSwitch packages
p   openvswitch-dev              - Open vSwitch development package
p   openvswitch-ipsec            - Open vSwitch IPsec tunneling support
p   openvswitch-pki              - Open vSwitch public key infrastructure dependency package
p   openvswitch-switch           - Open vSwitch switch implementations
p   openvswitch-switch-dpdk      - DPDK enabled Open vSwitch switch implementation
v   openvswitch-test             -
p   openvswitch-testcontroller   - Simple controller for testing OpenFlow setups
p   openvswitch-vtep             - Open vSwitch VTEP utilities
```

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

```
$ sudo apt install openvswitch-switch
```

Q42. Quel est le fichier de documentation qui fournit les directives de configuration d'un commutateur intégré au fichier système `/etc/network/interfaces` ?

Rechercher dans la liste des fichiers des paquets installés à la question précédente.

Voici un exemple de recherche.

```
$ dpkg -L openvswitch-switch | grep README
/usr/share/doc/openvswitch-switch/README.Debian.gz
```

Q43. Quelles sont les modiifications à apporter au fichier `/etc/network/interfaces` pour configurer le commutateur `asw-host` ?

Voici une copie du fichier de configuration réseau système dans le contexte de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- VLAN VIOLET ----------
auto enp0s6.430
iface enp0s6.430 inet6 static
 address fe80:1ae::2/64

# ---------- VLAN ORANGE ----------
auto enp0s6.431
iface enp0s6.431 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

# ---------- PPPoE ----------------
auto pppoe-provider
iface pppoe-provider inet ppp
        pre-up ifup enp0s6.431
        provider pppoe-provider

#auto enp0s6.217
#iface enp0s6.217 inet dhcp

# ---------- VLAN VERT ------------
allow-ovs asw-host
iface asw-host inet manual
 ovs_type OVSBridge
 ovs_ports sw-vlan40
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

allow-asw-host sw-vlan40
iface sw-vlan40 inet static
 ovs_type OVSBridge
 ovs_bridge asw-host
 ovs_options asw-host 40
 address 203.0.113.1/24

iface sw-vlan40 inet6 static
 ovs_type OVSBridge
 ovs_bridge asw-host
 ovs_options asw-host 40
 address fda0:7a62:28::1/64
```

7.5. Activation de la configuration IPv6 automatique pour le réseau de conteneurs

Pour que les hôtes du réseau de conteneurs obtiennent automatiquement une configuration IPv6, il faut que le routeur assure les annonces auprès de ces voisins. Un moyen simple pour assurer la configuration SLAAC des hôtes voisins du routeur consiste à utiliser le paquet radvd.

On débute par l'installation de ce paquet.

```
$ sudo apt install radvd
...
Préparation du dépaquetage de .../radvd_1%3a2.17-2+b1_amd64.deb ...
Dépaquetage de radvd (1:2.17-2+b1) ...
Paramétrage de radvd (1:2.17-2+b1) ...
Job for radvd.service failed because the control process exited with error code.
See "systemctl status radvd.service" and "journalctl -xe" for details.
invoke-rc.d: initscript radvd, action "start" failed.
● radvd.service - Router advertisement daemon for IPv6
     Loaded: loaded (/lib/systemd/system/radvd.service; disabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Sun 2020-09-13 22:32:10 CEST; 17ms ago
       Docs: man:radvd(8)
    Process: 2814 ExecStartPre=/usr/sbin/radvd --logmethod stderr_clean --configtest (code=exited, status=1/FAILURE)
```

On voit que le lancement du service a échoué.

Q44. Comment configurer le service radvd pour publier les annonces côté conteneurs ?

Rechercher les options utiles dans les pages de manuel du service : `man radvd.conf`.

Voici une copie du fichier de configuration `/etc/radvd.conf` de la maquette.

```
interface sw-vlan40
{
        AdvSendAdvert on;

        prefix fda0:7a62:28::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };

        RDNSS 2620:fe::fe
        {
        };
};
```

Attention ! Une fois le fichier créé, il ne faut pas oublier de redémarrer le service et de contrôler l'état de son fonctionnement.

```
$ sudo systemctl enable radvd
$ sudo systemctl restart radvd
$ systemctl status radvd
● radvd.service - Router advertisement daemon for IPv6
     Loaded: loaded (/lib/systemd/system/radvd.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-09-13 22:39:34 CEST; 5s ago
       Docs: man:radvd(8)
    Process: 2890 ExecStartPre=/usr/sbin/radvd --logmethod stderr_clean --configtest (code=exited, status=0/SUC
    Process: 2891 ExecStart=/usr/sbin/radvd --logmethod stderr_clean (code=exited, status=0/SUCCESS)
   Main PID: 2892 (radvd)
      Tasks: 2 (limit: 1142)
     Memory: 1.3M
     CGroup: /system.slice/radvd.service
             ├─2892 /usr/sbin/radvd --logmethod stderr_clean
             └─2893 /usr/sbin/radvd --logmethod stderr_clean

sept. 13 22:39:34 rtr systemd[1]: Starting Router advertisement daemon for IPv6...
sept. 13 22:39:34 rtr radvd[2890]: config file, /etc/radvd.conf, syntax ok
sept. 13 22:39:34 rtr radvd[2891]: version 2.17 started
sept. 13 22:39:34 rtr systemd[1]: Started Router advertisement daemon for IPv6.
```

7.6. Ajout des routes par défaut vers le réseau opérateur

Pour joindre l'Internet situé au delà du routeur bleu, il est nécessaire d'ajouter une route par défaut pour chaque protocole de la couche réseau : IPv4 et IPv6.

Attention ! Les tests de connectivité vers l'Internet supposent que le routeur bleu soit fonctionnel.

Q45. Comment ajouter manuellement les routes par défaut IPv4 et IPv6 vers le routeur bleu ?

Consulter les pages de manuel sur le routage avec la commande : `man ip-route`.

Sachant que le site distant est raccordé via une liaison point à point unique, on choisit de désigner la destination par l'interface de la liaison.

```
$ sudo ip route add default dev ppp0
$ sudo ip -6 route add default dev ppp0
```

Q46. Quels sont les tests de connectivité qui permettent valider la communication vers l'Internet en passant par le routeur bleu ?

Au niveau de la couche réseau, on lance les requêtes ICMP classques.

Attention ! Les deux exemples de tests ci-dessous prennent les conteneurs comme point de départ. L'idée est de parcourir la chaîne de communication la plus longue. Si les conteneurs ne sont pas disponibles, il est tout à possible de prendre le routeur vert comme origine.

Voici un exemple de test pour IPv4.

```
$ for i in {0..2}; do lxc exec container$i -- ping -q -c2 9.9.9.9; done
```

On change l'adresse de destination IPv6.

```
$ for i in {0..2}; do lxc exec container$i -- ping -q -c2 2620:fe::fe; done
```

Q47. Comment appliquer ces routes statiques dans la configuration système pour qu'elles soient activées à chaque établissement de session PPP ?

Il faut parcourir l'arborescence du répertoire `/etc/ppp/` pour repérer les scripts exécutés lors de l'ouverture de session. Créer un script pour chaque protocole de couche réseau qui ajoute la route statique voulue.

- Pour IPv4, le répertoire est `/etc/ppp/ip-up.d/`. Voici une copie du script exécutable `staticroute`.

```
#!/bin/sh

if [ -z "${CONNECT_TIME}" ]; then
 ip route add default dev ${PPP_IFACE}
fi
```

- Pour IPv6, le répertoire est `/etc/ppp/ipv6-up.d/`. Voici une copie du script exécutable `staticroute`.

```
#!/bin/sh

if [ -z "${CONNECT_TIME}" ]; then
 ip -6 route add default dev ${PPP_IFACE}
fi
```

7.7. Installation du gestionnaire de conteneurs LXD

Sur le routeur vert, la gestion des conteneurs est confiée à LXD. Pour des raisons de rapidité de mise en œuvre, on choisit de passer par le gestionnaire de paquets snapd pour l'installation des outils.

Q48. Comment installer le gestionnaire de paquets snap sur une distribution Debian GNU/Linux ?

Effectuer une recherche dans les paquets fournis via APT.

Il existe tout simplement un paquet appelé snapd.

```
$ sudo apt install snapd
```

Q49. Comment installer le gestionnaire de conteneurs LXD ?

Rechercher dans la liste des snaps.

Le snap s'appelle tout simplement lxd.

```
$ sudo snap install lxd
2020-09-21T18:12:31+02:00 INFO Waiting for automatic snapd restart...
Warning: /snap/bin was not found in your $PATH. If you've not restarted your session since you
         installed snapd, try doing that. Please see https://forum.snapcraft.io/t/9469 for more
         details.

lxd 4.6 from Canonical✓ installed
```

On peut lister les snaps installés.

```
$ snap list
Name    Version   Rev    Tracking       Publisher   Notes
core18  20200724  1885   latest/stable  canonical✓  base
lxd     4.6       17320  latest/stable  canonical✓  -
snapd   2.46.1    9279   latest/stable  canonical✓  snapd
```

Q50. Comment faire pour que l'utilisateur normal `etu` ait la capacité à gérer les conteneurs ?

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

Il faut que l'utilisteur normal appartienne au groupe système `lxd` pour qu'il est tous les droits sur la gestion des conteneurs.

```
$ sudo adduser etu lxd
```

Attention ! il faut se déconnecter/reconnecter pour bénéficier de la nouvelle attribution de groupe. On peut utiliser la commande groups pour vérifier le résultats.

```
$ groups
etu adm cdrom floppy sudo audio dip video plugdev staff netdev lxd
```

7.8. Configuration du gestionnaire de conteneurs LXD

Q51. Quelle est l'instruction de configuration initiale du gestionnaire LXD ?

Utiliser l'aide de la commande lxd.

C'est l'instruction `lxd init` qui nous intéresse.

Voici une copie d'écran de son exécution.

```
$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (btrfs, dir, lvm, ceph) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty disk or partition? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=13GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]: no
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: sw-vlan40
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: yes
config: {}
networks: []
storage_pools:
- config:
    size: 13GB
  description: ""
  name: default
  driver: btrfs
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: macvlan
      parent: sw-vlan40
      type: nic
    root:
      path: /
      pool: default
      type: disk
  name: default
cluster: null
```

Q52. Comment changer le type de raccordement défini par le paramètre `nictype` de `macvlan` à `bridged` ?

Rechercher dans les options d'édition des paramètres du profil avec la commande lxc.

Il faut suivre les champs du fichier yaml de description du profil.

```
$ lxc profile device set default eth0 nictype bridged
$ lxc profile device get default eth0 nictype
bridged
```

Q53. Quelle est l'instruction qui permet d'afficher le profil par défaut des conteneur ?

Rechercher dans les options de la commande lxc.

Voici un exemple d'exécution.

```
$ lxc profile show default
config: {}
description: Default LXD profile
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: sw-vlan40
    type: nic
  root:
    path: /
    pool: default
    type: disk
name: default
used_by: []
```

Q54. Quelle est l'instruction de lancement d'un conteneur ?

Rechercher dans les options de la commande lxc.

Tester son exécution avec un conteneur de type `debian/bullseye`.

Voici un exemple d'exécution.

```
$ lxc launch images:debian/bullseye container0
Creating container0
Starting container0
$ lxc launch images:debian/bullseye container1
Starting container1
$ lxc launch images:debian/bullseye container2
Starting container2
$ lxc ls
+------------+---------+------+-----------------------------------------+-----------+-----------+
|    NAME    |  STATE  | IPV4 |                  IPV6                   |   TYPE    | SNAPSHOTS |
+------------+---------+------+-----------------------------------------+-----------+-----------+
| container0 | RUNNING |      | fda0:7a62:28:0:216:3eff:feda:e1a (eth0) | CONTAINER | 0         |
+------------+---------+------+-----------------------------------------+-----------+-----------+
| container1 | RUNNING |      | fda0:7a62:28:0:216:3eff:fec4:d325 (eth0) | CONTAINER | 0         |
+------------+---------+------+-----------------------------------------+-----------+-----------+
| container2 | RUNNING |      | fda0:7a62:28:0:216:3eff:fe66:86fb (eth0) | CONTAINER | 0         |
+------------+---------+------+-----------------------------------------+-----------+-----------+
```

Q55. Comment appliquer une configuration IPv4 statique à chaque conteneur ?

Identifier le fichier de configuration système et modifier ce fichier pour chaque conteneur

En mode “manuel“, on édite directement le fichier `/etc/network/interfaces` dans chacun des trois conteneurs avec une commande comme celle-ci :

```
$ lxc exec container0 -- vim /etc/network/interfaces
```

pour ce qui est du resolver DNS qui est identique dans chaque conteneur, on peut utiliser une boucle.

```
$ for i in {0..2}; do lxc exec container$i -- sh -c "echo nameserver 9.9.9.9 > /etc/resolv.conf"; done
```

Une fois la configuration complétée, on redémarre les conteneurs pour vérifier que tous les paramètres ont bien été appliqués.

```
$ for i in {0..2}; do lxc restart container$i; done
```

Enfin, on peut relever le résultat avec la commande `lxc ls`.

```
$ lxc ls
+------------+---------+---------------------+-----------------------------------------+-----------+----------
|    NAME    |  STATE  |        IPV4         |                  IPV6                   |   TYPE    | SNAPSHOTS
+------------+---------+---------------------+-----------------------------------------+-----------+----------
| container0 | RUNNING | 203.0.113.10 (eth0) | fda0:7a62:28:0:216:3eff:feda:e1a (eth0) | CONTAINER | 0
+------------+---------+---------------------+-----------------------------------------+-----------+----------
| container1 | RUNNING | 203.0.113.11 (eth0) | fda0:7a62:28:0:216:3eff:fec4:d325 (eth0) | CONTAINER | 0
+------------+---------+---------------------+-----------------------------------------+-----------+----------
| container2 | RUNNING | 203.0.113.12 (eth0) | fda0:7a62:28:0:216:3eff:fe66:86fb (eth0) | CONTAINER | 0
+------------+---------+---------------------+-----------------------------------------+-----------+----------
```