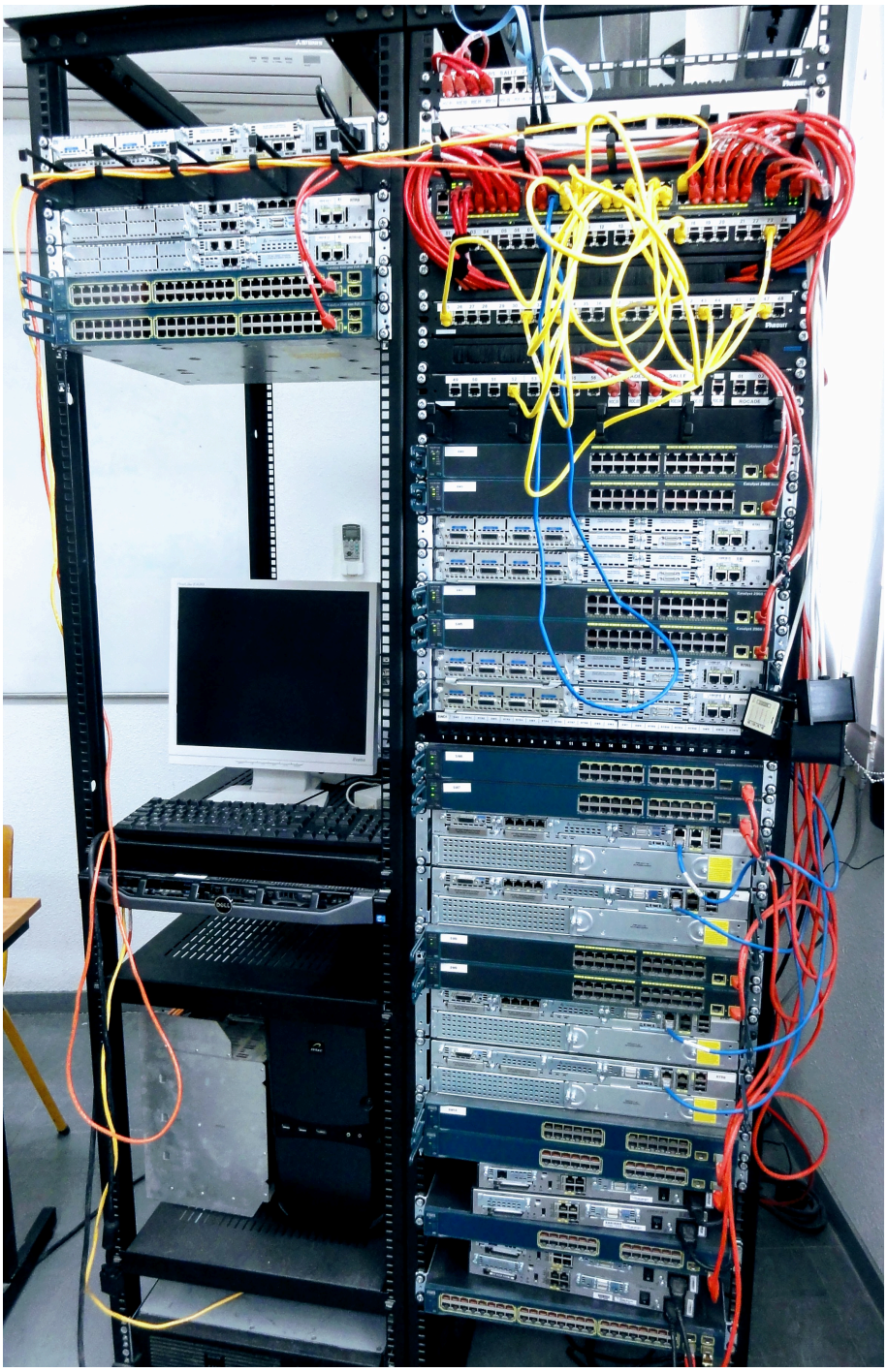
# Manuel de Travaux Pratiques Interconnexion LAN/WAN

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Ce manuel regroupe les supports du cycle de travaux pratiques sur le thème de l'interconnexion réseau LAN/WAN. La partie WAN utilise un commutateur RNIS comme support de trames HDLC.



# Table des matières

CHAPITRE 1

# Configuration des fonctions réseau & compilation du noyau Linux

**Résumé**

**Dans ce support de travaux pratiques, on se propose de préparer un système GNU/Linux pour être utilisé comme équipement d'interconnexion réseau. Après avoir passé en revue les fonctions réseau utiles du noyau Linux et sélectionné les pilotes des périphériques effectivement présents sur la plateforme matérielle, on construit un paquet de noyau Linux à partir de ses sources.**

# Table des matières



## 1.1. Le noyau courant et son arborescence

Avant d'attaquer la compilation d'un nouveau noyau à partir de ses sources, on doit identifier et localiser les différents composants du noyau en cours d'exécution sur le système.

Le jeu de questions ci-dessous suppose que la configuration système est directement issue de l'installation de la distribution Debian GNU/Linux. Le noyau courant exécuté est fourni via un paquet de la distribution.

Q1. Quelle est la commande `UNIX` usuelle qui identifie le noyau et sa version ?

Effectuer une recherche dans les pages de manuels des commandes installées sur le système avec une requête du type : `apropos informations, système`.

C'est la commande uname qui identifie le noyau courant. Pour interroger les pages de manuels à l'aide de la commande apropos, il faut que les paquets correspondant soient installés et que l'index de recherche soit construit.

Pour interroger les pages de manuels, on contrôle la liste des paquets correspondants installés et on lance manuellement la construction de l'index de recherche :

```
$ aptitude search ~imanpages
i   manpages          - Pages de manuel pour le système GNU/Linux
i A manpages-dev      - Pages de manuel sur l'utilisation de GNU/Linux pour le développement
i   manpages-fr       - Version française des pages de manuel sur l'utilisation de GNU/Linux
i   manpages-fr-extra - Version française des pages de manuel

<snip/>
# /etc/cron.daily/man-db

<snip/>
$ apropos -a informations système
dumpe2fs (8)    - Afficher des informations sur le système de fichiers ext2/ext3/ext4
fstab (5)       - Informations statiques sur les systèmes de fichiers
proc (5)        - Pseudosystème de fichiers d'informations sur les processus
uname (1)       - Afficher des informations sur le système
```

Pour obtenir la version courante du noyau exécuté :

```
$ uname -a
Linux vm0 4.12.0-2-686-pae #1 SMP Debian 4.12.12-2 (2017-09-11) i686 GNU/Linux
```

Q2. Où est placée l'image de la partie monolithique du noyau courant ?

Repérer le paquet Debian correspondant au noyau et retrouver l'image dans la liste des fichiers de ce paquet.

Une fois la version courante du noyau identifiée à l'aide de la commande uname, on peut faire la correspondance avec les paquets de noyau installés.

```
$ aptitude search ~ilinux-image
i A linux-image-4.12.0-1-686-pae - Linux 4.12 for modern PCs
i   linux-image-4.12.0-2-686-pae - Linux 4.12 for modern PCs
i   linux-image-686-pae          - Linux pour PC modernes - métapaquet
```

Connaissant le nom du paquet de noyau installé on peut lister les fichiers qu'il contient. À partir de cette liste on peut localiser la partie monolithique du noyau ainsi que ses modules dans l'arborescence du système de fichiers.

C'est dans le répertoire `/boot` que sont placées les images des noyaux disponibles sur un système GNU/Linux.

```
$ ls -A1 /boot/ | grep 4.12.0-2
config-4.12.0-2-686-pae ❶
initrd.img-4.12.0-2-686-pae ❷
System.map-4.12.0-2-686-pae ❸
vmlinuz-4.12.0-2-686-pae ❹
```

❶ Fichier de configuration du noyau de la distribution. Il contient l'ensemble des options qui ont été sélectionnées par le responsable du paquet. C'est une configuration très complète dans la mesure où un noyau publié dans une distribution doit supporter le maximum de matériel.

❷ Image compressée du disque RAM d'initialisation contenant une arborescence racine simplifée, des outils et l'ensemble des modules du noyau. Cette technique d'initialisation est la seule qui puisse fonctionner sur des systèmes sans disque dur où sur lesquels aucun système GNU/Linux n'a encore été installé.

❸ Fichier de cartographie des appels de fonctions du noyau. Cette cartographie est une aide à la mise au point pour les développeurs. On y trouve une identification nominative des fonctions en cas de problème au lieu d'adresses numériques en hexadécimal.

❹ Fichier image de la partie monolithique du noyau. C'est ce fichier qui est utilisé par le gestionnaire de démarrage pour lancer le système d'exploitation. Le gestionnaire de démarrage y accède directement à l'aide d'un appel BIOS.

Q3. Où sont placés les fichiers des modules correspondant au noyau courant ?

Comme dans le cas précédent, la liste des fichiers du paquet permet de retrouver l'arborescence de stockage des modules.

On peut parcourir la liste des fichiers contenus dans le paquet de noyau et effectuer des recherches par mots clés en utilisant la commande suivante :

```
$ dpkg -L linux-image-4.12.0-2-686-pae | egrep -e 'kernel$'
/lib/modules/4.12.0-2-686-pae/kernel
/lib/modules/4.12.0-2-686-pae/kernel/arch/x86/kernel
```

La liste ci-dessus montre que les modules du noyau sont placés dans le répertoire `/lib/modules/4.12.0-2-686-pae/kernel`.

Q4. Dans quel cas de figure utilise-t-on l'arborescence ou le disque RAM ?

Il faut bien différencier l'utilisation du disque RAM `initrd-*` de l'arborescence installée sur le disque du système.

Le fichier image du disque RAM d'initialisation a déjà été identifié ci-dessus.

Ce fichier est utilisé lors du lancement du système d'exploitation. Il est reconnu par le gestionnaire de démarrage de la même façon que la partie monolithique du noyau. Une fois le système complètement initialisé, les opérations de (chargement|déchargement) des modules utilisent l'arborescence du dique dur : `/lib/modules/`uname -r`/`.

Q5. Que contiennent les arborescences `/proc` et `/sys` ?

Consulter les documents ressource sysfs et Linux Filesystem Hierarchy

L'arborescence `/sys` est une représentation visible de l'arbre des périphériques physiques vus par le noyau. Elle est construite dynamiquement en fonction des branchements «à

chaud» effectués sur les différents bus de la machine. Les informations répertoriées dans cette arborescence sont du type : nom de périphérique, canal DMA, vecteur d'interruption, tensions d'alimentation, etc.

L'arborescence `/proc` comprend l'ensemble des paramètres du noyau en cours d'exécution. Ces paramètres sont modifiables en cours de fonctionnement. L'exemple emblématique, vis-à-vis de ces travaux pratiques est donné par l'ensemble des «réglages» possibles sur les machines d'états de la pile des protocoles réseau. La liste des paramètre donnée par la commande `ls /proc/sys/net/ipv4/` en donne un aperçu.

Q6. Quelle est la commande qui permet de lister les modules chargés en mémoire ?

À quel paquet appartient elle ?

Rechercher dans la base de données des paquets de la distribution les informations relatives aux manipulations sur les modules.

```
aptitude search '?description("modules du noyau Linux")'
i A kmod   - outils pour gérer les modules du noyau Linux
```

Le paquet `kmod` fournit la commande lsmod ainsi que les autres outils de manipulation sur les modules.

```
$ dpkg -L kmod | grep sbin/
/sbin/depmod
/sbin/insmod
/sbin/lsmod
/sbin/modinfo
/sbin/modprobe
/sbin/rmmod
```

Q7. Quelles sont les commandes qui permettent de charger un module en mémoire «manuellement» ?

Identifier celle qui traite automatiquement les dépendances entre modules.

Rechercher les informations dans la liste des fichiers du paquet ainsi que dans les pages de manuels des commandes.

On dispose de deux commandes : insmod et modprobe. Seule la commande modprobe traite les dépendances au (chargement|déchargement) d'un module. Illustration avec un module de gestion des dispositifs de stockage sur le bus USB.

```
# modprobe -v usb-storage
insmod /lib/modules/4.12.0-2-686-pae/kernel/drivers/usb/storage/usb-storage.ko
```

Q8. Quelles sont les commandes qui permettent de retirer un module de la mémoire «manuellement» ?

Identifier les options de la commande qui traite automatiquement les dépendances entre modules.

Rechercher les informations dans les pages de manuels des commandes.

Comme dans le cas précédent, c'est la commande modprobe qui retire de la mémoire les modules associés au déchargement.

```
# modprobe -rv usb-storage
rmmod usb_storage
```

## 1.2. Les sources du noyau Linux

Il faut bien reconnaître que s'attaquer à toutes les options de configuration du noyau Linux en partant de zéro est une tâche particulièrement ardue. Pour rendre la démarche plus aisée, on se propose de partir de la configuration fournie avec le paquet de la distribution. En procédant par modifications élémentaires à partir de cette configuration réputée sûre puisque permettant le fonctionnement du système actuel, on limite ainsi les possibilités d'erreurs.

Q9. Quels sont les principaux canaux de diffusion des sources du noyau Linux ?

Rechercher un site web, un dépôt de code en ligne et le nom du paquet de la distribution.

- Le site principal de publication des sources du noyau Linux est à l'adresse http://www.kernel.org/.
- Le développement du système de contrôle de version git a été initié par les développeurs du noyau Linux. Depuis, des services en lignes ont été bâtis à partir de git. Les branches de développement du noyau sont disponibles sur le site GitHub à l'adresse https://github.com/torvalds/linux.
- La distribution Debian GNU/Linux propose des paquets contenant les sources qui on servi à construire les paquets de noyau. Pour identifier ces paquets, on effectue une recherche dans le catalogue de la distribution.

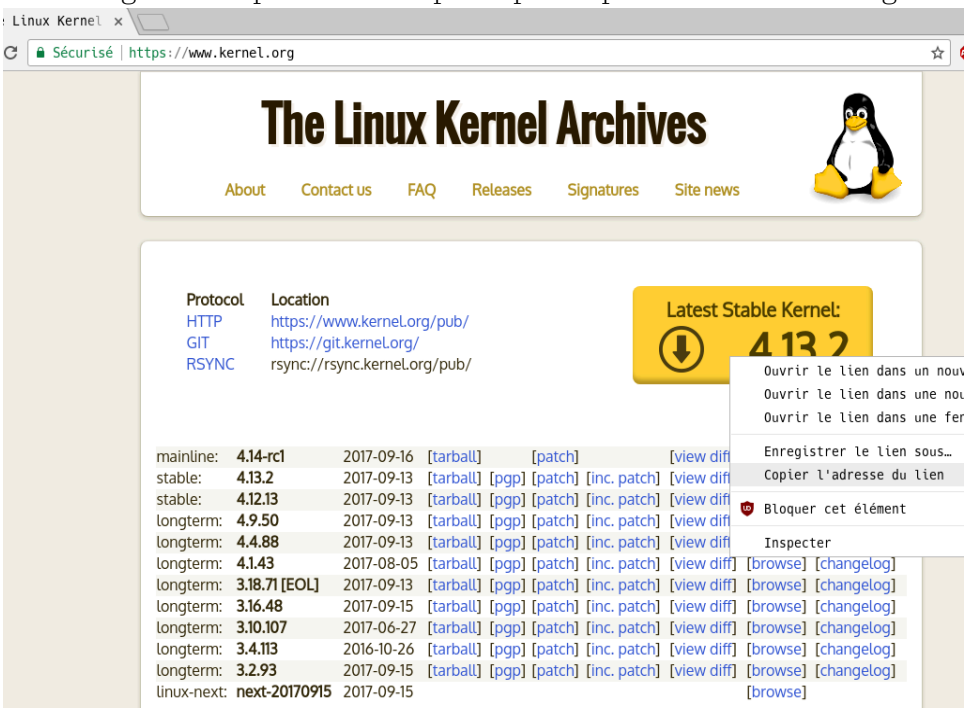
```
$ aptitude search linux-source
p   linux-source      - Linux kernel source (meta-package)
p   linux-source-4.12 - Linux kernel source for version 4.12 with Debian patches
```

Q10. Donner un exemple de téléchargement des sources du noyau sans passer par une interface graphique ?

Rechercher un outil permettant de lancer un téléchargement HTTP(s).

Lorsque l'on utilise des serveurs qui ne possèdent ni écran ni clavier, il est nécessaire d'effectuer les opérations sans recours à une interface graphique. Les outils les plus courants dans ce contexte sont `url` et `wget`.

- Téléchargement à partir du site principal de publication kernel.org.




Téléchargement des sources du noyau Linux - vue complète

```
$ wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.13.2.tar.xz
--  https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.13.2.tar.xz
Résolution de cdn.kernel.org (cdn.kernel.org)… 2a04:4e42:1d::432, 151.101.121.176
Connexion à cdn.kernel.org (cdn.kernel.org)|2a04:4e42:1d::432|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 100574388 (96M) [application/x-xz]
Sauvegarde en : « linux-4.13.2.tar.xz »
<snip/>
```

- Téléchargement à partir du gestionnaire de paquets de la distribution.

```
# aptitude install linux-source
Les NOUVEAUX paquets suivants vont être installés :
  linux-source linux-source-4.12{a} make{a}
0 paquets mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 102 Mo d'archives. Après dépaquetage, 103 Mo
seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

Q11. Quel est le groupe système qui permet de compiler un noyau ou des modules ?

Rechercher le groupe consacré aux manipulations des sources dans la liste des groupes système.

On cherche la chaîne `src` dans le fichier `/etc/group` et on ajoute l'utilisateur normal dans ce groupe.

```
# grep src /etc/group
src:x:40:

# adduser etu src
Ajout de l'utilisateur « etu » au groupe « src »...
Ajout de l'utilisateur etu au groupe src
Fait.

# id etu
uid=1000(etu) gid=1000(etu) groupes=1000(etu),24(cdrom),25(floppy),
29(audio),30(dip),40(src),44(video),46(plugdev)
```

Q12. Quel est le répertoire du système dédié au stockage des sources du noyau Linux ?

Faire une recherche dans le document Linux Filesystem Hierarchy.

Vérifier que ce répertoire appartient bien au groupe `src`.

C'est le répertoire `/usr/src` qui doit accueillir les sources du noyau.

On vérifie que les membres du groupe système `src` ont bien accès en écriture à ce répertoire.

```
# chgrp -R src /usr/src

# chmod 2775 /usr/src
```

Q13. Quelles sont les commandes «rituelles» d'installation des sources du noyau Linux ?

Pour chaque commande, expliquer les opérations réalisées et justifier le choix des options.

Il faut consulter les ressources suivantes : Debian Linux Kernel Handbook et Manuel de référence Debian - Chapitre 9.

Pour traiter cette question, on utilise le fichier source obtenu à l'aide du gestionnaire de paquets. D'après les documents de référence on doit utiliser la séquence de commandes suivante.

```
$ cd /usr/src/
$ tar xf linux-source-4.12.tar.xz❶
$ ln -s linux-source-4.12 linux❷
$ cd linux
$ cp /boot/config-4.12.0-2-686-pae .config❸
$ make menuconfig❹
```

❶ Extraction de l'arborescence des sources du noyau.

❷ Création d'un lien symbolique sur l'arborescence de travail. L'utilisation de ce lien permet de conserver plusieurs arborescences de sources. De cette façon, on peut travailler sur plusieurs versions de noyau.

❸ Copie du fichier de configuration fourni avec le paquet de noyau. Ce fichier est réputé fiable puisqu'il correspond au noyau en cours d'exécution et que le système est opérationnel.

Cette opération est optionnelle. En l'absence du fichier `.config` dans l'arborescence des sources du noyau, la commande suivante procède à la copie de la configuration du noyau courant.

❹ Lancement de l'interface des menus de configuration des options du noyau Linux. C'est à ce niveau que les «choses sérieuses» commencent.

La dernière commande n'est utilisable que si le paquet de bibliothèques de développement ncurses est installé. `aptitude install libncurses-dev`.

## 1.3. La configuration du noyau Linux

On se propose de configurer un système d'interconnexion. Le noyau correspondant doit donc comprendre les éléments suivants.

- Un cœur système monolithique : microprocesseur, périphériques non réseau et système de fichiers
- Le support des fonctions réseau nécessaires au routage
- Le support du filtrage netfilter sous forme modulaire
- Un pilote d'interface réseau Ethernet sous forme modulaire
- Les fonctions de l'ancien sous-système RNIS sous forme modulaire
- Un pilote d'interface RNIS sous forme modulaire

Q14. Quelle est la commande utilisée pour les opérations de configuration et de compilation ?

Toutes les opérations de compilation du noyau étant basées sur des `Makefiles`, c'est la commande make qui sert aussi pour la configuration.

Q15. Comment obtenir la liste des options de cette commande ?

La commande `make help` donne la liste des options disponibles.

Q16. Quelles sont les 3 options de configuration du noyau ?
Préciser les différences entre ces 3 options.

Les 3 commandes sont `make config`, `make menuconfig` et `make xconfig`.

Il est préférable d'utiliser la commande `make menuconfig`. C'est le meilleur compromis entre facilité de navigation et administration distante. Les bibliothèques de développement `ncurses` ne consomment que très peu de ressources CPU et l'utilisation d'une interface graphique sur un serveur est à proscrire.

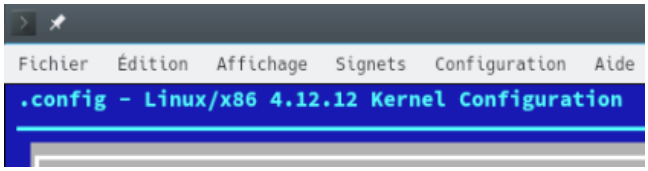
Q17. Sans opération préalable, quel est le fichier contenant les options de configuration du noyau utilisé ?

C'est le fichier texte `.config` qui contient l'ensemble des options de configuration du noyau Linux courant. Il est placé à la racine de l'arborescence des sources du noyau ; soit le répertoire `/usr/src/linux` dans notre cas.

Le fichier «patron» de configuration pour ces travaux pratiques doit donc être copié dans le répertoire `/usr/src/linux` et renommé `.config`. L'opération a déjà été effectuée à la Q : Q13.

Q18. Une fois la commande de configuration exécutée, comment identifier la version du noyau à compiler ?

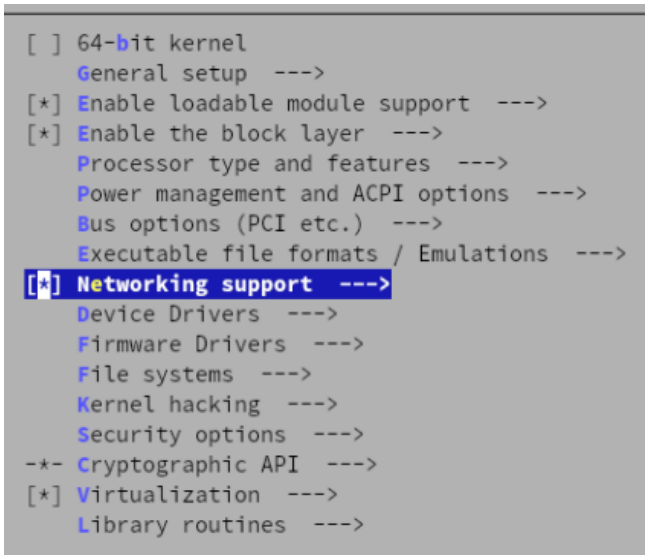
La version du noyau en cours de configuration est indiquée en haut à gauche de l'écran.




Identification version noyau Linux - vue complète

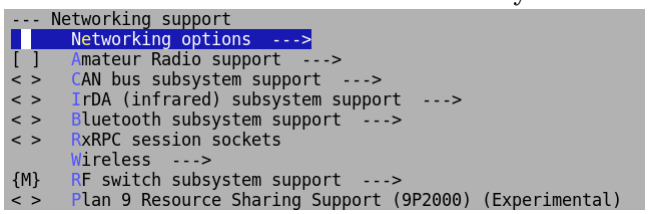
Q19. Quelles sont les options utiles des rubriques Networking Support et Networking options ?

On accède aux différents types de réseaux supportés par le noyau Linux via l'item Networking Support.




Accès aux fonctions réseau - vue complète

On accède aux fonctions réseau du noyau Linux via l'item Networking options.




Accès aux fonctions réseau du noyau Linux - vue complète

À partir du support Fonctions réseau du noyau Linux et de l'organisation des menus, on distingue les options génériques, telles que le support des sockets, des options spécifiques telles que celles relatives au filtrage.

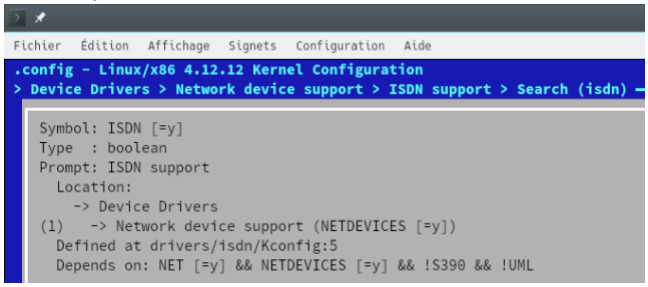
Q20. Quelles sont les options utiles des rubriques Device Drivers puis Network device support ?

Voir le support Fonctions réseau du noyau Linux pour s'orienter dans les options à sélectionner.

Pour accéder au catalogue des interfaces réseau supportées par le noyau il faut passer par la catégorie des pilotes de périphériques ou Device Drivers pour accéder à l'item Network device support.

Q21. Quelles sont les options utiles de la rubrique ISDN subsystem ?

À partir de la liste des pilotes de périphériques du noyau, on accède aux paramétrage du sous-système RNIS/ISDN.

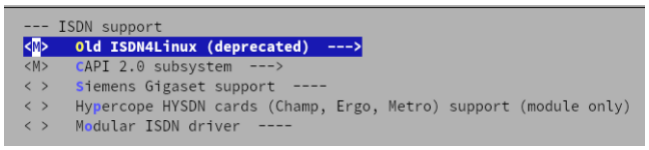



Accès au sous-système RNIS/ISDN - vue complète

Il existe trois types d'utilisation des connexions RNIS/ISDN dans le noyau Linux.

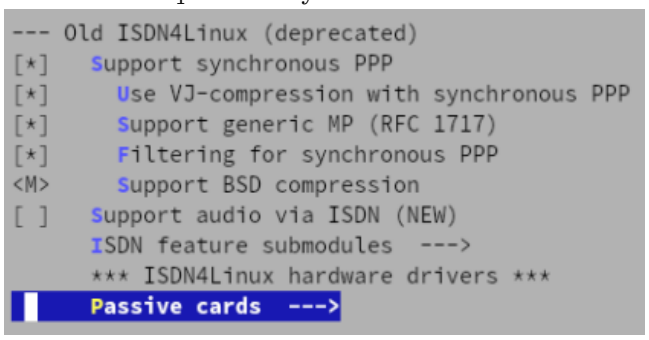
- Le plus récent utilise un mécanisme de sockets adapté aux fonctions réseau actuelles du noyau.
- Le plus ancien hérite des noyaux de la série 2.2.xx. Il comprend une machine d'état logicielle autonome de gestion de l'étbalissement du maintien et de la libération des connexions.

  C'est ce type de connexion que l'on utilise dans la suite des travaux pratiques de la série.
- Il existe un troisième type qui utilise le standard CAPI. Il s'agit d'une interface logicielle normalisée entre le noyau et le périphérique matériel.




Types de connexions RNIS/ISDN - vue complète

Le catalogue des paramètres utilisables avec le protocole PPP associé au sous-système RNIS/ISDN historique du noyau Linux est donné ci-dessous.




Paramètres PPP du sous-système RNIS/ISDN - vue complète

Le modèle des cartes implantées dans les postes de travaux pratiques est de type `AVM Fritz/PCI 2.0`.

## 1.4. La compilation & l'installation du nouveau noyau Linux

Q22. Quelle est l'option à utiliser avec les sources du noyau pour construire des paquets de la distribution Debian GNU/Linux ?

Rechercher la clé `deb` dans la liste des options du `Makefile` des sources du noyau.

La recherche dans les options permet d'identifier la directive de construction des paquets binaires de la distribution : `bindeb-pkg`.

```
$ make help | less
```

Suivant l'état antérieur de l'installation système, la liste des dépendances est plus ou moins importante lors du lancement de la compilation du noyau.

```
# sudo aptitude install -R fakeroot bison flex libelf-dev libssl-dev
```

Q23. Comment lancer la compilation du noyau ?

Pour faciliter les opérations de (dé|ré)installation du noyau, on se propose de construire un paquet Debian de noyau Linux. L'utilisation d'un paquet permet de s'assurer que tous les fichiers nécessaires ont bien été (copiés|supprimés) dans l'arborescence du système.

```
$ pwd
/usr/src/linux
$ make -j$(grep  -c  '^processor'  /proc/cpuinfo) bindeb-pkg
```

Q24. Quelles sont les étapes d'installation du noyau compilé ?

Quel outil faut-il utiliser pour gérer les paquets localement sur le système ?

Une fois les paquets de noyau construits, il ne reste plus qu'à procéder à l'installation de ces paquets locaux. Cette étape fait appel à l'outil de gestion de bas niveau des paquets Debian : dpkg. Cette opération nécessite les droits du super-utilisateur.

```
# pwd
/usr/src
# dpkg -i linux-image*.deb linux-libc*.deb
```

Après cette installation de paquet de noyau on peut valider la liste des paquets correspondant installés.

```
$ aptitude search ~ilinux-
```

Q25. Comment vérifier que le nouveau noyau sera disponible lors de l'initialisation du système ?

Identifier le gestionnaire d'amorce installé sur le système.

L'opération d'installation du paquet de noyau intègre l'ajout d'une nouvelle entrée dans le gestionnaire de démarrage.

On peut valider la liste des noyaux disponibles au niveau du gestionnaire d'amorce en faisant appel à la commande update-grub.

```
$$ sudo update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.2.14
Found initrd image: /boot/initrd.img-5.2.14
Found linux image: /boot/vmlinuz-5.2.0-2-amd64
Found initrd image: /boot/initrd.img-5.2.0-2-amd64
done
```

Une fois toutes ces étapes franchies, il ne reste plus qu'à relancer le système et vérifier que le noyau exécuté est bien celui qui a été recompilé à partir des sources.

## 1.5. Documents de référence

Debian Linux Kernel Handbook

Debian Linux Kernel Handbook : guide sur les techniques de construction d'un paquet Debian de noyau Linux.

Manuel de référence Debian

Manuel de référence Debian - Chapitre 9 : La section 9.7 traite des opérations de configuration et de compilation d'un noyau Linux.

CHAPITRE 2

# Configuration d'une interface RNIS en mode rawip

**Résumé**

**L'objectif de ce support de travaux pratiques est d'apprendre à configurer une interface de réseau étendu (WAN). À la différence d'une interface de réseau local, une interface de réseau étendu possède un très grand nombre d'options au niveau de la couche liaison. Dans le contexte de ces travaux pratiques on utilise la technologie RNIS en mode `rawip`. Les adresses IPv4 sont configurées manuellement à chaque extrémité de la liaison point à point.**

## Table des matières



## 2.1. Les outils de configuration d'une interface réseau

Avant d'aborder l'outil spécifique de configuration des options de l'interface RNIS au niveau liaison, voici un premier jeu de questions sur l'identification des interfaces réseau, la configuration IP et la résolution des noms de domaines.

Les questions ci-dessous reprennent les éléments de configuration abordés dans le support Configuration d'une interface de réseau local.

Voici une liste réduite des commandes qui permettent de traiter les questions. Les pages de manuels de ces commandes contiennent toutes les informations utiles au paramétrage des interfaces.

- dmesg : messages du système depuis son démarrage
- lspci : liste des périphériques connectés sur le bus PCI
- lsmod : liste des modules de pilotage de périphériques chargés
- ip : commande de visualisation et de configuration des paramètres réseau

Q26. Comment identifier les éléments matériels des interfaces réseau du poste de travaux pratiques ?

Utiliser les messages système de démarrage et surtout la liste des périphériques connectés sur le bus PCI.

La commande `$ dmesg | less` permet d'identifier les interfaces Ethernet et sans-fil. Aucune information n'est donnée sur les autres types d'interfaces.

Une recherche avec le mot clé `eth` dans les messages système permet de localiser les informations relatives au chargement du module de pilotage de l'interface Ethernet

```
[    0.528002] sky2 0000:02:00.0: Yukon-2 EC Ultra chip revision 3
[    0.528119]   alloc irq_desc for 28 on node -1
[    0.528121]   alloc kstat_irqs on node -1
[    0.528134] sky2 0000:02:00.0: irq 28 for MSI/MSI-X
[    0.528597] sky2 eth0: addr 00:1f:c6:01:26:71
```

De la même façon, on localise les informations sur l'interface sans-fil à l'aide du mot clé `wireless`.

```
[    5.570589] cfg80211: Using static regulatory domain info
[    5.570635] cfg80211: Regulatory domain: US
[    5.570677]  (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp)
[    5.570731]  (2402000 KHz - 2472000 KHz @ 40000 KHz), (600 mBi, 2700 mBm)
[    5.570776]  (5170000 KHz - 5190000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
[    5.570821]  (5190000 KHz - 5210000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
[    5.570866]  (5210000 KHz - 5230000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
[    5.570911]  (5230000 KHz - 5330000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
[    5.570956]  (5735000 KHz - 5835000 KHz @ 40000 KHz), (600 mBi, 3000 mBm)
[    5.571156] cfg80211: Calling CRDA for country: US
[    5.811100] usbcore: registered new interface driver usbserial
[    5.811158] USB Serial support registered for generic
[    6.051825] phy0: Selected rate control algorithm 'minstrel'
[    6.052315] phy0: hwaddr 00:15:af:51:d0:7d, RTL8187vB (default) V1 + rtl8225z2, rfkill mask 2
[    6.063101] rtl8187: Customer ID is 0x00
[    6.063176] Registered led device: rtl8187-phy0::tx
[    6.063240] Registered led device: rtl8187-phy0::rx
[    6.063716] rtl8187: wireless switch is on
```

Note

Bien sûr, les copies d'écran ci-dessus ne sont que des exemples, les références de composants changent d'une plateforme à l'autre.

La commande lspci liste les composants connectés au bus de la carte mère. À la différence des informations produites par la commande dmesg, cette liste est exhaustive.

```
$ lspci
<snipped/>
02:00.0 Ethernet controller: Marvell Technology Group Ltd. 88E8056 PCI-E Gigabit Ethernet Controller (re
03:00.0 SATA controller: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
03:00.1 IDE interface: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
05:01.0 Network controller: AVM GmbH Fritz!PCI v2.0 ISDN (rev 02)
05:03.0 FireWire (IEEE 1394): Agere Systems FW322/323 (rev 70)
```

On voit apparaître ci-dessus l'interface WAN.

Q27. Quelles sont les informations disponibles sur le type de média et le débit de l'interface LAN ? Est-il possible d'obtenir les mêmes information pour l'interface WAN ?

Rechercher les résultats de la négociation de bande passante, soit avec l'outil ethtool.

```
# ethtool eth4
Settings for eth4:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: Unknown
        Supports Wake-on: g
        Wake-on: d
        Link detected: yes
```

Il n'est pas possible d'obtenir les mêmes informations pour une interface WAN. Pour l'interface LAN tous les éléments du niveau liaison de données sont définis : le réseau Ethernet et le format de trame associé. Il ne reste que le débit à négocier sur les médias filaires en paires torsadées cuivre. À l'inverse, pour une interface WAN pratiquement tous les éléments du niveau liaison de données sont à paramétrer manuellement avant qu'un échange soit possible.

Q28. Quel est le script général d'initialisation des interfaces LAN réseau utilisé au démarrage du poste de travaux pratiques ? Ce script est-il utilisé pour l'interface WAN RNIS ?

Rechercher dans le répertoire des scripts d'initialisation des niveaux de démarrage (runlevels). Consulter la documentation Manuel de référence Debian : configuration du réseau. Retrouver dans les messages système si les interfaces réseau LAN et WAN sont initialisées en même temps.

Le script général utilisé lors de l'initialisation du système est le fichier `/etc/init.d/networking`. Il applique les paramètres de configuration contenus dans le fichier `/etc/network/interfaces`.

```
# cat /etc/network/interfaces
# The loopback interface
# Interfaces that comes with Debian Potato does not like to see
# "auto" option before "iface" for the first device specified.
iface lo inet loopback
auto lo

auto eth0
iface eth0 inet dhcp
```

Q29. Quelle est la syntaxe de la commande de configuration ip permettant d'affecter l'adresse IP du poste ?

Choisir les paramètres nécessaires à partir des options listées dans les pages de manuels. Revoir le support Configuration d'une interface de réseau local.

Q30. Quelle est la syntaxe de la commande route permettant d'affecter la passerelle par défaut du réseau local ?

Choisir les paramètres nécessaires à partir des options listées dans les pages de manuels. Revoir le support Configuration d'une interface de réseau local.

Q31. Comment valider le fonctionnement du protocole IP de la couche réseau ?

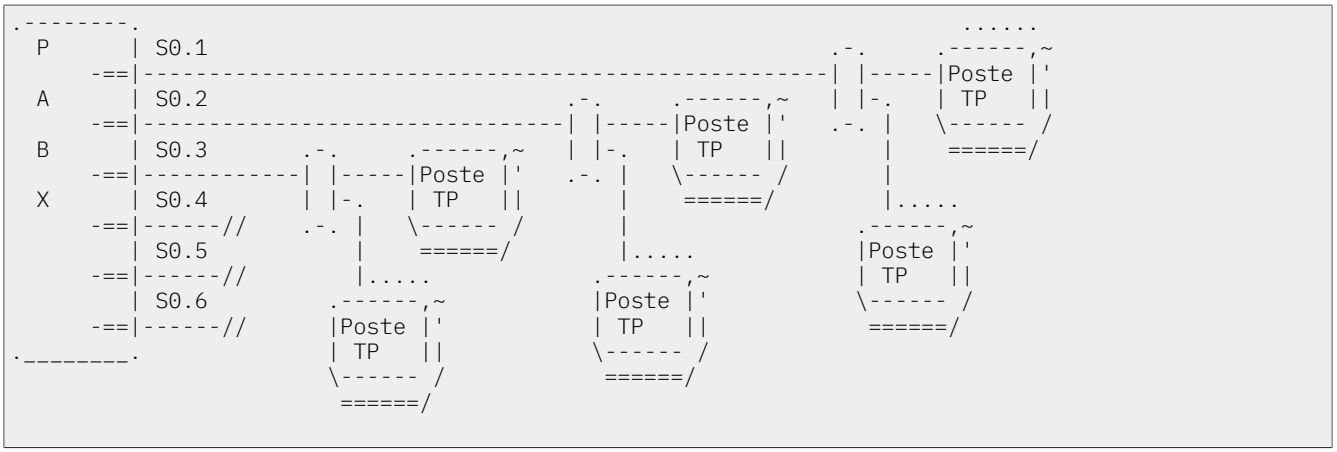
Attention au «piège du débutant» cette validation doit impérativement se faire au niveau réseau sans utiliser un service des couches supérieures tel que la résolution des noms par exemple.

Q32. Quel est le fichier de configuration utilisé par le resolver DNS pour faire la correspondance entre adresses IP et noms de domaines ?

Revoir le support Configuration d'une interface de réseau local.

## 2.2. La topologie RNIS et le sous-système du noyau LINUX

La topologie de base de la technologie RNIS est le bus. Il est donc nécessaire de réaliser une adaptation de la topologie étoile du câblage en paires torsadées cuivre du réseau Ethernet. On utilise des boîtiers de «mise en parallèle» des 8 fils du câble Ethernet.



Une fois la topologie physique en place, il faut identifier les éléments du noyau LINUX relatifs au sous-système RNIS. Que le noyau en cours d'exécution provienne de la distribution ou bien de la séance de travaux pratiques précédente, le sous-système RNIS a été compilé sous forme modulaire. C'est la méthode la plus pratique pour la mise au point des connexions réseau. On peut (charger|décharger) les modules autant de fois que nécessaire.

Comme le travail à effectuer traite des périphériques matériels du système, la documentation se trouve dans l'arborescence des sources du noyau. Cette documentation peut se présenter sous deux formes.

Le tarball des sources du noyau LINUX

Si les sources du noyau ont été directement téléchargés et installés dans le répertoire `/usr/src/linux/`, les fichiers de documentation sont placés dans le sous-répertoire `Documentation/isdn`.

Le paquet `linux-doc-2.6.xx`

Dans ce cas, le répertoire principal est `/usr/share/doc/linux-doc-2.6.xx/` et le sous-répertoire est le même que précédemment : `Documentation/isdn`.

Q33. Quel est le nom du module de pilotage de la carte RNIS ?

Consulter la liste des modules chargés et l'arborescence de stockage des modules disponibles : le répertoire `/lib/modules/2.6.xx`.

Dans le sous-système RNIS/ISDN «historique» du noyau Linux, le principal pilote utilisé est baptisé `HiSax`. On peut donc rechercher ce mot clé dans l'arborescence des modules du noyau courant.

```
# find /lib/modules/`uname -r`/ |grep -i hisax
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/elsa_cs.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hfc_usb.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/teles_cs.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/sedlbauer_cs.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hfc4s8s_l1.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax_fcpcipnp.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/avma1_cs.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax_isac.ko
/lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax_st5481.ko
```

En faisant correspondre la liste ci-dessus avec les informations données auparavant par la commande lspci, on identifie le module `hisax_fcpcipnp`.

Q34. Quelle est la commande à utiliser pour charger le module pilote de la carte RNIS ?

Consulter la liste des fichiers du paquet `kmod`.

C'est la commande modprobe qui permet de charger un module ainsi que ses dépendances.

```
# modprobe -v hisax_fcpcipnp
insmod /lib/modules/2.6.32-5-amd64/kernel/drivers/net/slhc.ko
insmod /lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/i4l/isdn.ko
insmod /lib/modules/2.6.32-5-amd64/kernel/lib/crc-ccitt.ko
insmod /lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax.ko
insmod /lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax_isac.ko
insmod /lib/modules/2.6.32-5-amd64/kernel/drivers/isdn/hisax/hisax_fcpcipnp.ko
```

Q35. Quels sont les messages systèmes qui indiquent que le module pilote de carte RNIS est correctement configuré ?

Rechercher dans les fichiers de messages systèmes contenant les informations sur le matériel. Vérifier que les messages systèmes annoncent que le canal D et les 2 canaux B sont disponibles.

L'analyse des messages système donne les informations suivantes.

```
[ 3315.748866] ISDN subsystem Rev: 1.1.2.3/1.1.2.3/1.1.2.2/1.1.2.3/1.1.2.2/1.1.2.2 loaded ❶
[ 3315.778268] HiSax: Linux Driver for passive ISDN cards
[ 3315.778270] HiSax: Version 3.5 (module)
[ 3315.778272] HiSax: Layer1 Revision 2.46.2.5 ❷
[ 3315.778274] HiSax: Layer2 Revision 2.30.2.4
[ 3315.778276] HiSax: TeiMgr Revision 2.20.2.3
[ 3315.778277] HiSax: Layer3 Revision 2.22.2.3
[ 3315.778279] HiSax: LinkLayer Revision 2.59.2.4
[ 3315.779643] hisax_isac: ISAC-S/ISAC-SX ISDN driver v0.1.0
[ 3315.781407] hisax_fcpcipnp: Fritz!Card PCI/PCIv2/PnP ISDN driver v0.0.1
[ 3315.781442] HiSax: Card 1 Protocol EDSS1 Id=fcpcipnp0 (0) ❸
[ 3315.781448] HiSax: DSS1 Rev. 2.32.2.3
[ 3315.781450] HiSax: 2 channels added
[ 3315.781452] HiSax: MAX_WAITING_CALLS added
[ 3315.781456] hisax_fcpcipnp: found adapter Fritz!Card PCI v2 at 0000:05:01.0
```

❶ Le sous-système RNIS du noyau Linux comprend la machine d'état d'établissement, de maintien et de libération des connexions.

❷ Tous les éléments de cette liste correspondent aux fonctions de gestion de la signalisation sur le canal D du bus RNIS.

❸ Ces messsages indiquent que les deux canaux B du bus RNIS sont ouverts et que l'interface RNIS est prête à être configurée.

Q36. Quels sont les paquets qui contiennent les outils de configuration d'interface RNIS/ISDN ?

Effectuer une recherche dans la base de données des paquets avec l'empreinte `isdn`. Installer les paquets relatifs à la configuration d'interface.

```
# aptitude search isdn
p   isdnactivecards   - ISDN utilities - active ISDN card support
p   isdnlog           - ISDN utilities - connection logger
p   isdnlog-data      - ISDN utilities - connection logger data
p   isdnutils         - ISDN utilities - dependency package
p   isdnutils-base    - ISDN utilities - minimal set
p   isdnutils-doc     - ISDN utilities - documentation
p   isdnutils-xtools  - ISDN utilities - graphical tools
p   isdnvbox          - ISDN utilities - answering machine dependency package
p   isdnvboxclient    - ISDN utilities - answering machine client
p   isdnvboxserver    - ISDN utilities - answering machine server
```

C'est le paquet `isdnutils-base` qui nous intéresse ici.

```
# aptitude install isdnutils-base
```

Q37. Quels sont les fichiers de périphériques ou device files associés aux interfaces RNIS ? Comment créer ces entrées ?

Effectuer des recherches dans le répertoire `/dev`. Rechercher le paquet qui contient le script MAKEDEV.

Si la commande `# find /dev/ -name \*isdn\*` ne donne aucun résultat, c'est qu'aucune entrée de périphérique n'a été créée auparavant. Dans ce cas, on doit procéder à une création manuelle à l'aide du script `/dev/MAKEDEV/`. La recherche des directives de création d'entrées RNIS dans le code de ce script permet d'identifier l'option `isdnbri`. On exécute alors les instructions suivantes.

```
# cd /dev && WRITE_ON_UDEV=yes MAKEDEV isdnbri && ln -s /dev/isdnctrl0 /dev/isdnctrl
# ls /dev/isdn[0-9]
/dev/isdn0  /dev/isdn1  /dev/isdn2  /dev/isdn3
/dev/isdn4  /dev/isdn5  /dev/isdn6  /dev/isdn7
/dev/isdn8  /dev/isdn9
```

Pour pouvoir utiliser MAKEDEV, il faut que le paquet correspondant ait été installé.

```
# aptitude show makedev
```

Q38. Quel est l'utilitaire de paramétrage des messages du sous-système RNIS ?

Utiliser la documentation `README.HiSax`.

C'est la commande isdnctrl qui sert à configurer les différents types d'interfaces.

Q39. Quelles sont les interfaces du sous-système qui transmettent les messages ?

Utiliser la documentation `README.HiSax`.

Par défaut, c'est le fichier `/dev/isdnctrl` qui sert de canal d'information. Il doit exister dans le répertoire `/dev/`.

```
# isdnctrl addif isdn0
Can't open /dev/isdnctrl or /dev/isdn/isdnctrl: No such file or directory
# ln -s /dev/isdnctrl0 /dev/isdnctrl
# isdnctrl addif isdn0
isdn0 added
```

Q40. Quelle commande utiliser pour envoyer les messages sur une console ?

Utiliser la documentation `README.HiSax`.

Comme l'entrée de périphérique `/dev/isdnctrl0` est de type caractère, il est possible d'afficher son contenu directement à la console.

```
# ls -lAh /dev/isdnctrl0
crw-rw---- 1 root dialout 45, 64 12 oct.  00:16 /dev/isdnctrl0
```

En phase de mise au point d'une connexion, la méthode d'affichage la plus simple consiste à dédier une console à cet usage.



Avertissement

La commande ci-dessous verrouille l'accès au prériphérique. Il faut impérativement libérer la ressource rapidement avec la séquence Ctrl+C.

```
# cat /dev/isdnctrl
85:31.79 L3DC State ST_L3_LC_REL Event EV_ESTABLISH_REQ
85:31.79 L3DC ChangeState ST_L3_LC_ESTAB_WAIT
85:31.79 tei State ST_TEI_NOP Event EV_IDREQ
85:31.79 tei assign request ri 60784
85:31.79 Card1 -> PH_DATA_REQ: UI[0]C (sapi 63, tei 127)
85:31.79 tei ChangeState ST_TEI_IDREQ
85:31.97 tei State ST_TEI_IDREQ Event EV_ASSIGN
85:31.97 tei identity assign ri 60784 tei 73
85:31.97 tei ChangeState ST_TEI_NOP
85:31.97 Card1 -> PH_DATA_REQ: SABME[1]C (sapi 0, tei 73)
85:32.07 L3DC State ST_L3_LC_ESTAB_WAIT Event EV_ESTABLISH_CNF
85:32.07 L3DC ChangeState ST_L3_LC_ESTAB
85:32.07 Card1 -> PH_DATA_REQ: I[0](ns 0, nr 0)C (sapi 0, tei 73)
85:32.48 Card1 -> PH_DATA_REQ: RR[0](nr 1)R (sapi 0, tei 73)
85:52.91 Card1 -> PH_DATA_REQ: I[0](ns 1, nr 1)C (sapi 0, tei 73)
85:53.18 Card1 -> PH_DATA_REQ: I[0](ns 2, nr 2)C (sapi 0, tei 73)
85:53.35 L3DC State ST_L3_LC_ESTAB Event EV_RELEASE_REQ
85:53.35 L3DC ChangeState ST_L3_LC_REL_DELAY
85:53.35 Card1 -> PH_DATA_REQ: RR[0](nr 3)R (sapi 0, tei 73)
86:06.05 Card1 -> PH_DATA_REQ: UA[1]R (sapi 0, tei 73)
86:06.05 L3DC State ST_L3_LC_REL_DELAY Event EV_RELEASE_IND
86:06.05 L3DC ChangeState ST_L3_LC_REL
```

La copie d'écran ci-dessus fait apparaître les différentes étapes d'un appel téléphonique qui n'aboutit pas.

## 2.3. La connexion directe en mode rawip

Dans cette partie, on teste la communication de bout en bout avec l'encapsulation `rawip`. Cette encapsulation utilise uniquement les numéros de téléphone pour établir la connexion. La configuration réseau des interfaces doit être établie avant la connexion «téléphonique» RNIS.

Tableau 2.1. Plan d'adressage IP & téléphonique

| Bus | Poste 1 | N° Tél. | Adresse IP | Poste 2 | N° Tél. | Adresse IP |
|---|---|---|---|---|---|---|
| S0.1 | alderaan | 104 | `192.168.100.1/29` | bespin | 105 | `192.168.100.2/29` |
| S0.2 | centares | 106 | `192.168.100.9/29` | coruscant | 107 | `192.168.100.10/29` |

| Bus | Poste 1 | N° Tél. | Adresse IP | Poste 2 | N° Tél. | Adresse IP |
|---|---|---|---|---|---|---|
| S0.3 | dagobah | 108 | `192.168.100.17/29` | endor | 109 | `192.168.100.18/29` |
| S0.4 | felucia | 110 | `192.168.100.25/29` | geonosis | 111 | `192.168.100.26/29` |
| S0.5 | hoth | 112 | `192.168.100.33/29` | mustafar | 113 | `192.168.100.34/29` |
| S0.6 | naboo | 114 | `192.168.100.41/29` | tatooine | 115 | `192.168.100.42/29` |

Comme il existe une grande variété de paramètres pour les connexions RNIS, il existe un outil de configuration dédié : isdnctrl. Il faut l'utiliser pour :

1. créer une nouvelle interface RNIS nommée `isdn0`,
2. attribuer le numéro de téléphone de cette interface,
3. attribuer l'identifiant MSN/EAZ (Multiple Subscriber Number) à partir du numéro de téléphone entrant,
4. fixer le numéro de téléphone du correspondant,
5. choisir le protocole HDLC pour la couche 2,
6. choisir l'encapsulation `rawip`,
7. fixer à 60 secondes le temps d'inactivité à l'issue duquel la connexion doit être libérée.
8. fixer le mode de connexion automatique

Au niveau réseau, on utilise ip pour configurer les adresses IP de l'interface `isdn0` et du correspondant. C'est une configuration en mode point à point.

La mise au point de la connexion se fait à l'aide des messages émis par le sous-système RNIS.

Q41. Quelle est la liste des paramètres de la commande isdnctrl à utiliser pour configurer l'interface RNIS ?

Utiliser les pages de manuels de la commande isdnctrl. Les numéros téléphoniques des bus S0 sont fournis dans le tableau ci-dessus.

Voici un exemple de configuration complète.

```
# isdnctrl list all

Current setup of interface 'isdn0':

EAZ/MSN:                104
Phone number(s):
  Outgoing:             105
  Incoming:             105
Dial mode:              auto
Secure:                 off
Callback:               off
Reject before Callback: on
Callback-delay:         5
Dialmax:                1
Hangup-Timeout:         10
Incoming-Hangup:        on
ChargeHangup:           off
Charge-Units:           0
Charge-Interval:        0
Layer-2-Protocol:       hdlc
Layer-3-Protocol:       trans
Encapsulation:          rawip
Slave Interface:        None
Slave delay:            10
Master Interface:       None
Pre-Bound to:           Nothing
PPP-Bound to:           Nothing
```

Q42. Quelle est la syntaxe de configuration IP de l'interface `isdn0` ?

Consulter le support Configuration d'une interface de réseau local ainsi que les pages de manuels. Les adresses IP à utiliser sont fournies dans le tableau ci-dessus.

À partir des exemples d'utilisation de la commande ip, on peut utiliser des instructions du type suivant.

```
# ip link set dev isdn0 up
# ip addr add 192.168.100.1/29 brd + dev isdn0
# ip addr ls dev isdn0
4: isdn0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 30
    link/ether fc:fc:c0:a8:64:02 peer ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/29 scope global isdn0
       valid_lft forever preferred_lft forever
    inet6 fe80::fefc:c0ff:fea8:6402/64 scope link
       valid_lft forever preferred_lft forever
```

Q43. Quelle est la signification de l'option `isdnctrl secure on` ?

Utiliser les pages de manuels de la commande isdnctrl.

Cette fonction a pour but de figer la paire de numéros de téléphone utilisées entre les deux hôtes en communication. Un hôte n'accepte un appel que si le numéro de l'appelant figure dans la liste des numéros autorisés en entrée.

Une fois la configuration établie on peut tester la connectivité téléphonique au niveau liaison et les communications IP au niveau réseau.

## 2.4. Documents de référence

Configuration d'une interface de réseau local

Configuration d'une interface de réseau local : identification du type d'interface, de ses caractéristiques et manipulations des paramètres. Ce support fournit une méthodologie de dépannage simple d'une connexion réseau.

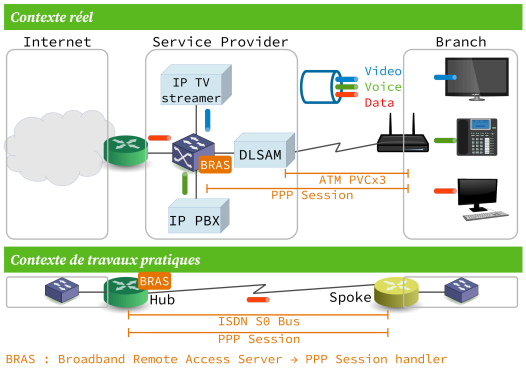
Manuel de référence Debian

Manuel de référence Debian : configuration du réseau : chapitre du manuel de référence Debian consacré à la configuration réseau.

CHAPITRE 3

# Topologie Hub & Spoke avec le protocole PPP


## Résumé




**L'objectif de ce support de travaux pratiques est l'interconnexion de réseaux locaux et de réseaux étendus. On utilise une topologie classique baptisée Hub & Spoke dans laquelle le routeur Hub est relié à plusieurs routeurs Spoke via une liaison point à point qui utilise le protocole PPP. Le support physique utilsé pour illustrer la topologie est la technologie RNIS qui permet de transmettre des trames HDLC en couche liaison.**


# Table des matières



## 3.1. Aide à la mise au point

Afin de résoudre les problèmes de connexion et de configuration, il existe différents canaux d'information système. Voici trois exemples de consultation de messages :

Messages système émis par le noyau Linux

L'affichage des messages système est géré par le démon `rsyslogd`. Pour consulter ces messages, il faut lire le contenu des fichiers du répertoire `/var/log/`. Dans le cas des travaux pratiques, les informations nécessaires à la mise au point des connexions réseau se trouvent dans le fichier `/var/log/syslog`. Pour visualiser les dernières lignes du fichier à la console on utilise la commande tail : `tail -50 /var/log/syslog`.

Du point de vue droits sur le système de fichiers, la commande tail peut être utilisée au niveau utilisateur normal dès lors que celui-ci appartient au groupe `adm`. Les commandes id et groups permettent de connaître les groupes auxquels l'utilisateur courant appartient.

Messages système émis par le sous-système RNIS

Les messages du sous-système RNIS sont transmis vers les interfaces `/dev/isdnctrl*`. On peut les consulter à l'aide de la commande : `cat /dev/isdnctrl` ou les renvoyer automatiquement sur une

console : `cat /dev/isdnctrl0 >/dev/tty10 &`. Les différents niveaux d'informations produits sont paramétrés à l'aide de l'utilitaire de contrôle du pilote d'interface RNIS : hisaxctrl. Ces niveaux sont détaillés dans les pages de manuels : `man hisaxctrl`. En ce qui concerne l'établissement des connexions téléphoniques, des codes sont renvoyés directement à la console en cas d'échec. Leur signification est donnée dans les pages de manuels `isdn_cause` : `man isdn_cause`.

Messages émis par le gestionnaire de connexion ipppd

Ces messages sont obtenus en configurant le démon de journalisation système `rsyslogd`. Les détails sur la configuration du service de journalisation système sont obtenus à l'aide des pages de manuels : `man syslog.conf`. Vérifier que la ligne suivante est bien présente dans le fichier `/etc/rsyslog.conf`.

```
# grep ^daemon /etc/rsyslog.conf
daemon.*                        -/var/log/daemon.log
```

## 3.2. Interface RNIS & protocole PPP

La connexion directe à l'aide du mode `rawip` (Voir Configuration d'une interface RNIS en mode rawip) présente l'avantage de la simplicité : authentification basée sur les numéros de téléphone sans échange d'adresses IP. Ce mode de connexion présente cependant des limitations importantes.

- La configuration des adresses IP doit être effectuée avant l'établissement de la connexion téléphonique. Il est donc impératif que les postes soient en état de marche au moment de la connexion.
- La sécurité de connexion étant basée sur les numéros de téléphone, il est impossible de se connecter depuis une autre installation.
- Comme la configuration réseau est effectuée manuellement à chaque extrémité, le plan d'adressage IP doit être connu de toutes les entités en communication.

Le protocole PPP permet de dépasser ces limitations en offrant une configuration indépendante de la technologie du réseau étendu après authentification et autorise une plus grande mobilité.

Les mécanismes de fonctionnement de ce protocole sont décrits dans le document RFC1661 The Point-to-Point Protocol (PPP). Dans le contexte de ces travaux pratiques, il doit remplir trois fonctions pour les deux configurations types étudiées :

- La possibilité de se connecter au serveur d'appel depuis n'importe quel poste ou numéro de téléphone.
- L'authentification de l'utilisateur appelant.
- L'attribution de l'adresse IP du poste appelant.

Relativement à la configuration `rawip`, il faut changer quelques paramètres de configuration au niveau liaison de l'interface RNIS.

Q44. Quelle est l'encapsulation à configurer sur l'interface RNIS pour utiliser le protocole PPP ?

Consulter les pages de manuels de la commande isdnctrl en effectuant une recherche avec la clé : `ppp`.

L'option recherchée dans les pages de manuels est : `syncppp`.

Q45. Quel est le démon de gestion de connexion qui utilise le mode de transmission synchrone des interfaces RNIS avec le protocole PPP ?

Lister les paquets liés au sous-système (RNIS|ISDN) et retrouver le gestionnaire de connexion associé.

On peut, par exemple;, effectuer la recherche suivante.

```
$ aptitude search ppp | grep -i isdn
p   ipppd              - ISDN utilities - PPP daemon
p   pppdcapiplugin     - ISDN utilities - pppd plug-in for CAPI sup
```

C'est le paquet `ipppd` qui contient le démon du même nom qui correspond à l'utilisation du sous-système RNIS du noyau Linux.

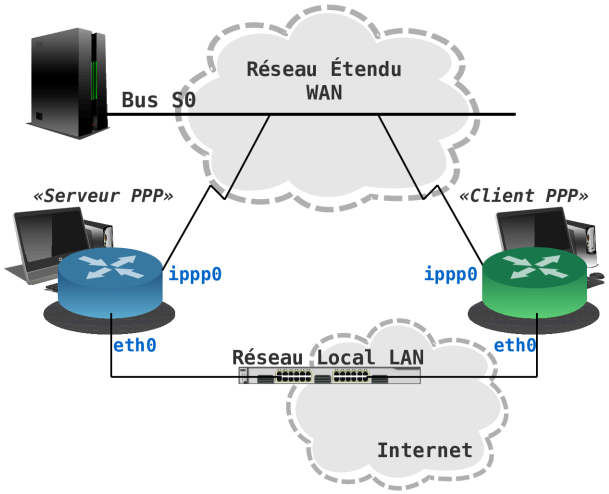
Q46. Quelles sont les noms d'interface RNIS à utiliser avec ce démon de gestion de connexion ?

Voir les pages de manuels de l'outil de configuration d'interface isdnctrl.

Le sous-système RNIS du noyau Linux dispose d'entrées spécifiques par type de communication. L'utilisation du démon `ipppd` impose une communication via un descripteur de périphérique nommé `/dev/ipppX` où X désigne un numéro d'interface.

## 3.3. Connexion avec le protocole PPP

Pour valider le fonctionnement de l'interface RNIS avec le protocole PPP, on utilise les postes de travaux pratiques par paires. Dans ce contexte, les deux modes : client et serveur ne se distinguent que par l'attribution d'adresses IP.




Topologie équivalente entre serveur et client PPP

C'est le serveur qui doit fournir les adresses données dans le tableau ci-dessous.

Tableau 3.1. Plans d'adressage IP et RNIS des liaisons WAN

| Bus | Serveur PPP | N° tél. | Adresses IP serveur:client | N° tél. | Client PPP |
|---|---|---|---|---|---|
| S0.1 | alderaan | 104 | `192.168.104.1:192.168.105.2` | 105 | bespin |
| S0.2 | centares | 106 | `192.168.106.1:192.168.107.2` | 107 | coruscant |
| S0.3 | dagobah | 108 | `192.168.108.1:192.168.109.2` | 109 | endor |
| S0.4 | felucia | 110 | `192.168.110.1:192.168.111.2` | 111 | geonosis |
| S0.5 | hoth | 112 | `192.168.112.1:192.168.113.2` | 113 | mustafar |
| S0.6 | naboo | 114 | `192.168.114.1:192.168.115.2` | 115 | tatooine |

Attention, les adresses données dans ce tableau étant utilisées par des liens point à point, le masque réseau occupe les 32 bits de l'espace d'adressage.



Saisie des options du démon PPP

Pour l'ensemble de ces travaux pratiques, les options du gestionnaire de connexion PPP ipppd doivent être saisies directement sur la ligne de commande. Il faut s'assurer que les fichiers `/etc/ppp/ioptions*` sont vides. Dans le cas contraire, les paramètres contenus dans ces fichiers peuvent être utilisés par défaut sans tenir compte de ceux saisis sur la ligne de commande.

### 3.3.1. Sans authentification

Q47. Quelles sont les options de configuration à fournir au gestionnaire de connexion pour ce mode de fonctionnement ?

Consulter les pages de manuels du démon ipppd.

Le protocole PPP n'a pas été conçu suivant le modèle Client/Serveur. Il suppose que deux processus pairs échangent des informations. Pour cette question c'est l'option `auth` qui définit si un hôte requiert une authentification de l'hôte pair. Cette authentification peut être requise par chacune des extrémités en communication. Pour désactiver l'authentification à chaque extrémité, on ajoute le préfixe `no` à l'option `auth`.

Q48. Quelles sont les options qui permettent de visualiser en détails le dialogue PPP dans les journaux systèmes ?

C'est à nouveau dans les pages de manuels que la réponse se trouve.

C'est l'option `debug` qui permet l'affichage des informations relatives aux différentes étapes de l'établissement de la connexion PPP.

Q49. 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 fait apparaître les informations suivantes.

```
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 IP qui est utilisé ; d'où l'acronyme IPCP.

Q50. 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.

### 3.3.2. Avec authentification PAP

Relativement à la section précédente, on ajoute ici le volet authentification au dialogue PPP en utilisant le protocole PAP.

Pour l'ensemble des postes de travaux pratiques les paramètres d'authentification login/password sont : `etu/stri`.

> Journalisation des échanges de mots de passe
>
> Il existe une option spécifique du gestionnaire de connexion PPP ipppd qui permet de journaliser les échanges sur les mots de passe : `+pwlog`. En ajoutant cette option à celles déjà utilisées lors de l'appel à ipppd sur la ligne de commande, on peut observer l'état des transactions d'authentification.

Q51. Quelles sont les options de configuration spécifiques à l'authentification PAP à fournir au démon PPP ?

Consulter les pages de manuels du démon ipppd.

C'est l'option `pap` qui permet de spécifier ce type d'authentification.

Q52. Dans quel fichier sont stockés les paramètres d'authentification login/password utilisés par le protocole PAP ?

Consulter les pages de manuels du démon ipppd.

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

Q53. Quels sont les en-têtes du dialogue de la couche LCP qui identifient les requêtes d'authentification échangées entre les deux processus pairs ?

Voici une copie d'écran de connexion qui fait apparaître les directives `Conf*` relatives à la partie authentification.

```
pppd[5259]: sent [LCP ConfReq id=0x1 <auth pap> <magic 0x53c04a36>]
pppd[5259]: rcvd [LCP ConfAck id=0x1 <auth pap> <magic 0x53c04a36>]
pppd[5259]: rcvd [LCP ConfReq id=0x1 <mru 1492> <magic 0x3f810ce9>]
pppd[5259]: sent [LCP ConfAck id=0x1 <mru 1492> <magic 0x3f810ce9>]
pppd[5259]: sent [LCP EchoReq id=0x0 magic=0x53c04a36]
pppd[5259]: rcvd [LCP EchoReq id=0x0 magic=0x3f810ce9]
pppd[5259]: sent [LCP EchoRep id=0x0 magic=0x53c04a36]
pppd[5259]: rcvd [PAP AuthReq id=0x1 user="etu" password=<hidden>]
```

Q54. Quelles sont les informations échangées sur les mots de passe avec le protocole PAP ? Est-il possible de relever le mot de passe avec ce protocole ?

L'utilisation de la méthode d'authentification PAP implique que le mot de passe circule en clair. Une simple capture du trafic permet de «relever» le mot de passe.

Voici un extrait de capture réseau effectué avec le démon `pppd` à l'aide de la commande `# tshark -V -i eth0 -R "ppp" > grepMyPassword.txt`.

```
Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    0001 .... = Version: 1
    .... 0001 = Type: 1
    Code: Session Data (0x00)
    Session ID: 0x0003
    Payload Length: 16
Point-to-Point Protocol
    Protocol: Password Authentication Protocol (0xc023)
PPP Password Authentication Protocol
    Code: Authenticate-Request (1)
    Identifier: 1
    Length: 14
    Data
        Peer-ID-Length: 3
        Peer-ID: etu
        Password-Length: 5
        Password: stri
```

### 3.3.3. Avec authentification CHAP

On reprend exactement le cas précédent en changeant le protocole d'authentification. On utilise maintenant le protocole CHAP qui est nettement plus intéressant que PAP. Nous allons voir pourquoi !

Les paramètres d'authentification login/password ne changent pas : `etu/stri`.

Q55. Quelles sont les options de configuration spécifiques à l'authentification CHAP à fournir au démon PPP ?

Consulter les pages de manuels du démon ipppd.

C'est l'option `chap` qui permet de spécifier ce type d'authentification.

Q56. Dans quel fichier sont stockés les paramètres d'authentification login/password utilisés par le protocole CHAP ?

Consulter les pages de manuels du démon ipppd.

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

Q57. Quels sont les en-têtes du dialogue de la couche LCP qui identifient les requêtes d'authentification échangées entre les deux processus pairs ?

Voici une copie d'écran de connexion qui fait apparaître les directives `Conf*` relatives à la partie authentification.

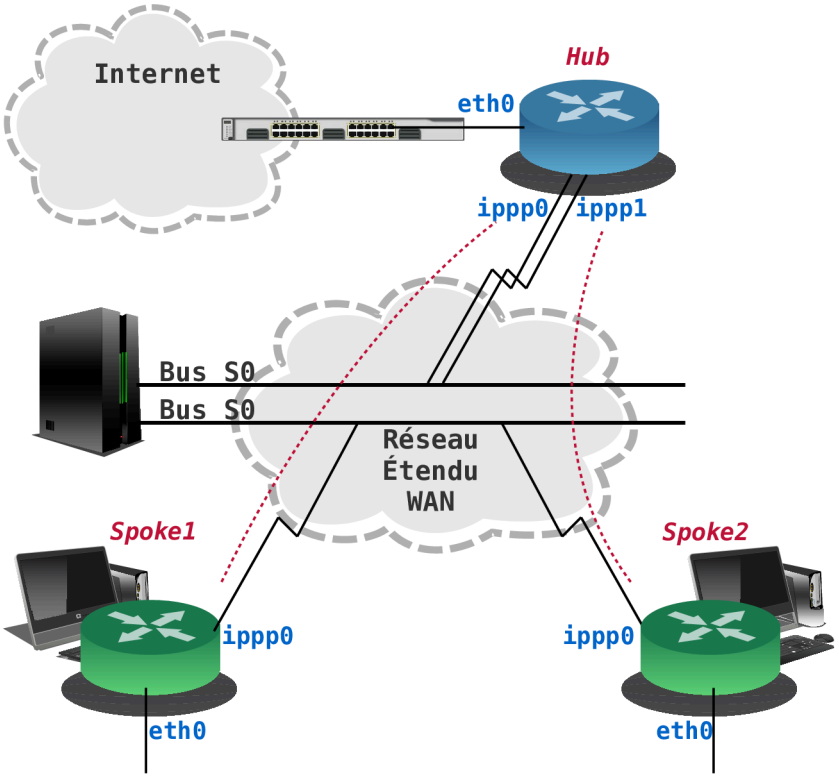
```
pppd[6037]: pppd 2.4.5 started by root, uid 0
pppd[6037]: using channel 28
pppd[6037]: Using interface ppp0
pppd[6037]: Connect: ppp0 < /dev/pts/1
pppd[6037]: sent [LCP ConfReq id=0x1 <auth chap MD5> <magic 0x46c97184>]
pppd[6037]: rcvd [LCP ConfAck id=0x1 <auth chap MD5> <magic 0x46c97184>]
pppd[6037]: rcvd [LCP ConfReq id=0x1 <mru 1492> <auth chap MD5> <magic 0x1f157ebf>]
pppd[6037]: sent [LCP ConfAck id=0x1 <mru 1492> <auth chap MD5> <magic 0x1f157ebf>]
pppd[6037]: sent [LCP EchoReq id=0x0 magic=0x46c97184]
pppd[6037]: sent [CHAP Challenge id=0x86 <ed6d9c0c022dbe008b2d3332fb275f5af1ca499393>,\
     name = "ppp-hub"]
pppd[6037]: rcvd [LCP EchoReq id=0x0 magic=0x1f157ebf]
pppd[6037]: sent [LCP EchoRep id=0x0 magic=0x46c97184]
pppd[6037]: rcvd [CHAP Challenge id=0x4f <29b6227da7da53d7ee2b4f6f7ec9a1900d5c9ac33f14>,\
     name = "etu"]
pppd[6037]: sent [CHAP Response id=0x4f <cb5bf4fbb0f9afd98a477f1f8a2e4c1f>,\
     name = "etu"]
pppd[6037]: rcvd [LCP EchoRep id=0x0 magic=0x1f157ebf]
pppd[6037]: rcvd [CHAP Response id=0x86 <de265c472d38a441fdbd2228314e0d86>,\
     name = "etu"]
pppd[6037]: sent [CHAP Success id=0x86 "Access granted"]
pppd[6037]: rcvd [CHAP Success id=0x4f "Access granted"]
pppd[6037]: CHAP authentication succeeded: Access granted
pppd[6037]: CHAP authentication succeeded
```

Q58. Quelles sont les informations échangées sur les mots de passe avec le protocole CHAP ? Est-il possible de relever le mot de passe avec ce protocole ?

Les éléments donnés dans le copie d'écran ci-dessus montrent qu'il n'y a pas d'échange de mot de passe entre les deux systèmes en communication. Seuls des Challenges sont échangés.

## 3.4. Topologie Hub & Spoke

La topologie dite Hub & Spoke est une forme de topologie étoile dans laquelle tous les liens sont de type point à point. Le rôle du Hub est de concentrer tous les accès depuis les sites distants ou les Spokes. Du point de vue routage le Hub détient la totalité du plan d'adressage alors que les Spokes ne disposent que d'un accès unique vers les autres réseaux.




Topologie Hub & Spoke

Dans le contexte de ces travaux pratiques, le routeur Hub dispose d'un accès au réseau local (LAN) via son interface Ethernet et doit fournir un accès à Internet par ses interfaces d'accès au réseau étendu (WAN). Ce réseau étendu est modélisé par les deux canaux B de l'interface RNIS du Hub. Côté Spokes, les interfaces Ethernet sont provisoirement inutilisées et le seul accès aux autres réseau se fait par un canal B de l'interface RNIS.

### 3.4.1. Établissement de la route par défaut

La configuration par défaut des paquets *pppd* suppose que le poste utilisé est un client pour lequel la route par défaut doit être établie à chaque nouvelle connexion PPP.

Dans le cas présent, le routeur d'accès (Hub) doit conserver sa route par défaut sur le réseau local indépendamment des demandes de connexion PPP. Il est donc nécessaire de modifier le script de connexion `/etc/ppp/ip-up.d/ipppd`. Voici un extrait avec les lignes à commenter :

```
PPP_NET=`echo $PPP_LOCAL | sed 's,\.[0-9]*\.[0-9]*$,.0.0/16,'`

case "$PPP_IFACE" in
 ippp0)  route del default  ❶
                # route add default netmask 0 $PPP_IFACE   # usually necessary
                route add default netmask 0 gw $PPP_REMOTE ❷
                # The next lines are for simple firewalling.
```

❶ Commenter cette ligne pour éviter l'effacement de la route par défaut.
❷ Commenter cette ligne pour éviter l'établissement d'une nouvelle route par défaut.

### 3.4.2. Plan d'adressage

Pour mettre en œuvre la topologie voulue, on distingue 4 groupes de 3 postes de travaux pratiques. Le rôle de chaque poste est défini dans le tableau ci-dessous.

Tableau 3.2. Affectation des rôles, des numéros de bus S0 et des adresses IP

| Groupe | Poste | Rôle | Bus S0 | N° Tél. | Interface | Réseau/Authentification |
|---|---|---|---|---|---|---|
| 1 | centares | Hub | S0.1 | 104 | ippp0 | 192.168.104.1:192.168.104.2 |
| | | | S0.1 | 105 | ippp1 | 192.168.105.1:192.168.105.2 |
| | bespin | Spoke 1 | S0.2 | 106 | ippp0 | etu_s1 / Sp0k3.1 |
| | | | - | - | dummy0 | 10.106.0.1/29 |
| | alderaan | Spoke 2 | S0.2 | 107 | ippp0 | etu_s2 / Sp0k3.2 |
| | | | - | - | dummy0 | 10.107.0.1/29 |
| 2 | endor | Hub | S0.3 | 108 | ippp0 | 192.168.107.1:192.168.107.2 |
| | | | S0.3 | 109 | ippp1 | 192.168.108.1:192.168.108.2 |
| | dagobah | Spoke 1 | S0.4 | 110 | ippp0 | etu_s1 / Sp0k3.1 |
| | | | - | - | dummy0 | 10.109.0.1/29 |
| | coruscant | Spoke 2 | S0.4 | 111 | ippp0 | etu_s2 / Sp0k3.2 |
| | | | - | - | dummy0 | 10.110.0.1/29 |
| 3 | hoth | Hub | S0.5 | 112 | ippp0 | 192.168.111.1:192.168.111.2 |
| | | | S0.5 | 113 | ippp1 | 192.168.112.1:192.168.112.2 |
| | geonosis | Spoke 1 | S0.6 | 114 | ippp0 | etu_s1 / Sp0k3.1 |
| | | | - | - | dummy0 | 10.113.0.1/29 |
| | felucia | Spoke 2 | S0.6 | 115 | ippp0 | etu_s2 / Sp0k3.2 |
| | | | - | - | dummy0 | 10.114.0.1/29 |
| 4 | naboo | Hub | S0.7 | 116 | ippp0 | 192.168.115.1:192.168.115.2 |
| | | | S0.7 | 117 | ippp1 | 192.168.116.1:192.168.116.2 |
| | mustafar | Spoke 1 | S0.8 | 118 | ippp0 | etu_s1 / Sp0k3.1 |
| | | | - | - | dummy0 | 10.117.0.1/29 |
| | tatooine | Spoke 2 | S0.8 | 119 | ippp0 | etu_s2 / Sp0k3.2 |
| | | | - | - | dummy0 | 10.118.0.1/29 |

Comme dans le tableau d'adressage précédent, les adresses données ci-dessus étant utilisées par des liens point à point, le masque réseau occupe les 32 bits de l'espace d'adressage.

> Avertissement
>
> Les connexions RNIS des routeurs (Hubs doivent se faire directement sur les ports de l'autocommutateur RNIS. En effet, ces connexions utilisent les deux canaux B du port BRI.

## 3.5. Configuration d'un routeur Hub

Compte tenu de la topologie définie dans la section précédente, on doit configurer les interfaces LAN et WAN du Hub. Ce routeur doit fournir l'accès Internet via son interface LAN et attribuer les adresses IP aux Spokes via ses interfaces WAN.

### 3.5.1. Connexion au réseau local

Le routeur Hub accède à l'Internet via son interface LAN. Cet accès doit être permanent et indépendant de l'état des interfaces WAN.

Q59. Comment activer le routage au niveau dans le noyau du routeur ?

Consulter le document Configuration d'une interface de réseau local et les pages de manuels de la commande sysctl pour trouver les options d'activation du routage IPv4.

Les paramètres de réglage des fonctions réseau du noyau Linux sont situés dans l'arborescence `/proc`. L'ensemble de ces paramètres est géré par la commande sysctl. On commence par effectuer une recherche avec la chaîne de caractères `ip_forward`.

```
# sysctl -a --pattern ip_forward
net.ipv4.ip_forward = 0
net.ipv4.ip_forward_use_pmtu = 0
```

Activer le routage revient à placer l'indicateur à la valeur `1`.

```
# sysctl -w net.ipv4.ip_forward=1
```

Pour rendre ce réglage permanent, il est possible d'éditer le fichier `/etc/sysctl.conf`. De cette façon, le routage sera activé à chaque redémarrage du système.

Q60. Quelles sont les opérations nécessaires à la configuration de la traduction des adresses sources des paquets sortant par l'interface LAN ?

Consulter la documentation Guide Pratique du NAT.

On utilise ici la méthode de traduction d'adresses sources la plus simple. Elle est connue sous le nom de masquerading.

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Q61. Quels sont les tests à réaliser pour s'assurer du fonctionnement de l'accès Internet ?

Consulter la documentation Configuration d'une interface de réseau local.

Les tests peuvent se décomposer en deux parties : l'utilisation du protocole ICMP et l'analyse réseau.

Avec ICMP, il faut reprendre la séquence «rituelle» des requêtes echo en allant de la destination la plus proche à la plus éloignée. On débute avec l'interface de boucle locale (loopback), puis l'interface locale, puis la passerelle par défaut et enfin une adresse IP située sur un autre réseau. Ce n'est qu'en dernier lieu que l'on doit effectuer un test avec le service de noms de domaines à l'aide des commandes host ou dig. Enfin, si le protocole ICMP n'est pas disponible au delà du réseau local, on peut utiliser un outil du type tcptraceroute pour tester la connectivité inter réseau.

Pour la partie analyse réseau, tshark permet d'analyser à la console le trafic passant par chacune des interfaces d'un routeur. Il permet notamment de caractériser le fonctionnement de la traduction d'adresses entre les interfaces LAN et WAN.

### 3.5.2. Connexion au réseau étendu

Chaque routeur Hub utilise les deux canaux B d'un bus S0. On doit donc configurer deux interfaces `ipppX` pour établir les connexions point à point avec les deux Spokes.

Q62. Donner la liste des options de la commande isdnctrl pour la configuration des deux interfaces du Hub ?

Reprendre les instructions vues dans le support Configuration d'une interface RNIS en mode rawip et la Section 3.3, « Connexion avec le protocole PPP ».

Comme dans les questions précédentes du même type, on doit effectuer les opérations suivantes pour chacune des deux interfaces `/dev/ippp0` et `/dev/ippp1`.

1. Créer l'interface.
2. Attribuer le numéro de téléphone entrant.
3. Attribuer l'identifiant MSN/EAZ (Multiple Subscriber Number) à partir du numéro de téléphone entrant.
4. Attribuer le numéro de téléphone du correspondant.
5. Choisir le protocole HDLC pour la couche liaison.
6. Choisir l'encapsulation `syncppp`.
7. Fixer le mode de connexion automatique.

Q63. Quelles sont les opérations supplémentaires nécessaires à la configuration des interfaces RNIS du routeur Hub ?

Consulter les pages de manuels de la commande isdnctrl en effectuant une recherche avec la clé : `pppbind`.

De façon à éviter les «croisements» entre les affectations d'adresses IP des deux Spokes, il est nécessaire de lier les interfaces réseau avec le plan de numérotation téléphonique. Par exemple, sur le Hub Centares, on peut exécuter les commandes suivantes.

```
# isdnctrl pppbind ippp0 0
# isdnctrl pppbind ippp1 1
```

Q64. Quelle est l'option de la commande isdnctrl qui permet de sauvegarder/restituer la configuration de l'interface RNIS ?

Utiliser les pages de manuel de l'outil isdnctrl. Sauvegarder le fichier de configuration de l'interface pour les utilisations ultérieures.

Ce sont les options `readconf` et `writeconf` de la commande isdnctrl quit permettent respectivement de lire et d'écrire dans un fichier de configuration l'ensemble des paramètres d'une ou plusieurs interfaces RNIS.

Q65. Quelles sont les opérations à effectuer pour mettre en œuvre le protocole PPP avec une authentification CHAP ?

Reprendre les questions de la Section 3.3, « Connexion avec le protocole PPP » pour chacune des deux interfaces. Les couples d'authentification login/password sont donnés dans le Section 3.4.2, « Plan d'adressage ».

### 3.5.3. Routage statique

Pour que les réseaux desservis par les routeurs Spokes soient accessibles depuis toutes les extrémités en communication, le routeur Hub doit disposer d'une table de routage complète. Comme le nombre des réseaux de chaque Spoke est limité, on utilise des entrées statiques dans la table de routage du Hub.

Q66. Comment ajouter une entrée statique dans la table de routage du Hub ?

Rechercher les options de la commande ip dans le Manuel de référence Debian : configuration du réseau.

C'est l'instruction `# ip route add ...` qui permet l'ajout de routes statiques. Par exemple, dans le cas du routeur Hub `centares`, les deux instructions suivantes permettent d'ajouter les deux routes correspondant aux deux réseaux des routeurs Spokes `alderaan` et `bespin`.

```
# ip route add 10.106.0.0/29 dev ippp0
# ip route add 10.107.0.0/29 dev ippp1
```

Q67. Comment tester la disponibilité des différents réseaux interconnectés ?

Reprendre les séquences de tests ICMP entre les différents hôtes.

## 3.6. Configuration d'un routeur Spoke

Dans ce scénario, le routeur accède à Internet par son interface WAN et redistribue cet accès sur un réseau local. Ce genre de routeur est appelé «routeur d'agence».

### 3.6.1. Connexion au réseau local

Compte tenu de la topologie définie dans la Section 3.4, « Topologie Hub & Spoke », l'interface LAN du routeur Spoke n'est pas utilisée. Il faut donc désactiver cette interface.

Q68. Comment supprimer la configuration d'une interface réseau au niveau système ?

Rechercher les outils systèmes proposés dans le Manuel de référence Debian : configuration du réseau.

Le paquet `ifupdown` propose deux scripts baptisés ifup et ifdown qui assurent un contrôle d'état sur la configuration des interfaces listées dans le fichier de configuration système `/etc/network/interfaces`.

Dans le cas de l'interface LAN du poste de travaux pratiques, `eth0` est configurée via le protocole DHCP. Pour résilier le bail DHCP et désactiver l'interface, on utilise l'instruction suivante.

```
# ifdown eth0
```

### 3.6.2. Connexion au réseau étendu

Chaque routeur Spoke utilise un canal B d'un bus S0. On doit donc configurer une interface `ippp0` pour établir la connexion point à point avec le Hub.

Q69. Donner la liste des options de la commande isdnctrl pour la configuration de l'interface du Spoke ?

Reprendre les instructions vues dans le support Configuration d'une interface RNIS en mode rawip et la Section 3.3, « Connexion avec le protocole PPP ».

Comme dans les questions précédentes du même type, on doit effectuer les opérations suivantes pour l'interface `/dev/ippp0`.

1. Créer l'interface.
2. Attribuer le numéro de téléphone entrant.
3. Attribuer l'identifiant MSN/EAZ (Multiple Subscriber Number) à partir du numéro de téléphone entrant.
4. Attribuer le numéro de téléphone du correspondant.
5. Choisir le protocole HDLC pour la couche liaison.
6. Choisir l'encapsulation `syncppp`.
7. Fixer le mode de connexion automatique.

Q70. Quelle est l'option de la commande isdnctrl qui permet de sauvegarder/restituer la configuration de l'interface RNIS ?

Utiliser les pages de manuel de l'outil isdnctrl. Sauvegarder le fichier de configuration de l'interface pour les utilisations ultérieures.

Ce sont les options `readconf` et `writeconf` de la commande isdnctrl quit permettent respectivement de lire et d'écrire dans un fichier de configuration l'ensemble des paramètres d'une ou plusieurs interfaces RNIS.

Q71. Quelles sont les opérations à effectuer pour mettre en œuvre le protocole PPP avec une authentification CHAP ?

Reprendre les questions de la Section 3.3, « Connexion avec le protocole PPP ». Les couples d'authentification login/password sont donnés dans le Section 3.4.2, « Plan d'adressage ».

### 3.6.3. Ajout d'un réseau fictif

L'ajout de nouvelles entrées fictives dans les tables de routage est une pratique très répandue. Elle permet de qualifier le bon fonctionnement d'un service ou d'un filtrage sans ajouter de matériel. Dans le cas de ces travaux pratiques, c'est le service Web qui est utilisé pour valider la disponibilité d'un réseau au niveau application.

Q72. Quelles sont les opérations à effectuer pour pouvoir utiliser des interfaces réseau virtuelles de type boucle locale sur un système GNU/Linux ?

Avec le noyau Linux, il est conseillé d'utiliser des interfaces baptisées dummy pour ce genre d'usage. Rechercher le module correspondant à charger en mémoire.

On charge le module `dummy` suivi de l'option `numdummies` pour créer les interfaces. Il suffit ensuite d'appliquer une nouvelle configuration IP pour ajouter un ou plusieurs nouveaux réseaux.

```
# ip link add dummy0 type dummy
```

En prenant l'exemple du Spoke `bespin`, on ajoute le réseau `10.106.0.0/29` en configurant l'interface `dummy0`.

```
# ip link set dev dummy0 up
# ip addr add 10.106.0.1/29 brd + dev dummy0
# ip route ls
default via 192.0.2.1 dev eth0
10.106.0.0/29 dev dummy0  proto kernel  scope link  src 10.106.0.1
192.0.2.0/26 dev eth0  proto kernel  scope link  src 192.0.2.10
```

Q73. Quelles sont les opérations à effectuer pour installer un service Web en écoute exclusivement sur l'adresse IP de l'interface `dummy0` ?

Installer le paquet `apache2` et modifier sa configuration pour que le service ne soit accessible que sur une adresse IP.

```
# aptitude install apache2
<snipped/>
Paramétrage de apache2-mpm-worker (2.2.21-2) ...
Starting web server: apache2apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1 for ServerName
.
Paramétrage de apache2 (2.2.21-2) ...
<snipped/>
```

On modifie ensuite le fichier de configuration `/etc/apache2/ports.conf` de façon à limiter l'accès à l'adresse IP voulue.

```
# cd /etc/apache2/
# diff -uBb ports.conf.orig ports.conf
--- ports.conf.orig     2011-10-31 20:37:09.000000000 +0100
+++ ports.conf  2011-10-31 20:37:39.000000000 +0100
@@ -6,7 +6,7 @@
 # README.Debian.gz

 NameVirtualHost *:80
-Listen 80
+Listen 10.106.0.1:80

 <IfModule mod_ssl.c>
     # If you add NameVirtualHost *:443 here, you will also have to change
```

On redémarre le service et on affiche le liste des sockets inet ouverts sur le système pour confirmer que l'adresse IP choisie est bien affectée.

```
# /etc/init.d/apache2 restart
<snipped/>
# lsof -i | grep apache2
apache2  22206        root    3u  IPv4  30721      0t0  TCP 10.106.0.1:www (LISTEN)
apache2  22211    www-data    3u  IPv4  30721      0t0  TCP 10.106.0.1:www (LISTEN)
apache2  22212    www-data    3u  IPv4  30721      0t0  TCP 10.106.0.1:www (LISTEN)
```

Q74. Comment valider l'accès à ce service Web depuis les autres routeurs ?

Si la table de routage du routeur Hub est complète, on décrit les couches de la modélisation en partant de la couche réseau vers la couche application. Les tests ICMP valident le niveau réseau. Les tests traceroute valident le fonctionnement des protocoles de la couche transport. Enfin, le navigateur web permet de tester la couche application.

Voici trois exemples de tests.

- Test ICMP.

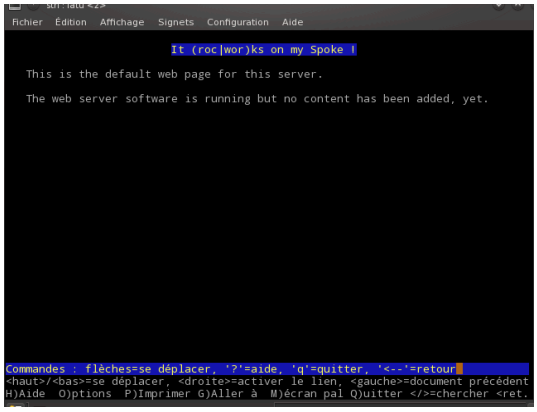
```
$ ping -c 2 10.106.0.1
PING 10.106.0.1 (10.106.0.1) 56(84) bytes of data.
64 bytes from 10.106.0.1: icmp_req=1 ttl=64 time=0.435 ms
64 bytes from 10.106.0.1: icmp_req=2 ttl=64 time=0.360 ms

--- 10.106.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.360/0.397/0.435/0.042 ms
```

- Test traceroute.

```
$ traceroute 10.106.0.1
traceroute to 10.106.0.1 (10.106.0.1), 30 hops max, 60 byte packets
 1  10.106.0.1 (10.106.0.1)  0.467 ms  0.256 ms  0.262 ms
```

- Test HTTP.




Copie d'écran navigateur Web

## 3.7. Documents de référence

The Point-to-Point Protocol (PPP)

RFC1661 The Point-to-Point Protocol (PPP) : Le protocole point-à-point PPP fournit une méthode standard de transport de datagrammes multi-protocoles sur des liaisons point à point. PPP comprend 3 composants principaux :

1. Une méthode d'encapsulation des datagrammes multi-protocoles.
2. Un protocole de contrôle de niveau liaison ou Link Control Protocol (LCP) pour établir, configurer et tester une connexion de données à ce niveau.
3. Une famille de protocoles de contrôle de niveau réseau pour établir et configurer différents protocoles de niveau réseau.

Dans la plupart des cas, on retrouve des trames HDLC au niveau liaison et IP est le seul protocole réseau utilisé.

Configuration d'une interface RNIS en mode rawip

Configuration d'une interface RNIS en mode rawip : support de travaux pratiques utilisant la connexion directe sur le réseau téléphonique.

Configuration d'une interface de réseau local

Configuration d'une interface de réseau local : identification du type d'interface, de ses caractéristiques et manipulations des paramètres. Ce support fournit une méthodologie de dépannage simple d'une connexion réseau.

Debian Reference Chapter 10 - Network configuration

Manuel de référence Debian : configuration du réseau : chapitre du manuel de référence Debian consacré à la configuration réseau.

Fonctions réseau du noyau Linux

Configuration des fonctions réseau & compilation du noyau Linux : présentation et configuration des fonctions réseau du noyau LINUX

Guide Pratique du NAT sous Linux 2.4

Guide Pratique du NAT : Ce document décrit comment réaliser du camouflage d'adresse IP, un serveur mandataire transparent, de la redirection de ports ou d'autres formes de traduction d'adresse réseau (Network Address Translation ou NAT) avec le noyau Linux 2.4.

Linux PPP HOWTO

Linux PPP HOWTO : Ce guide est relativement ancien. On y trouve cependant des exemples utiles sur le paramétrage de l'authentification avec la protocole PPP.

CHAPITRE 4

# Filtrage réseau avec netfilter/iptables

**Résumé**

**Ce support de travaux pratiques est une introduction au filtrage réseau. Il reprend la topologie Hub & Spoke des autres supports de la série. Les questions débutent par l'identification des outils et passent à l'application des règles de filtrage avec et sans suivi de communication (stateful vs stateless inspection). On introduit aussi les fonctions de traduction d'adresses (NAT).**

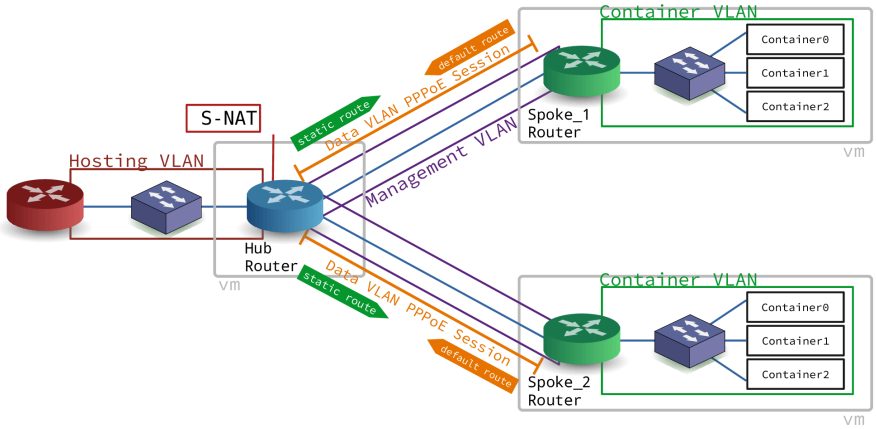
## Table des matières



## 4.1. Architecture réseau étudiée et filtrage

Les manipulations sur le système de filtrage réseau présentées ici s'appuient sur la topologie Hub and Spoke étudiée dans le support précédent de la série : Topologie Hub & Spoke avec le protocole PPPoE.

La topologie étudiée associe trois routeurs qui ont deux rôles distincts.



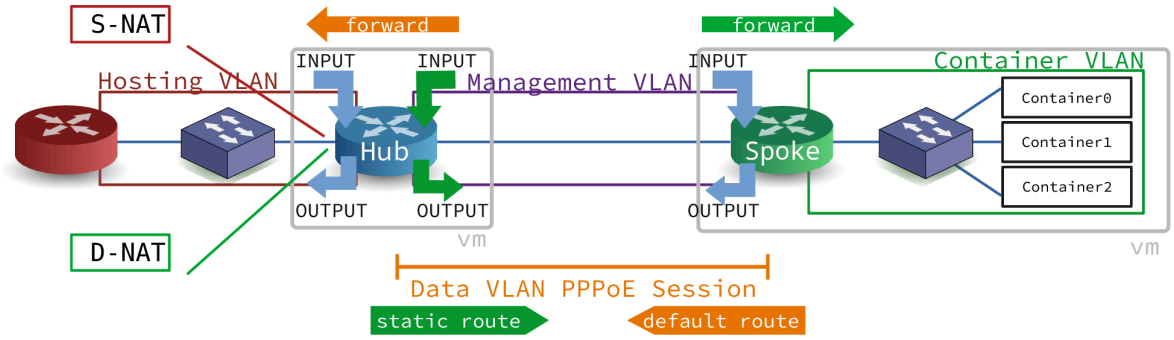

Topologie entre deux routeurs Hub et Spoke avec PPPoE

Routeur central, Hub, Broadband Remote Access Server, BRAS

> Ce routeur réalise une interconnexion LAN/WAN. Il fournit un accès Internet aux routeurs de sites distants via ses interfaces WAN. Il dispose de son propre accès Internet via son interface LAN.

Routeur d'extrémité, Spoke, Customer Premises Equipment, CPE

> Ce routeur réalise aussi une interconnexion LAN/WAN. À la différence du routeur Hub, il obtient l'accès Internet sur son interface WAN et il met cet accès à disposition d'un réseau local de site représenté par des conteneurs LXD.




Topologie Hub & Spoke et filtrage

## Routage et traduction d'adresses (situation de départ)

Les manipulations qui suivent supposent que la topologie Hub & Spoke est en place et fonctionnelle. On s'appuie sur le support précédent de la série : Topologie Hub & Spoke avec le protocole PPPoE

- Le routeur Hub doit s'assurer que le trafic réseau qu'il route vers et depuis l'Internet correspond bien au plan d'adressage défini. Dans ce but, il attribue les adresses du lien point à point ainsi qu'une route statique à destination du réseau d'extrémité distant.

  Le routeur Hub assure la traduction des adresses sources du réseau distant vers l'Internet.

- Le routeur Spoke doit obtenir son adresse IPv4 de réseau étendu via PPP et assurer le routage de son réseau local. Il dispose d'une route par défaut qui désigne le lien point à point comme seul accès vers l'Internet.

Les questions ci-dessous ont pour objectif de valider le fonctionnement du routage et de la traduction des adresses sources en sortie du routeur Hub vers l'Internet.

Q75. Comment tracer le chemin suivi par les paquets IPv4 et IPv6 d'un conteneur à un autre conteneur du site distant de l'autre branche de la topologie ?

Rechercher le paquet contenant la commande tracepath qui permet d'afficher le chemin suivi par le trafic réseau.

Partant de la topologie de la maquette, on commence par se placer sur le routeur Spoke2Vert et on accède à la console du container2.

```
etu@Spoke2Vert:~$ lxc ls
+------------+---------+------------------+-----------------------------------------+-----------+------
|    NAME    |  STATE  |       IPV4       |                  IPV6                   |   TYPE    | SNAPSH
+------------+---------+------------------+-----------------------------------------+-----------+------
| container0 | RUNNING | 10.0.2.10 (eth0) | fda0:7a62:2:0:216:3eff:feda:e1a (eth0)  | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+------
| container1 | RUNNING | 10.0.2.11 (eth0) | fda0:7a62:2:0:216:3eff:fec4:d325 (eth0) | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+------
| container2 | RUNNING | 10.0.2.12 (eth0) | fda0:7a62:2:0:216:3eff:fe66:86fb (eth0) | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+------
```

```
etu@Spoke2Vert:~$ lxc exec container2 -- /bin/bash
root@container2:~# apt install iputils-tracepath
```

une fois le paquet `iputils-tracepath` installé, on peut contacter les adresses IPv4 et IPv6 du container0 desservi par le routeur Spoke1Vert.

Toujours dans le contexte de la maquette, on affiche la liste des adresses des conteneurs côté Spoke1Vert.

```
etu@Spoke1Vert:~$ lxc ls
+------------+---------+------------------+-----------------------------------------+-----------+-------
|    NAME    |  STATE  |       IPV4       |                  IPV6                   |   TYPE    | SNAPSH
+------------+---------+------------------+-----------------------------------------+-----------+-------
| container0 | RUNNING | 10.0.1.10 (eth0) | fda0:7a62:1:0:216:3eff:feda:e1a (eth0)  | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+-------
| container1 | RUNNING | 10.0.1.11 (eth0) | fda0:7a62:1:0:216:3eff:fec4:d325 (eth0) | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+-------
| container2 | RUNNING | 10.0.1.12 (eth0) | fda0:7a62:1:0:216:3eff:fe66:86fb (eth0) | CONTAINER | 0
+------------+---------+------------------+-----------------------------------------+-----------+-------
```

On connait maintenant les adresses à contacter depuis le conteneur numéro 2 côté Spoke2Vert.

```
root@container2:~# tracepath 10.0.1.10
 1?: [LOCALHOST]                      pmtu 1500
 1:  10.0.2.1                                              0.937ms
 1:  10.0.2.1                                              0.115ms
 2:  10.0.2.1                                              0.306ms pmtu 1492
 2:  10.47.3.1                                             0.843ms
 3:  10.47.1.2                                             1.853ms
 4:  10.0.1.10                                             2.506ms reached
     Resume: pmtu 1492 hops 4 back 4
```

```
root@container2:~# tracepath fda0:7a62:1:0:216:3eff:feda:e1a
 1?: [LOCALHOST]                       0.046ms pmtu 1500
 1:  fda0:7a62:2::1                                        1.403ms
 1:  fda0:7a62:2::1                                        0.292ms
 2:  fda0:7a62:2::1                                        0.315ms pmtu 1492
 2:  2001:678:3fc:12c::2                                   0.788ms
 3:  fda0:7a62:1::1                                        2.491ms
 4:  fda0:7a62:1:0:216:3eff:feda:e1a                       2.539ms reached
     Resume: pmtu 1492 hops 4 back 4
```

Les résultats obtenus avec l'exécution de la commande tracepath montrent que le routage des paquets IPv4 et IPv6 est fonctionnel sur la topologie Hub & Spoke.

Q76. Comment caractériser la traduction d'adresses source en sortie du routeur Hub ?

La fonction de traduction d'adresse entre dans cadre du filtrage réseau et fait appel aux mêmes outils : netfilter/iptables.

Rechercher le paquet qui contient la commande conntrack puis rechercher les options de cette commande qui permettent d'afficher les états des enregistrements de la table NAT.

On se place sur le routeur Hub de la maquette et on installe le paquet `conntrack`.

```
etu@HubBleu:~$ sudo apt install conntrack
```

Dans le même temps, on accède au conteneur numéro 0 desservi par le routeur Spoke1Vert. C'est à partir de cette console que l'on lance des téléchargements depuis le serveur inetdoc.net à l'aide de la commande wget.

```
etu@Spoke1Vert:~$ lxc exec container0 -- /bin/bash
root@container0:~# apt install wget
```

Sur le routeur Hub, on affiche la liste des enregistrements de la table NAT.

- Requête IPv4 depuis le conteneur :

```
root@container0:~# while true
do
 wget -4 -O /dev/null https://inetdoc.net/pdf/iproute-cheatsheet.pdf
 sleep 3
done
```

Liste des enregistrements :

```
etu@HubBleu:~$ sudo conntrack -f ipv4 -L
udp      17 17 src=10.0.1.10 dst=9.9.9.9 sport=49165 dport=53
    src=9.9.9.9 dst=10.141.0.162 sport=53 dport=49165 mark=0 use=1
tcp      6 432000 ESTABLISHED src=172.16.0.230 dst=10.141.0.162 sport=40278 dport=22
    src=10.141.0.162 dst=172.16.0.230 sport=22 dport=40278 [ASSURED] mark=0 use=1
udp      17 20 src=10.0.1.10 dst=9.9.9.9 sport=36074 dport=53
    src=9.9.9.9 dst=10.141.0.162 sport=53 dport=36074 mark=0 use=1
tcp      6 1 CLOSE src=10.0.1.10 dst=89.234.156.195 sport=44860 dport=443
    src=89.234.156.195 dst=10.141.0.162 sport=443 dport=44860 [ASSURED] mark=0 use=1
tcp      6 7 CLOSE src=10.0.1.10 dst=89.234.156.195 sport=44864 dport=443
    src=89.234.156.195 dst=10.141.0.162 sport=443 dport=44864 [ASSURED] mark=0 use=1
udp      17 27 src=10.0.1.10 dst=9.9.9.9 sport=45443 dport=53
    src=9.9.9.9 dst=10.141.0.162 sport=53 dport=45443 mark=0 use=1
udp      17 24 src=10.0.1.10 dst=9.9.9.9 sport=33499 dport=53
    src=9.9.9.9 dst=10.141.0.162 sport=53 dport=33499 mark=0 use=1
tcp      6 4 CLOSE src=10.0.1.10 dst=89.234.156.195 sport=44862 dport=443
    src=89.234.156.195 dst=10.141.0.162 sport=443 dport=44862 [ASSURED] mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 9 flow entries have been shown.
```

- Requête IPv6 depuis le conteneur :

```
root@container0:~# while true
do
 wget -6 -O /dev/null https://inetdoc.net/pdf/iproute-cheatsheet.pdf
 sleep 3
done
```

Liste des enregistrements :

```
etu@HubBleu:~$ sudo conntrack -f ipv6 -L
tcp      6 5 CLOSE src=fda0:7a62:1:0:216:3eff:feda:e1a dst=2a03:7220:8083:c300::1 sport=47472 dport=4
    src=2a03:7220:8083:c300::1 dst=2001:678:3fc:12c::2 sport=443 dport=47472 [ASSURED] mark=0 use=1
tcp      6 8 CLOSE src=fda0:7a62:1:0:216:3eff:feda:e1a dst=2a03:7220:8083:c300::1 sport=47474 dport=4
    src=2a03:7220:8083:c300::1 dst=2001:678:3fc:12c::2 sport=443 dport=47474 [ASSURED] mark=0 use=1
tcp      6 431998 ESTABLISHED src=fe80:1d6::1 dst=fe80:1d6::2 sport=39218 dport=2222
    src=fe80:1d6::2 dst=fe80:1d6::1 sport=2222 dport=39218 [ASSURED] mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 3 flow entries have been shown.
```

## 4.2. Les outils de filtrage réseau

Sur un système GNU/Linux, les fonctions de filtrage réseau sont réparties entre les espaces mémoire noyau (kernelspace) et utilisateur (userspace). Les fonctions de filtrage réseau sont disponibles sous forme de modules qui sont chargés dynamiquement dans la mémoire du système en cours d'exécution en fonction de la syntaxe des règles de filtrage ajoutées.

Q77. Quel est le paquet le plus important pour les manipulations sur les fonctions de filtrage réseau ?

Rechercher dans la liste des paquets les mots clés tels que `iptables` ou `firewall`.

La partie userspace des fonctions de filtrage réseau s'appelle iptables. On lance donc une recherche avec ce mot clé dans la base de données des paquets Debian.

```
$ aptitude search ~iiptables
i   iptables              - administration tools for packet filtering and NAT
i   iptables-persistent   - boot-time loader for netfilter rules, iptables plugin
```

Q78. Comment visualiser les modules chargés dynamiquement en fonction de l'utilisation des règles de filtrage réseau ?

Utiliser la commande qui sert à lister les modules chargés en mémoire avant et après avoir consulté les tables de filtrage réseau pour la première fois.

La commande lsmod sert à lister les modules chargés en mémoire. Voici un exemple de liste de modules relatifs au filtrage.

```
$ $ lsmod | egrep '(ip_|nf_)' | fmt -t -w80
nf_conntrack_netlink    57344  0
nf_nat                  49152  2 nft_chain_nat,xt_MASQUERADE
nf_conntrack           176128  3 nf_nat,nf_conntrack_netlink,xt_MASQUERADE
nf_defrag_ipv6          24576  1 nf_conntrack
nf_defrag_ipv4          16384  1 nf_conntrack
libcrc32c               16384  2 nf_conntrack,nf_nat
nf_tables              241664  7 nft_compat,nft_counter,nft_chain_nat
nfnetlink               16384  3 nft_compat,nf_conntrack_netlink,nf_tables
ip_tables               32768  0
x_tables                53248  4 nft_compat,ip_tables,xt_limit,xt_MASQUERADE
```

Q79. Quels sont les outils de sauvegarde et de restauration des jeux de règles de filtrage réseau fournis avec le paquet iptables-persistent ?

Consulter la liste des fichiers du paquet.

La liste des fichiers du paquet fait apparaître les outils iptables-save et iptables-restore qui permettent respectivement de sauvegarder et de restaurer l'ensemble des règles de toutes les tables utilisées.

Ces programmes sont indispensables pour éditer, insérer ou retirer des règles sans avoir à se préoccuper de l'ordre de saisie. De plus, le programme de restauration se charge de l'effacement des règles précédentes.

Q80. Comment visualiser les enregistrements d'états de suivi des communications réseau ?

Rechercher la chaîne `conntrack` dans la liste des paquets.

La section «7.2 Les entrées de conntrack» du Tutoriel iptables décrit précisément les différents champs du suivi de communication.

Voici un échantillon capturé sur le routeur HubBleu après avoir lancé une mise à jour du catalogue des paquets sur les conteneurs du routeur Spoke2Vert.

```
$ sudo conntrack -L | fmt -t -w80
conntrack v1.4.6 (conntrack-tools): 7 flow entries have been shown.
tcp      6 431999 ESTABLISHED src=172.16.0.230 dst=10.141.0.162 sport=40626
   dport=22 src=10.141.0.162 dst=172.16.0.230 sport=22 dport=40626 [ASSURED]
   mark=0 use=1
udp      17 23 src=10.0.2.10 dst=9.9.9.9 sport=53336 dport=53 src=9.9.9.9
   dst=10.141.0.162 sport=53 dport=53336 [ASSURED] mark=0 use=1
tcp      6 113 TIME_WAIT src=10.0.2.10 dst=151.101.12.204 sport=48494
   dport=80 src=151.101.12.204 dst=10.141.0.162 sport=80 dport=48494 [ASSURED]
   mark=0 use=1
udp      17 27 src=10.0.2.11 dst=9.9.9.9 sport=39790 dport=53 src=9.9.9.9
   dst=10.141.0.162 sport=53 dport=39790 mark=0 use=1
udp      17 23 src=10.0.2.10 dst=9.9.9.9 sport=59674 dport=53 src=9.9.9.9
   dst=10.141.0.162 sport=53 dport=59674 mark=0 use=1
tcp      6 117 TIME_WAIT src=10.0.2.11 dst=151.101.12.204 sport=50252
   dport=80 src=151.101.12.204 dst=10.141.0.162 sport=80 dport=50252 [ASSURED]
   mark=0 use=1
udp      17 27 src=10.0.2.11 dst=9.9.9.9 sport=43617 dport=53 src=9.9.9.9
   dst=10.141.0.162 sport=53 dport=43617 [ASSURED] mark=0 use=1
```

## 4.3. Protection de base des routeurs Hub et Spoke

Le but de cette section est de mettre en place le routage avant de passer aux fonctions de filtrage réseau proprement dites. Elle correspond à la vue Topologie PPP et routage.

Voici une liste de fonctions de protection à mettre en œuvre sur tous les types de routeurs.

Protection contre l'usurpation des adresses sources, rpfilter, BCP38

Ces fonctions de protection comprennent une partie noyau ainsi qu'une partie filtrage avec le module `rpfilter` à implanter dans la table `raw` qui assure un filtrage sans état. Voir Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.

Les tests de validation de ces mécanismes peuvent se faire à l'aide de la commande hping3. Les résultats doivent être visibles aussi bien dans les journaux systèmes que sur les compteurs des règles de la table `raw`. En avant pour la chasse aux martiens !

Protection contre les dénis de services ICMP, module `netfilter limit`

Les routeurs doivent s'assurer que le volume de trafic qui est présenté en entrée est compatible avec un fonctionnement nominal des services.

Protection contre les robots de connexion au service SSH, fail2ban

Les routeurs ont besoin d'un accès d'administration à distance via SSH. Pour autant, cet accès doit être protégé contre les tentatives d'intrusion par dictionnaire de couples d'authentifiants.

L'outil fail2ban fourni avec le paquet du même nom introduit une chaîne de filtrage dédiée à ces tentatives d'intrusion.

### 4.3.1. Protection contre l'usurpation d'adresse source

Q81. Comment afficher la liste des règles de filtrage de la table `raw` dédiée au filtrage sans état (stateless) ?

Rechercher dans les pages de manuels de la commande iptables les options relatives aux listes et aux compteurs.

La visualisation des compteurs de correspondance des règles de filtrage est indispensable pour qualifier le fonctionnement du filtrage

C'est l'option `-L` qui permet l'affichage des listes.

C'est l'option `-v` qui permet d'obtenir les valeurs des compteurs de correspondance avec chaque règle.

Voici un exemple dans le contexte de la maquette sur le routeur Spoke1Vert. Une règle a déjà été insérée dans la table `raw`. Elle permet de visualiser les compteurs de correspondance qui montrent que la règle a bien été utilisée.

```
etu@Spoke1Vert:~$ sudo iptables -vL -t raw
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain PREROUTING (policy ACCEPT 112K packets, 113M bytes)
 pkts bytes target     prot opt in     out     source       destination
   60  4996 DROP       all  --  any    any     anywhere     anywhere  rpfilter invert /* BCP38 */

Chain OUTPUT (policy ACCEPT 20480 packets, 1365K bytes)
 pkts bytes target     prot opt in     out     source       destination
```

Q82. Comment activer la protection contre l'usurpation des adresses sources au niveau du noyau ?

Rechercher les informations relatives à la fonction Reverse Path Forwarding du noyau Linux. Identifier les rôles des 3 valeurs possibles de cette fonction.

La documentation est à cette adresse : Kernel IP sysctl.

Le fichier de configuration principal `/etc/sysctl.conf` dispose de plusieurs entrées relatives à cette fonction. Voici un extrait dans le contexte de la maquette.

```
$ grep rp_filter /etc/sysctl.conf
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
```

Voici la liste des valeurs actives au moment de l'exécution de la commande.

```
etu@Spoke1Vert:~$ sudo sysctl -ar '\.rp_filter'
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.asw-host.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.enp0s6.rp_filter = 1
net.ipv4.conf.enp0s6/470.rp_filter = 1
net.ipv4.conf.enp0s6/471.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 0
net.ipv4.conf.ovs-system.rp_filter = 1
net.ipv4.conf.ppp0.rp_filter = 1
net.ipv4.conf.sw-vlan1.rp_filter = 1
net.ipv4.conf.veth52dfe1cc.rp_filter = 1
net.ipv4.conf.veth73d0058f.rp_filter = 1
net.ipv4.conf.vethc394c229.rp_filter = 1
```

Voici l'extrait de la documentation officielle qui donne les explications sur les 3 valeurs possibles du paramètre `rp_filter`.

```
rp_filter - INTEGER
 0 - No source validation.
 1 - Strict mode as defined in RFC3704 Strict Reverse Path
     Each incoming packet is tested against the FIB and if the interface
     is not the best reverse path the packet check will fail.
     By default failed packets are discarded.
 2 - Loose mode as defined in RFC3704 Loose Reverse Path
     Each incoming packet's source address is also tested against the FIB
     and if the source address is not reachable via any interface
     the packet check will fail.

 Current recommended practice in RFC3704 is to enable strict mode
 to prevent IP spoofing from DDos attacks. If using asymmetric routing
 or other complicated routing, then loose mode is recommended.

 The max value from conf/{all,interface}/rp_filter is used
 when doing source validation on the {interface}.

 Default value is 0. Note that some distributions enable it
 in startup scripts.
```

Q83. Comment enregistrer les tentatives d'usurpation d'adresses dans les journaux système ?

Rechercher les entrées de l'arborescence `/proc` relatives aux paquets “martiens“.

Rechercher aussi le paramètre relatifs aux “martiens“ dans le fichier `/etc/sysctl.conf`.

On a activé la “journalisation des martiens“ en éditant le fichier `/etc/sysctl.conf`.

```
$ grep martians /etc/sysctl.conf
net.ipv4.conf.all.log_martians = 1
```

On vérifie que le paramètre est bien actif sur le système.

```
$ sudo sysctl -ar 'all.*martians'
net.ipv4.conf.all.log_martians = 1
```

Si ce n'est pas le cas, il ne faut pas oublier de parcourir à nouveau les fichiers de paramètres à l'aide de la commande sysctl.

```
$ sudo sysctl --system
```

Q84. Comment valider la fonction de blocage des tentatives d'usurpation d'adresses entre le routeur Hub et les routeurs Spoke ?

Installer le paquet `hping3` sur le routeur Hub.

Rechercher dans les pages de manuels de la commande hping3 les options qui permettent de générer du trafic ICMP avec des adresses source aléatoires à destination d'un conteneur hébergé sur un routeur Spoke.

Voici un premier exemple de test effectué sur le routeur Hub dans le contexte de la maquette.

L'option `-a` désigne l'adresse IPv4 source usurpée tandis que l'adresse en bout de ligne désigne la destination. Ici, on cherche à contacter un conteneur avec l'adresse source d'un conteneur voisin en étant placé “à l'extérieur“ du VLAN vert.

```
etu@HubBleu:~$ sudo hping3 -1 -a 10.0.2.12 --fast -c 10 10.0.2.11
HPING 10.0.2.11 (ppp0 10.0.2.11): icmp mode set, 28 headers + 0 data bytes

--- 10.0.2.11 hping statistic ---
10 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Côté routeur Spoke, on peut consulter les traces des tentatives d'usurpation d'adresses à l'aide de la commande suivante.

```
etu@Spoke2Vert:~$ grep martian /var/log/kern.log
```

Il est aussi possible de lancer un test avec une série d'adresses IP source aléatoires. Voici un exemple de commande qui provoquera un nombre de blocages aléatoire en fonction des correspondances.

```
etu@HubBleu:~$ sudo hping3 -1 --rand-source --fast -c 100 10.0.1.11
```

Q85. Comment filtrer les tentatives d'usurpation d'adresses source au plus tôt de façon à limiter le coût de traitement de ces paquets falsifiés sur le système ?

Identifier le nom de la table de filtrage sans état et rechercher la fonction associée au filtrage des adresses sources usurpées. Rechercher dans les pages de manuels `iptables-extensions` les informations relatives au module `rpfilter`.

Ajouter une règle spécifique dans la table de traitement sans état pour les protocoles IPv4 et IPv6.

La table de filtrage sans état est appelée : `raw`. Après consultation des exemples donnés dans les pages de manuels, on aboutit aux deux règles suivantes que l'on applique sur les routeurs Spoke.

```
$ sudo iptables -t raw -A PREROUTING -i ppp0 -m rpfilter --invert -m comment --comment "BCP38" -j DROP
$ sudo ip6tables -t raw -A PREROUTING -i ppp0 -m rpfilter --invert -m comment --comment "BCP38" -j DROP
```

On peut ensuite sauvegarder ces règles dans les fichiers systèmes utilisés par le service `iptables-persistent`.

```
$ sudo sh -c "iptables-save >/etc/iptables/rules.v4"
$ sudo sh -c "ip6tables-save >/etc/iptables/rules.v6"
```

Q86. Comment caractériser les nouvelles règles de filtrage entre le routeur Hub et les routeurs Spoke ?

Pour les tests IPv4, il suffit de reprendre les mêmes tests que ceux effectués plus haut avec la commande hping3.

Installer le paquet `thc-ipv6` sur le routeur Hub pour disposer des outils de tests spécifiques au protocole IPv6.

Rechercher dans les pages de manuels de la commande atk6-thcping6 les options qui permettent de générer du trafic ICMP avec une adresse source falsifiée à destination d'un conteneur hébergé sur un routeur Spoke.

Dans le contexte de la maquette, les requêtes falsifiées sont émises depuis le routeur HubBleu à destination du routeur Spoke1Vert sur lequel les règles de filtrage IPv4 et IPv6 ont été implantées.

- Pour le protocole IPv4, on reprend la commande hping3 avec les mêmes paramètres de dans la question sur la protection au niveau du noyau Linux et on relève le compteur des paquets “jetés“ sur le routeur Spoke.

```
etu@HubBleu:~$ sudo hping3 -1 -a 10.0.2.12 --fast -c 100 10.0.2.11
HPING 10.0.2.11 (ppp1 10.0.2.11): icmp mode set, 28 headers + 0 data bytes

--- 10.0.2.11 hping statistic ---
100 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

```
etu@Spoke1Vert:~$ sudo iptables -vL -t raw
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain PREROUTING (policy ACCEPT 393K packets, 457M bytes)
 pkts bytes target     prot opt in     out     source    destination
   71  5304 DROP       all  --  any    any     anywhere  anywhere   rpfilter invert /* BCP38 */

Chain OUTPUT (policy ACCEPT 94964 packets, 5522K bytes)
 pkts bytes target     prot opt in     out     source    destination
```

- Pour le protocole IPv6, on utilise la commande atk6-thcping6 avec l'adresse d'un conteneur comme source et l'adresse du routeur Spoke dans le VLAN supervision (violet) comme destination.

```
etu@HubBleu:~$ sudo atk6-thcping6 -n 10 enp0s6.470 fda0:7a62:1:0:216:3eff:feda:e1a fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
0000.000  ping packet sent to fe80:1d6::2
*** buffer overflow detected ***: terminated
Abandon
```

```
etu@Spoke1Vert:~$ sudo ip6tables -vL -t raw
# Warning: ip6tables-legacy tables present, use ip6tables-legacy to see them
Chain PREROUTING (policy ACCEPT 37580 packets, 6219K bytes)
 pkts bytes target     prot opt in     out     source     destination
   14   784 DROP       all      any    any     anywhere   anywhere     rpfilter invert /* BCP38 */

Chain OUTPUT (policy ACCEPT 7599 packets, 1505K bytes)
 pkts bytes target     prot opt in     out     source     destination
```

### 4.3.2. Protection contre les dénis de service ICMP

Q87. Comment peut-on se protéger contre un nombre de sollicitations ICMP trop important ?

Rechercher dans le guide Tutoriel iptables la correspondance Limit qui permet de définir un seuil au delà duquel les nouveaux flux réseau ne sont plus acceptés.

Il faut ajouter une règle spécifique au protocole ICMP après celle qui assure le traitement des flux déjà enregistrés dans les tables de suivi d'état (Stateful).

Dans le contexte de la maquette, les nouvelles règles de filtrage sont appliquées sur le routeur Spoke2Vert et le trafic “malveillant“ est généré sur le routeur HubBleu à destination des conteneurs du réseau local du site distant.

On commence par afficher la liste des règles de la table par défaut appelée netfilter de façon à vérifier si la règle générale de suivi des enregistrements est présente ou non dans les chaînes `INPUT` et `FORWARD`.

Dans la copie d'écran ci-dessous, on constate qu'aucune règle de filtrage n'a été appliquée au moment du test.

```
etu@Spoke2Vert:~$ sudo iptables -vL
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source     destination

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

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

Le résultat de la commande `sudo ip6tables -vL` doit être identique à la copie d'écran ci-dessus.

On ajoute les deux règles générales de suivi des conversations en premier dans les chaînes `INPUT` et `FORWARD` pour le protocole IPv4.

```
etu@Spoke2Vert:~$ sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
etu@Spoke2Vert:~$ sudo iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
etu@Spoke2Vert:~$ sudo iptables -vL
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source      destination
    0     0 ACCEPT     all  --  any    any     anywhere    anywhere     ctstate RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source      destination
    0     0 ACCEPT     all  --  any    any     anywhere    anywhere     ctstate RELATED,ESTABLISHED

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

On ajoute aussi deux règles identiques pour le protocole IPv6.

```
etu@Spoke2Vert:~$ sudo ip6tables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

Maintenant que le trafic relatif ou appartenant à un flux enregistré dans la table de suivi d'état est accepté, nous pouvons définir les conditions dans lesquelles un nouveau flux entre de le système de suivi d'état. Ici, on s'intéresse au protocole ICMP et au module Limit. Voici un exemple de règle qui restreint le trafic ICMP à 2 nouvelles entrées par seconde sur les chaînes `INPUT` et `FORWARD`.

- Pour le protocole IPv4.

```
etu@Spoke2Vert:~$ sudo iptables -A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j
etu@Spoke2Vert:~$ sudo iptables -A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW
```

- Pour le protocole IPv6.

```
etu@Spoke2Vert:~$ sudo ip6tables -A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
```

Q88. Comment qualifier le fonctionnement des règles de limitation du nombre de nouvelles requêtes ICMP ?

Rechercher les options de la commande hping3 qui permettent de générer des flux ICMP en utilisant des adresses IPv4 source aléatoires.

Attention ! Il faut positionner la politique par défaut en mode "tout ce qui n'est pas autorisé est interdit“ sur le routeur cible le temps du test de qualification.

On rappelle que dans le contexte de la maquette, les règles de filtrage sont appliquées sur le routeur Spoke2Vert et le trafic “malveillant“ est généré sur le routeur HubBleu à destination des conteneurs du réseau local du site distant.

- On commence par modifier la politique par défaut dans la chaîne `FORWARD` sur le routeur Spoke2Vert.

```
$ sudo iptables -P FORWARD DROP
$ sudo ip6tables -P FORWARD DROP
```

- On lance la génération de trafic ICMP à partir du routeur HubBleu. Dans l'exemple ci-dessous, ce sont 100 paquets ICMP `echo-request` qui sont envoyés avec une adresse IPv4 source aléatoire à destination du conteneur `10.0.2.11`.

```
etu@HubBleu:~$ sudo hping3 -1 --rand-source --fast -c 100 10.0.2.11
```

À la fin de l'émission, les résultats montrent que 76% des 100 requêtes ont été rejetées.

```
--- 10.0.2.11 hping statistic ---
100 packets transmitted, 24 packets received, 76% packet loss
round-trip min/avg/max = 1.1/5.0/9.3 ms
```

- On se place sur le routeur Spoke2Vert et on affiche la liste des règles de la chaîne `FORWARD` avec les compteurs de paquets.

```
etu@Spoke2Vert:~$ sudo iptables -vL FORWARD
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain FORWARD (policy DROP 127 packets, 3556 bytes)
 pkts bytes target     prot opt in     out     source     destination
  143  4004 ACCEPT     all  --  any    any     anywhere   anywhere     ctstate RELATED,ESTABLISHED
   72  2016 ACCEPT     icmp --  any    any     anywhere   anywhere     limit: avg 2/sec burst 5 ctsta
```

Cet échantillon montre que 127 paquets ont été mis à la poubelle et non routés jusqu'au conteneur.

- Comme le jeu de règles sur le routeur Spoke2Vert est trop restreint pour être acceptable par les conteneurs, on replace la politique par défaut à `ACCEPT` sur la chaîne `FORWARD`.

```
$ sudo iptables -P FORWARD ACCEPT
$ sudo ip6tables -P FORWARD ACCEPT
```

### 4.3.3. Protection contre les robots de connexion au service SSH

Q89. Quel est la fonction du paquet fail2ban ?

Afficher la description du paquet fail2ban après l'avoir installé.

```
$ apt install fail2ban
```

```
$ sudo aptitude show fail2ban | grep -A2 Desc | fmt -t -w80
Description : ban hosts that cause multiple authentication errors Fail2ban
 monitors log files (e.g. /var/log/auth.log, /var/log/apache/access.log)
 and temporarily or persistently bans failure-prone addresses by updating
 existing firewall rules.  Fail2ban allows easy specification of different
 actions to be taken such as to ban an IP using iptables or hostsdeny rules,
 or simply to send a notification email.
```

Le rôle du service fail2ban est de repérer les erreurs d'authentification dans les journaux des différents services actifs et de créer une chaîne `iptables` qui bloque les tentatives de connexion suivantes.

Q90. Quel est le numéro de port utilisé par le service SSH sur les routeurs ?

Il est important de connaître les caractéristiques du service qui doit être surveillé par fail2ban. Rechercher dans la liste des ports réseau ouverts celui qui concerne le service SSH.

Dans le contexte de la maquette, le service SSH a été paramétré pour utiliser le port numéro 2222. On obtient la liste des ports en écoute avec les commandes lsof ou ss.

```
$ sudo lsof -i tcp:2222 -sTCP:listen
COMMAND   PID USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
sshd    43975 root    3u  IPv4 7613072      0t0  TCP *:2222 (LISTEN)
sshd    43975 root    4u  IPv6 7613074      0t0  TCP *:2222 (LISTEN)
```

```
$ ss -tapl '( sport = :2222 )' | fmt -t -w80
State  Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
LISTEN 0      128          0.0.0.0:2222      0.0.0.0:*
LISTEN 0      128             [::]:2222         [::]:*
```

Ce sont donc les tentatives de connexion au service SSH sur le port numéro 2222 que le service fail2ban doit surveiller.

Q91. Quel est le fichier de configuration du service SSH qui permet de définir le numéro de port en écoute avec le protocole TCP ?

Repérer le répertoire qui contient les éléments de configuration du service SSH.

C'est le fichier `/etc/ssh/sshd_config` qui contient les paramètres du serveur. Dans le cas de ces manipulations, on a décommenté la ligne avec le mot clé `Port`.

```
$ grep ^Port /etc/ssh/sshd_config
Port 2222
```

Attention ! Si on édite ce fichier de configuration, les modifications ne sont prises en compte qu'au redémarrage du service via la commande `sudo systemctl restart ssh`.

Q92. Quels sont les deux fichiers de configuration principaux fournis à l'installation du paquet `fail2ban` ?

Rechercher dans l'arborescence des fichiers de configuration, les informations relatives aux traitements assurés en cas de détection d'erreurs de connexion à n'importe quel service, puis les informations spécifiques au service SSH.

Dans le répertoire `/etc/fail2ban`, on identifie les deux fichiers demandés.

- Le fichier `/etc/fail2ban/jail.conf` donne la liste des paramètres par défaut en cas de détection de tentatives de connexions en erreur. Voici une extraction des premiers paramètres généraux. Ici, le temps de maintien d'une adresse source en prison (`bantime`) est de 10 minutes mais ce temps est multiplié par deux en cas de récidive.

```
etu@Spoke1Vert:~$ sed -n '/^\[DEF/,/fail2ban_agent/p' /etc/fail2ban/jail.conf | egrep -v '(^#|^$)'
[DEFAULT]
bantime.increment = true
bantime.factor = 2
ignoreself = true
ignorecommand =
bantime  = 10m
findtime  = 10m
maxretry = 5
maxmatches = %(maxretry)s
backend = auto
usedns = warn
logencoding = auto
enabled = false
mode = normal
filter = %(__name__)s[mode=%(mode)s]
destemail = root@localhost
sender = root@<fq-hostname>
mta = sendmail
protocol = tcp
chain = <known/chain>
port = 0:65535
fail2ban_agent = Fail2Ban/%(fail2ban_version)s
```

- Le fichier `/etc/fail2ban/jail.d/defaults-debian.conf` contient les paramètres par défaut pour la distribution avec une section spécifique au service SSH. C'est ce fichier que l'on édite pour l'adapter ua contexte des routeurs de la topologie. On spécifie le numéro de port identifié dans les questions précédentes ainsi qu'un traitement particulier. Voici un exemple de configuration appliqué à la maquette.

```
[sshd]
enabled = true
port    = 2222
filter  = sshd
logpath  = /var/log/auth.log
action = %(action_)s
maxretry = 3
banaction = iptables-new
```

  Là encore, les modifications effectuées sur la configuration ne sont prises en compte qu'au redémarrage du service : `sudo systemctl restart fail2ban`.

Q93. Comment caractériser le fonctionnement du service fail2ban ?

Si le service a été installé et configuré sur un routeur Spoke, il est possible de lancer plusieurs tentatives de connexion SSH depuis le routeur Hub en se trompant de mot de passe.

On peut alors afficher les règles de filtrage iptables et consulter l'état de la prison fail2ban.

On commence par lancer plusieurs tentatives (au moins 3) de connexion SSH à partir du routeur Hub.

```
etu@HubBleu:~$ ssh -p 2222 etu@10.47.1.2
etu@10.47.1.2's password:
Permission denied, please try again.
etu@10.47.1.2's password:
Permission denied, please try again.
etu@10.47.1.2's password:
etu@10.47.1.2: Permission denied (publickey,password).
etu@HubBleu:~$ ssh -p 2222 etu@10.47.1.2
ssh: connect to host 10.47.1.2 port 2222: Connection refused
```

On relève ensuite les résultats côté routeur Spoke.

La liste des règles de filtrage montre qu'une nouvelle chaîne a été ajoutée. Dans cette chaîne, on reconnaît l'adresse IPv4 du lien PPP côté Hub.

```
etu@Spoke1Vert:~$ sudo iptables -vL
Chain INPUT (policy ACCEPT 335K packets, 401M bytes)
 pkts bytes target     prot opt in     out     source      destination
    2   120 f2b-sshd   tcp  --  any    any     anywhere    anywhere   state NEW tcp dpt:2222

Chain FORWARD (policy ACCEPT 60353 packets, 58M bytes)
 pkts bytes target     prot opt in     out     source      destination

Chain OUTPUT (policy ACCEPT 95465 packets, 5602K bytes)
 pkts bytes target     prot opt in     out     source      destination

Chain f2b-sshd (1 references)
 pkts bytes target     prot opt in     out     source      destination
    2   120 REJECT     all  --  any    any     10.47.1.1   anywhere   reject-with icmp-port-unreachable
    0     0 RETURN     all  --  any    any     anywhere    anywhere
```

Toujours sur le routeur Spoke, on relève l'état du service fail2ban et plus particulièrement celui de la "prison" spécifique au protocole SSH.

```
etu@Spoke1Vert:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed: 3
|  `- File list: /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned: 1
   `- Banned IP list: 10.47.1.1
```

Enfin, on répète l'opération avec l'adresse IPv6 du routeur Spoke sur le lien PPP.

```
etu@HubBleu:~$ ssh -p 2222 fe80::5c93:b536:53e7:f976%ppp0
etu@fe80::5c93:b536:53e7:f976%ppp0's password:
Permission denied, please try again.
etu@fe80::5c93:b536:53e7:f976%ppp0's password:
Permission denied, please try again.
etu@fe80::5c93:b536:53e7:f976%ppp0's password:
etu@fe80::5c93:b536:53e7:f976%ppp0: Permission denied (publickey,password).
etu@HubBleu:~$ ssh -p 2222 fe80::5c93:b536:53e7:f976%ppp0
ssh: connect to host fe80::5c93:b536:53e7:f976%ppp0 port 2222: Connection refused
```

On voit apparaître une nouvelle adresse dans la liste sur le routeur Spoke.

```
etu@Spoke1Vert:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 1
|  |- Total failed: 7
|  `- File list: /var/log/auth.log
`- Actions
   |- Currently banned: 2
   |- Total banned: 2
   `- Banned IP list: 10.47.1.1 fe80::490a:39d4:1a05:f33d
```

Les règles de filtrage pour le protocole IPv6 ont aussi été complétées.

```
etu@Spoke1Vert:~$ sudo ip6tables -vL
# Warning: ip6tables-legacy tables present, use ip6tables-legacy to see them
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source    destination
    3   240 f2b-sshd   tcp      any    any     anywhere  anywhere     state NEW tcp dpt:2222

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

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

Chain f2b-sshd (1 references)
 pkts bytes target     prot opt in     out     source    destination
    2   160 REJECT     all      any    any     fe80::490a:39d4:1a05:f33d  anywhere   reject-with icmp6-p
    1    80 RETURN     all      any    any     anywhere  anywhere
```

## 4.4. Règles de filtrage communes à toutes les configurations

La mise en place du filtrage réseau sur les équipements doit répondre à deux principes.

- On considère que les équipements d'interconnexion mis en œuvre dans ces travaux pratiques délimitent des périmètres de dimension moyenne. Par conséquent, on a une connaissance exhaustive des flux réseaux sur le système. On adopte donc la règle : tout trafic réseau non autorisé est interdit.
- On fait le choix d'un filtrage basé sur le suivi de communication (stateful inspection). On cherche donc à écrire des règles qui décrivent le plus précisément possible le premier paquet qui doit être enregistré dans la table de suivi de communication. Ces règles de description du premier paquet doivent être placées après celle qui laisse passer le trafic qui correspond ou qui est relatif à une communication déjà enregistrée dans les tables.
- Dans le but de simplifier l'étude du filtrage, on fait le choix d'autoriser tous les flux sortants émis par les routeurs Hub et Spoke. On laisse donc la politique par défaut à `ACCEPT` pour les chaînes `OUTPUT` des routeurs.

On commence par afficher les règles actives sur les différents routeurs à l'issue des questions de la section précédente : Section 4.3, « Protection de base des routeurs Hub et Spoke ».

Attention ! Les noms d'interfaces correspondent à la maquette de test.

- Régles de filtrage IPv4 côté Hub : fichier `/etc/iptables/rules.v4`.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

- Régles de filtrage IPv6 côté Hub : fichier `/etc/iptables/rules.v6`.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s fe80::/10 -j ACCEPT
-A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
COMMIT
```

- Régles de filtrage IPv4 côté Spoke : fichier `/etc/iptables/rules.v4`.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

- Régles de filtrage IPv6 côté Spoke : fichier `/etc/iptables/rules.v6`.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s fe80::/10 -j ACCEPT
-A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
COMMIT
```

Q94. Dans les jeux de règles déjà en place, comment identifier les règles qui traitent les flux réseau dèjà enregistrés dans le suivi de communication ?

La section «7.3. États de l'espace utilisateur» du Tutoriel iptables décrit les correspondances entre les états et les flux réseau.

Le tableau de la section «7.3. États de l'espace utilisateur» permet de sélectionner les états `ESTABLISHED` et `RELATED` que l'on retrouve en première position dans les chaînes `INPUT` et `FORWARD`.

Voici un exemple qui illustre l'utilisation de ces règles dans le contexte de la maquette. L'évolution des compteurs, montre qu'une règle est effectivement utilisée dans le traitement du trafic réseau.

```
etu@Spoke2Vert:~$ sudo ip6tables -vL
# Warning: ip6tables-legacy tables present, use ip6tables-legacy to see them
Chain INPUT (policy ACCEPT 26 packets, 2192 bytes)
 pkts bytes target     prot opt in     out     source     destination
  591 54216 ACCEPT     all      any    any     anywhere   anywhere     ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     icmp     any    any     anywhere   anywhere     limit: avg 2/sec burst 5

Chain FORWARD (policy ACCEPT 6 packets, 480 bytes)
 pkts bytes target     prot opt in     out     source     destination
    6   360 ACCEPT     all      any    any     anywhere   anywhere     ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     icmp     any    any     anywhere   anywhere     limit: avg 2/sec burst 5

Chain OUTPUT (policy ACCEPT 505 packets, 97048 bytes)
 pkts bytes target     prot opt in     out     source     destination
```

Q95. Quelles règles faut-il ajouter pour autoriser les nouveaux flux réseau depuis et vers l'interface de boucle locale (chaîne `INPUT`) ?

Pour que les processus locaux au système puissent communiquer entre eux, il est essentiel d'autoriser le trafic sur l'interface de boucle locale `lo`.

On insère une nouvelle règle sur la chaîne `INPUT` qui admet tous les nouveaux paquets entrant sur l'interface de `lo` sans tenir compte de la table de suivi des communications.

La même règle est insérée pour les protocoles IPv4 et IPv6. On utilise les numéros de lignes pour insérer les nouvelles règles en postion 2.

```
etu@Spoke2Vert:~$ sudo iptables -I INPUT 2 -i lo -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -I INPUT 2 -i lo -j ACCEPT
```

On relève un exemple de résultat en affichant la liste des règles actives avec leurs numéros.

```
etu@Spoke2Vert:~$ sudo iptables -vL --line-numbers
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain INPUT (policy ACCEPT 939 packets, 364K bytes)
num   pkts bytes target     prot opt in     out     source     destination
1      955 1336K ACCEPT     all  --  any    any     anywhere   anywhere     ctstate RELATED,ESTABLISHED
2        0     0 ACCEPT     all  --  lo     any     anywhere   anywhere
3        0     0 ACCEPT     icmp --  any    any     anywhere   anywhere     limit: avg 2/sec burst 5

Chain FORWARD (policy ACCEPT 12 packets, 825 bytes)
num   pkts bytes target     prot opt in     out     source     destination
1      864  653K ACCEPT     all  --  any    any     anywhere   anywhere     ctstate RELATED,ESTABLISHED
2        0     0 ACCEPT     icmp --  any    any     anywhere   anywhere     limit: avg 2/sec burst 5

Chain OUTPUT (policy ACCEPT 573 packets, 34052 bytes)
num   pkts bytes target     prot opt in     out     source     destination
```

À partir de ce jeu de règles, on peut lancer un test ICMP et relever les compteurs d'utilisation de la nouvelle règle.

```
etu@Spoke2Vert:~$ ping -q -c 4 ::1
PING ::1(::1) 56 data bytes

--- ::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.116/0.139/0.162/0.022 ms
```

```
etu@Spoke2Vert:~$ sudo ip6tables -vL INPUT --line-numbers
# Warning: ip6tables-legacy tables present, use ip6tables-legacy to see them
Chain INPUT (policy ACCEPT 33 packets, 2968 bytes)
num   pkts bytes target     prot opt in     out     source    destination
1     1445  131K ACCEPT     all      any    any     anywhere  anywhere     ctstate RELATED,ESTABLISHED
2        2   208 ACCEPT     all      lo     any     anywhere  anywhere
3        0     0 ACCEPT     ipv6-icmp  any    any     anywhere  anywhere     limit: avg 2/sec burst 5
```

Q96. Quelles règles faut-il ajouter pour autoriser les nouvelles connexions SSH et les intégrer dans la table de suivi des communications ?

Le protocole de couche transport utilisé est TCP et le numéro de port utilisé par le service SSH est `2222`.

La section «7.3. États de l'espace utilisateur» du Tutoriel iptables décrit les correspondances entre les états et les flux réseau. Rechercher la clé relative aux nouveaux flux entrants.

Le tableau de la section «7.3. États de l'espace utilisateur» permet de sélectionner l'état `NEW`.

Voici un exemple d'ajout de règles dans le contexte de la maquette.

```
etu@Spoke2Vert:~$ sudo iptables -A INPUT -p tcp --syn --dport 2222 \
  -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -A INPUT -p tcp --syn --dport 2222 \
  -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
```

Comme précédemment, on peut relever les compteurs suite à une nouvelle connexion SSH.

```
etu@Spoke2Vert:~$ sudo ip6tables -vL INPUT
# Warning: ip6tables-legacy tables present, use ip6tables-legacy to see them
Chain INPUT (policy ACCEPT 41 packets, 3608 bytes)
 pkts bytes target     prot opt in     out     source    destination
 2227  206K ACCEPT     all      any    any     anywhere  anywhere     ctstate RELATED,ESTABLISHED
    2   208 ACCEPT     all      lo     any     anywhere  anywhere
    0     0 ACCEPT     ipv6-icmp  any    any     anywhere  anywhere     limit: avg 2/sec burst 5
    1    80 ACCEPT     tcp      any    any     anywhere  anywhere     tcp dpt:2222 flags:FIN,SYN,RST,ACK
```

Q97. Quelle est l'instruction qui définit la politique par défaut à appliquer sur les chaînes de la table `netfilter` ?

Il s'agit d'appliquer le principe de filtrage énoncé en début de section qui veut que tout trafic non autorisé soit interdit.

La section «9.3. Commandes» du Tutoriel iptables donne la syntaxe de configuration de cible par défaut pour les chaînes : `INPUT`, `FORWARD` et `OUTPUT`.

On consulte la documentation et on relève la commande -P. Ensuite, on sélectionne la politique par défaut adaptée au contexte : `DROP`.

Voici un exemple sur un routeur Spoke.

```
etu@Spoke2Vert:~$ sudo iptables -P INPUT DROP
etu@Spoke2Vert:~$ sudo ip6tables -P INPUT DROP
etu@Spoke2Vert:~$ sudo iptables -P FORWARD DROP
etu@Spoke2Vert:~$ sudo ip6tables -P FORWARD DROP
etu@Spoke2Vert:~$ sudo iptables -P OUTPUT ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -P OUTPUT ACCEPT
```

Une fois ces règles basiques en place, on peut aborder les filtrages réseau spécifiques à la topologie de travaux pratiques.

## 4.5. Règles de filtrage sur le routeur Hub

Dans cette section, on doit compléter les règles de filtrage pour répondre à deux objectifs :

- Le routeur Hub doit autoriser le trafic issu des routeurs Spoke vers l'Internet.
- Les demandes de connexion aux services Web hébergés sur les conteneurs desservis par les routeurs Spoke doivent être redirigées via la traduction des adresses destination.

  Voici un exemple de correspondances de numéros de ports pour l'accès aux différents services web.

Tableau 4.1. Correspondance entre numéro de port et service Web

| numéros de port Hub : http,https | conteneur |
|---|---|
| 8010,8453 | `10.0.1.10`<br>`fda0:7a62:1:0:216:3eff:feda:e1a` |
| 8011,8454 | `10.0.1.11`<br>`fda0:7a62:1:0:216:3eff:fec4:d325` |
| 8012,8455 | `10.0.1.12`<br>`fda0:7a62:1:0:216:3eff:fe66:86fb` |
| 8020,8463 | `10.0.2.10`<br>`fda0:7a62:2:0:216:3eff:feda:e1a` |
| 8021,8464 | `10.0.2.11`<br>`fda0:7a62:2:0:216:3eff:fec4:d325` |
| 8022,8465 | `10.0.2.12`<br>`fda0:7a62:2:0:216:3eff:fe66:86fb` |

Avant d'aborder les questions, on commence par afficher le contenu des deux fichiers `/etc/iptables/rules.v4` et `/etc/iptables/rules.v6` qui correspondent à la situation initiale avant de répondre aux objectifs de cette section.

- Jeu de règles pour le protocole IPv4.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

- Jeu de règles pour le protocole IPv6.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s fe80::/10 -j ACCEPT
-A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
COMMIT
```

Q98. Comment autoriser et enregistrer dans le mécanisme de suivi des états les flux entrants par les interfaces WAN du routeur Hub ?

Rechercher dans les pages de manuels de la commande iptables le moyen de désigner plusieurs interfaces en une seule règle.

C'est le symbole `+` qui permet de regrouper les interfaces `ppp0` et `ppp1` dans une même règle de filtrage.

On ajoute donc les deux règles suivantes sur le routeur Hub.

```
etu@HubBleu:~$ sudo iptables -A FORWARD -i ppp+ -m conntrack --ctstate NEW -j ACCEPT
etu@HubBleu:~$ sudo ip6tables -A FORWARD -i ppp+ -m conntrack --ctstate NEW -j ACCEPT
```

Q99. Comment valider l'utilisation de ces deux nouvelles règles à partir d'un routeur Spoke ?

Il suffit de lancer un téléchargement depuis un routeur Spoke en utilisant successivement les protocoles IPv4 et IPv6. Ensuite, on relève les enregistrements sur le routeur Hub à l'aide de la commande conntrack.

Voici un exemple de relevé avec un téléchargement suffisamment volumineux pour collecter la liste des entrées de suivi d'état sur le routeur Hub.

```
etu@Spoke2Vert:~$ wget -4 -O /dev/null https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz
```

```
etu@HubBleu:~$ sudo conntrack -f ipv4 -L
tcp      6 300 ESTABLISHED src=10.47.3.2 dst=151.101.121.176 sport=60962 dport=443 \
   src=151.101.121.176 dst=10.141.0.162 sport=443 dport=60962 [ASSURED] mark=0 use=2
```

```
etu@Spoke2Vert:~$ wget -6 -O /dev/null https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz
```

```
etu@HubBleu:~$ sudo conntrack -f ipv6 -L
tcp      6 300 ESTABLISHED src=fda0:7a62:2::1 dst=2a04:4e42:1d::432 sport=49156 dport=443 \
   src=2a04:4e42:1d::432 dst=2001:678:3fc:12c::2 sport=443 dport=49156 [ASSURED] mark=0 use=2
```

Q100. Comment implanter les règles de traduction d'adresses IPv4 et IPv6 destination de façon à rendre accessibles les services Web configurés dans les conteneurs situés dans les réseaux desservis par les routeurs Spoke ?

Il faut rechercher la syntaxe des règles de la cible DNAT à appliquer dans la table des règles de traduction d'adresses (nat) ainsi que la syntaxe des règles à ajouter dans la chaîne `FORWARD` de la table `netfilter`.

Ces nouvelles règles doivent être conformes au tableau de correspondance donné en début de section. Bien sûr, les adresses doivent être modifiées en fonction du plan d'adressage du document Topologie Hub & Spoke avec le protocole PPPoE.

Comme indiqué dans l'énoncé de la question, l'ajout des règles comprend deux parties : les règles de la table `nat` et les règles de la table `netfilter`.

Dans le contexte de la maquette, on a édité les fichiers `/etc/iptables/rules.v4` et `/etc/iptables/rules.v6`.

- Pour le protocole IPv4.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8010 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C0 -j DNAT --to 10.0.1.10:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8453 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C0 -j DNAT --to 10.0.1.10:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8011 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C1 -j DNAT --to 10.0.1.11:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8454 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C1 -j DNAT --to 10.0.1.11:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8012 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C2 -j DNAT --to 10.0.1.12:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8455 -m conntrack --ctstate NEW \
  -m comment --comment Spoke1C2 -j DNAT --to 10.0.1.12:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8020 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C0 -j DNAT --to 10.0.2.10:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8463 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C0 -j DNAT --to 10.0.2.10:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8021 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C1 -j DNAT --to 10.0.2.11:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8464 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C1 -j DNAT --to 10.0.2.11:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8022 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C2 -j DNAT --to 10.0.2.12:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8465 -m conntrack --ctstate NEW \
  -m comment --comment Spoke2C2 -j DNAT --to 10.0.2.12:443
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp --syn --dport 2222 -m conntrack --ctstate NEW \
  -m comment --comment SSH -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -p tcp --syn --dport 2222 -m conntrack --ctstate NEW \
  -m comment --comment SSH -j ACCEPT
-A FORWARD -i ppp+ -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -d 10.0.1.10/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke1C0 -j ACCEPT
-A FORWARD -d 10.0.1.11/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke1C1 -j ACCEPT
-A FORWARD -d 10.0.1.12/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke1C2 -j ACCEPT
-A FORWARD -d 10.0.2.10/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke2C0 -j ACCEPT
-A FORWARD -d 10.0.2.11/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke2C1 -j ACCEPT
-A FORWARD -d 10.0.2.12/32 -p tcp --syn -m multiport --dports 80,443 \
  -m comment --comment Spoke2C2 -j ACCEPT
COMMIT
```

- Pour le protocole IPv6.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ N A T
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8010 \
  -m comment --comment Spoke1C0 -j DNAT --to [fda0:7a62:1:0:216:3eff:feda:e1a]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8453 \
  -m comment --comment Spoke1C0 -j DNAT --to [fda0:7a62:1:0:216:3eff:feda:e1a]:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8011 \
  -m comment --comment Spoke1C1 -j DNAT --to [fda0:7a62:1:0:216:3eff:fec4:d325]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8454 \
  -m comment --comment Spoke1C1 -j DNAT --to [fda0:7a62:1:0:216:3eff:fec4:d325]:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8012 \
  -m comment --comment Spoke1C2 -j DNAT --to [fda0:7a62:1:0:216:3eff:fe66:86fb]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8455 \
  -m comment --comment Spoke1C2 -j DNAT --to [fda0:7a62:1:0:216:3eff:fe66:86fb]:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8020 \
  -m comment --comment Spoke2C0 -j DNAT --to [fda0:7a62:2:0:216:3eff:feda:e1a]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8463 \
  -m comment --comment Spoke2C0 -j DNAT --to [fda0:7a62:2:0:216:3eff:feda:e1a]:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8021 \
  -m comment --comment Spoke2C1 -j DNAT --to [fda0:7a62:2:0:216:3eff:fec4:d325]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8464 \
  -m comment --comment Spoke2C1 -j DNAT --to [fda0:7a62:2:0:216:3eff:fec4:d325]:443
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8022 \
  -m comment --comment Spoke2C2 -j DNAT --to [fda0:7a62:2:0:216:3eff:fe66:86fb]:80
-A PREROUTING -i enp0s6.300 -p tcp --syn --dport 8465 \
  -m comment --comment Spoke2C2 -j DNAT --to [fda0:7a62:2:0:216:3eff:fe66:86fb]:443
-A POSTROUTING -o enp0s6.300 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s fe80::/10 -j ACCEPT
-A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A INPUT -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
-A INPUT -m limit --limit 1/sec -m conntrack --ctstate INVALID -j DROP
-A INPUT -m limit --limit 1/sec -j NFLOG
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A FORWARD -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
-A FORWARD -i ppp+ -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -d fda0:7a62:1:0:216:3eff:feda:e1a/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke1C0 -j ACCEPT
-A FORWARD -d fda0:7a62:1:0:216:3eff:fec4:d325/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke1C1 -j ACCEPT
-A FORWARD -d fda0:7a62:1:0:216:3eff:fe66:86fb/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke1C2 -j ACCEPT
-A FORWARD -d fda0:7a62:2:0:216:3eff:feda:e1a/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke2C0 -j ACCEPT
-A FORWARD -d fda0:7a62:2:0:216:3eff:fec4:d325/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke2C1 -j ACCEPT
-A FORWARD -d fda0:7a62:2:0:216:3eff:fe66:86fb/128 -p tcp --syn -m multiport --dports 80,443 \
  -m conntrack --ctstate NEW -m comment --comment Spoke2C2 -j ACCEPT
-A FORWARD -m limit --limit 1/sec -m conntrack --ctstate INVALID -j DROP
-A FORWARD -m limit --limit 1/sec -j NFLOG
COMMIT
```

Pour rétablir les lignes des copies d'écran ci-dessus, il est possible d'utiliser la commande ci-dessous avec laquelle le fichier `rules.txt` contient les lignes coupées avec le caractère `\`.

```
$ sudo sed '/^[ \-].*\\$/N;s/\\\n *//' rules.txt
```

Comme pour toutes les autres sections, on n'oublie pas de sauvegarder le jeu des règles qui ont été validées.

```
$ sudo sh -c "iptables-save >/etc/iptables/rules.v4"
$ sudo sh -c "ip6tables-save >/etc/iptables/rules.v6"
```

## 4.6. Règles de filtrage sur le routeur Spoke

Comme pour la section précédente sur le routeur Hub, on doit compléter le jeu de règles de filtrage pour répondre à deux objectifs :

- Le routeur Spoke doit autoriser et enregistrer dans la table de suivi d'état les flux réseaux sortants issus du réseau des conteneurs.
- Ce même routeur Spoke doit autoriser et enregistrer dans la table de suivi d'état les flux réseaux entrants à destination des services Web hébergés par les conteneurs.

On commence par afficher le contenu des deux fichiers `/etc/iptables/rules.v4` et `/etc/iptables/rules.v6` d'un routeur Spoke qui correspondent à la situation initiale avant de traiter les questions de cette section.

- Jeu de règles pour le protocole IPv4.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m limit --limit 2/sec -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
COMMIT
```

- Jeu de règles pour le protocole IPv6.

```
#~~~~~~~~~~~~ R A W
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -m rpfilter --invert -m comment --comment BCP38 -j DROP
COMMIT
#~~~~~~~~~~~~ F I L T E R
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s fe80::/10 -j ACCEPT
-A INPUT -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A INPUT -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -m limit --limit 2/sec -j ACCEPT
-A FORWARD -p tcp --syn --dport 2222 -m conntrack --ctstate NEW -m comment --comment SSH -j ACCEPT
COMMIT
```

Q101. Comment autoriser et enregistrer dans le mécanisme de suivi des états les flux sortants par l'interface WAN du routeur Spoke ?

Rechercher dans les pages de manuels de la commande iptables le moyen de désigner une interface ainsi que le sens des flux qui transitent par cette interface.

C'est la directive `-o` qui permet de désigner les flux sortants par l'intreface `ppp0`.

On ajoute donc les deux règles suivantes sur les routeurs Spoke.

```
etu@Spoke2Vert:~$ sudo iptables -A FORWARD -o ppp0 -m conntrack --ctstate NEW -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -A FORWARD -o ppp0 -m conntrack --ctstate NEW -j ACCEPT
```

Q102. Comment valider l'utilisation de ces deux nouvelles règles à partir d'un routeur Spoke ?

Il suffit de lancer un téléchargement depuis un conteneur desservi par le routeur Spoke en utilisant successivement les protocoles IPv4 et IPv6. Ensuite, on relève les enregistrements sur le même routeur Spoke à l'aide de la commande conntrack.

Voici un exemple de relevé avec un téléchargement suffisamment volumineux pour collecter la liste des entrées de suivi d'état sur le routeur Spoke.

On commence par s'assurer que le paquet wget est bien installé sur le conteneur depuis lequel on effectue le test.

```
etu@Spoke2Vert:~$ lxc exec container0 -- apt install wget
```

On passe ensuite au téléchargement et au relevé de la table de suivi d'état.

```
etu@Spoke2Vert:~$ lxc exec container0 -- \
  wget -4 -O /dev/null https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz
```

```
etu@Spoke2Vert:~$ sudo conntrack -f ipv4 -L
tcp      6 2 CLOSE src=10.0.2.10 dst=151.101.113.176 sport=59888 dport=443 \
    src=151.101.113.176 dst=10.0.2.10 sport=443 dport=59888 [ASSURED] mark=0 use=1
```

```
etu@Spoke2Vert:~$ lxc exec container0 -- \
  wget -6 -O /dev/null https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz
```

```
etu@Spoke2Vert:~$ sudo conntrack -f ipv6 -L
tcp      6 300 ESTABLISHED src=fda0:7a62:2:0:216:3eff:feda:e1a dst=2a04:4e42:3::432 sport=38384 dport=44
    src=2a04:4e42:3::432 dst=fda0:7a62:2:0:216:3eff:feda:e1a sport=443 dport=38384 [ASSURED] mark=0 use=
```

Q103. Comment autoriser les flux Web entrants par l'interface WAN vers les conteneurs ?

Rechercher dans les options de la commande iptables celles qui permettent de désigner les interfaces d'entrée et de sortie ainsi que les numéros de ports associés au service Web.

Les options utiles pour les interfaces sont `-i` pour l'entrée et `-o` pour la sortie. Les numéros de ports `80` et `443` sont regroupés avec le module `multiport`.

Voici un exemple des deux règles à ajouter.

```
etu@Spoke2Vert:~$ sudo iptables -A FORWARD -i ppp0 -o sw-vlan2 \
  -p tcp --syn -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
etu@Spoke2Vert:~$ sudo ip6tables -A FORWARD -i ppp0 -o sw-vlan2 \
  -p tcp --syn -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
```

Q104. Comment valider l'utilisation des deux règles ajoutées dans la question précédente ?

Reprendre, depuis le routeur Hub, l'utilsation de la commande wget telle qu'elle a été présentée dans la section Routeurs Spoke du support Topologie Hub & Spoke avec le protocole PPPoE.

Voici un exemple des résultats obtenus sur le routeur Hub de la maquette. Le code HTTP `200` montre que la requête a bien été traitée par le serveur Web de chaque conteneur.

```
etu@HubBleu:~$ for addr in 10.0.2.10 10.0.2.11 10.0.2.12;\
  do sh -c "wget -O /dev/null http://$addr 2>&1 | grep \"HTTP\" "; done
requête HTTP transmise, en attente de la réponse… 200 OK
requête HTTP transmise, en attente de la réponse… 200 OK
requête HTTP transmise, en attente de la réponse… 200 OK
```

```
etu@HubBleu:~$ for addr in fda0:7a62:2:0:216:3eff:feda:e1a \
  fda0:7a62:2:0:216:3eff:fec4:d325 \
  fda0:7a62:2:0:216:3eff:fe66:86fb; \
  do sh -c "wget -O /dev/null http://[$addr] 2>&1 | grep \"HTTP\" "; done
requête HTTP transmise, en attente de la réponse… 200 OK
requête HTTP transmise, en attente de la réponse… 200 OK
requête HTTP transmise, en attente de la réponse… 200 OK
```

On se place ensuite sur le routeur Spoke pour relever les compteurs des règles de filtrage.

```
etu@Spoke2Vert:~$ sudo iptables -vL FORWARD | grep http
    6   360 ACCEPT     tcp  --  ppp0   sw-vlan2  anywhere  anywhere \
    tcp flags:FIN,SYN,RST,ACK/SYN multiport dports http,https ctstate NEW
```

```
etu@Spoke2Vert:~$ sudo ip6tables -vL FORWARD | grep http
    3   240 ACCEPT     tcp      ppp0   sw-vlan2  anywhere  anywhere \
    tcp flags:FIN,SYN,RST,ACK/SYN multiport dports http,https ctstate NEW
```

## 4.7. Documents de référence

### IETF & IANA

Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, BCP 38, `rp_filter`

Le document standard RFC2827 Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing est un guide de bonne pratiques pour se protéger contre l'usurpation des adresses sources. Dans le monde GNU/Linux, la fonction clé est appélée `rp_filter` pour Reverse Path Filtering.

### Distribution Debian GNU/Linux

Manuel de référence Debian

Manuel de référence Debian : configuration du réseau : chapitre du manuel de référence Debian consacré à la configuration réseau.

### Site inetdoc.net

Configuration d'une interface de réseau local

Configuration d'une interface de réseau local : identification du type d'interface, de ses caractéristiques et manipulations des paramètres. Ce support fournit une méthodologie de dépannage simple d'une connexion réseau.

Fonctions réseau du noyau Linux

Configuration des fonctions réseau & compilation du noyau Linux : présentation et configuration des fonctions réseau du noyau LINUX

Didacticiel sur Iptables

Tutoriel iptables : guide très complet sur le fonctionnement du filtrage réseau avec les noyaux Linux.

Guide Pratique du NAT

Guide Pratique du NAT : Ce document décrit comment réaliser du camouflage d'adresse IP, un serveur mandataire transparent, de la redirection de ports ou d'autres formes de Traduction d'adresse réseau (Network Address Translation ou NAT) avec le noyau Linux 2.4.

CHAPITRE 5

# Introduction au routage inter-VLAN

**Résumé**

**Le routage inter-VLAN sur les systèmes GNU/Linux présente de nombreux intérêts tant du point de vue conception que du point de vue exploitation. Avec un système GNU/Linux on peut combiner les fonctions de cloisonnement des domaines de diffusion avec d'autres services tels que le filtrage réseau netfilter/iptables. De plus, avec une infrastructure hétérogène associant plusieurs générations et|ou marques de commutateurs, GNU/Linux permet d'homogénéiser l'exploitation.**

## Table des matières



## 5.1. Réseaux locaux virtuels et routage

Les définitions importantes sur les réseaux locaux virtuels et le routage associé sont présentées dans l'article Routage Inter-VLAN

On rappelle simplement que la notion de réseau local virtuel ou VLAN permet de constituer des groupes logiques dans les réseaux Ethernet au niveau liaison de la modélisation OSI. Sans l'ajout d'une balise définie dans le standard IEEE 802.1Q, le format des adresses MAC ne permet aucun découpage en sous-ensembles (à l'exception du trafic multicast qui ne nous concerne pas ici). Une fois que l'on peut repérer l'appartenance à un groupe logique sur la base des étiquettes ajoutées aux trames il est possible de distribuer un domaine de diffusion entre plusieurs équipements physiques distincts.

On atteint ainsi un objectif très important. Il est possible de concevoir une topologie logique de réseau totalement indépendante de la topologie physique.

Réseau virtuel ou pas, il ne faut pas oublier les éléments suivants sur la segmentation des réseaux locaux.

- Une interface de commutateur délimite un domaine de collision.
- Une interface de routeur délimite à la fois un domaine de collision et un domaine de diffusion.

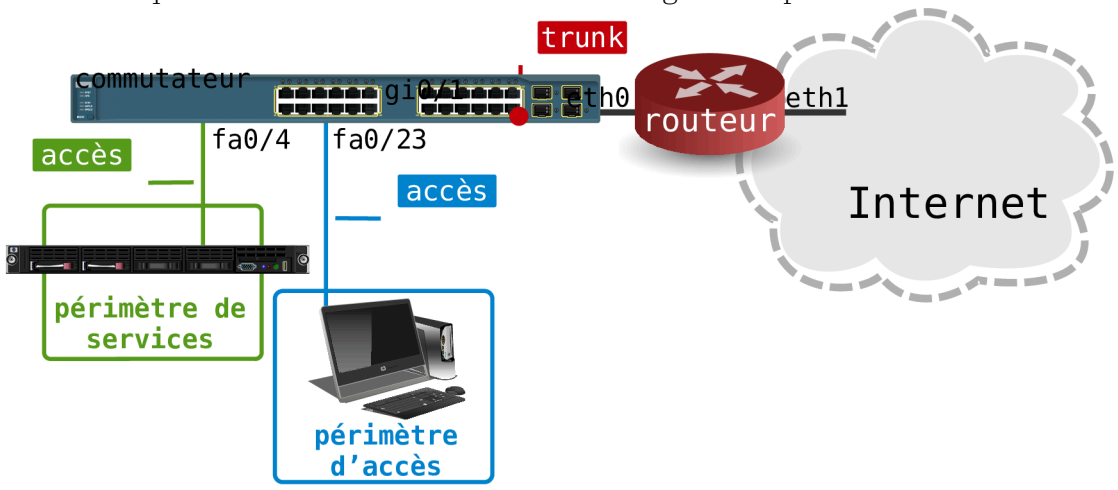
## 5.2. Etude d'une configuration type

La configuration type étudiée ici est une maquette réduite qui comprend un routeur et un commutateur physique. Pour les besoin des l'illustration, on dissocie l'équipement responsable de la commutation de paquets de l'équipement en charge de la commutation de trames.

Le routeur unique correspond bien à la réalité des réseaux modernes. Du réseau d'agence d'une centaine d'hôtes au réseau de campus de plusieurs milliers d'hôtes, seule la capacité de traitement de l'équipement varie.

Le commutateur unique correspond beaucoup moins à la réalité. Même dans un réseau d'agence, on dépasse très vite le cap des 48 ports connectés. On utilise alors un équipement avec une bonne capacité de commutation qui assure la distribution vers des commutateurs dédiés aux accès des hôtes. Tous ces commutateurs sont reliés entre eux à l'aide de trunks qui véhiculent les flux marqués des réseaux virtuels.

Dans l'illustration présentée ici, les deux couches distribution et accès sont «synthétisées» sur un seul équipement. Un trunk sur un lien gigabit relie le routeur au commutateur. En véhiculant les flux marqués entre le routeur et le commutateur il assure la liaison entre routage et commutation de trames. Les hôtes directement connectés au commutateur n'ont aucune connaissance des balises IEEE 802.1q. Ils ne nécessitent donc aucune configuration particulière.




Topologie type

Cette infrastructure type comprend 2 périmètres reliés au réseau public Internet. Un premier périmètre de services utilisé pour l'hébergement des services accessibles depuis le réseau public : DNS, Web, courrier électronique, etc. Un second périmètre pour les postes de travail qui ne doivent pas être accessibles depuis le réseau public.

On ajoute aux deux périmètres classiques, un réseau particulier dédié à la gestion de l'infrastructure : configuration des équipements, métrologie, journalisation, etc.

Tableau 5.1. Plan d'adressage des périmètres

| Nom | VLAN numéro | Adresse IP |
|---|---|---|
| Management | 1 | `192.168.2.0/24` |
| Services | 100 | `192.168.100.0/24` |
| Accès | 200 | `192.168.200.0/24` |

Le tableau ci-dessus établit la correspondance entre les périmètres, les réseaux virtuels et les réseaux IP à interconnecter.

## 5.2.1. Configuration du trunk

### Communications réseau dans le périmètre Management

Du point de vue configuration, ce réseau est très particulier. Il véhicule les trames sans balises IEEE 802.1q entre le routeur et le commutateur. On associe à ce périmètre le VLAN natif du trunk.

Côté routeur GNU/Linux, on configure l'interface de façon classique puisqu'il s'agit de traiter des trames Ethernet standard.

```
# ip addr add 192.168.2.2/24 brd + dev eth0
```

Côté commutateur, on utilise la notion de VLAN «natif» pour configurer l'interface en mode trunk.

```
!
interface GigabitEthernet0/1
 switchport trunk native vlan 1
 switchport mode trunk
 no cdp enable

<snipped/>
!
interface Vlan1
 ip address 192.168.2.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip route-cache
```

La configuration du trunk est la suivante :

```
#sh int gi0/1 trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/1       on           802.1q         trunking      1

Port      Vlans allowed on trunk
Gi0/1       1-4094

Port        Vlans allowed and active in management domain
Gi0/1       1,100,200

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/1       1,100,200
```

Les règles d'utilisation des trames sans balises IEEE 802.1q sont les suivantes :

- Toute trame appartenant au VLAN natif peut être émise sans balise IEEE 802.1q sur un port en mode trunk par le commutateur.
- Toute trame reçue sans balise IEEE 802.1q sur un port en mode trunk du commutateur appartient au VLAN natif.

On complétera la configuration du commutateur de façon à ce que toutes les opérations de gestion de l'équipement passent par ce VLAN natif.

À ce niveau, les tests de communication réseau sont très simples.

- Côté routeur :

```
RouterA:~$ ping -c 2 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=255 time=19.4 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=255 time=1.22 ms

--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.226/10.355/19.484/9.129 ms
```

- Côté commutateur :

```
Switch#ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

## 5.2.2. Configuration IEEE 802.1Q sur le Routeur GNU/Linux

**Communications réseau dans les périmètres Services et Accès**

Cette fois ci, il est indispensable de traiter les flux marqués avec les balises IEEE 802.1q. Aujourd'hui, tous les noyaux fournis avec les distributions Linux comme Debian GNU/Linux disposent d'un module appelé `8021q`.

```
$ find /lib/modules/`uname -r` -name 8021q
/lib/modules/4.13.0-1-amd64/kernel/net/8021q
```

Le chargement de ce module se fait automatiquement dès qu'une opération relative aux étiquettes IEEE 802.1q est effectuée. Il suffit alors de consulter la liste des modules pour vérifier sa présence. Il est toujours possible de charger manuellement ce module. Voici un exemple.

```
# modprobe -v 8021q
modprobe -v 8021q
insmod /lib/modules/4.13.0-1-amd64/kernel/net/llc/llc.ko
insmod /lib/modules/4.13.0-1-amd64/kernel/net/802/stp.ko
insmod /lib/modules/4.13.0-1-amd64/kernel/net/802/mrp.ko
insmod /lib/modules/4.13.0-1-amd64/kernel/net/802/garp.ko
insmod /lib/modules/4.13.0-1-amd64/kernel/net/8021q/8021q.ko

# grep 8021q /var/log/kern.log
kernel: [  439.345617] 8021q: 802.1Q VLAN Support v1.8
kernel: [  439.345723] 8021q: adding VLAN 0 to HW filter on device eth0
```

Une fois la partie kernelspace traitée, on passe logiquement à la partie userspace. La commande ip du paquet iproute2 dispose de toutes les options utiles pour créer les sous-interfaces associées aux étiquettes IEEE 802.1q.

Dans notre exemple, la syntaxe pour les deux sous-interfaces des deux périmètres définis est la suivante :

```
# ip link add link eth0 name eth0.100 type vlan id 100
# ip link add link eth0 name eth0.200 type vlan id 200
```

On visualise aussi le résultat avec la commande ip :

```
$ ip addr ls
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.2/24 brd 192.168.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::b8ad:ff:feca:fe00/64 scope link
       valid_lft forever preferred_lft forever
3: eth0.100@eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
4: eth0.200@eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
```

Les deux nouvelles sous-interfaces se configurent manuellement de façon classique.

```
# ip addr add 192.168.100.1/24 brd + dev eth0.100
# ip addr add 192.168.200.1/24 brd + dev eth0.200
```

Sur un système de la famille Debian GNU/Linux, il est possible de rendre cette configuration permanente en éditant le fichier `/etc/network/interfaces` comme suit :

```
<snipped/>
auto eth0
iface eth0 inet static
        address 192.168.2.2/24

auto eth0.100
iface eth0.100 inet static
        address 192.168.100.1/24

auto eth0.200
iface eth0.200 inet static
        address 192.168.200.1/24
```

Une fois la configuration des interfaces en place, on obtient la table de routage suivante :

```
# ip route ls
default via aaa.bbb.ccc.1 dev eth1❶
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.2❷
192.168.100.0/24 dev eth0.100  proto kernel  scope link  src 192.168.100.1❸
192.168.200.0/24 dev eth0.200  proto kernel  scope link  src 192.168.200.1❹
aaa.bbb.ccc.0/24 dev eth1  proto kernel  scope link  src aaa.bbb.ccc.7❺
```

❶ L'interface `eth1` a la possibilité d'acheminer le trafic issu des deux périmètres vers l'Internet via la passerelle par défaut.

❷ L'interface physique `eth0` sert de trunk entre le routeur et le commutateur. Sa configuration réseau correspond au périmètre Management. Le réseau auquel appartient l'interface utilise des trames sans balises IEEE 802.1q. Dans le jargon, ce VLAN est qualifié de natif.
❸ La sous-interface `eth0.100` est associée au VLAN numéro 100. Sa configuration réseau correspond au périmètre Services. Les trames de ce réseau qui circulent sur le trunk sont complétées avec une balise IEEE 802.1q qui comprend l'identificateur de VLAN 100.
❹ La sous-interface `eth0.200` est associée au VLAN numéro 200. Sa configuration réseau correspond au périmètre Accès. Les trames de ce réseau qui circulent sur le trunk sont complétées avec une balise IEEE 802.1q qui comprend l'identificateur de VLAN 200.
❺ L'interface `eth1` est directement connectée au réseau «public». Elle n'a aucune connaissance du trafic issu des différents périmètres sans configuration spécifique.

Côté commutateur, il faut que la base de données des VLANs connus contienne les mêmes identificateurs que ceux affectés sur le Routeur GNU/Linux.

Le fichier de configuration du commutateur doit contenir les informations suivantes si le protocole VTP a préalablement été configuré en mode transparent :

```
vlan 1
 name management
!
vlan 100
 name services
!
vlan 200
 name access
```

Ensuite, on affecte les ports du commutateurs aux différents VLANs ou périmètres.

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int range fastEthernet 0/1 - 12
Switch(config-if-range)#switchport access vlan 100
Switch(config-if-range)#exit
Switch(config)#int range fastEthernet 0/13 - 48
Switch(config-if-range)#switchport access vlan 200
Switch(config-if)#^Z
Switch#
07:10:45: %SYS-5-CONFIG_I: Configured from console by console
```

On visualise le résultat des affectations de ports en mode accès de la façon suivante.

```
Switch#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
100  services                         active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
200  access                           active    Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Fa0/25, Fa0/26, Fa0/27, Fa0/28
                                                Fa0/29, Fa0/30, Fa0/31, Fa0/32
                                                Fa0/33, Fa0/34, Fa0/35, Fa0/36
                                                Fa0/37, Fa0/38, Fa0/39, Fa0/40
                                                Fa0/41, Fa0/42, Fa0/43, Fa0/44
                                                Fa0/45, Fa0/46, Fa0/47, Fa0/48
```

### 5.2.3. Activation de la fonction routage

Avec la configuration actuelle, le Routeur GNU/Linux ne remplit pas sa fonction. Par exemple, les hôtes du périmètre Accès ne peuvent pas communiquer avec les serveurs du périmètre Services. Il est nécessaire d'activer la fonction routage au niveau du noyau Linux pour que les paquets IP puissent être transmis (ou routés) entre des réseaux différents.

La présentation des fonctions réseau d'une interface pilotée par le noyau Linux sort du cadre de ce document. Il faut consulter le support Configuration d'une interface de réseau local pour obtenir les informations nécessaires.

Voici un extrait du fichier `/etc/sysctl.conf` comprenant l'ensemble des réglages appliqués au noyau Linux du Routeur de la configuration type. Pour appliquer ces paramètres, il suffit d'utiliser la

commande `sysctl --system` et de valider la valeur de la «clé» `ip_forward` pour IPv4. Si cette valeur est à `1`, le routage est actif au niveau du noyau Linux.

```
$ 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
```

Les tests de communication entre les réseaux des différents périmètres peuvent être effectués depuis le commutateur.

```
Switch#ping 192.168.2.2❶

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Switch#ping 192.168.100.1❷

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1000 ms
Switch#ping 192.168.200.1❸

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.200.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#ping aaa.bbb.ccc.7❹

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to aaa.bbb.ccc.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1004 ms
Switch#ping aaa.bbb.ccc.1❺

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to aaa.bbb.ccc.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

❶ Test de communication ICMP sur le périmètre Management. Ce test n'utilise pas la fonction routage puisqu'il est effectué entre les deux extrémités du trunk.
❷ Test de communication ICMP sur le périmètre Services. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `192.168.100.0/24`.
❸ Test de communication ICMP sur le périmètre Accès. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `192.168.200.0/24`.
❹ Test de communication ICMP vers le réseau public. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `aaa.bbb.ccc.0/24`.
❺ Test de communication ICMP vers l'Internet. Ce test échoue puisque le Routeur GNU/Linux n'échange pas sa table de routage avec les autres routeurs de l'Internet.

Ces tests montrent qu'il faut compléter la configuration pour que les échanges réseau entre les périmètres et l'Internet soient possibles. Comme ces échanges réseau entre l'Internet et les périmètres ne peuvent pas se faire dans n'importe quelles conditions, il est nécessaire d'introduire la fonction de filtrage pour obtenir une interconnexion satisfaisante.

## 5.3. Interconnexion et filtrage réseau

L'étude du filtrage réseau avec le noyau Linux sort du cadre de ce document. Il faut consulter les versions françaises du Guide Pratique du Filtrage de Paquets sous Linux 2.4 et du Guide Pratique du NAT sous Linux 2.4 pour obtenir les informations nécessaires.

D'un point de vue général, on dispose de deux solutions distinctes pour interconnecter les périmètres réseau administrés avec l'Internet.

- Partager la table de routage des périmètres administrés avec d'autres routeurs via un protocole de routage dynamique tel qu'OSPF.

- Camoufler les périmètres administrés derrière une adresse IP publique accessible depuis l'Internet. Cette opération est réalisée avec les fonctions de filtrage réseau du noyau Linux : netfilter pour la partie kernelspace et iptables pour la partie userspace.

C'est la seconde proposition qui offre le plus de facilités de contrôle immédiat sur les flux réseau. L'outil de camouflage (masquerading) généralement utilisé est appelé traduction d'adresses (Native Address Translation ou NAT).

### 5.3.1. Fonctionnement minimal

Après avoir activé le routage au niveau noyau (voir Section 5.2.3, « Activation de la fonction routage »), la fontion de camouflage est simple à mettre en oeuvre :

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Cette règle réalise une traduction d'adresse source. Tout paquet IPv4 sortant par l'interface `eth1` voit son adresse IPv4 source réécrite avec l'adresse IPv4 de l'interface.

L'exécution de la règle entraîne le chargement des modules de gestion de la traduction d'adresses et du suivi dynamique de communication (stateful inspection).

```
# lsmod | grep nat | fmt -t -w80
nf_nat_masquerade_ipv4    16384  1 ipt_MASQUERADE
iptable_nat               16384  1
nf_nat_ipv4               16384  1 iptable_nat
nf_nat                    28672  2 nf_nat_masquerade_ipv4,nf_nat_ipv4
nf_conntrack             131072  7
   nf_conntrack_ipv4,ipt_MASQUERADE,nf_conntrack_netlink,nf_nat_masquerade_ipv4,xt_conntrack,nf_nat_ipv4,nf_na
libcrc32c                 16384  2 nf_conntrack,nf_nat
ip_tables                 24576  2 iptable_filter,iptable_nat
```

Le suivi d'état des échanges réseau consiste à conserver une empreinte de paquet sortant de façon à identifier les paquets retour relatifs à cette «demande». Les empreintes sont stockées dans la table `/proc/net/nf_conntrack` du système de fichiers virtuel du noyau Linux.

```
# cat /proc/net/nf_conntrack | fmt -t -w80
ipv4     2 udp❶       17 8 src=198.168.200.2❷  dst=176.9.82.67❸  sport=36322 dport=123
   src=176.9.82.67 dst=aaa.bbb.ccc.7❹ sport=123 dport=36322 mark=0 zone=0 use=2
ipv4     2 tcp      6 113 TIME_WAIT❺  src=203.0.113.1 dst=203.0.113.4 sport=38940
   dport=22❻  src=203.0.113.4 dst=203.0.113.1 sport=22 dport=38940 [ASSURED]
   mark=0 zone=0 use=2
```

❶ Protocole de transport utilisé.
❺ État de la connexion TCP.
❷ Adresse IPv4 source. Cette adresse correspond à un poste client appartenant au périmètre Accès.
❸ Adresse IPv4 destination. Il s'agit d'une adresse publique sur l'Internet.
❻ Le numéro de port destination du paquet identifie service Internet utilisé : SSH.
❹ L'adresse IPv4 destination attendue pour un paquet retour est l'adresse publique du Routeur GNU/Linux. Cette ligne montre bien que le routeur à la connaissance des réseaux internes et du réseau public. C'est à partir de ces correspondances d'adresses IPv4 que les décisions d'acheminement sont prises. Dans le cas de la traduction d'adresses par camouflage, l'adresse IPv4 retour est réécrite avec l'adresse IPv4 de l'hôte du périmètre Accès.

Si cette configuration a le mérite d'illustrer le fonctionnement du routage inter-VLAN de façon simple, elle ne correspond pas à un niveau de contrôle d'accès suffisant. L'objet de la section suivante est justement de chercher à augmenter ce niveau de contrôle.

### 5.3.2. Meilleur contrôle d'accès

Dans un premier temps, il faut garantir que tous les paquets IP non autorisés sont bloqués ; ce qui revient à appliquer la règle «tout ce qui n'est pas autorisé est interdit».

La traduction de cette règle en termes de configuration revient à jeter tous les nouveaux paquets par défaut sur les «chaînes» d'entrée et de traversée des interfaces réseau

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

En toute rigueur, il faudrait faire de même avec la chaîne de sortie OUTPUT. Cette présentation ayant pour but premier d'illustrer les concepts, ajouter les traitements de la chaîne OUTPUT ne ferait qu'alourdir les scripts sans apporter d'élément nouveau.

Dans un deuxième temps, il faut affiner la configuration du suivi de communication dynamique. La règle d'or du filtrage avec la fonction stateful inspection, c'est la description la plus fine possible du premier paquet qu'on autorise à passer.

La traduction de cette règle en termes de configuration contient 2 parties :

- Un bloc de règles qui organise le suivi de communication pour chaque chaîne sur laquelle on appliqué la politique par défaut DROP.

```
-A <CHAINE> -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

- Des règles spécifiques à chaque flux autorisé. C'est à la rédaction de ces règles qui correspondent au premier paquet autorisé qu'il faut apporter le plus grand soin. Un exemple pour les paquets IP émis depuis le périmètre Accès sur la chaîne FORWARD :

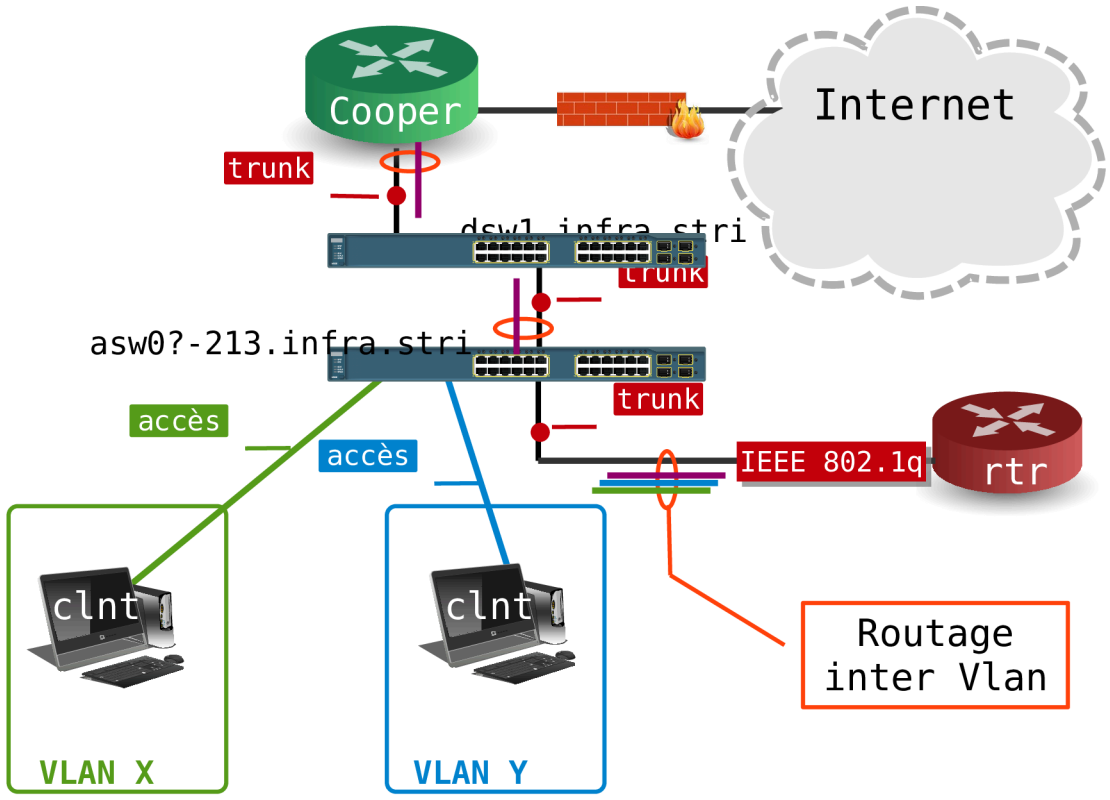
```
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -m conntrack --ctstate NEW -j ACCEPT
```

Voici une version intermédiaire de script de configuration du filtrage pour le périmètre Accès. En supposant que le fichier des règles est stocké dans le répertoire /etc/iptables/, on active les règles avec une commande du type iptables-restore </etc/iptables/rules.v4.

```
# Filtrage réseau du périmètre Accès
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Tables de traduction d'adresses
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth1 -p tcp --syn -m tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Tables de filtrage
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
#
# -> Chaîne INPUT
# . suivi de communication
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
# . toutes les communications internes sont autorisées
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i eth0.200 -m conntrack --ctstate NEW -j ACCEPT
# . administration du Routeur GNU/Linux avec SSH
-A INPUT -i eth0 -p tcp -m tcp --syn --dport 22 -m conntrack --ctstate NEW -j ACCEPT
# . services de gestion du commutateur vers le Routeur GNU/Linux
-A INPUT -i eth0 -p udp -m multiport --dports 69,123,162,514 -m conntrack --ctstate NEW -j ACCEPT
# . poubelle propre
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -p tcp -j REJECT --reject-with tcp-reset
-A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
#
# -> Chaîne FORWARD
# . suivi de communication
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
# . communications des hôtes du périmètre Accès
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -m conntrack --ctstate NEW -j ACCEPT
# . poubelle propre
-A FORWARD -m conntrack --ctstate INVALID -j DROP
-A FORWARD -p tcp -j REJECT --reject-with tcp-reset
-A FORWARD -p udp -j REJECT --reject-with icmp-port-unreachable
COMMIT
```

## 5.4. Travaux pratiques

### 5.4.1. Topologie type de travaux pratiques




Topologie type TP

Pour les besoins de ces travaux pratiques, les configurations de 4 commutateurs de chaque salle de travaux pratiques sont effacées ainsi que leurs bases de données de VLANs.

Comme indiqué dans la topologie type ci-dessus, trois postes de travaux pratiques sont associés à un commutateur. Un poste joue le rôle de routeur inter-VLAN et les deux autres sont des postes clients appartenant chacun à un VLAN ou réseau IP différent.

Le seul point de configuration imposé est le raccordement au réseau d'interconnexion avec le routeur principal de la salle de travaux pratiques. Ce raccordement utilise le port `gi0/2` de chaque commutateur qui doit être configuré en mode trunk en utilisant le VLAN natif numéro 1. Le réseau IPv4 correspondant au VLAN numéro 1 au préfixe réseau `172.16.0.0/20`

Point important, la lecture de la section «Plan d'adressage» du document Architecture réseau des travaux pratiques donne les adresses, dans le VLAN numéro 4, des deux routeurs ayant accès au réseau de Campus.

- Routeur `cooper.stri` : `172.16.16.1/20`
- Routeur `casper.stri` : `172.16.16.2/20`

### 5.4.2. Affectation des postes de travail

Les affectations données dans la table ci-dessous ne sont pas figées pour la durée des travaux pratiques. Une fois la configuration validée sur un groupe de trois postes, il est vivement conseillé de permuter les rôles de façon à mieux maîtriser les étapes de configuration.

Tableau 5.2. Affectation des rôles, des numéros de VLANs et des adresses IP dans la salle 211

| Groupe | Commutateur | Poste | Rôle | VLAN | Adresse ip du poste |
|---|---|---|---|---|---|
| 1 | `asw05-211.infra.stri` | christophsis | client | 450 | `192.168.10.2/25` |

| Groupe | Commutateur | Poste | Rôle | VLAN | Adresse ip du poste |
|---|---|---|---|---|---|
| | 172.16.0.33/20<br>Interface vlan3 | corellia | client | 451 | 192.168.10.130/25 |
| | | delaya | routeur | 4 | 172.16.18.1/20 |
| | | | | 450 | 192.168.10.1/25 |
| | | | | 451 | 192.168.10.129/25 |
| 2 | asw06-211.infra.stri<br>172.16.0.34/20<br>Interface vlan3 | kashyyyk | client | 460 | 192.168.20.2/25 |
| | | korriban | client | 461 | 192.168.20.130/25 |
| | | kessel | routeur | 4 | 172.16.18.2/20 |
| | | | | 460 | 192.168.20.1/25 |
| | | | | 461 | 192.168.20.129/25 |
| 3 | asw07-211.infra.stri<br>172.16.0.35/20<br>Interface vlan3 | mygeeto | client | 470 | 192.168.30.2/26 |
| | | nelvaan | client | 471 | 192.168.30.66/26 |
| | | rattatak | client | 472 | 192.168.30.130/26 |
| | | saleucami | routeur | 4 | 172.16.18.3/20 |
| | | | | 470 | 192.168.30.1/26 |
| | | | | 471 | 192.168.30.65/26 |
| | | | | 472 | 192.168.30.129/26 |
| 4 | asw08-211.infra.stri<br>172.16.0.36/20<br>Interface vlan3 | taris | client | 480 | 192.168.40.2/26 |
| | | teth | client | 481 | 192.168.40.66/26 |
| | | utapau | client | 482 | 192.168.40.130/26 |
| | | yavin | routeur | 4 | 172.16.18.4/20 |
| | | | | 480 | 192.168.40.1/26 |
| | | | | 481 | 192.168.40.65/26 |
| | | | | 482 | 192.168.40.129/26 |

Tableau 5.3. Affectation des rôles, des numéros de VLANs et des adresses IP dans la salle 213

| Groupe | Commutateur | Poste | Rôle | VLAN | Adresse ip du poste |
|---|---|---|---|---|---|
| 1 | asw05-213.infra.stri<br>172.16.0.15/20<br>Interface vlan3 | alderaan | client | 250 | 192.168.1.2/25 |
| | | bespin | client | 251 | 192.168.1.130/25 |
| | | centares | routeur | 4 | 172.16.17.1/20 |
| | | | | 250 | 192.168.1.1/25 |
| | | | | 251 | 192.168.1.129/25 |
| 2 | asw06-213.infra.stri<br>172.16.0.16/20<br>Interface vlan3 | coruscant | client | 260 | 192.168.2.2/25 |
| | | dagobah | client | 261 | 192.168.2.130/25 |
| | | endor | routeur | 4 | 172.16.17.2/20 |

| Groupe | Commutateur | Poste | Rôle | VLAN | Adresse ip du poste |
|---|---|---|---|---|---|
| | | | | 260 | 192.168.2.1/25 |
| | | | | 261 | 192.168.2.129/25 |
| 3 | asw07-213.infra.stri<br>172.16.0.17/20<br>Interface vlan3 | felucia | client | 270 | 192.168.3.2/25 |
| | | geonosis | client | 271 | 192.168.3.130/25 |
| | | hoth | routeur | 4 | 172.16.17.3/20 |
| | | | | 270 | 192.168.3.1/25 |
| | | | | 271 | 192.168.3.129/25 |
| 4 | asw08-213.infra.stri<br>172.16.0.18/20<br>Interface vlan3 | mustafar | client | 280 | 192.168.4.2/25 |
| | | naboo | client | 281 | 192.168.4.130/25 |
| | | tatooine | routeur | 4 | 172.16.17.4/20 |
| | | | | 280 | 192.168.4.1/25 |
| | | | | 281 | 192.168.4.129/25 |

Le positionnement des 4 commutateurs est référencé dans le support Architecture réseau des travaux pratiques.

### 5.4.3. Configuration des postes de travaux pratiques

Q105. Dans un groupe de trois postes tel qu'il a été défini ci-dessus, quel(s) poste(s) nécessite(nt) une configuration spécifique pour l'utilisation des réseaux locaux virtuels ?

Q106. Toujours dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les postes clients sont raccordés ?

Q107. Encore dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les routeurs sont raccordés ?

Q108. Dans la configuration d'un trunk, qu'est-ce qui distingue un VLAN natif ?

Q109. À partir du tableau des affectations ci-dessus, pourquoi les trois postes d'un groupe ne peuvent-ils pas appartenir au même réseau IP ?

Q110. Quel type de poste reçoit les trames complétées par des balises IEEE 802.1Q ?

Une fois le plan d'adressage IP défini, reprendre la Section 5.2, « Etude d'une configuration type » pour le groupe de postes de travaux pratiques.

Q111. Quel est le paquet qui contient les outils de configuration des interfaces réseau correspondant à chaque VLAN à router ?

Q112. Une fois les interfaces de chaque VLAN configurées sur le poste routeur, quelles sont les opérations à effectuer pour que le transfert des paquets IP d'un réseau local à l'autre soit effectif ?

Q113. Pourquoi doit-on utiliser la traduction d'adresses pour les flux réseau sortants du poste routeur vers l'Internet ? Que deviennent les paquets IP de ces flux sans traduction d'adresses ? Si la traduction d'adresses n'était pas disponible, quelle autre technique faudrait-il employer ?

Q114. Donner la séquence des tests ICMP à effectuer pour valider la connectivité entre :

- les postes clients et le poste routeur,
- les postes clients et l'ensemble des autres interfaces du routeur,
- les postes clients entre eux,
- les postes clients et l'Internet.

Q115. À l'aide de l'analyseur Wireshark, capturer des flux réseau mettant en évidence le marquage des trames avec les balises IEEE 802.1Q. Relever les numéros d'identification des VLANs vus par les interfaces du routeur. Quelle interface faut-il utiliser pour la capture de façon à visualiser l'ensemble du trafic ?

Q116. Pourquoi les flux réseau capturés contiennent-ils autant de trames STP (Spanning Tree Protocol) ?

Q117. Pourquoi la majorité des trames STP capturées sont-elles considérées comme ayant le type Ethernet II ? Quel aurait du être le type d'une trame STP si les balises IEEE 802.1Q n'étaient pas utilisées ?

## 5.5. Documents de référence

IEEE 802.1Q Standard

IEEE 802.1Q Standard

Standards d'encapsulation dans les trunks

Documentation Cisco™ : InterSwitch Link and IEEE 802.1Q Frame Format

Configuring InterVLAN Routing and ISL/802.1Q Trunking, Document ID: 14976

Documentation Cisco™ décrivant une configuration simple sur le routage inter-VLAN : Configuring InterVLAN Routing and ISL/802.1Q Trunking.

La segmentation des réseaux locaux

Segmentation des réseaux locaux : argumentation sur les fonctions de commutation et de routage.

Configuration d'une interface de réseau local

Configuration d'une interface de réseau local : présentation complète sur la configuration des interfaces réseau avec un système GNU/Linux. La section sur les Fonctions réseau d'un interface traite des réglages possibles au niveau du noyau Linux. C'est à ce niveau que l'on retrouve l'activation du routage. Voir Section 5.2.3, « Activation de la fonction routage ».

Guide Pratique du Filtrage de Paquets sous Linux 2.4

Guide Pratique du Filtrage de Paquets : présentation des concepts du filtrage réseau avec le noyau Linux.

Guide Pratique du NAT sous Linux 2.4

Guide Pratique du NAT : présentation des concepts de la fonction de traduction d'adresses IP avec le noyau Linux.

Architecture réseau des travaux pratiques

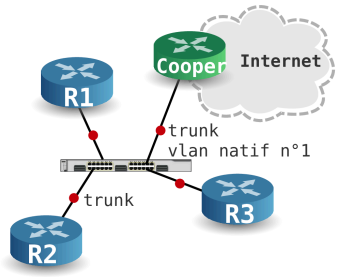
Le support Architecture réseau des travaux pratiques présente la topologie physique de la salle de travaux pratiques avec la Disposition des équipements dans l'armoire de brassage ainsi que les configurations par défaut des équipements. On y trouve aussi le plan d'adressage IP utilisé avec les autres supports de travaux pratiques, le plan de numérotations des VLANs et les affectations des groupes de ports des commutateurs.

CHAPITRE 6

# Introduction au routage dynamique OSPF avec Bird


**Résumé**

**L'objectif de ce support de travaux pratiques est d'étudier le protocole de routage dynamique OSPF. Cette illustration s'appuie sur une topologie minimale très classique : le triangle. L'originalité consiste à utiliser les VLANs pour distinguer la topologie physique (l'étoile) de la topologie logique (le triangle). Cette version du support utilise le logiciel Bird.**


# Table des matières



## 6.1. Préparer les systèmes pour le routage IPv4 et IPv6

La première étape consiste à installer les outils sur les trois routeurs, à appliquer une configuration commune et à mettre en place la topologie physique.

1. Installer le paquet bird avant de brasser les postes sur les commutateurs attribués avec le plan d'adressage de la salle de travaux pratiques.

```
$ aptitude search ~ibird
i   bird              - démon de routage internet
```

Sans configuration particulière, les services bird et bird6 sont lancés.

```
R1:~# systemctl status bird
# bird.service - BIRD Internet Routing Daemon (IPv4)
   Loaded: loaded (/lib/systemd/system/bird.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2018-10-20 12:58:52 UTC; 2min 23s ago
 Main PID: 751 (bird)
   Memory: 632.0K
   CGroup: /system.slice/bird.service
           ##751 /usr/sbin/bird -f -u bird -g bird

oct. 20 12:58:52 R1 systemd[1]: Starting BIRD Internet Routing Daemon (IPv4)...
oct. 20 12:58:52 R1 systemd[1]: Started BIRD Internet Routing Daemon (IPv4).
oct. 20 12:58:52 R1 bird[751]: Started
```

```
R1:~# systemctl status bird6
# bird6.service - BIRD Internet Routing Daemon (IPv6)
   Loaded: loaded (/lib/systemd/system/bird6.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2018-10-20 17:18:50 UTC; 3h 9min ago
  Process: 1728 ExecStartPre=/usr/sbin/bird6 -p (code=exited, status=0/SUCCESS)
  Process: 1722 ExecStartPre=/usr/lib/bird/prepare-environment (code=exited, status=0/SUCCESS)
 Main PID: 1729 (bird6)
   Memory: 788.0K
   CGroup: /system.slice/bird6.service
           ##1729 /usr/sbin/bird6 -f -u bird -g bird

oct. 20 17:18:50 R1 systemd[1]: Starting BIRD Internet Routing Daemon (IPv6)...
oct. 20 17:18:50 R1 systemd[1]: Started BIRD Internet Routing Daemon (IPv6).
oct. 20 17:18:50 R1 bird6[1729]: Started
```

2. Activer le routage IPv4 et IPv6 au niveau noyau.

Il faut éditer le fichier `/etc/sysctl.conf` pour fixer les valeurs des paramètres de configuration du routage. Voir la section Fonctions réseau d'une interface du support Configuration d'une interface de réseau local.

```
#sysctl -p
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.secure_redirects = 1
net.ipv4.conf.all.log_martians = 1
```

3. Créer les sous-interfaces associées aux VLANs sur chacun des routeurs R1, R2 et R3 à l'aide du script suivant :

```
#!/bin/bash

for vlan in $*
do
 ip link add link eth0 name eth0.$vlan type vlan id $vlan
 ip link set dev eth0.$vlan up
done
```

Sur le routeur R1, on utilise le script avec les numéros de VLANs 12 et 13 par exemple.

```
R1:~# sh ./subinterfaces.sh 12 13
```

On adapte l'utilisation du même script aux routeurs R2 et R3 avec les numéros de VLANs concernés.

Il est aussi possible d'éditer le fichier `/etc/network/interfaces` de façon à rendre cette configuration permanente. Voici une copie de ce fichier pour le routeur R1.

```
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 192.0.2.7/27
        gateway 192.0.2.1
        dns-nameservers 192.0.2.1 9.9.9.9

iface eth0 inet6 static
        address 2001:678:3fc:a::7/64
        gateway fe80::dc02:44ff:fe64:4834
        dns-nameservers 2001:678:3fc:a::1 2620:fe::fe

auto eth0.12
iface eth0.12 inet static
        address 10.1.12.1/26

iface eth0.12 inet6 static
        address 2001:678:3fc:c::1/64

auto eth0.13
iface eth0.13 inet static
        address 10.1.13.1/26

iface eth0.13 inet6 static
        address 2001:678:3fc:d::1/64
```

## 6.2. Valider les communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider le raccordement des routeurs aux commutateurs désignés, les communications entre chaque routeur et la visualisation des tables de routage pour les interfaces réseau configurées.

Q118. Quelles sont les opérations à effectuer pour implanter les adresses IPv4 et IPv6 des interfaces correspondant à chacun des VLANs routés ?

Au niveau liaison, les sous-interfaces ont déjà été configurées avec le script subinterfaces.sh. Il reste à paramétrer les adresses de ces sous-interfaces.

Routeur R1

```
R1:~# ip addr add 10.1.12.1/26 brd + dev eth0.12
R1:~# ip -6 addr add 2001:678:3fc:c::1/64 dev eth0.12
R1:~# ip addr add 10.1.13.1/26 brd + dev eth0.13
R1:~# ip -6 addr add 2001:678:3fc:d::1/64 dev eth0.13
```

Routeur R2

```
R2:~# ip addr add 10.1.12.2/26 brd + dev eth0.12
R2:~# ip -6 addr add 2001:678:3fc:c::2/64 dev eth0.12
R2:~# ip addr add 10.1.23.2/26 brd + dev eth0.23
R2:~# ip -6 addr add 2001:678:3fc:17::2/64 dev eth0.23
```

Routeur R3

```
R3:~# ip addr add 10.1.13.3/26 brd + dev eth0.13
R3:~# ip -6 addr add 2001:678:3fc:d::3/64 dev eth0.13
R3:~# ip addr add 10.1.23.3/26 brd + dev eth0.23
R3:~# ip -6 addr add 2001:678:3fc:17::3/64 dev eth0.23
```

Q119. Quelles sont les opérations à effectuer pour valider les communications IP entre routeurs ?

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

Exemple entre R1 et R2

```
R1:~# ping -qc2 10.1.12.2
PING 10.1.12.2 (10.1.12.2) 56(84) bytes of data.

--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 0.068/0.309/0.551/0.242 ms
```

```
R1:~# ping -qc2 2001:678:3fc:c::2
PING 2001:678:3fc:c::2(2001:678:3fc:c::2) 56 data bytes

--- 2001:678:3fc:c::2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 19ms
rtt min/avg/max/mdev = 0.070/0.295/0.521/0.226 ms
```

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

Q120. Comment visualiser la table de routage au niveau système ?

Utiliser la commande ip pour visualiser la table de routage

Toutes les routes affichées correspondent à des réseaux IPv4 et IPv6 sur lesquels le routeur est directement connecté via une interface active correctement configurée.

Routeur R1 - niveau système

```
R1:~# ip route ls
default via 192.0.2.1 dev eth0 onlink
10.1.12.0/26 dev eth0.12 proto kernel scope link src 10.1.12.1
10.1.13.0/26 dev eth0.13 proto kernel scope link src 10.1.13.1
192.0.2.0/27 dev eth0 proto kernel scope link src 192.0.2.7
```

```
R1:~# ip -6 route ls
2001:678:3fc:a::/64 dev eth0 proto kernel metric 256 pref medium
2001:678:3fc:c::/64 dev eth0.12 proto kernel metric 256 pref medium
2001:678:3fc:d::/64 dev eth0.13 proto kernel metric 256 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth0.12 proto kernel metric 256 pref medium
fe80::/64 dev eth0.13 proto kernel metric 256 pref medium
default via fe80::dc02:44ff:fe64:4834 dev eth0 metric 1024 onlink pref medium
```

Routeur R2 - niveau système

```
R2:~# ip route ls
10.1.12.0/26 dev eth0.12 proto kernel scope link src 10.1.12.2
10.1.23.0/26 dev eth0.23 proto kernel scope link src 10.1.23.2
```

```
R2:~# ip -6 route ls
2001:678:3fc:c::/64 dev eth0.12 proto kernel metric 256 pref medium
2001:678:3fc:17::/64 dev eth0.23 proto kernel metric 256 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth0.12 proto kernel metric 256 pref medium
fe80::/64 dev eth0.23 proto kernel metric 256 pref medium
```

Routeur R3 - niveau système

```
R3:~# ip route ls
10.1.13.0/26 dev eth0.13 proto kernel scope link src 10.1.13.3
10.1.23.0/26 dev eth0.23 proto kernel scope link src 10.1.23.3
```

```
R3:~# ip -6 route ls
2001:678:3fc:d::/64 dev eth0.13 proto kernel metric 256 pref medium
2001:678:3fc:17::/64 dev eth0.23 proto kernel metric 256 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth0.13 proto kernel metric 256 pref medium
fe80::/64 dev eth0.23 proto kernel metric 256 pref medium
```

Q121. Comment activer la fonction routage du noyau Linux ?

Reprendre l'instruction présentée dans le document Configuration d'une interface de réseau local : activation du routage.

L'opération doit être répétée sur chacun des trois routeurs pour que le protocole de routage dynamique puisse fonctionner normalement.

Si cette fonction n'est pas active dans le noyau Linux, aucune décision d'acheminement d'un paquet d'une interface vers l'autre ne sera prise. Les paquets à router sont simplement jetés.

Les instructions d'activation de la fonction de routage sont données dans la section Préparation des routeurs.

## 6.3. Configurer les démons OSPF Bird

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

Q122. Quels sont les fichiers de configuration à éditer pour activer les protocoles OSPFv2 et OSPFv3 sur le routeur ?

Une fois le paquet `bird` installé, deux démons distincts sont lancés : `bird` pour IPv4 et `bird6` pour IPv6. Rechercher dans la liste des fichiers fournis avec le paquet, les exemples de fichiers de configuration.

Les fichiers de configuration sont au nombre de deux. Ils sont placés dans le dossier `/etc/bird/`. Deux exemples de ces fichiers sont données dans le dossier de documentation du paquet.

```
R1:~# dpkg -L bird | grep example
/usr/share/doc/bird/examples
/usr/share/doc/bird/examples/bird.conf.gz
/usr/share/doc/bird/examples/bird6.conf.gz
```

Q123. Comment accéder à l'état des différents protocoles actifs pour chaque démon ?

À chaque édition d'un fichier de configuration, il faut relancer le démon correspondant. C'est à nouveau dans la liste des fichiers du paquet que l'on identifie les outils d'accès à la configuration active des deux démons.

Il faut consulter la section Remote control de la documentation Bird. Les commandes utiles pour cette question sont les suivantes.

show status
show protocols

Chaque démon dispose d'une console propre avec les outils birdc et birdc6. Ce sont ces deux consoles qui permettent de connaître le statut du démon, la liste des protocoles actifs et les informations relatives au fonctionnement de ces protocoles.

```
R1:~# birdc
BIRD 1.6.4 ready.
bird> sh status
BIRD 1.6.4
Router ID is 0.0.4.1
Current server time is 2018-10-20 20:59:04
Last reboot on 2018-10-20 16:54:04
Last reconfiguration on 2018-10-20 20:58:58
Daemon is up and running
```

```
R1:~# birdc6
BIRD 1.6.4 ready.
bird> sh status
BIRD 1.6.4
Router ID is 0.0.6.1
Current server time is 2018-10-20 21:00:22
Last reboot on 2018-10-20 17:18:49
Last reconfiguration on 2018-10-20 17:18:49
Daemon is up and running
```

De la même façon, on peut connaître la liste des protocoles actifs de chaque démon.

```
bird> sh protocols
name     proto    table    state  since       info
device1  Device   master   up     16:54:04
kernel1  Kernel   master   up     20:58:58
```

Q124. Comment activer le protocole de routage OSPF et attribuer l'identifiant du routeur ?

Consulter le document BIRD User's Guide à la section OSPF pour activer le protocole. Consulter les tableaux des plans d'adressage pour obtenir la valeur de l'identifiant du routeur à configurer.

On édite les fichiers `/etc/bird.conf` et `/etc/bird6.conf` avec les paramètres suivants.

router id IPv4 address
protocol ospf <name>
area <id>

Voici une copie des fichiers du routeur `R1`.

```
R1:~# grep -v ^# /etc/bird/bird.conf

router id 0.0.1.4;

protocol kernel {
        scan time 10;
        import none;
}

protocol device {
        scan time 10;
}

protocol ospf OSPFv2R1 {
        area 0 {
        };
}
```

```
R1:~# grep -v ^# /etc/bird/bird6.conf

router id 0.0.1.6;

protocol kernel {
        scan time 10;
        import none;
}

protocol device {
        scan time 10;
}

protocol ospf OSPFv3R1 {
```

```
        area 0 {
        };
}
```

Une fois les deux services relancés, on peut vérifier que les éléments demandés sont bien présents dans la configuration des démons de routage OSPF.

```
R1:~# systemctl restart bird

R1:~# birdc sh protocols
BIRD 1.6.4 ready.
name     proto    table    state  since       info
kernel1  Kernel   master   up     09:38:34
device1  Device   master   up     09:38:34
OSPFv2R1 OSPF     master   up     09:38:34    Alone

R1:~# birdc sh ospf state
BIRD 1.6.4 ready.

area 0.0.0.0

        router 0.0.1.4
                distance 0
```

```
R1:~# systemctl restart bird6

R1:~# birdc6 sh protocols
BIRD 1.6.4 ready.
name     proto    table    state  since       info
kernel1  Kernel   master   up     09:42:33
device1  Device   master   up     09:42:33
OSPFv3R1 OSPF     master   up     09:42:33    Alone

R1:~# birdc6 sh ospf state
BIRD 1.6.4 ready.

area 0.0.0.0

        router 0.0.1.6
                distance 0
```

Q125. Comment activer et valider le protocole de routage OSPF pour les réseaux IPv4 et IPv6 connus de chaque routeur ?

Consulter la section OSPF de la documentation BIRD User's Guide ainsi que l'exemple OSPF example. Il suffit d'adapter les exemples avec les noms d'interfaces en fonction du contexte.

On édite les fichiers `/etc/bird.conf` et `/etc/bird6.conf` avec les paramètres suivants.

interface <interface pattern>
authentication none|simple|cryptographic;

On vérifie au niveau console (Voir Remote control) l'état de la base de connaissance des deux processus OSPF avec la commande suivante.

show ospf state

Routeur R1 : OSPFv2 & interfaces

```
R1:~# grep -v ^# /etc/bird/bird.conf

router id 0.0.1.4;

protocol kernel {
        scan time 10;
        import none;
}

protocol device {
        scan time 10;
}

protocol ospf OSPFv2R1 {
        rfc1583compat yes;
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
```

```
                };
        };
}
```

Routeur R1 : base de connaissance OSPFv2

Dans la copie d'écran ci-dessous, on relève les deux routeurs voisins de `R1` ainsi que le réseau distant `10.1.23.0/26`.

```
R1:~# birdc sh ospf state
BIRD 1.6.4 ready.

area 0.0.0.0

        router 0.0.1.4
                distance 0
                network 10.1.12.0/26 metric 10
                network 10.1.13.0/26 metric 10

        router 0.0.2.4
                distance 10
                network 10.1.12.0/26 metric 10
                network 10.1.23.0/26 metric 10

        router 0.0.3.4
                distance 10
                network 10.1.13.0/26 metric 10
                network 10.1.23.0/26 metric 10

        network 10.1.12.0/26
                dr 0.0.2.4
                distance 10
                router 0.0.2.4
                router 0.0.1.4

        network 10.1.13.0/26
                dr 0.0.3.4
                distance 10
                router 0.0.3.4
                router 0.0.1.4

        network 10.1.23.0/26
                dr 0.0.2.4
                distance 20
                router 0.0.2.4
                router 0.0.3.4
```

Routeur R1 : OSPFv3 & interfaces

```
R1:~# grep -v ^# /etc/bird/bird6.conf

router id 0.0.1.6;

protocol kernel {
        scan time 10;
        import none;
}

protocol device {
        scan time 10;
}

protocol ospf OSPFv3R1 {
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
                };
        };
}
```

Routeur R1 : base de connaissance OSPFv3

Dans la copie d'écran ci-dessous, on relève les deux routeurs voisins de `R1` ainsi que le réseau distant `2001:678:3fc:17::/64`.

```
R1:~# birdc6 sh ospf state
```

```
BIRD 1.6.4 ready.

area 0.0.0.0

        router 0.0.1.6
                distance 0
                network [0.0.1.6-2] metric 10
                network [0.0.1.6-3] metric 10

        router 0.0.2.6
                distance 10
                network [0.0.1.6-2] metric 10
                network [0.0.3.6-3] metric 10

        router 0.0.3.6
                distance 10
                network [0.0.1.6-3] metric 10
                network [0.0.3.6-3] metric 10

        network [0.0.1.6-2]
                distance 10
                router 0.0.1.6
                router 0.0.2.6
                address 2001:678:3fc:c::/64

        network [0.0.1.6-3]
                distance 10
                router 0.0.1.6
                router 0.0.3.6
                address 2001:678:3fc:d::/64

        network [0.0.3.6-3]
                distance 20
                router 0.0.3.6
                router 0.0.2.6
                address 2001:678:3fc:17::/64
```

Q126. Comment identifier le type de réseau des interfaces actives d'un routeur pour chaque version du protocole de routage OSPF ?

La question précédente montre que la configuration des deux processus `bird` et `bird6` est basée sur l'activation du protocole par interface. Il faut donc rechercher dans la section Remote control l'instruction qui donne l'état des interfaces actives.

show ospf interface

Comme on utilise uniquement des liens Ethernet dans ce contexte de travaux pratiques, le type de réseau est nécessairement diffusion.

Routeur R1 : OSPFv2 & interfaces

```
R1:~# birdc sh ospf interface
BIRD 1.6.4 ready.
OSPFv2R1:
Interface eth0.12 (10.1.12.0/26)
        Type: broadcast
        Area: 0.0.0.0 (0)
        State: Backup
        Priority: 1
        Cost: 10
        Hello timer: 10
        Wait timer: 40
        Dead timer: 40
        Retransmit timer: 5
        Designated router (ID): 0.0.2.4
        Designated router (IP): 10.1.12.2
        Backup designated router (ID): 0.0.1.4
        Backup designated router (IP): 10.1.12.1
Interface eth0.13 (10.1.13.0/26)
        Type: broadcast
        Area: 0.0.0.0 (0)
        State: Backup
        Priority: 1
        Cost: 10
        Hello timer: 10
        Wait timer: 40
```

```
        Dead timer: 40
        Retransmit timer: 5
        Designated router (ID): 0.0.3.4
        Designated router (IP): 10.1.13.3
        Backup designated router (ID): 0.0.1.4
        Backup designated router (IP): 10.1.13.1
```

Routeur R1 : OSPFv3 & interfaces

```
R1:~# birdc6 sh ospf interface
BIRD 1.6.4 ready.
OSPFv3R1:
Interface eth0.12 (IID 0)
        Type: broadcast
        Area: 0.0.0.0 (0)
        State: DR
        Priority: 1
        Cost: 10
        Hello timer: 10
        Wait timer: 40
        Dead timer: 40
        Retransmit timer: 5
        Designated router (ID): 0.0.1.6
        Designated router (IP): fe80::70de:4fff:fe1d:68b4
        Backup designated router (ID): 0.0.2.6
        Backup designated router (IP): fe80::943a:41ff:fe65:7307
Interface eth0.13 (IID 0)
        Type: broadcast
        Area: 0.0.0.0 (0)
        State: DR
        Priority: 1
        Cost: 10
        Hello timer: 10
        Wait timer: 40
        Dead timer: 40
        Retransmit timer: 5
        Designated router (ID): 0.0.1.6
        Designated router (IP): fe80::70de:4fff:fe1d:68b4
        Backup designated router (ID): 0.0.3.6
        Backup designated router (IP): fe80::3032:e9ff:fe73:6322
```

Q127. 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 routeur avec un démon OSPF envoie aussi des paquets HELLO dans le même VLAN, les deux routeurs cherchent à former une adjacence.

La commande utile de la section Remote control est la suivante.

show ospf neighbors

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

```
R1:~# birdc sh ospf neighbors
BIRD 1.6.4 ready.
OSPFv2R1:
Router ID       Pri          State      DTime   Interface  Router IP
0.0.2.4   1     Full/DR         00:31   eth0.12    10.1.12.2
0.0.3.4   1     Full/DR         00:34   eth0.13    10.1.13.3
```

```
R1:~# birdc6 sh ospf neighbors
BIRD 1.6.4 ready.
OSPFv3R1:
Router ID       Pri          State      DTime   Interface  Router IP
0.0.2.6   1     Full/BDR        00:33   eth0.12    fe80::943a:41ff:fe65:7307
0.0.3.6   1     Full/BDR        00:36   eth0.13    fe80::3032:e9ff:fe73:6322
```

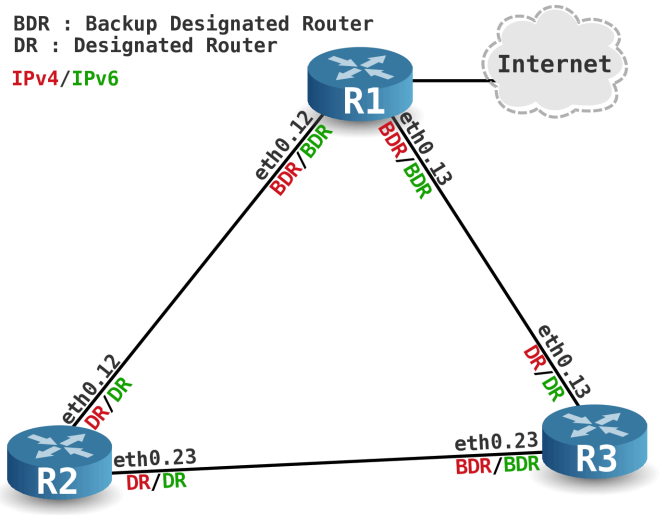
Q128. Comment identifier le rôle des différentes interfaces des routeurs pour chacun des liens du triangle de la topologie logique ?

Avertissement

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

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

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




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

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

Q129. Quelles sont les réseaux IPv4 et IPv6 présents dans la base calcul du protocole OSPF ?

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

La commande utile dans les deux consoles est la suivante.

show route

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

Voici la vue depuis le routeur `R1`.

```
R1:~# birdc sh route
BIRD 1.6.4 ready.
10.1.12.0/26       dev eth0.12 [OSPFv2R1 16:47:15] * I (150/10) [0.0.2.4]
10.1.13.0/26       dev eth0.13 [OSPFv2R1 16:47:10] * I (150/10) [0.0.3.4]
10.1.23.0/26       via 10.1.13.3 on eth0.13 [OSPFv2R1 16:47:10] * I (150/20) [0.0.2.4]
```

Les valeurs notées entre parenthèses correspondent à la métrique du lien pour joindre le réseau noté à gauche. Pour le protocole OSPF, le calcul de métrique se fait à partir du coût de lien par défaut pour chaque interface active. La valeur par défaut est `10`

Les deux premiers réseaux de la table sont joignable via un lien direct ; soit une métrique de `10`. Le troisième réseau est joignable via deux liens Ethernet ; d'où la métrique de `20`.

Pour les réseaux IPv6, on retrouve les mêmes métriques puisque la topologie est identique pour les deux version du protocole IP.

```
R1:~# birdc6 sh route
BIRD 1.6.4 ready.
2001:678:3fc:d::/64 dev eth0.13 [OSPFv3R1 10:01:30] * I (150/10) [0.0.1.6]
2001:678:3fc:c::/64 dev eth0.12 [OSPFv3R1 10:05:36] * I (150/10) [0.0.1.6]
2001:678:3fc:17::/64 via fe80::3032:e9ff:fe73:6322 on eth0.13 [OSPFv3R1 10:05:38] * I (150/20) [0.0.3.6]
```

Avec OSPFv3, les relations de voisinage entre routeurs utilisent nécessairement les adresses de lien local appartenant au préfixe `fe80::/10`.

Q130. Comment utiliser toutes les solutions disponibles pour joindre le réseau distant depuis chacun des sommets de la topologie triangle ?

Avec la topologie logique triangle, le réseau du côté opposé à un sommet (au routeur) doit être joignable depuis les deux réseaux locaux raccordés à ce routeur. Nous sommes donc dans un contexte multi chemins.

Consulter la section OSPF de la documentation BIRD User's Guide et rechercher l'intsruction qui permet l'utilisation de plusieurs chemins à coût égal.

ecmp switch [limit number]

La fonction réseau du noyau Linux recherchée est connue sous le nom Equal Cost Multi Path ou ECMP.

Sur le routeur `R1`, la configuration du bloc d'instructions OSPF de chaque processus est la suivante.

```
R1:~# sed -n '/protocol ospf/,//p' /etc/bird/bird.conf
protocol ospf OSPFv2R1 {
        rfc1583compat yes;
        ecmp yes;
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
                };
        };
}
```

```
R1:~# sed -n '/protocol ospf/,//p' /etc/bird/bird6.conf
protocol ospf OSPFv3R1 {
        ecmp yes;
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
                };
        };
}
```

Relativement aux questions précédentes, les tables de routage proposées par les processus `bird` et `bird6` font apparaître les deux chemins disponibles pour joindre le réseau distant du sommet de la topologie triangle.

```
R1:~# birdc sh route
BIRD 1.6.4 ready.
10.1.12.0/26       dev eth0.12 [OSPFv2R1 19:49:42] * I (150/10) [0.0.1.4]
10.1.13.0/26       dev eth0.13 [OSPFv2R1 19:50:37] * I (150/10) [0.0.1.4]
10.1.23.0/26       multipath [OSPFv2R1 19:50:48] * I (150/20) [0.0.2.4]
        via 10.1.12.2 on eth0.12 weight 1
        via 10.1.13.3 on eth0.13 weight 1
```

```
R1:~# birdc6 sh route
BIRD 1.6.4 ready.
2001:678:3fc:d::/64 dev eth0.13 [OSPFv3R1 19:50:38] * I (150/10) [0.0.1.6]
2001:678:3fc:c::/64 dev eth0.12 [OSPFv3R1 19:49:44] * I (150/10) [0.0.1.6]
2001:678:3fc:17::/64 multipath [OSPFv3R1 19:50:50] * I (150/20) [0.0.2.6]
        via fe80::3032:e9ff:fe73:6322 on eth0.13 weight 1
        via fe80::943a:41ff:fe65:7307 on eth0.12 weight 1
```

## 6.4. Échanger les routes entre Bird et le système

Dans la section précédente, tous les échanges de préfixes réseau IPv4 et IPv6 se font entre les démons Bird installés sur les trois routeurs de la topologie étudiée. Il faut maintenant être capable d'échanger les résultats des traitements OSPFv2 et OSPFv3 avec le sous-système réseau du noyau de chaque routeur.

- Sur R1, les deux démons Bird doivent importer la route par défaut déjà connue au niveau système. De plus, les routes vers les réseaux fictifs de R2 et R3 apprise via OSPF doivent être exportées vers le sous-système réseau du noyau de R1.

- Sur les routeurs R2 et R3, les routes par défaut apprises via OSPF doivent être exportées vers le sous-système réseau du noyau. De plus les routes des réseaux fictifs doivent être importées dans les démons Bird pour être publiées via OSPF.

> **Avertissement**
>
> Les résultats des questions de cette section ne sont visibles que si les routes sont déjà présentes, soit dans les démons de routage Bird, soit au niveau système.

Q131. Comment faire pour que les routes connues du sous-système réseau du noyau Linux soient importées dans les deux démons `bird` et `bird6` ?

Consulter la section kernel de la documentation BIRD User's Guide à la recherche des paramètres d'importation. Les commandes utiles pour cette question sont les suivantes.

import
learn switch

Pour cette question, les configurations des démons `bird` et `bird6` sont identiques quel que soit le routeur considéré. Voici un extrait de fichier de configuration.

```
R1:~# sed -n '/protocol kernel/,/^}/p' /etc/bird/bird.conf
protocol kernel {
        scan time 10;
        import all;
        learn yes;
#       export all;   # Actually insert routes into the kernel routing table
}
```

Tant que les questions sur l'ajout de réseaux fictifs ne sont pas traitées, seul les démons du routeur R1 ont un résultat observable. Les routes par défaut sont importées dans les démons Bird.

```
R1:~# birdc show route 0.0.0.0/0
BIRD 1.6.4 ready.
0.0.0.0/0          via 192.0.2.1 on eth0 [kernel1 13:47:04] * (10)
```

```
R1:~# birdc6 show route ::/0
BIRD 1.6.4 ready.
::/0               via fe80::dc02:44ff:fe64:4834 on eth0 [kernel1 2018-10-21] * (10)
```

Q132. Comment faire pour que les routes calculées par les processus `bird` et `bird6` soient soumises au sous-système réseau du noyau Linux ?

Consulter la section kernel de la documentation BIRD User's Guide à la recherche des paramètres d'exportation. La commande utile pour cette question est la suivante.

export

En reprenant l'exemple du routeur R1, la configuration des blocs d'instructions kernel de chaque processus devient :

```
R1:~# sed -n '/protocol kernel/,/^}/p' /etc/bird/bird.conf
protocol kernel {
        scan time 10;
        import all;
        learn yes;
        export all;   # Actually insert routes into the kernel routing table
}
```

Avec la commande ip au niveau système, on voit apparaître les «sources» d'alimentation de la table de routage du système en question : kernel et bird.

```
R1:~# ip route ls
default via 192.0.2.1 dev eth0 onlink
10.1.12.0/26 dev eth0.12 proto kernel scope link src 10.1.12.1
10.1.13.0/26 dev eth0.13 proto kernel scope link src 10.1.13.1
10.1.23.0/26 proto bird
        nexthop via 10.1.12.2 dev eth0.12 weight 1
        nexthop via 10.1.13.3 dev eth0.13 weight 1
192.0.2.0/27 dev eth0 proto kernel scope link src 192.0.2.7
```

```
R1:~# ip -6 route ls
2001:678:3fc:a::/64 dev eth0 proto kernel metric 256 pref medium
2001:678:3fc:c::/64 dev eth0.12 proto kernel metric 256 pref medium
2001:678:3fc:d::/64 dev eth0.13 proto kernel metric 256 pref medium
2001:678:3fc:17::/64 proto bird metric 1024
        nexthop via fe80::3032:e9ff:fe73:6322 dev eth0.13 weight 1
        nexthop via fe80::943a:41ff:fe65:7307 dev eth0.12 weight 1 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth0.12 proto kernel metric 256 pref medium
fe80::/64 dev eth0.13 proto kernel metric 256 pref medium
default via fe80::dc02:44ff:fe64:4834 dev eth0 metric 1024 onlink pref medium
```

## 6.5. Publier les routes par défaut via OSPF

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

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

Avant publication de la route par défaut depuis le routeur R1, les démons OSPF n'utilisent que des annonces LSA (Link State Advertisement) de type 1 et 2. Voici un tableau de référence pour le codage des LSAs.

Tableau 6.1. Codage des annonces OSPF

| OSPFv2 | Description | OSPFv3 | Description |
|---|---|---|---|
| 1 | Router LSA | 0x2001 | Router LSA |
| 2 | Network LSA | 0x2002 | Network LSA |
| 3 | Network Summary LSA | 0x2003 | Inter-Area Prefix LSA for ABRs |
| 4 | ASBR Summary LSA | 0x2004 | Inter-Area Router LSA for ABRs |
| 5 | AS-external LSA | 0x4005 | AS-external LSA |
| 6 | Group Membership LSA | 0x2006 | Group Membership LSA |
| 7 | Not So Stubby Area LSA | 0x2007 | Type-7 LSA |
| 8 | | 0x2008 | Link LSA |
| 9 | | 0x2009 | Intra-Area Prefix LSA |

- ABR : Area Border Router

- ASBR : Autonomous System Border Router

Les listes des annonces connues du routeur `R1` avant publication des routes par défaut sont données dans les copies d'écran ci-dessous.

```
R1:~# birdc show ospf lsadb
BIRD 1.6.4 ready.

Area 0.0.0.0

 Type   LS ID           Router          Sequence   Age  Checksum
 0001  0.0.1.4         0.0.1.4         80000148  1465    88f0
 0001  0.0.2.4         0.0.2.4         80000146  1466    70f2
 0002  10.1.12.2       0.0.2.4         80000003  1466    3439
 0001  0.0.3.4         0.0.3.4         80000147  1469    a4b6
 0002  10.1.13.3       0.0.3.4         80000003  1470    2148
 0002  10.1.23.3       0.0.3.4         80000003   375    bf9e
```

```
R1:~# birdc6 show ospf lsadb
BIRD 1.6.4 ready.

Area 0.0.0.0

 Type   LS ID           Router          Sequence   Age  Checksum
 2001  0.0.0.0         0.0.1.6         8000015c  1643    116b
 2009  0.0.0.0         0.0.1.6         8000015c  1643    7e2b
 2001  0.0.0.0         0.0.2.6         80000151  1647    4243
 2009  0.0.0.0         0.0.2.6         80000151  1647    9a18
 2002  0.0.0.2         0.0.2.6         80000001  1647    9976
 2009  0.0.0.2         0.0.2.6         80000001  1647    a95c
 2001  0.0.0.0         0.0.3.6         80000153  1646    4c35
 2009  0.0.0.0         0.0.3.6         80000153  1646    9c12
 2002  0.0.0.2         0.0.3.6         80000001  1646    9b72
 2009  0.0.0.2         0.0.3.6         80000001  1646    c939
 2002  0.0.0.3         0.0.3.6         80000003   575    9a6f
 2009  0.0.0.3         0.0.3.6         80000003   575    ca2a

Link eth0.12

 Type   LS ID           Router          Sequence   Age  Checksum
 0008  0.0.0.2         0.0.1.6         80000137  1648    c5df
 0008  0.0.0.2         0.0.2.6         8000012e   575    5640

Link eth0.13

 Type   LS ID           Router          Sequence   Age  Checksum
 0008  0.0.0.3         0.0.1.6         80000137  1646    e1c1
 0008  0.0.0.2         0.0.3.6         8000012e   610    62dc
```

Q133. Quelle est la condition préalable à respecter pour que le routeur `R1` soit en mesure de publier une route par défaut via OSPF ?

Avant de procéder à l'importation de route dans les démons Bird, on doit s'assurer de la présence des deux routes par défaut IPv4 et IPv6 dans les tables de routage au niveau système.

Sur le routeur `R1` uniquement, on valide la présence des routes par défaut.

```
R1:~# ip route ls default
default via 192.0.2.1 dev eth0 onlink

R1:~# ip -6 route ls default
default via fe80::dc02:44ff:fe64:4834 dev eth0 metric 1024 onlink pref medium
```

Q134. Comment valider l'importation des routes par défaut dans les deux démons `bird` et `bird6` ?

L'importation des routes depuis le niveau système dans les démons Bird a été traitée à la Section 6.4, « Échanger les routes entre Bird et le système ». Ici, on se contente de vérifier la présence des routes par défaut au niveau des consoles de chaque démon.

On peut spécifier le préfixe réseau directement dans l'affichage de la table de routage de chaque démon.

```
R1:~# birdc show route 0.0.0.0/0
BIRD 1.6.4 ready.
0.0.0.0/0          via 192.0.2.1 on eth0 [kernel1 2018-10-27] * (10)

R1:~# birdc6 show route ::/0
BIRD 1.6.4 ready.
::/0               via fe80::dc02:44ff:fe64:4834 on eth0 [kernel1 2018-10-27] * (10)
```

Q135. Comment créer les filtres qui serviront à exporter les routes par défaut dans la configuration de chaque démon pour les protocoles OSPFv2 et OSPFv3 ?

Il faut consulter la documentation BIRD User's Guide aux sections Filters et OSPF pour trouver des exemples de syntaxe.

Voici une copie d'écran pour chaque démon.

```
R1:~# sed -n '/^filter export_OSPF/,/^}/p' /etc/bird/bird.conf
filter export_OSPF {
        if ( net = 0.0.0.0/0 ) then {
                ospf_metric2 = 100;
                accept;
        }
        reject;
}
```

```
R1:~# sed -n '/^filter export_OSPF/,/^}/p' /etc/bird/bird6.conf
filter export_OSPF {
        if ( net = ::/0 ) then {
                ospf_metric2 = 100;
                accept;
        }
        reject;
}
```

Q136. Comment appliquer les filtres de la question précédente pour que les routes par défaut soient exportées via OSPF à destination des autres routeurs ?

Il faut consulter la documentation BIRD User's Guide à la section OSPF et rechercher un exemple de la directive suivante.

export

Voici une copie d'écran pour chaque démon avec l'exportation dans le processus OSPF en fonction du filtre défini préalablement.

```
R1:~# sed -n '/^protocol ospf/,/^}/p' /etc/bird/bird.conf
protocol ospf OSPFv2R1 {
        rfc1583compat yes;
        ecmp yes;
        export filter export_OSPF;
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
                };
        };
}
```

```
R1:~# sed -n '/^protocol ospf/,/^}/p' /etc/bird/bird6.conf
protocol ospf OSPFv3R1 {
        ecmp yes;
        export filter export_OSPF;
        area 0 {
                interface "eth0.12", "eth0.13" {
                        authentication none;
                };
        };
}
```

Q137. Quelles sont les nouvelles annonces LSA apparues après exportation des routes par défaut depuis R1 vers les deux autres routeurs de la topologie triangle ?

À partir du Tableau 6.1, « Codage des annonces OSPF » donné en début de section, donner le nouveau rôle du routeur R1.

Une fois que l'exportation des routes par défaut dans OSPF est effective sur R1, ce routeur devient Autonomous System Border Router ou ASBR. Dès lors, il émet des annonces de type 5 que l'on peut identifier dans les bases de chaun des trois routeurs de l'aire OSPF.

Par exemple, on obtient les résultats suivants sur le routeur R2.

```
R2:~# birdc show ospf lsadb type 5
BIRD 1.6.4 ready.

Global

 Type   LS ID           Router          Sequence   Age  Checksum
 0005  0.0.0.0         0.0.1.4         80000001  1570    afb9
```

```
R2:~# birdc6 show ospf lsadb type 5
BIRD 1.6.4 ready.

Global

 Type   LS ID           Router          Sequence   Age  Checksum
 4005  0.0.0.0         0.0.1.6         80000001  1574    e4c8
```

Q138. Comment valider l'exportation des routes par défaut depuis les deux démons `bird` et `bird6` vers le niveau système sur les routeurs R2 et R3 ?

L'exportation des routes depuis les démons Bird vers le système a été traitée à la Section 6.4, « Échanger les routes entre Bird et le système ». Ici, on se contente de vérifier la présence des routes par défaut au niveau système sur R2 et R3.

Voici une copie d'écran pour le routeur R2 qui caractérise le fait que les routes par défaut ont été apprises via Bird.

```
R2:~# ip route ls default
default via 10.1.12.1 dev eth0.12 proto bird

R2:~# ip -6 route ls default
default via fe80::70de:4fff:fe1d:68b4 dev eth0.12 proto bird metric 1024 pref medium
```

Si la table de routage du routeur d'accès à Internet contient les routes statiques vers les réseaux de l'aire OSPF, il est possible de lancer les tests ICMP classiques. Voici deux exemples depuis le routeur R2.

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

--- 9.9.9.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 13.796/13.910/14.024/0.114 ms

R2:~# ping -qc2 2620:fe::fe
PING 2620:fe::fe(2620:fe::fe) 56 data bytes

--- 2620:fe::fe ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 43.986/44.802/45.618/0.816 ms
```

## 6.6. Consulter les documents de référence

Architecture réseau des travaux pratiques

Le support Architecture réseau des travaux pratiques présente la topologie physique des salles de travaux pratiques avec la Disposition des équipements dans l'armoire de brassage ainsi que les configurations par défaut des équipements. On y trouve aussi le plan d'adressage IP utilisé avec les autres supports de travaux pratiques, le plan de numérotations des VLANs et les affectations des groupes de ports des commutateurs.

Configuration d'une interface réseau

Le support Configuration d'une interface de réseau local présente les opérations de configuration d'une interface réseau et propose quelques manipulations sur les protocoles de la pile TCP/IP

## Introduction au routage inter-VLAN

Le support Routage Inter-VLAN introduit le principe du routage inter-VLAN ainsi que ses conditions d'utilisation. C'est aussi un support de travaux pratiques dans lequel on n'utilise que du routage statique et de la traduction d'adresses sources (S-NAT) pour acheminer le trafic utilisateur entre les différents réseaux.

CHAPITRE 7

# Étude de cas sur l'interconnexion LAN/WAN


Résumé

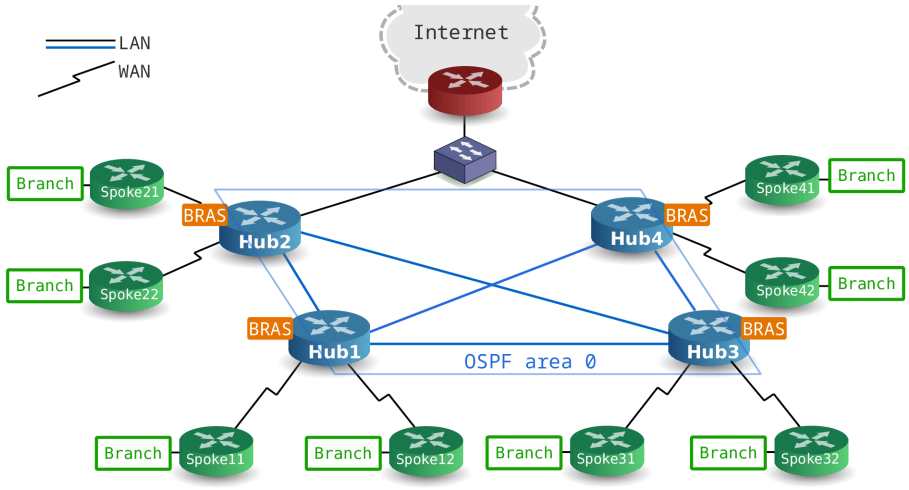
**L'objectif de cette étude de cas est de faire la synthèse sur l'ensemble du cycle de travaux pratiques sur le thème de l'interconnexion réseau LAN/WAN. Côté réseaux étendus, on retrouve la configuration des accès via PPP sur trames «HDLC synchrones» (RNIS) et le filtrage avec et sans traduction d'adresses. Côté réseaux locaux, on reprend le routage inter-VLAN avec le protocole de routage dynamique OSPF.**


## Table des matières

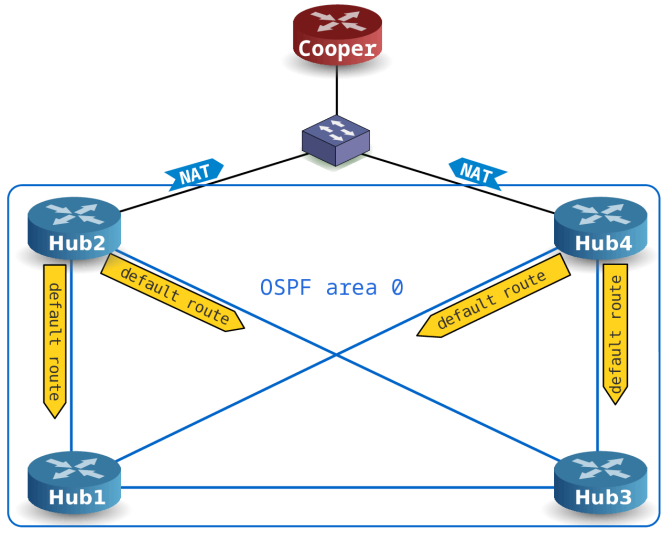


## 7.1. Topologies réseaux

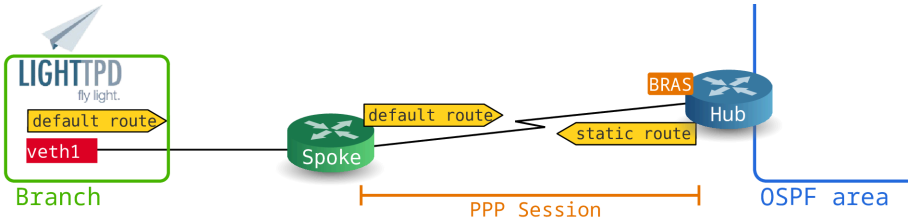
La topologie logique globale se présente comme une associations de topologies triangulaires LAN et WAN. L'ensemble des routeurs présentés appartiennent à une seule aire OSPF.




Les deux sous topologies LAN reprennent l'architecture étudiée dans le support Routage dynamique avec OSPF (Bird). La seule particularité ici réside dans la redondance de deux VLANs entre les routeurs Hub2 et Hub3.



Les quatre sous topologies WAN reprennent l'architecture étudiée dans les supports Topologie Hub & Spoke avec le protocole PPP ou Topologie Hub & Spoke avec le protocole PPPoE. La différence ici réside dans l'utilisation du protocole de routage dynamique OSPF sur des les liens WAN point à point.




L'objectif de cette séance de travaux pratiques est d'aboutir à un accès aux services Web hébergés sur les réseaux fictifs depuis un routeur Spoke vers n'importe quel autre routeur Spoke sans recourir une seule fois à la traduction d'adresses.

## 7.2. Plan d'adressage WAN

Les connexions RNIS des routeurs Hubs se font directement sur les ports de l'autocommutateur RNIS sachant que ces connexions utilisent les deux canaux B d'un port de type BRI.

Un routeur Spoke doit s'authentifier auprès d'un routeur Hub via le protocole PPP avec la méthode CHAP.

Un routeur Spoke doit aussi mettre en place un réseau fictif qui héberge un service Web dans un espace de noms réseau dédié.

Tableau 7.1. Salle 211 - Affectation des rôles, des numéros de VLANs et des adresses IP

| Groupe | Poste | Rôle | VLAN | Flux | Réseau/Authentification |
|---|---|---|---|---|---|
| 1 | christophsis | Hub1 | 400 | Management | `fe80:190::1/64` |
| | | | 401 | Data | `192.168.1.141:192.168.1.142` |
| | | | 402 | Data | `192.168.1.145:192.168.1.146` |
| | corellia | Spoke11 | 400 | Management | `fe80:190::11/64` |
| | | | 401 | Data | `etu_s1 / Sp0k3.1` |
| | | | | Branch | `10.4.0.1/26` |
| | | | | | `2001:678:3fc:191::1/64` |
| | delaya | Spoke12 | 400 | Management | `fe80:190::12/64` |
| | | | 402 | Data | `etu_s2 / Sp0k3.2` |
| | | | | Branch | `10.4.0.65/26` |
| | | | | | `2001:678:3fc:192::1/64` |
| 2 | kashyyyk | Hub2 | 405 | Management | `fe80:195::2/64` |
| | | | 406 | Data | `192.168.1.149:192.168.1.150` |
| | | | 407 | Data | `192.168.1.153:192.168.1.154` |
| | korriban | Spoke21 | 405 | Management | `fe80:195::21/64` |
| | | | 406 | Data | `etu_s1 / Sp0k3.1` |
| | | | | Branch | `10.4.0.129/26` |

| Groupe | Poste | Rôle | VLAN | Flux | Réseau/Authentification |
|---|---|---|---|---|---|
| | | | | | 2001:678:3fc:196::1/64 |
| | kessel | Spoke22 | 405 | Management | fe80:195::22/64 |
| | | | 407 | Data | etu_s2 / Sp0k3.2 |
| | | | | Branch | 10.4.0.193/26 |
| | | | | | 2001:678:3fc:197::1/64 |
| 3 | mygeeto | Hub3 | 410 | Management | fe80:19a::3/64 |
| | | | 411 | Data | 192.168.1.157:192.168.1.158 |
| | | | 412 | Data | 192.168.1.161:192.168.1.162 |
| | nelvaan | Spoke31 | 410 | Management | fe80:19a::31/64 |
| | | | 411 | Data | etu_s1 / Sp0k3.1 |
| | | | | Branch | 10.4.1.1/26 |
| | | | | | 2001:678:3fc:19b::1/64 |
| | rattatak | Spoke32 | 410 | Management | fe80:19a::32/64 |
| | | | 412 | Data | etu_s2 / Sp0k3.2 |
| | | | | Branch | 10.4.1.65/26 |
| | | | | | 2001:678:3fc:19c::1/64 |
| 4 | saleucami | Hub4 | 415 | Management | fe80:19f::4/64 |
| | | | 416 | Data | 192.168.1.165:192.168.1.166 |
| | | | 417 | Data | 192.168.1.169:192.168.1.170 |
| | taris | Spoke41 | 415 | Management | fe80:19f::41/64 |
| | | | 416 | Data | etu_s1 / Sp0k3.1 |
| | | | | Branch | 10.4.1.129/26 |
| | | | | | 2001:678:3fc:1a0::1/64 |
| | teth | Spoke42 | 415 | Management | fe80:19f::42/64 |
| | | | 417 | Data | etu_s2 / Sp0k3.2 |
| | | | | Branch | 10.4.1.193/26 |
| | | | | | 2001:678:3fc:1a1::1/64 |

Tableau 7.2. Salle 213 - Affectation des rôles, des numéros de bus S0 et des adresses IP

| Groupe | Poste | Rôle | Bus S0 | N° Tél. | Interface | Réseau/Authentification |
|---|---|---|---|---|---|---|
| 1 | alderaan | Hub1 | S0.1 | 104 | ippp0 | 192.168.104.1:192.168.104.2 |
| | | | S0.1 | 105 | ippp1 | 192.168.105.1:192.168.105.2 |
| | bespin | Spoke11 | S0.2 | 106 | ippp0 | etu_s11 / Sp0k3.11 |

| Groupe | Poste | Rôle | Bus S0 | N° Tél. | Interface | Réseau/Authentification |
|---|---|---|---|---|---|---|
| | | | | | veth0:veth1 | 10.106.0.1/30: 10.106.0.2/30 |
| | centares | Spoke12 | S0.2 | 107 | ippp0 | etu_s12 / Sp0k3.12 |
| | | | | | veth0:veth1 | 10.107.0.1/30: 10.107.0.2/30 |
| 2 | coruscant | Hub2 | S0.3 | 108 | ippp0 | 192.168.107.1:192.168.107.2 |
| | | | S0.3 | 109 | ippp1 | 192.168.108.1:192.168.108.2 |
| | dagobah | Spoke21 | S0.4 | 110 | ippp0 | etu_s21 / Sp0k3.21 |
| | | | | | veth0:veth1 | 10.110.0.1/30: 10.110.0.2/30 |
| | endor | Spoke22 | S0.4 | 111 | ippp0 | etu_s22 / Sp0k3.22 |
| | | | | | veth0:veth1 | 10.111.0.1/30: 10.111.0.2/30 |
| 3 | felucia | Hub3 | S0.5 | 112 | ippp0 | 192.168.111.1:192.168.111.2 |
| | | | S0.5 | 113 | ippp1 | 192.168.112.1:192.168.112.2 |
| | geonosis | Spoke31 | S0.6 | 114 | ippp0 | etu_s31 / Sp0k3.31 |
| | | | | | veth0:veth1 | 10.114.0.1/30: 10.114.0.2/30 |
| | hoth | Spoke32 | S0.6 | 115 | ippp0 | etu_s32 / Sp0k3.32 |
| | | | | | veth0:veth1 | 10.115.0.1/30: 10.115.0.2/30 |
| 4 | mustafar | Hub4 | S0.7 | 116 | ippp0 | 192.168.115.1:192.168.115.2 |
| | | | S0.7 | 117 | ippp1 | 192.168.116.1:192.168.116.2 |
| | naboo | Spoke41 | S0.8 | 118 | ippp0 | etu_s41 / Sp0k3.41 |
| | | | | | veth0:veth1 | 10.118.0.1/30: 10.118.0.2/30 |
| | tatooine | Spoke42 | S0.8 | 119 | ippp0 | etu_s42 / Sp0k3.42 |
| | | | | | veth0:veth1 | 10.119.0.1/30: 10.119.0.2/30 |

## 7.3. Plan d'adressage LAN

La section «Plan d'adressage» du document Architecture réseau des travaux pratiques donne les adresses, dans le VLAN numéro 4, des deux routeurs ayant accès au réseau du campus.

- Routeur `cooper.stri` : `172.16.16.1/20` et `2001:678:3fc:4::1/64`
- Routeur `casper.stri` : `172.16.16.2/20` et `2001:678:3fc:4::2/64`

Tous les ports utilisés pour raccorder les routeurs ayant le rôle Hub doivent être configurés en mode trunk puisqu'ils servent à véhiculer le trafic de plusieurs domaines de diffusion. On conserve le VLAN natif numéro 1.

Tableau 7.3. Salle 211 - Affectation d'un commutateur par groupe

| Groupe | Commutateur |
|---|---|
| 1 | asw05-211.infra.stri |
| 2 | asw06-211.infra.stri |
| 3 | asw07-211.infra.stri |

| Groupe | Commutateur |
|---|---|
| 4 | asw08-211.infra.stri |

Tableau 7.4. Salle 211 - Affectation des rôles, des numéros de VLANs et des adresses IP

| Groupe | Poste | Rôle | OSPF router-id | VLAN | Interface | Réseau |
|---|---|---|---|---|---|---|
| 1 | christophsis | Hub1 | OSPFv2 : 0.211.4.1<br>OSPFv3 : 0.211.6.1 | 421 | eth0.421 | 10.4.21.1/28<br>2001:678:3fc:1a5::1 |
| | | | | 431 | eth0.431 | 10.4.31.1/28<br>2001:678:3fc:1af::1 |
| | | | | 441 | eth0.441 | 10.4.41.1/28<br>2001:678:3fc:1b9::1 |
| 2 | kashyyyk | Hub2 | OSPFv2 : 0.211.4.2<br>OSPFv3 : 0.211.6.2 | 4 | eth0.4 | 172.16.18.2/20<br>2001:678:3fc:4::70a |
| | | | | 421 | eth0.421 | 10.4.21.2/28<br>2001:678:3fc:1a5::2 |
| | | | | 432 | eth0.432 | 10.4.32.2/28<br>2001:678:3fc:1b0::2 |
| 3 | mygeeto | Hub3 | OSPFv2 : 0.211.4.3<br>OSPFv3 : 0.211.6.3 | 431 | eth0.431 | 10.4.31.3/28<br>2001:678:3fc:1af::3 |
| | | | | 432 | eth0.432 | 10.4.32.3/28<br>2001:678:3fc:1b0::3 |
| | | | | 443 | eth0.443 | 10.4.43.3/28<br>2001:678:3fc:1bb::3 |
| 4 | saleucami | Hub4 | OSPFv2 : 0.211.4.4<br>OSPFv3 : 0.211.6.4 | 4 | eth0.4 | 172.16.18.4/20<br>2001:678:3fc:4::70c |
| | | | | 441 | eth0.441 | 10.4.41.4/28<br>2001:678:3fc:1b9::4 |
| | | | | 443 | eth0.443 | 10.4.43.4/28<br>2001:678:3fc:1bb::4 |

Tableau 7.5. Salle 213 - Affectation d'un commutateur par groupe

| Groupe | Commutateur |
|---|---|
| 1 | asw05-213.infra.stri |
| 2 | asw06-213.infra.stri |
| 3 | asw07-213.infra.stri |
| 4 | asw08-213.infra.stri |

Tableau 7.6. Salle 213 - Affectation des rôles, des numéros de VLANs et des adresses IP

| Groupe | Poste | Rôle | OSPF router-id | VLAN | Interface | Réseau |
|---|---|---|---|---|---|---|
| 1 | alderaan | Hub1 | OSPFv2 : 0.213.4.1<br>OSPFv3 : 0.213.6.1 | 221 | eth0.221 | 10.2.21.1/28 |
| | | | | | | 2001:678:3fc:dd::1 |
| | | | | 231 | eth0.231 | 10.2.31.1/28 |
| | | | | | | 2001:678:3fc:e7::1 |
| | | | | 241 | eth0.241 | 10.2.41.1/28 |
| | | | | | | 2001:678:3fc:f1::1 |
| 2 | coruscant | Hub2 | OSPFv2 : 0.213.4.2<br>OSPFv3 : 0.213.6.2 | 4 | eth0.4 | 172.16.17.2/20 |
| | | | | | | 2001:678:3fc:4::6a6 |
| | | | | 221 | eth0.221 | 10.2.21.2/28 |
| | | | | | | 2001:678:3fc:dd::2 |
| | | | | 232 | eth0.232 | 10.2.32.2/28 |
| | | | | | | 2001:678:3fc:e8::2 |
| 3 | felucia | Hub3 | OSPFv2 : 0.213.4.3<br>OSPFv3 : 0.213.6.3 | 231 | eth0.231 | 10.2.31.3/28 |
| | | | | | | 2001:678:3fc:e7::3 |
| | | | | 232 | eth0.232 | 10.2.32.3/28 |
| | | | | | | 2001:678:3fc:e8::3 |
| | | | | 243 | eth0.243 | 10.2.43.3/28 |
| | | | | | | 2001:678:3fc:f3::3 |
| 4 | mustafar | Hub4 | OSPFv2 : 0.213.4.4<br>OSPFv3 : 0.213.6.4 | 4 | eth0.4 | 172.16.17.4/20 |
| | | | | | | 2001:678:3fc:4::4 |
| | | | | 241 | eth0.241 | 10.2.41.4/28 |
| | | | | | | 2001:678:3fc:f1::4 |
| | | | | 243 | eth0.243 | 10.2.43.4/28 |
| | | | | | | 2001:678:3fc:f3::4 |

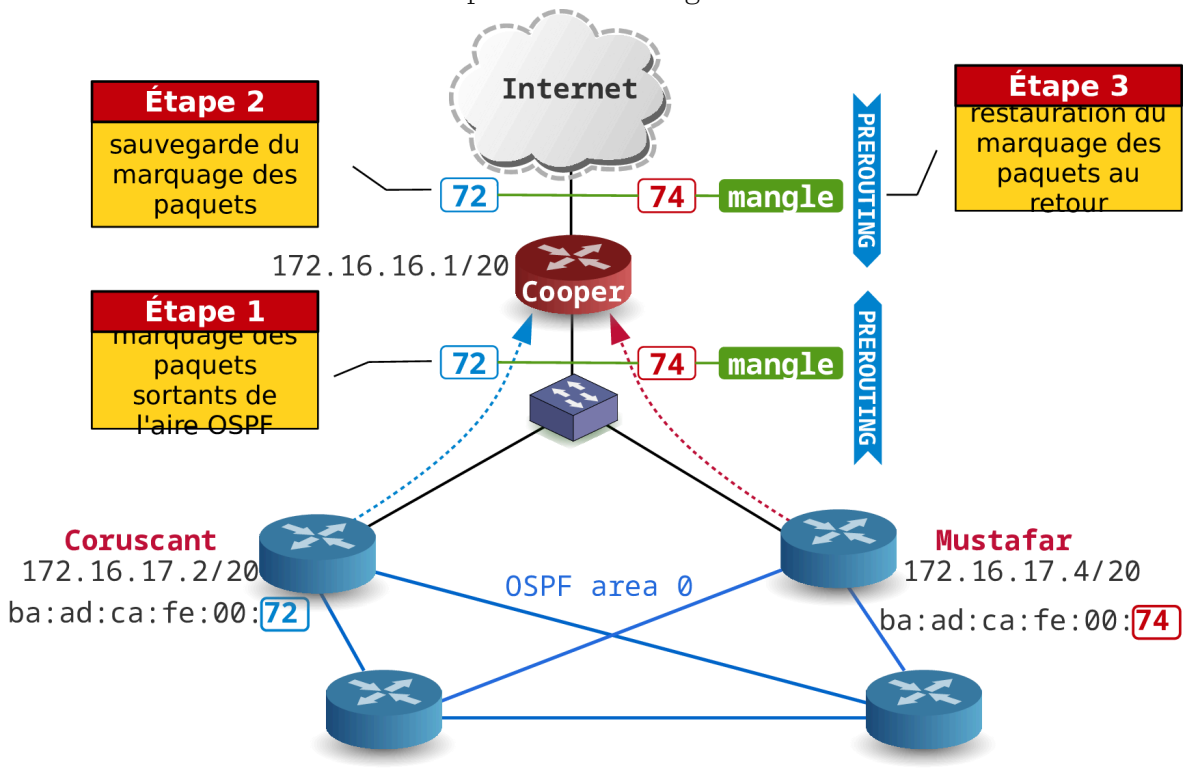
## 7.4. Interconnexion avec deux routeurs de bordure OSPF

Une fois que tous les routeurs (postes de travaux pratiques) sont actifs et que toutes les instances de routage OSPF ont convergé, les Spokes disposent de deux passerelles de sortie vers l'Internet. Les deux routeurs de bordure qui assurent cette fonction de passerelle sont Centares et Naboo. Du point de vue de l'aire OSPF, on dispose ainsi d'une tolérance aux pannes puisque les instances de routage OSPF effectuent automatiquement un recalcul de topologie dans le cas où l'une des deux passerelles viendrait à «tomber».

Le routeur de niveau supérieur, Cooper, ne dispose pas du même niveau d'information puisqu'il n'y a aucun échange de protocole de routage entre lui et les deux routeurs de bordure de l'aire OSPF. Pour associer ce routeur au mécanisme de tolérance aux pannes, on utilise le marquage de paquets. L'idée est que pour tout flux issu d'une passerelle, le flux retour soit renvoyé à cette même passerelle.

On identifie ainsi la source du trafic. Si une des deux passerelles «tombe», elle n'émet plus aucun flux et le routeur Cooper ne verra plus de nouveau flux provenant de son interface.

Comme les interfaces des trois routeurs appartiennent au même VLAN ou domaine de diffusion, on utilise les adresses MAC comme identifiant de marquage. Dans cet exemple, l'octet le plus à droite de l'adresse MAC sert à identifier la passerelle à l'origine du trafic.




> **Note**
>
> Les manipulations présentées ci-dessous sont réalisées par l'enseignant sur le routeur Cooper en début de séance. Compte tenu des «aléas de configuration» dans l'aire OSPF, le mécanisme de tolérance aux pannes est très utile dans le contexte des travaux pratiques.

Le processus de traitement suit les étapes suivante pour un flux sortant de l'aire OSPF.

1. Nouveau flux entrant sur l'interface de Cooper en provenance de l'une des deux passerelles
2. Marquage du premier paquet en fonction de l'adresse MAC source dans la chaîne PREROUTING de la table `mangle`.
3. Mémorisation du marquage de paquet dans le mécanisme suivi d'état du système de filtrage (connmark)
4. Entrée dans la table de routage dédiée au routeur de bordure à l'origine du flux.

Le processus de traitement suit les étapes suivante pour un flux retour vers l'aire OSPF.

1. Restauration du marquage de paquet en fonction des enregistrements effectués via le mécanisme suivi d'état du système de filtrage (connmark)
2. Entrée dans la table de routage dédiée au routeur de bordure à l'origine du flux.

Les opérations de configuration correspondantes sont données ci-après.

Création des tables de routage dédiées à chaque passerelle

- Édition du fichier `/etc/iproute2/rt_tables`.

```
# cat /etc/iproute2/rt_tables
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
72      centares
79      naboo
```

- Ajout des entrées dans les deux nouvelles tables de routage.

```
# ip route add 10.0.16.0/20 via 172.16.17.10 table centares
# ip route add 10.0.32.0/20 via 172.16.17.10 table centares
# ip route add default dev bond0 table centares
```

```
# ip route add 10.0.16.0/20 via 172.16.17.40 table naboo
# ip route add 10.0.32.0/20 via 172.16.17.40 table naboo
# ip route add default dev bond0 table naboo
```

Dans les deux copies d'écran ci-dessus on a agrégé tous les réseaux de l'aire OSPF en deux entrées.

Création des règles de marquage des flux

La table `mangle` est dédiée à l'altération des paquets. Ici, on s'intéresse uniquement à l'ajout d'un marquage de chaque paquet transitant par les interfaces réseau.

```
# iptables -t mangle -A PREROUTING -i bond0.4 \
 -m mac --mac-source 00:1f:c6:01:26:72 -j MARK --set-mark 72
# iptables -t mangle -A PREROUTING -i bond0.4 \
 -m mac --mac-source 00:1f:c6:01:26:79 -j MARK --set-mark 79
# iptables -t mangle -A PREROUTING -i bond0.4 -j CONNMARK --save-mark
# iptables -t mangle -A PREROUTING -i bond0 -j CONNMARK --restore-mark
```

Création des règles d'entrée dans les tables de routage

C'est le marquage qui détermine le choix de la table de routage à utiliser.

```
# ip rule add fwmark 72 table centares
# ip rule add fwmark 79 table naboo
```

Et voilà ! Il ne reste plus qu'à consulter les entrées `conntrack` à l'aide de la commande `# conntrack -L` pour voir apparaître les marquages.