Manuel de Travaux Pratiques Interconnexion LAN/WAN

Philippe Latu
philippe.latu(at)inetdoc.net

http://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. Les questions correspondantes ne peuvent donc être traitées que dans la salle de travaux pratiques de l'Université Toulouse 3.

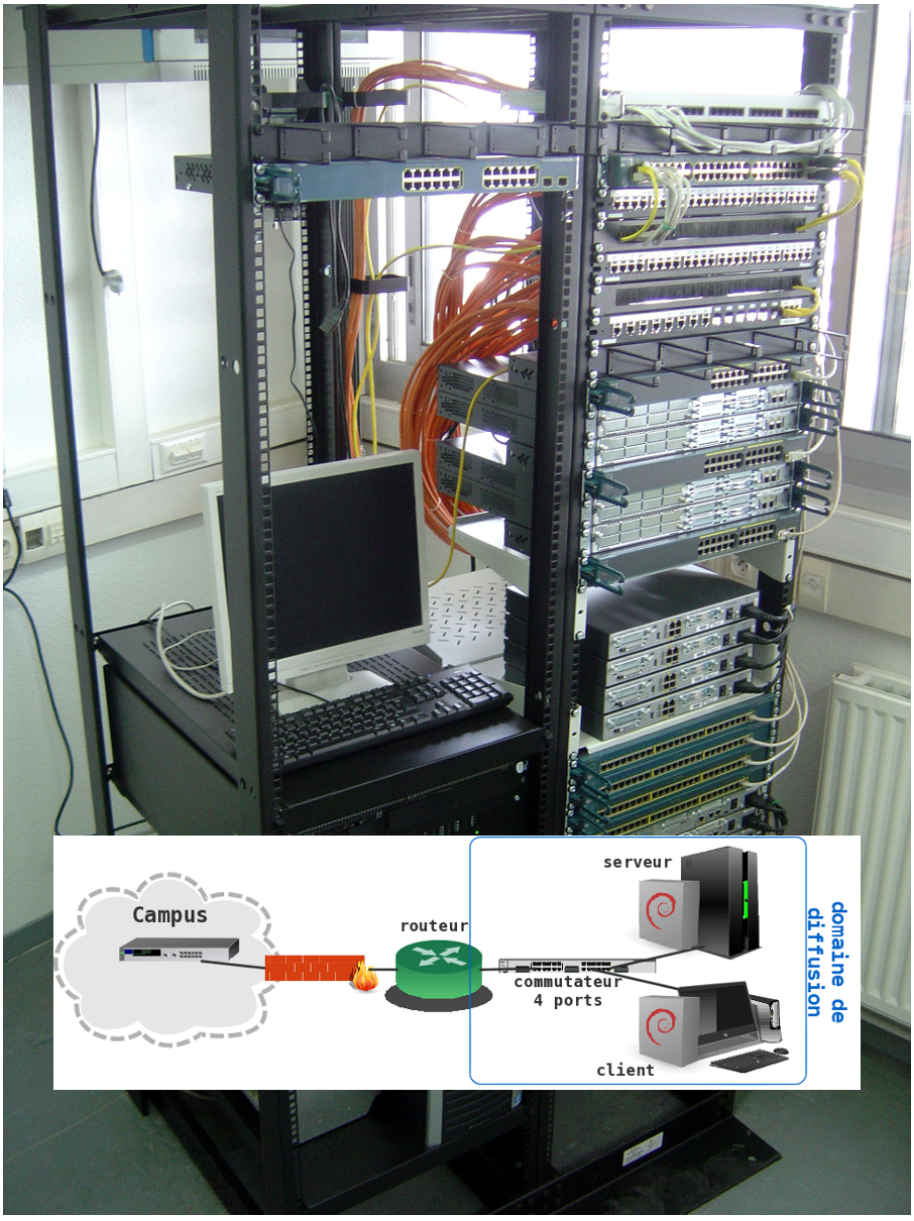


Table des matières

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.

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          - Manual pages about using a GNU/Linux system
i   manpages-fr       - French version of the manual pages about using GNU/Linux
i   manpages-fr-extra - French version of the manual pages

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

<snip/>
$ apropos informations, système | grep uname
uname (1)            - Afficher des informations sur le système
```

Pour obtenir la version courante du noyau exécuté :

```
$ uname -a
Linux vm0 3.0.0-1-amd64 #1 SMP Sat Aug 27 16:21:11 UTC 2011 x86_64 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   linux-image-2.6-amd64      - Linux for 64-bit PCs (dummy package)
i A linux-image-3.0.0-1-amd64  - Linux 3.0.0 for 64-bit PCs
i A linux-image-amd64          - Linux for 64-bit PCs (meta-package)
```

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/
config-3.0.0-1-amd64 ❶
grub
initrd.img-3.0.0-1-amd64 ❷
lost+found
System.map-3.0.0-1-amd64 ❸
vmlinuz-3.0.0-1-amd64 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-3.0.0-1-amd64 | egrep -e 'kernel$'
/lib/modules/3.0.0-1-amd64/kernel
/lib/modules/3.0.0-1-amd64/kernel/arch/x86/kernel
```

La liste ci-dessus montre que les modules du noyau sont placés dans le répertoire `/lib/modules/3.0.0-1-amd64/kernel/`.

Q4. Dans quels 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. Cette arborescence a été introduite avec les noyaux de la série 2.6.xx. 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 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 à l'aide d'une interrogation du type : **`aptitude search ~imodule`**.

La commande **lsmod** :

```
$ lsmod
Module                  Size  Used by
ext2                   63732  1
loop                   22711  0
joydev                 17262  0
snd_pcm                68104  0
evdev                  17558  2
snd_timer              22581  1 snd_pcm
snd                    52823  2 snd_pcm,snd_timer
soundcore              13152  1 snd
<snip/>
```

Cette commande appartient au paquet `module-init-tools`. En listant le contenu de ce paquet on obtient les noms des commandes associées et les pages de manuels correspondantes.

```
$ dpkg -S `which lsmod`
kmod: /bin/lsmod

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

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 pilote d'interface RNIS :

```
# modprobe -v hfcpci
insmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/mISDN/mISDN_core.ko
insmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/hardware/mISDN/hfcpci.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. Toujours avec le pilote d'interface RNIS :

```
# modprobe -rv hfcpci
rmmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/hardware/mISDN/hfcpci.ko
rmmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/mISDN/mISDN_core.ko
```

1.2. Les sources du noyau Linux

Dans cette partie, on s'appuie pas sur le gestionnaire de paquets de la distribution et on télécharge directement les sources du noyau Linux à partir du dépôt défini dans la liste des sources (fichier `/etc/apt/sources.list`).

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.

Les versions stables du noyau évoluent fréquemment. Les questions ci-dessous sont basées sur la version courante de la série 2.6.xx.

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-2.6      - Linux kernel source (dummy package)
p   linux-source-2.6.32   - Linux kernel source for version 2.6.32 with Debian patches
p   linux-source-3.0.0    - Linux kernel source for version 3.0.0 with Debian patches
```

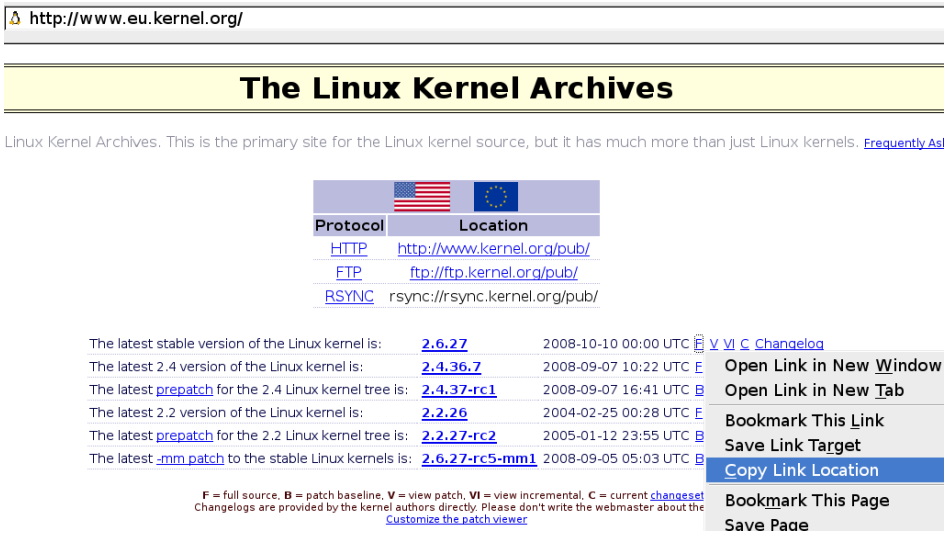
Q10. Quels sont les modes de téléchargement des sources qui permettent de s'affranchir d'une interface graphique ?

La grande majorité des téléchargements se font via le protocole HTTP. Pour trouver l'outil permettant de lancer un téléchargement HTTP, on peut faire une requête par mot clé dans les pages de manuels des outils installés sur le système : **`apropos "network download"`**.

Lorsque l'on utilise des serveurs rack, ceux-ci ne possèdent ni écran ni clavier. Il est donc nécessaire d'effectuer les opérations à distance sans recours à une interface graphique. On dispose de deux protocoles pour les transferts : FTP et HTTP. Les outils correspondant sont `ncftp` et `wget`.

Compte tenu des réponses à la question précédente, on peut utiliser les trois ressources suivantes.

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

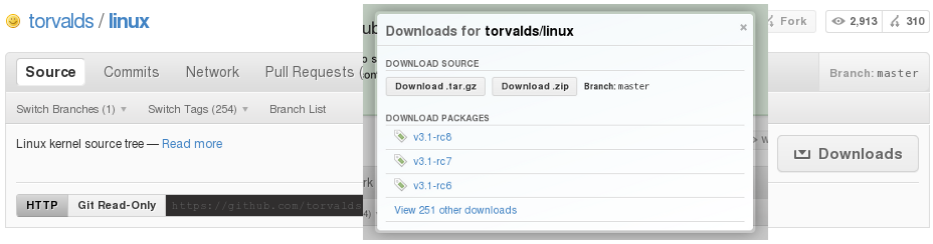


Téléchargement des sources du noyau Linux - vue complète

```
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.bz2
--  http://www.eu.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.bz2
Résolution de www.eu.kernel.org... 130.239.17.4, 199.6.1.164
Connexion vers www.eu.kernel.org|130.239.17.4|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 50355835 (48M) [application/x-bzip2]
Saving to: `linux-2.6.32.tar.bz2'

18% [==================>              ] 9 500 807    217K/s  eta 6m 46s
```

- Téléchargement à partir du dépôt GitHub.



Téléchargement des sources du noyau Linux sur GitHub - vue complète

```
$ wget https://github.com/torvalds/linux/tarball/v3.1-rc8
--2011-10-04 00:43:19--  https://github.com/torvalds/linux/tarball/v3.1-rc8
Résolution de github.com (github.com)... 207.97.227.239
Connexion vers github.com (github.com)|207.97.227.239|:443...connecté.
requête HTTP transmise, en attente de la réponse...302 Found
Emplacement: https://nodeload.github.com/torvalds/linux/tarball/v3.1-rc8 [suivant]
--2011-10-04 00:43:20--  https://nodeload.github.com/torvalds/linux/tarball/v3.1-rc8
Résolution de nodeload.github.com (nodeload.github.com)... 207.97.227.252
Connexion vers nodeload.github.com (nodeload.github.com)|207.97.227.252|:443...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 98197772 (94M) [application/octet-stream]
Sauvegarde en : «v3.1-rc8»

100%[======================================>] 98 197 772  6,64M/s   ds 18s

2011-10-04 00:43:39 (5,14 MB/s) - «v3.1-rc8» sauvegardé [98197772/98197772]

$ mv v3.1-rc8 linux-3.1-rc8.tar.bz2

$ tar tvf linux-3.1-rc8.tar.bz2
```

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

```
# aptitude install linux-source-3.0.0
Les NOUVEAUX paquets suivants vont être installés :
  binutils{a} bzip2{a} cpp{a} cpp-4.6{a} gcc{a} gcc-4.6{a} libc-dev-bin{a}
  libc6-dev{a} libcloog-ppl0{a} libgmp10{a} libgmpxx4ldbl{a} libgomp1{a}
  libmpc2{a} libmpfr4{a} libppl-c4{a} libppl9{a} libpwl5{a} libquadmath0{a}
  linux-libc-dev{a} linux-source-3.0.0 make{a} manpages-dev{a}
0 paquets mis à jour, 22 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 102 Mo d'archives. Après dépaquetage, 148 Mo seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

Q11. À quel groupe doit appartenir l'utilisateur normal pour pouvoir effectuer les opérations de compilation de modules ou du noyau ?

Rechercher dans la liste des groupes système, celui consacré à la manipulation des sources.

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 de l'arborescence système dédié au stockage des sources du noyau Linux ?

Faire une recherche dans le document Linux Filesystem Hierarchy.

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

On vérifie que les membres du groupe système `src` on bien accès 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 les fichiers sources obtenus 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-3.0.0.tar.bz2❶
$ ln -s linux-source-3.0.0 linux❷
$ cd linux
$ cp /boot/config-3.0.0-1-amd64 .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.
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.

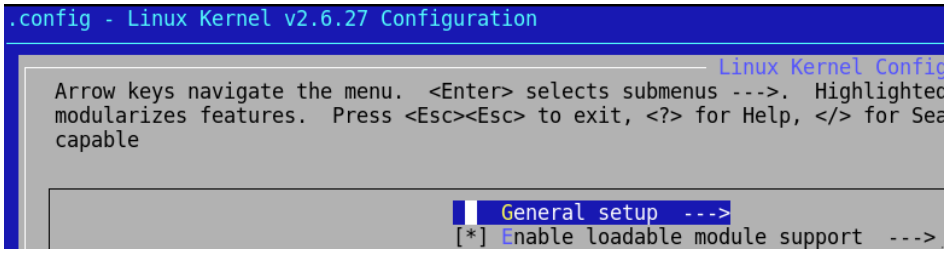
Q17. Sans opération préalable, quel est le fichier contenant les options de configuration du noyau utilisé ?

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 ?

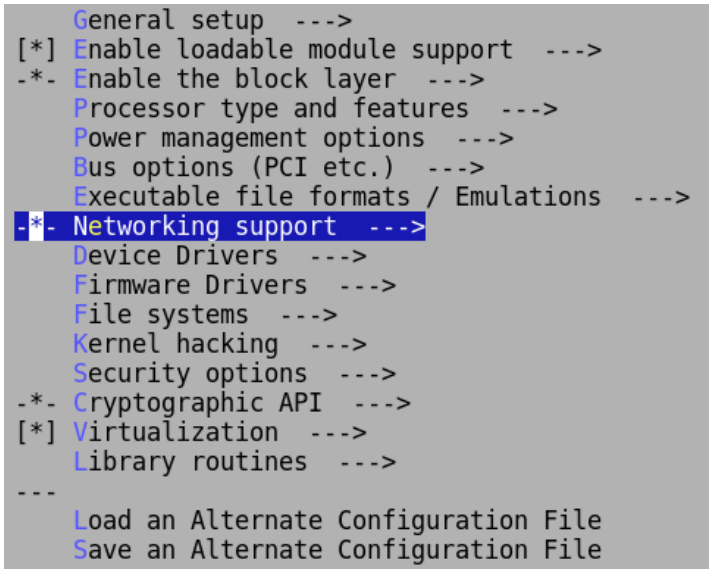
La version du noyau en cours de configuration est indiquée en haut à gauche de l'écran.



Identification version noyau Linux - vue complète

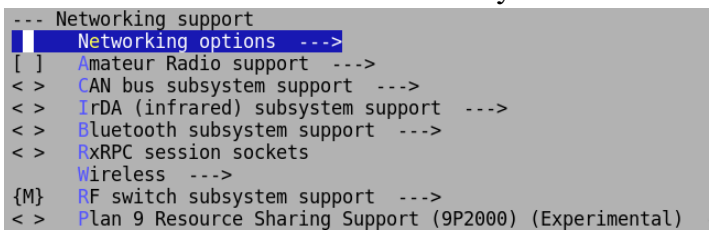
Q19. Quelles sont les options indispensables et facultatives des rubriques Networking Support puis Networking options ?

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



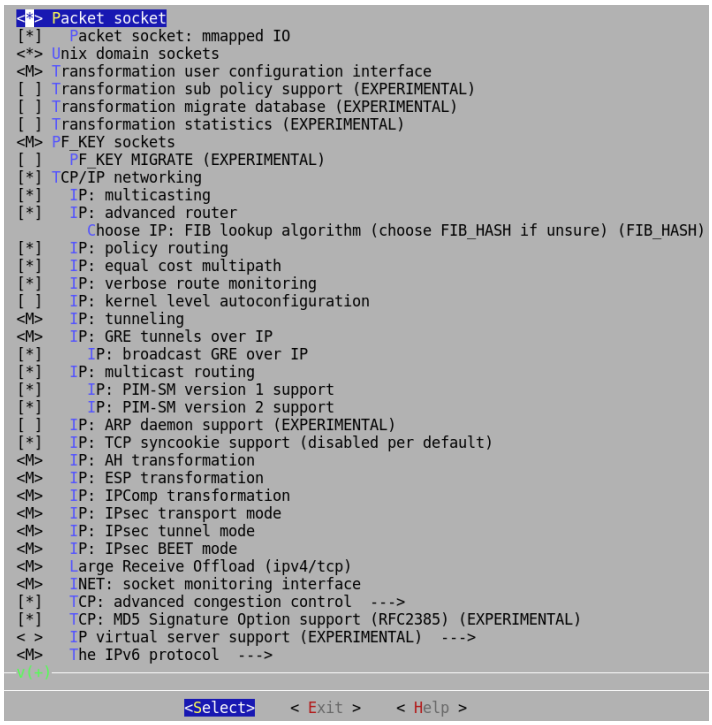
Accès aux types de réseaux supportés - vue complète

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.



Fonctions TCP/IP du noyau Linux - vue complète

Le menu principal de la partie filtrage netfilter se présente comme une longue liste de fonctions. Sachant que les modules dédiés à une fonction du filtrage se chargent dynamiquement à la demande lors de l'application des règles de filtrage, on sélectionne généralement la totalité de

ces fonctions sous forme modulaire. Seuls les modules effectivement utilisés seront chargés en mémoire.

```
--- Network packet filtering framework (Netfilter)
[ ]   Network packet filtering debugging
[ ]   Advanced netfilter configuration
      Core Netfilter Configuration  --->
      IP: Netfilter Configuration  --->
      IPv6: Netfilter Configuration  --->
```

Fonctions de filtrage réseau du noyau Linux - vue complète

Q20. Quelles sont les options indispensables et facultatives des rubriques Device Drivers puis Network device support ?

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

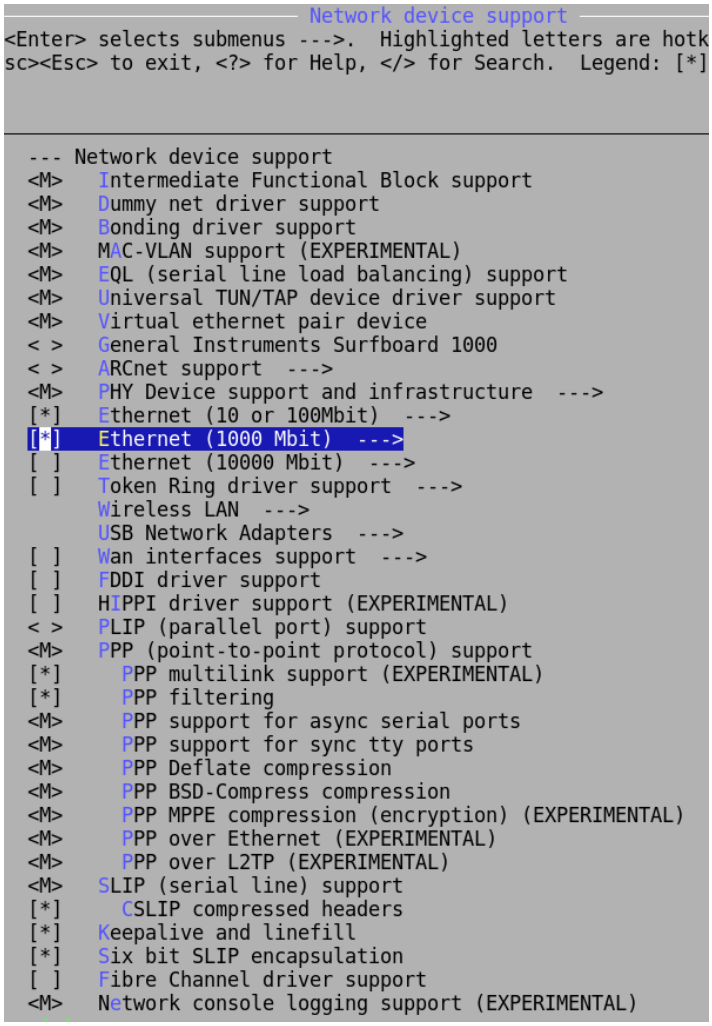
Pour accéder au catalogue des interfaces réseau supportées par le noyau il faut passer par la catégorie des pilotes de périphériques ou Device Drivers pour accéder à l'item Network device support.

```
                        Device Drivers
Enter> selects submenus --->.  Highlighted letters are hotkeys.  Press
><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in

    Generic Driver Options  --->
<M> Connector - unified userspace <-> kernelspace linker  --->
< > Memory Technology Device (MTD) support  --->
<M> Parallel port support  --->
-*- Plug and Play support  --->
[*] Block devices  --->
[*] Misc devices  --->
<M> ATA/ATAPI/MFM/RLL support  --->
    SCSI device support  --->
<M> Serial ATA (prod) and Parallel ATA (experimental) drivers  --->
[*] Multiple devices driver support (RAID and LVM)  --->
[*] Fusion MPT device support  --->
    IEEE 1394 (FireWire) support  --->
<M> I2O device support  --->
[*] Macintosh device drivers  --->
[*] Network device support  --->
[*] ISDN support  --->
< > Telephony support  --->
    Input device support  --->
    Character devices  --->
{M} I2C support  --->
[*] SPI support  --->
[ ] GPIO Support  --->
{M} Dallas's 1-wire support  --->
-*- Power supply class support  --->
{*} Hardware Monitoring support  --->
{M} Generic Thermal sysfs driver  --->
[*] Watchdog Timer Support  --->
    Sonics Silicon Backplane  --->
    Multifunction device drivers  --->
    Multimedia devices  --->
    Graphics support  --->
<M> Sound card support  --->
[*] HID Devices  --->
[*] USB support  --->
< > MMC/SD card support  --->
< > Sony MemoryStick card support (EXPERIMENTAL)  --->
```

Accès aux interfaces réseau supportées - vue complète

Le catalogue recense tous les types d'interfaces réseau. Dans notre cas, il faut choisir le bon modèle d'interface Ethernet.



Catalogue des types d'interfaces réseau - vue complète

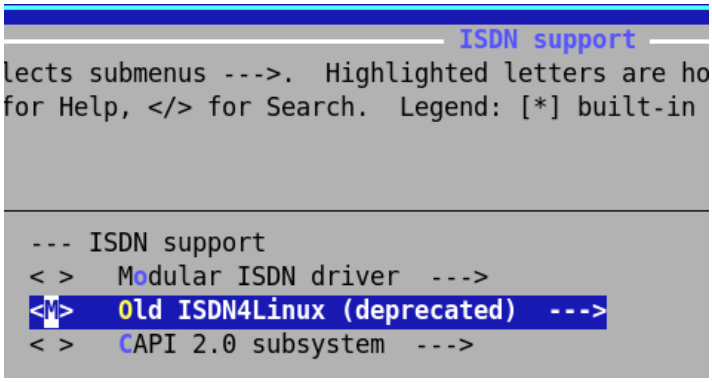
Q21. Quelles sont les options indispensables et facultatives de la rubrique ISDN subsystem ?

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

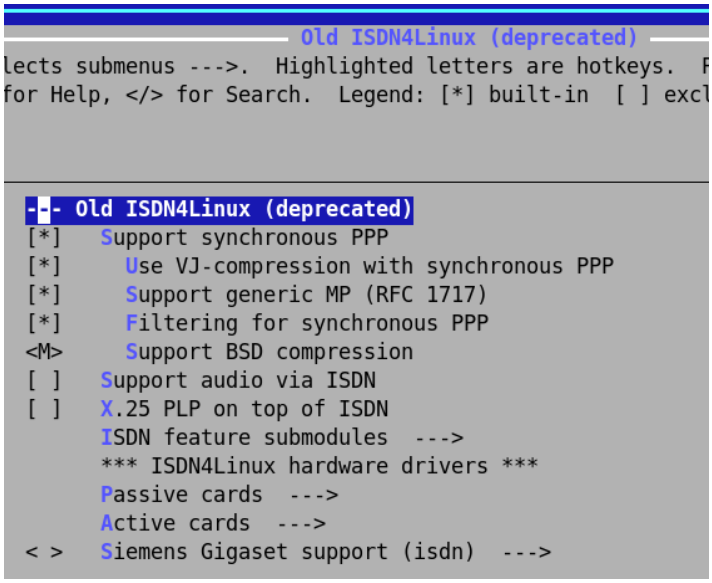
Il existe trois types d'utilisation des connexions RNIS/ISDN dans le noyau Linux.

- Le premier, le plus récent, utilise un mécanisme de sockets adapté aux reste des fonctions réseau du noyau.
- Le second est hérité 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.
- Le troisième utilise le standard CAPI. Il s'agit d'une interface logicielle normalisée entre le noyau et le périphérique matériel.



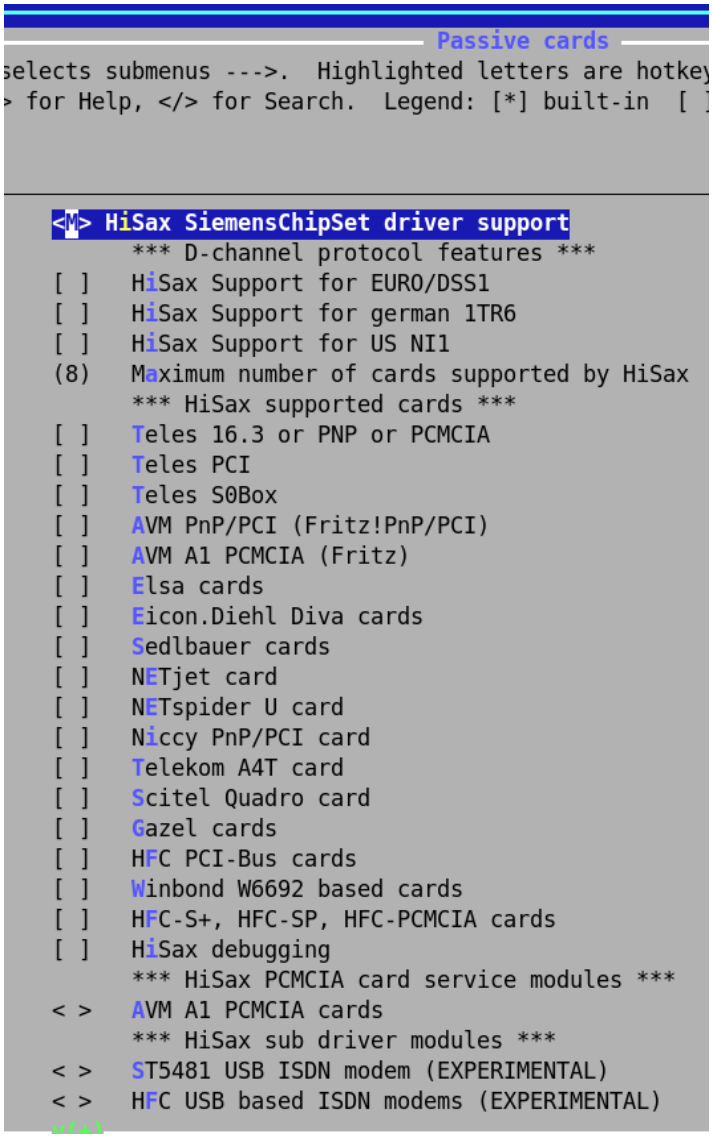
Types de connexions RNIS/ISDN - vue complète

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

La liste des modèles de cartes RNIS/ISDN supportés par le sous-système RNIS/ISDN historique du noyau Linux est donnée ci-dessous. Cette organisation est liée au fait que les mêmes composants Siemens™ ont été utilisés sur de nombreux modèles de cartes de marques différentes.



Modèles de cartes utilisant les mêmes composants Siemens - 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. Quel est le paquet qui contient les outils de construction de paquet de noyau ?

Rechercher le mot clé `kernel` à l'aide du gestionnaire de paquets. Installer le paquet correspondant

La recherche dans les attributs du gestionnaire de paquets permet d'identifier le paquet `kernel-package`.

```
$ aptitude search kernel | grep package
p   debian-kernel-handbook    - reference to Debian Linux kernel packages
p   kernel-package            - A utility for building Linux kernel relate
p   kernel-patch-grsecurity2  - transitional package for Debian Lenny
p   kernel-patch-scripts      - Scripts to help dealing with packaged kern
```

Suivant l'état antérieur de l'installation système, la liste des dépendances est plus ou moins importante lors de l'installation de `kernel-package`.

```
# aptitude install kernel-package
Les NOUVEAUX paquets suivants vont être installés :
  autopoint{a} build-essential{a} dpkg-dev{a} fakeroot{a} g++{a} g++-4.6{a}
  gettext{a} git{a} git-man{a} intltool-debian{a} kernel-package
  libalgorithm-diff-perl{a} libalgorithm-diff-xs-perl{a}
  libalgorithm-merge-perl{a} libcroco3{a} libcurl3-gnutls{a} libdpkg-perl{a}
  liberror-perl{a} libglib2.0-0{a} libglib2.0-data{a} libmail-sendmail-perl{a}
  librtmp0{a} libstdc++6-4.6-dev{a} libsys-hostname-long-perl{a}
  libtimedate-perl{a} libunistring0{a} po-debconf{a} rsync{a}
  shared-mime-info{a}
```

Q23. Quelles sont les commandes de compilation du noyau ?

Rechercher les commandes dans le support Manuel de référence Debian - Chapitre 9 et donner la signification de chacune des commandes.

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
$ export CONCURRENCY_LEVEL=`grep  -c  '^processor'  /proc/cpuinfo`
$ make-kpkg clean
$ make-kpkg --rootcmd fakeroot --initrd --append-to-version -1st-try kernel_image kernel_headers
```

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-*.deb
```

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

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

Q25. Que faut-il faire pour que le gestionnaire de démarrage propose le nouveau noyau compilé lors de l'initialisation du système ?

Identifier le gestionnaire d'amorce installé sur le système.

En fait, l'opération d'installation du paquet de noyau intègre l'ajout d'une nouvelle entrée dans le gestionnaire de démarrage. Aucune opération supplémentaire n'est donc nécessaire.

on peut tout de même valider la liste des noyaux disponibles au niveau du gestionnaire d'amorce. Dans le cas de grub avec la distribution Debian GNU/Linux, on obtient une liste du type suivant.

```
# update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.0.0-1st-try
Found initrd image: /boot/initrd.img-3.0.0-1st-try
Found linux image: /boot/vmlinuz-3.0.0-1-amd64
Found initrd image: /boot/initrd.img-3.0.0-1-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 (RNIS|ISDN). On s'intéresse uniquement au choix des paramètres du niveau liaison de données. En effet, à la différence d'une interface de réseau local Ethernet (LAN), une interface de réseau étendu (WAN) possède un très grand nombre d'options au niveau 2. Il est nécessaire de maîtriser ces options pour exploiter correctement une liaison de ce type. Pour les besoins de la séance de travaux pratiques, on se limite au mode `rawip` au niveau réseau. Dans ce mode chaque extrémité de la liaison WAN est configurée manuellement avec une adresse IP donnée.

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 au démarrage de la machine,
- **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 d'une interface,
- **route** : commande de visualisation et de configuration de la table de routage.

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 Control
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 les outils du paquet `net-tools`, soit avec l'outil ethtool.

L'exécution de la commande **mii-tool** donne le détail de la négociation entre le port du commutateur et celui du poste de travaux pratiques.

```
# mii-tool -v
eth0: negotiated 1000baseT-FD flow-control, link ok
  product info: vendor 00:50:43, model 11 rev 1
  basic mode:   autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: 1000baseT-HD 1000baseT-FD 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:  1000baseT-FD 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
  link partner: 1000baseT-HD 1000baseT-FD 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
```

```
# 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 ?

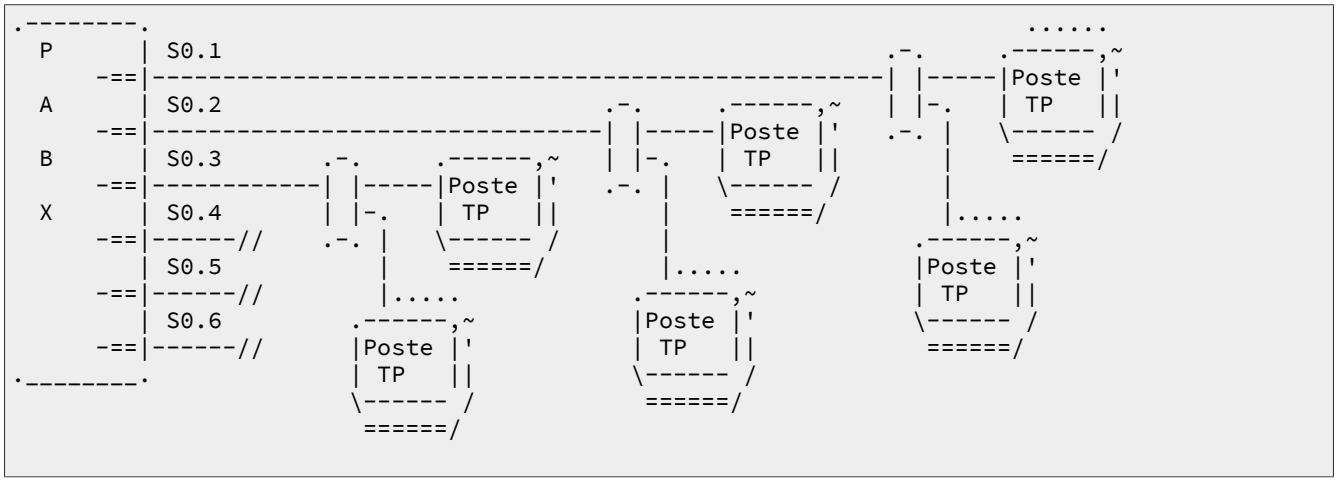
Attention au «piège du débutant» cette validation doit impérativement se faire au niveau réseau sans utiliser un service des couches supérieures tel que la résolution des noms par exemple.

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`
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 q
    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 d'étudier les configurations d'un routeur d'accès (*Hub*) et d'un ou plusieurs routeurs d'agence (*Spoke*). On assimile ces deux configurations types à des routeurs qui réalisent l'interconnexion entre un réseau local et un réseau étendu. La technologie RNIS sert de support au réseau étendu. C'est le moyen d'illustrer une communication à base de trames HDLC et le fonctionnement du protocole PPP.

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 `(r)syslogd`. 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 `(r)syslogd`. 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/syslog.conf`.

```
# grep ^daemon /etc/syslog.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.

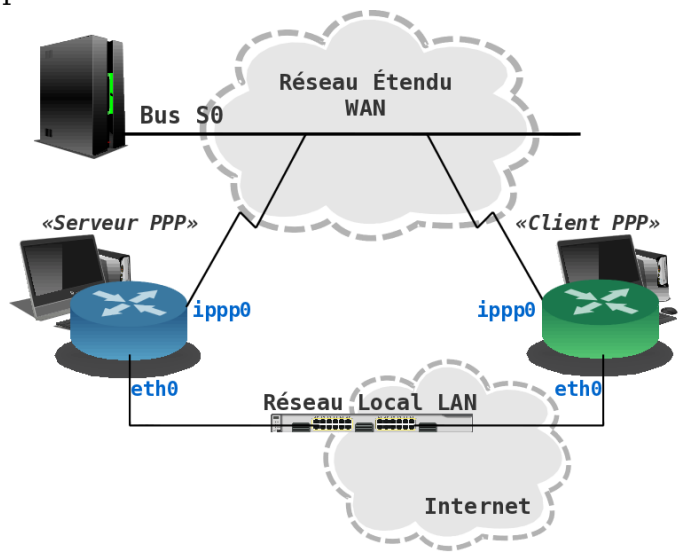
Q46. Quelles sont les noms d'interface RNIS à utiliser avec ce démon de gestion de connexion ?

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.

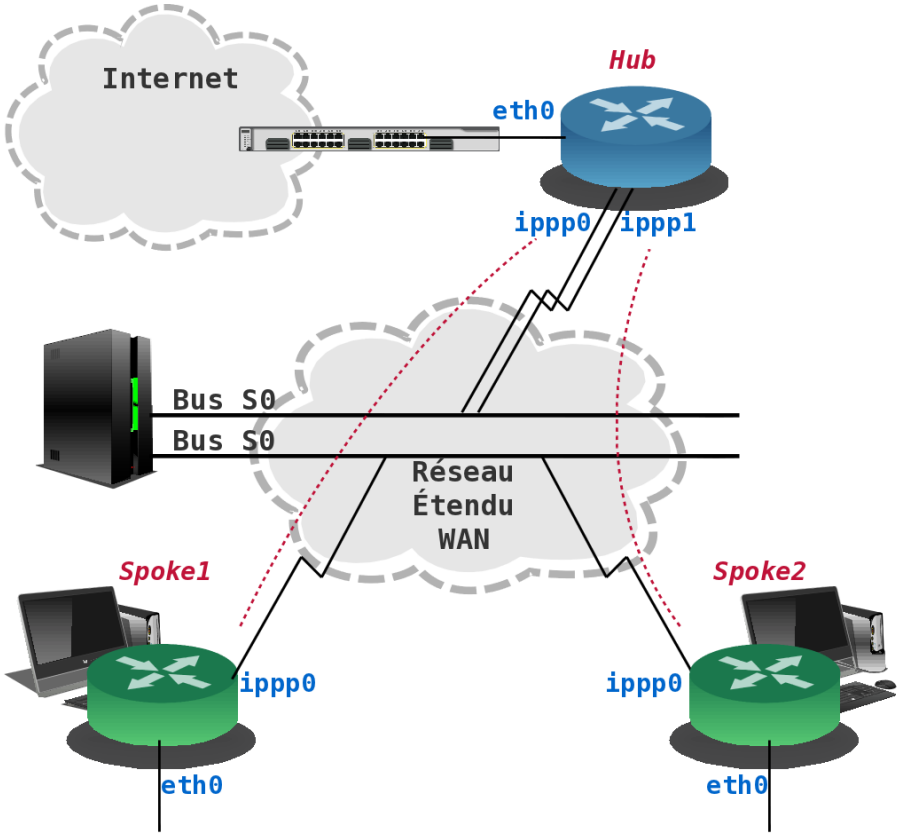
```
pppd[6037]: pppd 2.4.5 started by root, uid 0
pppd[6037]: using channel 28
pppd[6037]: Using interface ppp0
pppd[6037]: Connect: ppp0 < /dev/pts/1
pppd[6037]: sent [LCP ConfReq id=0x1 <auth chap MD5> <magic 0x46c97184>]
pppd[6037]: rcvd [LCP ConfAck id=0x1 <auth chap MD5> <magic 0x46c97184>]
pppd[6037]: 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 = "pp
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 = "
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

Groupe	Poste	Rôle	Bus S0	N° Tél.	Interface	Réseau/Authentification
			-	-	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 les documentations Configuration d'une interface de réseau local et Guide Pratique du NAT.

Les paramètres de réglage des fonctions réseau du noyau Linux sont situés dans l'arborescence `/proc`. Le paramètre qui nous intéresse est `ip_forward`.. Par défaut, sa valeur est à `0` et le routage est désactivé dans le noyau. On doit donc passer cette valeur à `1` pour activer le routage.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

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. Dans le cadre des travaux pratiques, ce n'est pas nécessaire.

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.

```
# modprobe -v dummy numdummies=2
insmod /lib/modules/3.16-2-amd64/kernel/drivers/net/dummy.ko numdummies=2
# ip addr ls | grep dummy
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default
4: dummy1: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default
```

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.

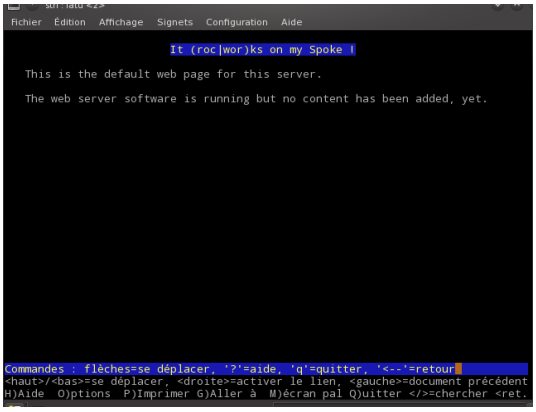
```
$ ping -c 2 10.106.0.1
PING 10.106.0.1 (10.106.0.1) 56(84) bytes of data.
64 bytes from 10.106.0.1: icmp_req=1 ttl=64 time=0.435 ms
64 bytes from 10.106.0.1: icmp_req=2 ttl=64 time=0.360 ms

--- 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

Étude du filtrage réseau dans le contexte d'un routeur central (*hub*) connecté à un routeur d'agence (*spoke*) via un lien WAN. Comme dans les autres supports de travaux pratiques de cette série, on assimile ces configurations types à des interconnexions entre réseaux locaux et réseaux étendus. Les questions de ce support sont présentées comme une introduction pas à pas au filtrage réseau. On débute avec les outils, on poursuit avec les fonctions de suivi de communication (*stateful inspection*) sur une interface, puis on ajoute à ce filtrage les fonctions de traduction d'adresse (NAT).

4.1. Architecture réseau étudiée

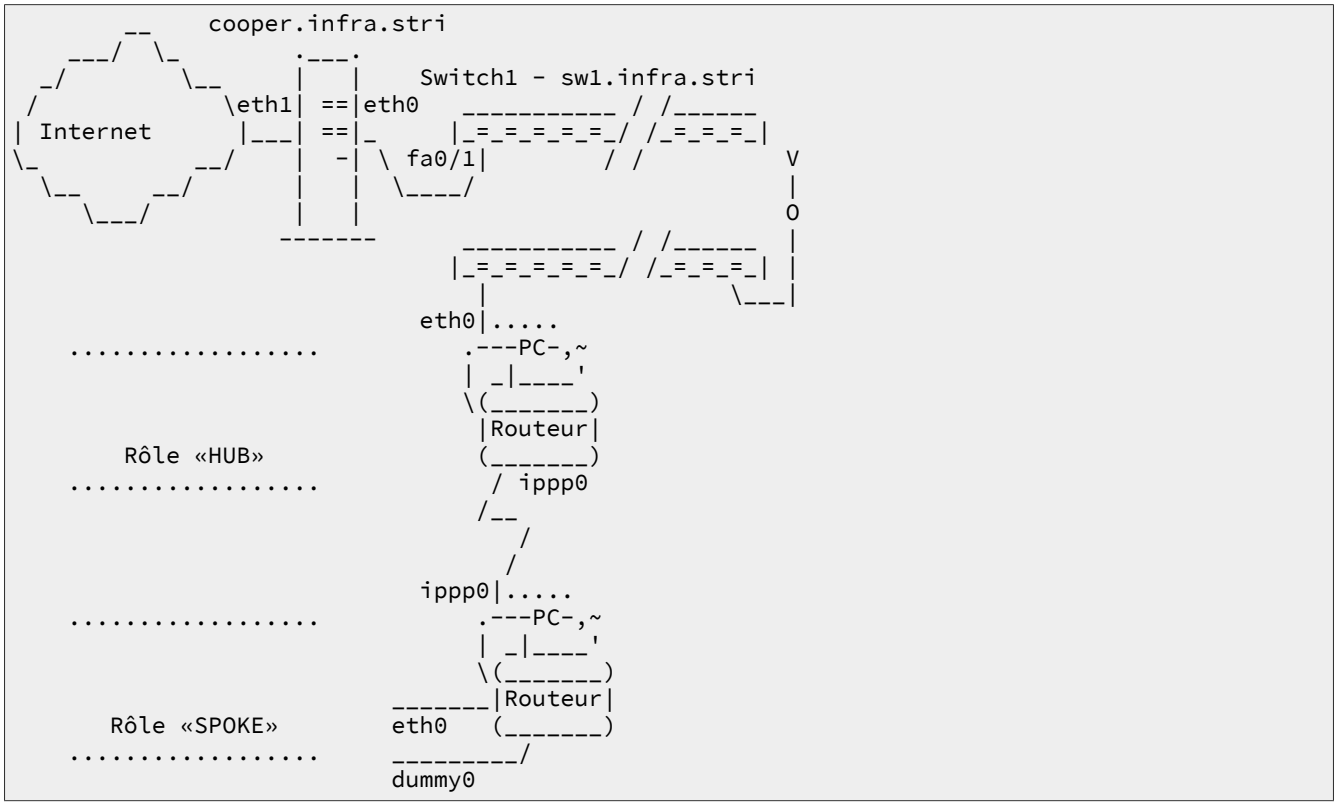
Les manipulations sur le système de filtrage réseau présentées ici s'appuient sur une maquette type qui illustre une «branche» de la topologie Hub and Spoke. Dans cette topologie en étoile, le routeur central joue le rôle d'un concentrateur qui fournit les accès réseau à tous les routeurs d'agence.

4.1.1. Topologie type

La topologie étudiée associe deux postes de travaux pratiques avec deux rôles distincts.

Routeur central, hub, Remote Access Server, RAS
: Ce poste réalise une interconnexion LAN/WAN. Il fournit un accès Internet au routeur d'agence client via son interface WAN. Il dispose de son propre accès Internet via son interface LAN Ethernet.

Routeur d'agence, spoke, Customer Premises Equipment, CPE
: Ce poste réalise aussi une interconnexion LAN/WAN. À la différence du routeur central, il obtient l'accès Internet sur son interface WAN et il met cet accès à disposition d'un réseau local d'extrémité via une interface LAN Ethernet (et|ou) une autre «pseudo» interface réseau.



Pour les besoins des questions de travaux pratiques ci-après, on se limite à un scénario simple d'utilisation des fonctions de filtrage réseau.

Le routeur central doit s'assurer que le trafic réseau qu'il route vers l'Internet provient bien de son routeur d'agence auquel il a attribué une adresse IP via PPP. Ce routeur central doit limiter le volume de

trafic ICMP pour éviter les éventuels dénis de services relatifs à ce protocole. Il peut aussi intercepter toutes les requêtes DNS du routeur d'agence en exploitant un service de type cache only pour éviter les falsifications des réponses aux questions émises par les postes client du réseau d'agence.

Le routeur d'agence doit s'assurer que le trafic réseau entrant sur son interface WAN est bien relatif à une demande qui a été émise via cette même interface. Ce routeur d'agence doit «masquer» son réseau local d'extrémité à l'aide des fonctions de traduction d'adresse source S-NAT. Il doit aussi ouvrir un accès d'administration SSH tout en se protégeant des attaques de type dictionnaire qui consistent à essayer de se connecter au poste avec des milliers de couples login/password connus. Enfin, ce même routeur d'agence doit donner accès à un service Web hébergé dans un périmètre dédié au sein de cette agence.

Pour traiter les questions ci-après, il est vivement conseillé de s'appuyer sur la Section 4.7, « Documents de référence ».

4.1.2. Plan d'adressage WAN

Le plan d'adressage des liaisons WAN reprend celui du support de travaux pratiques Topologie Hub & Spoke avec le protocole PPP.

Tableau 4.1. Plan d'adressage liaisons WAN

Poste routeur central	bus	N° tél.	Adresses IP hub:spoke	N° tél.	Poste routeur d'agence
alderaan	S0.1	104	`192.168.104.1:192.168.104.2`	105	bespin
centares	S0.2	106	`192.168.106.1:192.168.106.2`	107	coruscant
dagobah	S0.3	108	`192.168.108.1:192.168.108.2`	109	endor
felucia	S0.4	110	`192.168.110.1:192.168.110.2`	111	geonosis
hoth	S0.5	112	`192.168.112.1:192.168.112.2`	113	mustafar
naboo	S0.6	114	`192.168.114.1:192.168.114.2`	115	tatooine

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).

Que l'on utilise un noyau fourni par la distribution au le noyau construit à l'issue des travaux pratiques Configuration des fonctions réseau & compilation du noyau Linux, les fonctions de filtrage réseau sont disponibles sous forme de modules que l'on (charge|décharge) de la mémoire système en cours d'exécution. Les outils de filtrage réseau du noyau Linux chargent dynamiquement ces modules en fonction de la syntaxe des règles de filtrage saisies.

4.2.1. Questions sur iptables

Q75. Quels paquets contiennent les outils utilisateur principaux de manipulation des fonctions de filtrage réseau ?

Rechercher dans le cache du gestionnaire de paquets de la distribution des mots clés tels que `iptables` ou `firewall` à l'aide des commandes **apt-cache** ou **aptitude**.

Dans le cadre des travaux pratiques, seuls les outils élémentaires sont utilisés pour faciliter la compréhension des mécanismes de suivi de communication du système de filtrage réseau. On ne s'intéresse donc pas aux paquets qui fournissent des solutions de filtrage «clé en main».

On sait que 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 et on installe les paquets intéressants.

Une recherche simple dans le catalogue des paquets donne le résultat suivant.

```
$ aptitude search iptables
p   arno-iptables-firewall       - single- and multi-homed firewall script with DSL/ADSL support
i   iptables                     - Outils d'administration pour le filtrage de paquets et le NAT
p   iptables-dev                 - iptables development files
p   iptables-persistent          - Simple package to set up iptables on boot
p   libiptables-chainmgr-perl    - Perl extension for manipulating iptables policies
p   libiptables-parse-perl       - Perl extension for parsing iptables firewall rulesets
```

Une recherche plus précise dans les descriptions de paquets donne une liste plus étoffée.

```
# aptitude search '~diptables' | grep ipt
p   apf-firewall                    - easy iptables based firewall system
p   arno-iptables-firewall          - single- and multi-homed firewall script wi
p   firehol                         - An easy to use but powerful iptables state
p   fwsnort                         - Snort-to-iptables rule translator
p   ipkungfu                        - iptables-based Linux firewall
i   iptables                        - Outils d'administration pour le filtrage d
p   iptables-dev                    - iptables development files
p   iptables-persistent             - Simple package to set up iptables on boot
i   iptstate                        - Top-like state for netfilter/iptables
p   libiptables-chainmgr-perl       - Perl extension for manipulating iptables p
p   libiptables-parse-perl          - Perl extension for parsing iptables firewa
p   mxallowd                        - Anti-Spam-Daemon using nolisting/iptables
p   netscript-2.4                   - Linux 2.4.x (and 2.6.x) router/firewall ne
p   uif                             - Advanced iptables-firewall script
p   uruk                            - Very small firewall script, for configurin
i   xtables-addons-common           - Extensions targets and matches for iptable
p   xtables-addons-source           - Extensions targets and matches for iptable
```

Comme on doit rester à un faible niveau d'intégration des règles de filtrage de manière à bien illustrer les mécanismes de suivi de communication. Les deux paquets retenus sont iptables et iptstate. On utilise les commandes usuelles d'installation et de consultation des informations sur ces deux paquets.

```
# aptitude install iptables iptstate
<snipped/>
# apt-cache show iptables
<snipped/>
# apt-cache show iptstate
<snipped/>
```

Q76. Quelles sont les options de la commande iptables qui permettent de visualiser les règles de filtrage réseau actives ainsi que les compteurs correspondants ? Quelle option faut-il préciser pour spécifier la table consultée : netfilter ou nat.

Consulter les pages de manuels via **`# man iptables`**.

La consultation des pages de manuels permet de relever le jeu d'options `-vL`.

```
# iptables -vL
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
```

La table `netfilter` est utilisée de façon implicite alors que la table `nat` de traduction d'adresses doit être appelée explicitement.

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

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

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

Q77. 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. En exécutant cette commande avant et après avoir utilisé **iptables**, on visualise par différence les nouveaux modules chargés par chaque appel. Par exemple, l'exécution des deux commandes de consultation de la question précédente provoque le chargement des modules suivants.

- Après consultation de la table `netfilter` :

```
# lsmod |less
Module                  Size  Used by
iptable_filter          2258  0
ip_tables              13899  1 iptable_filter
x_tables               12845  1 ip_tables
<snipped/>
```

- Après consultation de la table `nat` :

```
# lsmod |less
iptable_nat             4299  0
nf_nat                 13388  1 iptable_nat
nf_conntrack_ipv4       9833  3 iptable_nat,nf_nat
nf_conntrack           46535  3 iptable_nat,nf_nat,nf_conntrack_ipv4
nf_defrag_ipv4          1139  1 nf_conntrack_ipv4
iptable_filter          2258  0
ip_tables              13899  2 iptable_nat,iptable_filter
x_tables               12845  2 iptable_nat,ip_tables
<snipped/>
```

Q78. Quels sont les outils de sauvegarde et de restauration des jeux de règles de filtrage réseau fournis avec le paquet iptables ?

Consulter la liste des fichiers du paquet iptables.

La liste des fichiers du paquet contient les outils recherchés : **`# dpkg -L iptables |grep bin`**. Les deux programmes **iptables-save** et **iptables-restore** permettent respectivement de sauvegarder et de restaurer l'ensemble des règles des tables `netfilter` et `nat`.

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.

4.2.2. Questions sur netfilter

Q79. Comment identifier la version du noyau utilisée et la disponibilité des fonctions de filtrage réseau de cette version ?

Après avoir utilisé la commande «historique» d'identification de la version du noyau, faire une recherche dans l'arborescence des modules de ce noyau avec la commande **find**. Trouver les fichiers dont le nom comprend la chaîne `netfilter`.

On utilise la commande **uname** que l'on associe à une recherche dans l'arborescence des modules du noyau en cours d'exécution.

```
$ uname -r
2.6.32-5-amd64
$ find /lib/modules/`uname -r` -type d -name netfilter
/lib/modules/2.6.32-5-amd64/kernel/net/bridge/netfilter ❶
/lib/modules/2.6.32-5-amd64/kernel/net/ipv4/netfilter   ❷
/lib/modules/2.6.32-5-amd64/kernel/net/decnet/netfilter
/lib/modules/2.6.32-5-amd64/kernel/net/ipv6/netfilter   ❸
/lib/modules/2.6.32-5-amd64/kernel/net/netfilter        ❹
```

❶ Fonctions de filtrage au niveau liaison. Ces fonctions ne sont pas utilisées ici.

❷ Fonctions de filtrage au niveau réseau utilisant le protocole IPv4. C'est dans ce répertoire que se trouvent les modules utilisés dans ces travaux pratiques.
❸ Fonctions de filtrage au niveau réseau utilisant le protocole IPv6. Ces fonctions ne sont pas utilisées ici.
❹ Fonctions communes de filtrage réseau indépendantes des niveaux et des protocoles utilisés. C'est notamment à ce niveau que l'on trouve les modules de la machine d'état de suivi de communications.

Q80. Quels sont les objets du système de fichiers virtuel `/proc` relatifs aux fonctions de filtrage réseau du noyau Linux ?

Avec le chargement des modules en mémoire système, de nouvelles entrées apparaissent dans l'arborescence `/proc`. Même si l'arborescence `/proc` n'est pas un véritable système de fichiers, il est possible d'effectuer des recherches toujours à l'aide de la commande **find**.

Le chargement des trois premiers modules entraîne la création des entrées relatives aux noms chaînes, aux correspondances et aux prises de décisions.

```
# find /proc/net/ -name "*tables*"
/proc/net/ip_tables_targets
/proc/net/ip_tables_matches
/proc/net/ip_tables_names
```

La consultation des règles de la table `nat` entraîne la création de toutes les entrées nécessaires à la machine d'état de suivi de communication.

```
# find /proc/net/ -name "*conntrack*"
/proc/net/ip_conntrack_expect
/proc/net/ip_conntrack
/proc/net/nf_conntrack
/proc/net/nf_conntrack_expect
/proc/net/stat/ip_conntrack
/proc/net/stat/nf_conntrack
```

Q81. Comment visualiser les informations de la machine d'état de suivi de communication ?

Rechercher les entrées de l'arborescence `/proc` dont le nom comprend la chaîne conntrack.

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

Le programme iptstate affiche les entrées de la table de suivi de communication sur le même mode que la commande **top**.

Les états sont directement consultables à partir du fichier virtuel `/proc/net/ip_conntrack`. Par exemple :

```
# cat /proc/net/ip_conntrack
<snipped/>
udp 17 27 src=192.0.2.3 dst=192.0.2.1 sport=54932 dport=53 \
      packets=1 bytes=59 src=192.0.2.1 dst=192.0.2.3 sport=53 dport=54932 \
      packets=1 bytes=259 mark=0 secmark=0 use=2
```

L'exemple ci-dessus donne l'état du suivi de communication d'une requête DNS entre un poste client avec l'adresse `192.0.2.3` et le port source `54932` et un serveur avec l'adresse `192.0.2.1`.

4.3. Règles de filtrage communes à toutes les configurations

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

- Comme les équipements d'interconnexion mis en œuvre dans ces travaux pratiques délimitent des périmètres de faible dimension, on a une connaissance exhaustive des flux réseaux sur le système. On adopte donc la règle : tout trafic réseau non autorisé est interdit.

- Pour exploiter au mieux les fonctionnalités offertes par le noyau Linux, on s'appuie sur le suivi de communication (stateful inspection) pour obtenir un filtrage réseau le plus efficace possible.

On cherche donc à suivre la règle d'or d'écriture des règles de filtrage qui consiste à décrire le plus précisément possible le premier paquet qui doit être enregistré dans la table de suivi de communication. Cette règle de description du premier paquet doit être placée après celle(s) qui laisse(nt) passer les flux réseau déjà enregistrés dans la machine d'état de suivi de communication.

Q82. Quelle est la syntaxe de la commande **iptables** sur la politique par défaut à appliquer sur les chaînes de la table `netfilter` pour respecter le premier principe de filtrage énoncé ci-dessus ?

De façon très classique, on consulte les pages de manuels de la commande **iptables** et on recherche le mot clé `policy`. La stratégie retenue suppose que l'on implante règles d'autorisation des flux réseaux valides et que tout autre trafic soit éliminé. La politique par défaut à appliquer sur les trois chaînes est donc : `DROP`.

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

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT DROP
# iptables -vL
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target   prot opt in    out   source     destination

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

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

Q83. Quelle est la syntaxe de la commande **iptables** qui autorise le trafic réseau déjà enregistré dans la machine d'état de suivi de communication sur les chaînes `INPUT` et `OUTPUT` ?

La recherche de la correspondance `state` dans les pages de manuel de la commande **iptables** permet de sélectionner les états `ESTABLISHED` et `RELATED` à appliquer sur les chaînes.

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

```
# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# iptables -vL
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target   prot opt in    out   source     destination
    0     0 ACCEPT   0    --  any   any   anywhere   anywhere  state RELATED,ESTABLISHED

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

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target   prot opt in    out   source     destination
    0     0 ACCEPT   0    --  any   any   anywhere   anywhere  state RELATED,ESTABLISHED
```

À partir de cette étape, on utilise les programmes **iptables-save** et **iptables-restore** pour optimiser les manipulations. Ces programmes présentent un grand intérêt dans la mesure où l'affichage des règles de filtrage est plus condensé.

```
# iptables-save >/var/lib/iptables/active
# vim /var/lib/iptables/active
# iptables-restore </var/lib/iptables/active
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
COMMIT
```

La commande **iptables-restore** doit être utilisée après chaque édition du fichier `/var/lib/iptables/active`.

Q84. À partir des règles de filtrage précédentes, est-il possible d'émettre ou de recevoir du trafic réseau non enregistré dans la machine d'état de suivi de communication ?

Faire des tests ICMP, DNS et HTTP. Conclure et justifier.

La réponse est non. La politique par défaut sur les chaînes `INPUT` et `OUTPUT` étant positionnée à `DROP`, tout nouveau paquet entrant ou sortant est rejeté. Pour qu'une communication soit possible, il faudrait avoir enregistré un flux réseau dans la machine d'état avant d'appliquer ce jeu de règles de filtrage.

Q85. Quelle est la syntaxe de la commande **iptables** qui autorise le trafic réseau depuis (chaîne `OUTPUT`) et vers (chaîne `INPUT`) l'interface de boucle locale de l'équipement ?

Pour que les processus locaux au système puissent communiquer entre eux via la pile de protocole TCP/IP, il est essentiel d'autoriser le trafic sur l'interface de boucle locale `lo`. La recherche de la correspondance `state` dans les pages de manuel de la commande **iptables** permet de sélectionner l'état `NEW` pour autoriser le premier paquet depuis et vers cette interface. Tous les autres paquets devront correspondre aux règles déjà écrites ci-dessus.

On ajoute deux nouvelles règles sur les chaînes `INPUT` et `OUTPUT` qui admettent respectivement tous les paquets entrant et sortant par l'interface de `lo` dans la table de suivi des communications.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

À partir de ce jeu de règles, on peut lancer un test ICMP : **`# ping -c 4 127.0.0.1`**.

Q86. Quelle est la syntaxe de la commande **iptables** qui autorise le trafic sortant sur l'interface LAN en sortie du système ?

Comme pour la question précédente, les processus locaux au système doivent pouvoir émettre du trafic via la pile de protocole TCP/IP. Il est donc préférable d'autoriser le trafic sortant sur l'interface LAN. On utilise à nouveau l'état `NEW` pour autoriser le premier paquet depuis l'interface. Tous les autres paquets devront correspondre aux communications déjà enregistrées dans les tables de suivi.

On ajoute une règle sur la chaîne `OUTPUT` qui admet, dans la table de suivi des communications, les paquets sortant par l'interface `eth0`.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q87. Quelle est la syntaxe de la commande **iptables** qui autorise le trafic ICMP en entrée du système en limitant le nombre des nouvelles requêtes à 5 par minute ?

Interdire tout trafic ICMP est une très mauvaise idée du point de vue administration réseau. Pour autant, il est très facile de se prémunir contre les tentatives de saturation du trafic sur les interfaces en limitant le nombre de requêtes simultanées en entrée sur toutes les interfaces. Dans un premier temps on se contente de cette règle unique très simple même s'il est judicieux de valider les types et les codes des messages ICMP. Dans un deuxième temps, on distinguera les types de messages ; voir Section 4.6.1, « Protocole ICMP ».

On peut qualifier le fonctionnement de la limitation de trafic à l'aide des commandes **ping** et **hping3** à partir d'un hôte distant.

La recherche de la correspondance `limit` dans les pages de manuel de la commande **iptables** permet de compléter la syntaxe de la règle d'autorisation du trafic ICMP avec l'état `NEW` pour le suivi de communication.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -m limit --limit 5/min -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Les tests de qualification de la nouvelle règle utilisent la commande usuelle **ping** puis un outil beaucoup moins classique qui offre de nombreuses «possibilités», **hping3**.

```
# ping -n -c 3 192.0.2.3
PING 192.0.2.3 (192.0.2.3) 56(84) bytes of data.
64 bytes from 192.0.2.3: icmp_req=1 ttl=64 time=0.958 ms
64 bytes from 192.0.2.3: icmp_req=2 ttl=64 time=0.746 ms
64 bytes from 192.0.2.3: icmp_req=3 ttl=64 time=0.803 ms

--- 192.0.2.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.746/0.835/0.958/0.095 ms

# hping3 -1 --rand-source --fast -c 10 192.0.2.3
HPING 192.0.2.3 (tap0 192.0.2.3): icmp mode set, 28 headers + 0 data bytes
len=28 ip=192.0.2.3 ttl=64 xml:id=5204 icmp_seq=0 rtt=1.0 ms
len=28 ip=192.0.2.3 ttl=64 xml:id=9948 icmp_seq=1 rtt=0.5 ms
len=28 ip=192.0.2.3 ttl=64 xml:id=31892 icmp_seq=2 rtt=0.6 ms
len=28 ip=192.0.2.3 ttl=64 xml:id=48870 icmp_seq=3 rtt=0.6 ms
len=28 ip=192.0.2.3 ttl=64 xml:id=40501 icmp_seq=4 rtt=0.7 ms

--- 192.0.2.3 hping statistic ---
10 packets transmitted, 5 packets received, 50% packet loss
```

On constate que le premier test ne produit aucune erreur alors que la tentative de spoofing rapide des adresses IP source entraîne des pertes de paquets ICMP dès que la limite fixée dans la règle de filtrage est atteinte.

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

4.4. Règles de filtrage sur le poste routeur d'agence (*spoke*)

Suivant le cahier des charges fixé, la première contrainte imposée au poste routeur d'agence est de s'assurer que le trafic entrant sur son interface WAN est bien relatif à une demande émise via cette même interface. Le seconde contrainte étant le routage pour le réseau local d'extrémité, on doit compléter le filtrage avec une traduction d'adresse source liée à cette même interface WAN.

Il est conseillé de travailler à partir du fichier de règles de filtrage établi dans la section précédente. Après chaque édition de ce fichier, la commande **iptables-restore** permet d'appliquer le nouveau jeu de règles après avoir effacé les règles précédentes et remis les compteurs de paquets à zéro.

Q88. Quelle est la syntaxe de la commande **iptables** qui autorise le trafic sortant sur l'interface WAN ?

Relativement à la configuration commune présentée précédemment, il suffit d'ajouter une règle d'autorisation sur la chaîne OUTPUT tout en enregistrant le premier paquet sortant avec l'état NEW.

Avec ce jeu de règles actif, on peut lancer la séquence habituelle des tests ICMP et caractériser l'utilisation des règles en visualisant l'évolution des compteurs avec la commande **iptables -vL**.

Comme dans le cas de l'interface LAN, on ajoute une règle qui admet les nouveaux paquets sortant par l'interface `ippp0` dans la table de suivi des communications via la chaîne `OUTPUT`.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/min -m conntrack --ctstate NEW -j AC
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o ippp0 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q89. Quelle est la syntaxe de la commande **iptables** qui active la traduction d'adresse source pour le trafic sortant sur l'interface WAN ?

Le but est de configurer le routeur d'agence pour qu'il ne soit visible de l'Internet qu'à travers l'adresse IP délivrée par le routeur central (ou le fournisseur d'accès). Cette opération utilise la table `nat`. On ajoute dans cette table une règle de traduction d'adresse source dynamique liée à l'interface WAN. Il faut rechercher les informations sur la cible `MASQUERADE` dans les pages de manuels de la commande **iptables**.

On ajoute une règle dans la chaîne `POSTROUTING` qui traduit l'adresse IP source de tous les paquets sortant par l'interface `ippp0` avec l'adresse attribuée dynamiquement via le protocole PPP à cette interface.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: POSTROUTING
-A POSTROUTING -o ippp0 -j MASQUERADE
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/min -m conntrack --ctstate NEW -j AC
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o ippp0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q90. Ce jeu de règles est-il suffisant pour que le poste se comporte comme un routeur avec une fonction de traduction d'adresses IP sources ?

Cette question comprend deux parties. Il faut s'intéresser dans un premier temps à la fonction de routage proprement dite et consulter l'indicateur d'état du noyau Linux qui correspond à la capacité à transmettre un paquet d'une interface vers une autre. Dans un second temps, il faut identifier dans le système de filtrage la chaîne dédiée au transit des paquets.

La réponse est non. Il manque au moins deux conditions pour que le routage et la transmission des paquets entre les interfaces soient actifs.

- Pour qu'un paquet soit transmis d'une interface réseau vers une autre, il faut s'assurer que le routage est actif au niveau du noyau. Cette fonction est paramétrée par la variable d'état

`ip_forward` du système de fichiers virtuel `/proc`. La valeur `1` indique que la fonction routage est active dans le noyau :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Comme la politique par défaut sur la chaîne `FORWARD` est `DROP`, aucun paquet ne peut traverser les règles de filtrage et transiter d'une interface vers l'autre. Sans règle supplémentaire, les tests ICMP doivent incrémenter le compteur `DROP` de la chaîne `FORWARD`.

Q91. Quelle est la syntaxe de la commande **iptables** qui autorise le transfert des paquets entrant par l'interface LAN vers l'interface WAN ?

Rechercher la syntaxe des règles correspondant à ce qui a déjà été vu dans la mise au point du jeu de règles communes pour les chaînes `INPUT` et `OUTPUT`. Il faut que tout trafic relatif à une demande enregistrée dans la table de suivi des communications soit accepté. Il faut aussi que les nouveaux paquets entrant par l'interface LAN soient admis et enregistré dans cette table.

On implante deux règles dans la chaîne `FORWARD`. Une première règle pour le trafic relatif à une demande déjà enregistrée et une seconde pour les paquets entrant par l'interface LAN.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~:: POSTROUTING
-A POSTROUTING -o ippp0 -j MASQUERADE
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/min -m conntrack --ctstate NEW -j AC
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~:: FORWARD
-A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i eth0 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o ippp0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Cette configuration ne peut être qualifiée qu'avec un trafic devant transiter entre deux interfaces. Avec la configuration de travaux pratiques proposée, il faut connecter un poste supplémentaire sur le même réseau local que celui de l'interface LAN du routeur d'agence. C'est le trafic réseau initié par ce nouvel hôte réseau qui utilise les deux règles de la chaîne `FORWARD` implantées dans le script ci-dessus.

L'instruction **`# iptables -vL FORWARD`** affiche les compteurs relatifs à la chaîne `FORWARD`. Ces compteurs évoluent lorsqu'un nouveau trafic à destination d'un autre réseau apparaît sur une interface.

Q92. Quelle est la syntaxe de la commande **iptables** qui permet l'administration à distance du routeur d'agence (Spoke) via son interface WAN en utilisant le protocole SSH ? Proposer une configuration qui offre une protection contre les attaques de type «dictionnaire».

Un premier niveau de réponse consiste à admettre les nouvelles demandes de connexions TCP sur le port numéro 22 sur l'interface WAN. On obtient alors le jeu de règles suivant.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: POSTROUTING
-A POSTROUTING -o ippp0 -j MASQUERADE
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/min -m conntrack --ctstate NEW -j AC
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i ippp0 -p tcp --syn --dport 22 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~:: FORWARD
-A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i eth0 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o ippp0 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Du point de vue sécurité, cette configuration n'est pas très satisfaisante. Sachant que toutes les nouvelles demandes de connexion TCP sont acceptées, on ouvre la porte à toutes les attaques de type «dictionnaire».

La section «10.3.19. Correspondance Recent» du Tutoriel iptables décrit précisément les différentes possibilités du module `recent`. En utilisant cette fonctionnalité, on peut remplacer la solution donnée ci-dessus par le jeu de règles suivant qui limite ne nombre de tentatives de connexions à 4 par minute.

```
# grep -v '^# ' /var/lib/iptables/active
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: POSTROUTING
-A POSTROUTING -o ippp0 -j MASQUERADE
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~:: INPUT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/min \
     -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i ippp0 -p tcp --dport 22 -m recent --set --name SSH \
     -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i ippp0 -p tcp --dport 22 -m recent --update --seconds 60 --hitcount 4 \
     --rttl --name SSH -m limit --limit 5/min -j LOG --log-prefix "SSH_brute_force "
-A INPUT -i ippp0 -p tcp --dport 22 \
     -m recent --update --seconds 60 --hitcount 4 --rttl --name SSH -j DROP
#~~~~~~~~~~~~~~~~~~~~~:: FORWARD
-A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i eth0 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~:: OUTPUT
-A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o ippp0 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Avertissement

Dans la copie d'écran ci-dessus, des lignes ont été coupées avec des caractères '\' dans le but d'optimiser l'affichage. Pour rétablir la syntaxe correcte des règles de filtrage, il est possible d'utiliser sed avec une instruction du type **`$ sed '/^[ \-].*\\$/N;s/\\\n *//' dump.iptables`** où le fichier `dump.iptables` contient la copie d'écran ci-dessus.

Non seulement la solution présentée ci-dessus s'est montrée très efficace ces dernières années, mais elle à le mérite de ne pas faire intervenir un outil tiers ; ce qui diminue le coût d'administration.

4.5. Règles de filtrage sur le routeur central (*hub*)

Suivant le cahier des charges fixé, le routeur central doit autoriser le trafic issu du poste client sur son interface WAN et le router sur l'interface LAN.

Tout comme dans le cas du routeur d'agence, on utilise le jeu de règles communes que l'on complète avec les besoins spécifiques à la configuration d'un routeur qui doit faire transiter le trafic d'un interface sur l'autre.

À la différence du routeur d'agence, le routeur central maîtrise l'attribution des adresses IP. On peut donc inclure le contrôle des adresses IP sources dans les règles de filtrage réseau.

Q93. Le jeu de règles communes est-il suffisant pour que le poste se comporte comme un routeur ?

Identifier les conditions nécessaires pour que la fonction routage du noyau soit active et que le filtrage réseau autorise le transit de l'interface WAN vers l'interface LAN.

Non. Il manque au moins 2 conditions pour que le routage et la traduction d'adresses sources soient actifs.

- Pour qu'un paquet soit transmis d'une interface réseau vers une autre, il faut s'assurer que le routage est actif au niveau du noyau. Cette fonction est paramétrée par la variable d'état `ip_forward` du système de fichiers virtuel `/proc`. La valeur `1` indique que la fonction routage est active dans le noyau :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Comme la politique par défaut sur la chaîne `FORWARD` est `DROP`, aucun paquet ne peut traverser les règles de filtrage et transiter d'une interface vers l'autre. Sans règle supplémentaire, les tests ICMP doivent incrémenter le compteur `DROP` de la chaîne `FORWARD`.

Q94. Quelle est la syntaxe de la commande **iptables** qui autorise le transfert des paquets entrant par l'interface WAN vers l'interface LAN ?

Il faut implanter deux règles dans la chaîne `FORWARD`. Une première règle qui correspond à ce qui a déjà été vu dans la mise au point du jeu de règles communes pour les chaînes `INPUT` et `OUTPUT` : tout trafic relatif à une demande enregistrée dans la machine d'état de suivi de communication est accepté. Une seconde règle qui accepte les paquets entrants par l'interface LAN en enregistrant les nouvelles communication dans la même machine d'état. On obtient le jeu de règles suivant :

```
# cat iptables.router
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# T a b l e   F I L T E R
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m limit --limit 5/sec -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# F O R W A R D
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i ippp0 -s 192.168.96.0/20 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q95. Après avoir initié des communications avec les différents protocoles usuels (ICMP, UDP et TCP), relever l'état des communications du routeur d'agence distant avec l'outil iptsate.

Q96. Est-il possible de visualiser à l'aide de l'analyseur réseau wireshark le trafic retour relatif aux requêtes émises par le client WAN ?

Non. Pour que le trafic retour aboutisse sur l'interface du client WAN, il faudrait que la route vers le réseau étendu soit connue du reste de l'Internet.

Q97. Sans protocole de routage dynamique assurant la publication de la route vers le réseau étendu sur l'Internet, quelle est la solution technique à utiliser pour que les postes clients distants puissent accéder aux autres réseaux ?

C'est la traduction d'adresse source qui permet d'utiliser l'adresse IP de l'interface LAN du routeur comme la seule interface visible de l'Internet.

Q98. Quelle est la syntaxe de la règle d'implantation de la traduction d'adresses IP source en sortie de l'interface LAN du routeur central ?

```
# cat iptables.router
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# T a b l e   N A T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# T a b l e   F I L T E R
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m limit --limit 5/sec -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# F O R W A R D
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i ippp0 -s 192.168.96.0/20 -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -o eth0 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

4.6. Règles de filtrage avec identification des protocoles

Pour les deux configurations étudiées ci-avant, aucune distinction de protocole n'a été effectuée. Pour affiner le processus d'enregistrement et de suivi des communications réseau, il est possible de distinguer les caractéristiques de chacun des protocoles autorisés.

4.6.1. Protocole ICMP

Le protocole ICMP décrit dans le document standard RFC792 Internet Control Message Protocol est une pièce essentielle du modèle TCP/IP. Il est principalement utilisé pour rapporter les conditions d'erreurs sur les réseaux. Cependant, les caractéristiques actuelles du protocole ne recommandent aucun contrôle de validation sur les messages d'erreur reçus. Ce protocole laisse donc la porte ouverte à une grande variété d'attaques qui peuvent être effectuées contre TCP à l'aide de messages ICMP. Ces attaques comprennent la réinitialisation de connexion, la réduction du débit de sortie, les dégradations de performances. Toutes ces attaques peuvent être réalisées depuis des réseaux distants, sans la nécessité d'analyser les paquets qui correspondent à la connexion TCP attaquée.

Alors que les implications sur la sécurité du protocole ICMP sont connues depuis longtemps, tous les systèmes n'ont pas mis en application des contrôles de validation sur les messages d'erreur reçus pour réduire au minimum l'impact de ces attaques.

Au niveau du noyau Linux, les responsables du sous-système réseau ont décidé de ne plus traiter les messages de type 4 `source-quench`.

On dispose des ressources suivantes pour débuter l'étude du protocole ICMP.

- La liste des types de messages ICMP est enregistrée par l'Internet Assigned Numbers Authority (IANA) : ICMP parameters.
- Le Tutoriel iptables contient une section complète de présentation des caractéristiques du protocole ICMP.

4.6.2. Règles de filtrage communes à toutes les configurations

- Avec le protocole TCP, il est possible d'identifier les phases d'établissement, de maintien et de libération de connexion.

- Avec le protocole UDP, il n'y a pas grand chose à identifier puisque ce protocole n'est pas orienté connexion et que le nombre des champs de l'en-tête est très limité.

Q99. Quelle est la syntaxe d'appel de la commande **iptables** qui permet d'afficher la liste des messages ICMP et leurs types connus du système de filtrage réseau ?

Après avoir recherché le mot clé `icmp` dans les pages de manuels de la commande **iptables**, on obtient l'instruction suivante : **`# iptables -p icmp -h`**.

Q100. Quelles sont les modifications à apporter sur le jeu de règles communes pour distinguer les messages ICMP les plus importants ?

On considère quatre types de messages ICMP :

- Type de message 8 : `echo-request` : on autorise les nouvelles requêtes ping à raison de 5 par seconde.

- Type de message 0 `echo-reply` : on autorise les réponses pong aux requêtes ping enregistrées dans la machine d'état de suivi de communication.

- Type de message 3 : `destination-unreachable` : on autorise toutes les notifications d'erreur sur la destination relatives à une demande émise à partir de ce système.

- Type de message 11 : `time-exceeded` : on autorise toutes les notification de débordement de temps relatives au trafic émis à partir de ce système.

```
# cat iptables.common.txt
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/s -m conntrack --ctstate NEW -j ACCE
-A INPUT -p icmp --icmp-type echo-reply -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q101. Quelles sont les modifications à apporter sur le jeu de règles communes pour distinguer les conditions sur les connexions TCP ?

On distingue les demandes d'ouverture de connexion avec l'option `--syn` des connexions déjà établies avec l'option inverse `! --syn`.

```
# cat iptables.common.txt
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/s -m conntrack --ctstate NEW -j ACCE
-A INPUT -p icmp --icmp-type echo-reply -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p tcp ! --syn -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p tcp --syn -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

Q102. Quelles sont les modifications à apporter sur le jeu de règles communes pour distinguer le protocole UDP ?

```
# cat iptables.common.txt
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A INPUT -p icmp --icmp-type echo-request -m limit --limit 5/s -m conntrack --ctstate NEW -j ACCE
-A INPUT -p icmp --icmp-type echo-reply -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p tcp ! --syn -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p tcp --syn -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
COMMIT
```

4.7. Documents de référence

4.7.1. IETF & IANA

Types de messages ICMP
: L'Internet Assigned Numbers Authority a enregistré les types de messages ICMP à la page ICMP parameters.

4.7.2. 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.

4.7.3. 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

Configuration des fonctions réseau & compilation du noyau LINUX
: Configuration des fonctions réseau & compilation du noyau Linux : travaux pratiques sur la préparation d'un système routeur GNU/Linux. Compilation d'un noyau LINUX à partir de ses sources après avoir passé en revue ses fonctions réseau et sélectionné les pilotes de périphériques nécessaires.

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.

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.

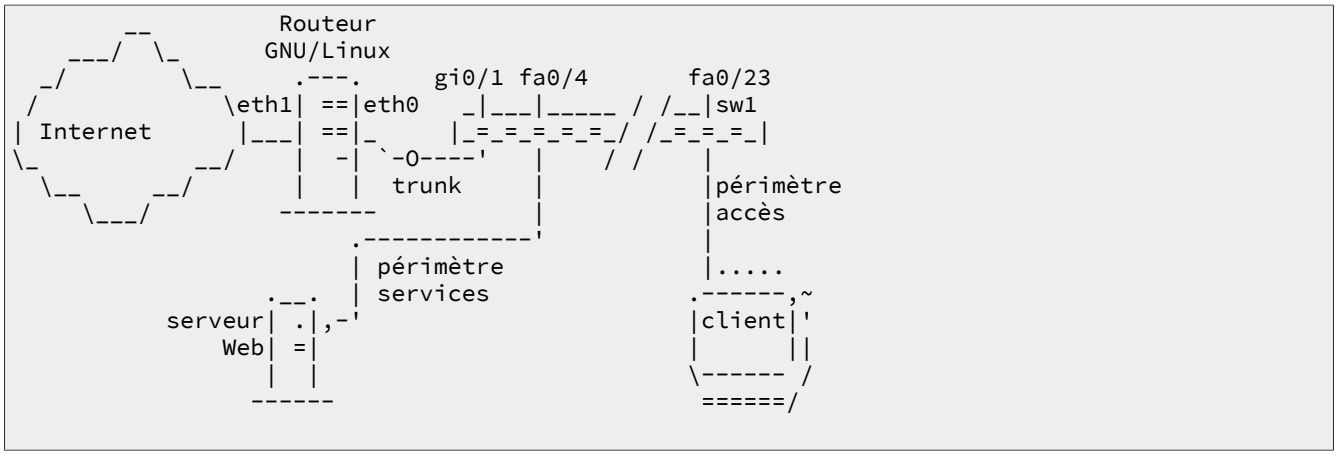
5.2. Etude d'une configuration type

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 IEEE802.1Q. Ils ne nécessitent donc aucune configuration particulière.



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	n° VLAN	Adresse IP
Management	2	`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 IEEE802.1Q 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 2
 switchport mode trunk
 no cdp enable

<snipped/>
!
interface Vlan2
 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      2

Port      Vlans allowed on trunk
Gi0/1       1-4094

Port        Vlans allowed and active in management domain
Gi0/1       1-2,100,200

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/1       1-2,100,200
```

Les règles d'utilisation des trames sans balises IEEE802.1Q sont les suivantes :

- Toute trame appartenant au VLAN natif est émise sans balise IEEE802.1Q sur un port en mode trunk par le commutateur.
- Toute trame reçue sans balise IEEE802.1Q 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 IEEE802.1Q. 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/4.2.0-1-amd64/kernel/net/8021q
```

Le chargement de ce module se fait automatiquement dès qu'une opération relative aux étiquettes IEEE802.1Q 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
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/llc/llc.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/stp.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/mrp.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/garp.ko
insmod /lib/modules/4.2.0-1-686-pae/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 IEEE802.1Q.

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 IEEE802.1Q. 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 IEEE802.1Q 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 IEEE802.1Q 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 2
 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
2    management                       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 une copie 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 -p`** et de valider la valeur de la «clé» `ip_forward`. Si cette valeur est à `1`, le routage est actif au niveau du noyau Linux.

```
# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information.
#
# Refuser la prise en charge des requêtes ARP pour d'autres hôtes
net.ipv4.conf.all.proxy_arp=0

# Ignorer les mauvais messages d'erreurs ICMP
net.ipv4.icmp_ignore_bogus_error_responses=1

# Ignorer les messages de diffusion ICMP
net.ipv4.icmp_echo_ignore_broadcasts=1

# Journaliser les adresses sources falsifiées ou non routables
net.ipv4.conf.all.log_martians=1

# Refuser les adresses sources falsifiées ou non routables
net.ipv4.conf.all.rp_filter=1

# Refuser les messages ICMP redirect
net.ipv4.conf.all.accept_redirects=0

net.ipv4.conf.all.send_redirects=0

# Refuser le routage source
net.ipv4.conf.all.accept_source_route=0

# Activer le routage
net.ipv4.ip_forward=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 les routeurs de l'Internet via un protocole de routage tel qu'OSPF. Consulter le guide Initiation au routage, 3ème partie pour obtenir des exemples complets d'exploitation du protocole de routage OSPF avec les services du logiciel GNU/Linux Quagga.

- 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 IP sortant par l'interface `eth1` voit son adresse IP source réécrite avec l'adresse IP 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).

```
# dmesg |grep ip_
ip_tables: (C) 2000-2002 Netfilter core team
ip_conntrack version 2.1 (4095 buckets, 32760 max) - 248 bytes per conntrack
```

```
# lsmod |grep ip
iptable_filter          3200  0
ipt_MASQUERADE          3712  1
iptable_nat            23516  2 ipt_MASQUERADE
ip_conntrack           44728  2 ipt_MASQUERADE,iptable_nat
ip_tables              22528  3 iptable_filter,ipt_MASQUERADE,iptable_nat
ipv6                  255936  12
```

Le suivi dynamique de communication 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/ip_conntrack` du système de fichiers virtuel du noyau Linux.

```
# cat /proc/net/ip_conntrack
tcp❶      6 431999 ESTABLISHED❷ src=192.168.200.2❸ dst=192.168.200.1❹ \
          sport=33450 dport=22❺ packets=417 bytes=31133 \
          src=192.168.200.1 dst=192.168.200.2 \
          sport=22 dport=33450 packets=306 bytes=111969 [ASSURED] mark=0 use=1
tcp       6 431999 ESTABLISHED src=192.168.200.2 dst=64.236.34.4 \
          sport=33449 dport=80❻ packets=7075 bytes=368009 \
          src=64.236.34.4 dst=aaa.bbb.ccc.7❼ \
          sport=80 dport=33449 packets=9219 bytes=12839148 [ASSURED] mark=0 use=1
```

❶ Protocole de transport utilisé.
❷ État de la connexion TCP.
❸ Adresse IP source. Cette adresse correspond à un poste client appartenant au périmètre Accès.
❹ Adresses IP destination. Dans le premier cas, la communication est interne au réseau du périmètre Accès. Dans le second cas, il s'agit d'une adresse sur l'Internet.
❺ Le numéro de port destination du paquet sortant identifie service Internet utilisé : SSH.
❻ Le numéro de port destination du paquet sortant identifie le service Internet utilisé : HTTP. Plus loin sur la même ligne, on retrouve les adresses IP source et destination attendues.
❼ L'adresse IP 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 IP que les décision d'acheminement sont prises. Dans le cas de la traduction d'adresses par camouflage, l'adresse IP retour est réécrite avec l'adresse IP 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> -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A <CHAINE> -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A <CHAINE> -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A <CHAINE> -p icmp -m conntrack --ctstate 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` :

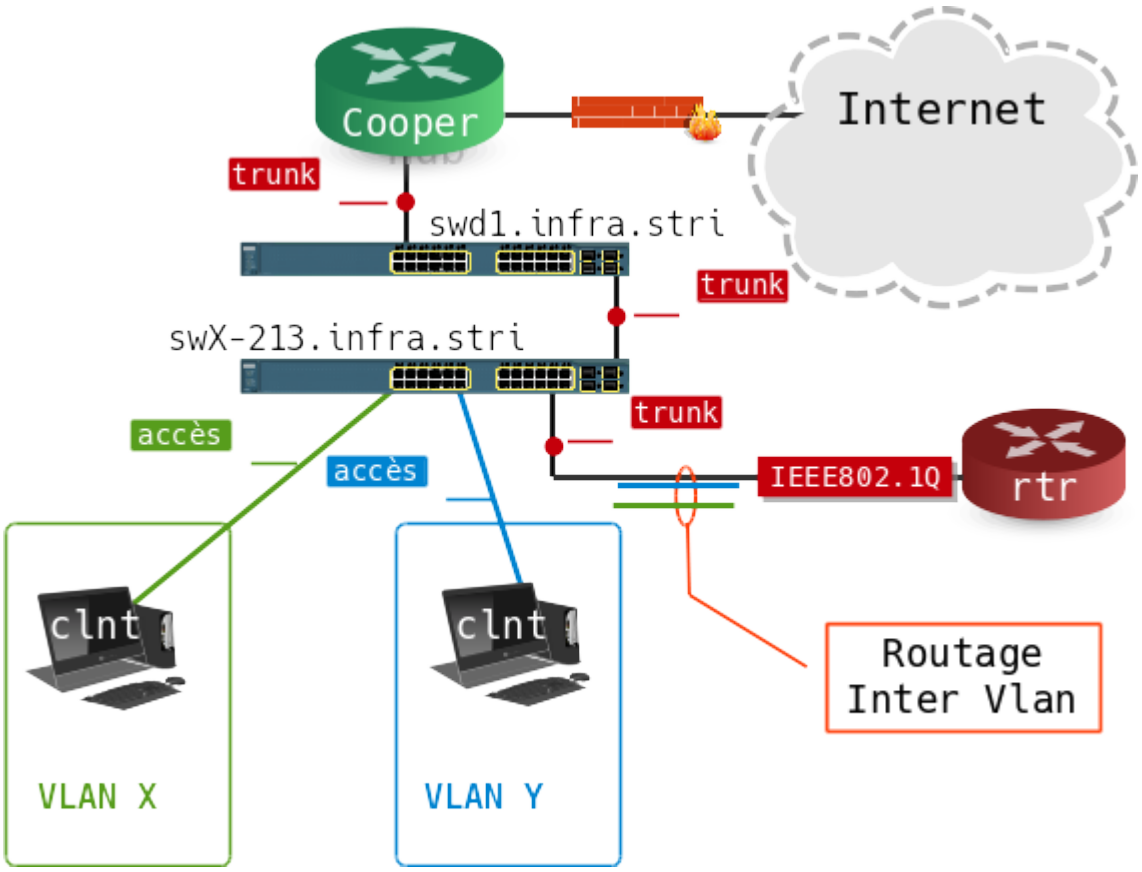
```
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p tcp -m tcp --syn --sport 1024: -m conntrack --ctstate N
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p udp -m udp --sport 1024: -m conntrack --ctstate NEW -j
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p icmp --icmp-type echo-request -m conntrack --ctstate NE
```

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 -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A INPUT -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A INPUT -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -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 eth1 -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
# . services de gestion du commutateur vers le Routeur GNU/Linux
-A INPUT -i eth0 -s 192.168.2.1 -p udp -m multiport --dports 69,123,162,514 -m conntrack --ctstate NEW
# . 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 -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A FORWARD -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A FORWARD -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A FORWARD -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
# . communications des hôtes du périmètre Accès
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p tcp --syn --sport 1024: -m conntrack --ctstate NEW -j ACC
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p udp --sport 1024: -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p icmp --icmp-type echo-request -m conntrack --ctstate NEW
# . 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 des 4 commutateurs : `sw5.infra.stri`, `sw6.infra.stri`, `sw7.infra.stri` et `sw8.infra.stri` 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 `fa0/24` de chaque commutateur qui doit être configuré en mode trunk en utilisant le VLAN natif numéro 1. Le réseau IP correspondant au VLAN numéro 1 a l'adresse : `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 des deux routeurs ayant accès au réseau de Campus.

- Routeur `casper.infra.stri` : `172.16.0.2/20`
- Routeur `cooper.infra.stri` : `172.16.0.4/20`

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

Groupe	Commutateur	Poste	Rôle	VLAN	Réseau
1	`sw5.infra.stri`	alderaan	client	250	`192.168.1.2/25`

Groupe	Commutateur	Poste	Rôle	VLAN	Réseau
		bespin	client	251	192.168.1.130/25
		centares	routeur	1 (natif)	172.16.0.30/20
				250	192.168.1.1/25
				251	192.168.1.129/25
2	sw6.infra.stri	coruscant	client	260	192.168.2.2/25
		dagobah	client	261	192.168.2.130/25
		endor	routeur	1 (natif)	172.16.0.32/20
				260	192.168.2.1/25
				261	192.168.2.129/25
3	sw7.infra.stri	felucia	client	270	192.168.3.2/25
		geonosis	client	271	192.168.3.130/25
		hoth	routeur	1 (natif)	172.16.0.34/20
				270	192.168.3.1/25
				271	192.168.3.129/25
4	sw8.infra.stri	mustafar	client	280	192.168.4.2/25
		naboo	client	281	192.168.4.130/25
		tatooine	routeur	1 (natif)	172.16.0.36/20
				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

Q103. 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 ?

Q104. Toujours dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les postes clients sont raccordés ?

Q105. Encore dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les routeurs sont raccordés ?

Q106. Dans la configuration d'un trunk, qu'est-ce qui distingue un VLAN natif ?

Q107. À partir du tableau des affectations ci-dessus, pourquoi les trois postes d'un groupe ne peuvent-ils pas appartenir au même réseau IP ?

Q108. 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.

Q109. Quel est le paquet qui contient les outils de configuration des interfaces réseau correspondant à chaque VLAN à router ?

Q110. 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 ?

Q111. 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 ?

Q112. 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.

Q113. 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 ?

Q114. Pourquoi les flux réseau capturés contiennent-ils autant de trames STP (Spanning Tree Protocol) ?

Q115. 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

How LAN Switches Work, Document ID: 10607
: Documentation Cisco™ : How LAN Switches Work

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.

Initiation au routage, 3ème partie
: Initiation au routage, 3ème partie : guide complet sur l'utilisation du logiciel Quagga pour transformer un système GNU/Linux en routeur OSPF.

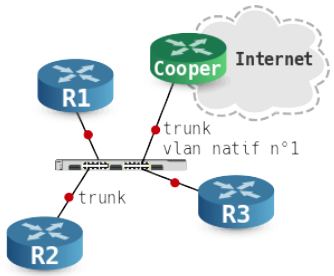
Architecture réseau des travaux pratiques
: Le support Architecture réseau des travaux pratiques présente la topologie physique de la salle de travaux pratiques avec la Disposition des équipements dans l'armoire de brassage ainsi que les configurations par défaut des équipements. On y trouve aussi le 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 avec OSPF

Résumé

L'objectif de ce support de travaux pratiques est de mettre en évidence les caractéristiques de fonctionnement du protocole de routage OSPF. Cette illustration s'appuie sur des liens de type Ethernet et sur l'utilisation des VLANs. Les questions sont présentées comme une introduction pas à pas au protocole de routage OSPF. On débute avec la mise en place d'une topologie réseau type basée sur le routage inter-VLAN, puis on implante les instances de démons de routage.



6.1. Architecture réseau étudiée

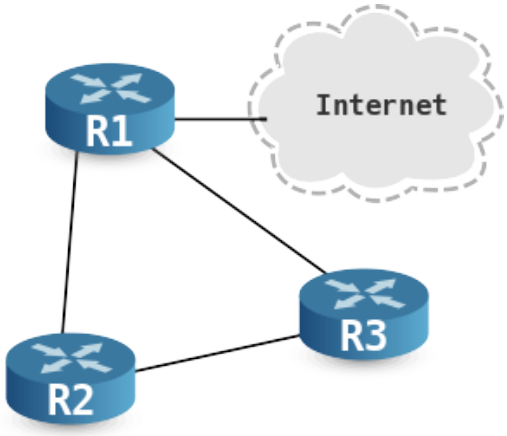
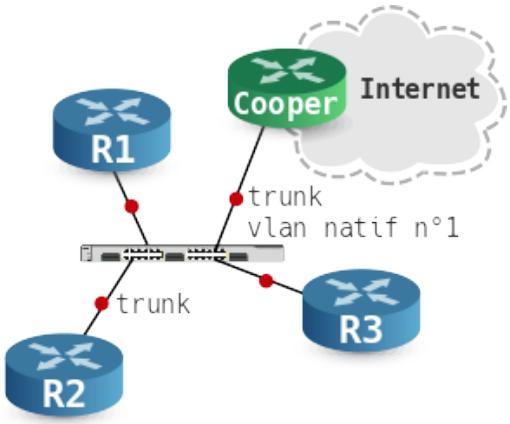
6.1.1. Topologie type

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

Topologie logique
: On retrouve un grand classique dans l'introduction aux protocoles de routage dynamiques : le triangle. Tous les liens sont de type LAN.

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

Tableau 6.1. Topologie type étudiée

Topologie logique	Topologie physique
	

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

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

On profite aussi de cette introduction pour employer une technique très répandue pour ajouter «artificiellement» des entrées de tables de routage en s'appuyant sur des interfaces virtuelles de type dummy équivalentes à des interfaces de boucle locale.

Pour les besoins de rédaction des questions et réponses de ce support, la topologie a été mise en œuvre sur machines virtuelles KVM avec le commutateur Open vSwitch fourni avec le paquet openvswitch-switch. Les éléments de réponse aux questions dépendent donc de cette mise en œuvre. Pour la séance de travaux pratiques «réelle», il convient donc de se conformer strictement au plan d'adressage fourni ci-après.

6.1.2. Plan d'adressage

Comme dans le support sur l'introduction au routage inter-VLAN, le seul point de configuration imposé est le raccordement au réseau d'interconnexion avec les routeurs de l'infrastructure de travaux pratiques.

- `swd2.infra.stri` en salle 211
- `swd1.infra.stri` en salle 213

Le raccordement au commutateur de la couche distribution utilise le port «de numéro le plus élevé» de chaque commutateur de couche accès ; `gi0/2` normalement. Ces liens montants doivent être configurés en mode trunk en utilisant le VLAN natif numéro 1. Le réseau IP correspondant au VLAN numéro 1 a pour adresse : `172.16.0.0/20`

Point important, la lecture de la section «Plan d'adressage» du document Architecture réseau des travaux pratiques donne l'adresse des deux routeurs connectés à l'Internet.

Tableau 6.2. Affectation des rôles, des numéros de VLANs et des adresses IP

Groupe	Commutateur	Poste	Rôle	router-id	VLAN	Interface	Réseau
1	`sw5.infra.stri`	alderaan	R10	0.0.0.10	1 (natif)	eth0	`172.16.1.1/20`
					221	eth0.221	`10.1.21.1/26`
					231	eth0.231	`10.1.31.1/26`
		bespin	R20	0.0.0.20	221	eth0.221	`10.1.21.2/26`
					232	eth0.232	`10.1.32.2/26`
		centares	R30	0.0.0.30	231	eth0.231	`10.1.31.3/26`
					232	eth0.232	`10.1.32.3/26`
2	`sw6.infra.stri`	coruscant	R40	0.0.0.40	1 (natif)	eth0	`172.16.2.1/20`
					241	eth0.241	`10.2.41.1/26`
					251	eth0.251	`10.2.51.1/26`
		dagobah	R50	0.0.0.50	241	eth0.241	`10.2.41.4/26`
					254	eth0.254	`10.2.54.4/26`
		endor	R60	0.0.0.60	251	eth0.251	`10.2.51.5/26`
					254	eth0.254	`10.2.54.5/26`
1	`sw7.infra.stri`	felucia	R70	0.0.0.70	1 (natif)	eth0	`172.16.3.1/20`
					261	eth0.261	`10.3.61.1/26`
					271	eth0.271	`10.3.71.1/26`

Groupe	Commutateur	Poste	Rôle	router-id	VLAN	Interface	Réseau
		geonosis	R80	0.0.0.80	261	eth0.261	10.3.61.6/26
					276	eth0.276	10.3.76.6/26
		hoth	R90	0.0.0.90	271	eth0.271	10.3.71.7/26
					276	eth0.276	10.3.76.7/26
4	sw8.infra.stri	mustafar	R100	0.0.0.100	1 (natif)	eth0	172.16.4.1/20
					281	eth0.281	10.4.81.1/26
					291	eth0.291	10.4.91.1/26
		naboo	R110	0.0.0.110	281	eth0.281	10.4.81.8/26
					298	eth0.298	10.4.98.8/26
		tatooine	R120	0.0.0.120	291	eth0.291	10.4.91.9/26
					298	eth0.298	10.4.98.9/26

Le positionnement des 4 commutateurs est référencé dans le support Architecture réseau des travaux pratiques.

6.2. Préparation des routeurs

La première étape consiste à mettre en place la topologie physique.

- Vérifier l'installation du paquet quagga avant de brasser les postes sur les commutateurs non programmés.

```
$ aptitude search ~iquagga
i   quagga              - BGP/OSPF/RIP routing daemon
```

- Vérifier que la fonction de routage des paquets IPv4 est active au niveau noyau.

```
$ cat /proc/sys/net/ipv4/ip_forward
1
```

Si ce n'est pas le cas, il est possible d'éditer le fichier `/etc/sysctl.conf` pour fixer les valeurs des paramètres de configuration des protocoles de la pile TCP/IP dans le noyau Linux. 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.log_martians = 1
net.ipv4.conf.all.proxy_arp = 0
```

- 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 txqueuelen 10000
 tc qdisc add dev eth0.$vlan root pfifo_fast
 ip link set dev eth0.$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.

- Activer les démons `zebra` et `ospfd` sur chaque routeur en éditant le fichier `/etc/quagga/daemons` et en remplaçant `no` par `yes`.

```
r1:~# grep -v '^#' /etc/quagga/daemons
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

- Créer les fichiers de configuration de base pour les deux démons `zebra` et `ospfd` sur chaque routeur en utilisant les patrons livrés avec le paquet quagga.

```
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/zebra.conf.sample zebra.conf
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/ospfd.conf.sample ospfd.conf
r1:/etc/quagga# chown quagga zebra.conf ospfd.conf
```

- Éditer le patron du fichier de configuration du démon `zebra` en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat zebra.conf
! -*- zebra -*-
!
hostname R1-zebra
password zebra
enable password zebra
!
log file /var/log/quagga/zebra.log
```

- Éditer le patron du fichier de configuration du démon `ospfd` en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat ospfd.conf
! -*- ospf -*-
!
hostname R1-ospfd
password zebra
enable password zebra
!
log file /var/log/quagga/ospfd.log
```

- Compléter la configuration des interfaces dans le démon `zebra` de façon à fixer la bande passante de chaque interface active.

```
r1:/etc/quagga# grep -1 bandwidth zebra.conf
interface eth0
 bandwidth 100000
 ipv6 nd suppress-ra
--
interface eth0.12
 bandwidth 100000
 ipv6 nd suppress-ra
--
interface eth0.13
 bandwidth 100000
 ipv6 nd suppress-ra
```


Note

Contrairement à un routeur «intégré» avec un système d'exploitation dédié, le démon de routage statique n'a pas directement accès aux interfaces matérielles. Or, sur un système GNU/Linux, le débit d'une interface nommée `eth0` peut aller de 10Mbps à 10Gbps. Sans information spécifique du noyau, l'application «service de routage» n'a aucun moyen de connaître le débit exact de l'interface `eth0`. C'est la raison pour laquelle il est nécessaire de

paramétrer manuellement les débits de chaque interface dans la configuration du démon `zebra`.

Avertissement

Ce dernier paramétrage est essentiel dans le calcul des métriques et le fonctionnement du protocole de routage OSPF. Si les calculs de métriques pour les liens actifs sont erronés, le choix des routes à emprunter pour faire transiter le trafic utilisateur entre deux routeurs peut lui aussi être erroné.

6.3. Communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider les communications IP entre chaque routeur et de visualiser les tables de routage déjà connues.

Q116. Quelles sont les opérations à effectuer pour implanter les adresses IP 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 IP de ces sous-interfaces.

Routeur R1

```
r1:~# ip addr add 10.1.12.1/26 brd + dev eth0.12
r1:~# ip addr add 10.1.13.1/26 brd + dev eth0.13
```

Routeur R2

```
r2:~# ip addr add 10.1.12.2/26 brd + dev eth0.12
r2:~# ip addr add 10.1.23.2/26 brd + dev eth0.23
```

Routeur R3

```
r3:~# ip addr add 10.1.23.3/26 brd + dev eth0.23
r3:~# ip addr add 10.1.13.3/26 brd + dev eth0.13
```

Q117. 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 -c2 10.1.12.2
PING 10.1.12.2 (10.1.12.2) 56(84) bytes of data.
64 bytes from 10.1.12.2: icmp_seq=1 ttl=64 time=0.810 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=64 time=0.715 ms

--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.715/0.762/0.810/0.054 ms
```

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

Q118. Quelles sont les opérations à effectuer pour visualiser la table de routage existante d'un routeur au niveau système et au niveau du démon de routage statique zebra ?

Utiliser la commande **ip** pour visualiser la table de routage puis afficher la même table de routage à partir de la connexion au démon zebra avec la commande usuelle du système Cisco™ IOS.

Routeur R1 - niveau système

```
r1:~$ ip route ls
default via 192.0.2.1 dev eth0
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
192.0.2.0/26 dev eth0  proto kernel  scope link  src 192.0.2.10
```

Toutes les routes affichées correspondent à des réseaux IPv4 sur lesquels le routeur est directement connecté via une interface active et correctement configurée.

Routeur R1 - démon zebra

```
R1-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.0.2.1, eth0
C>* 10.1.12.0/26 is directly connected, eth0.12
C>* 10.1.13.0/26 is directly connected, eth0.13
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.0.2.0/26 is directly connected, eth0
```

On retrouve les mêmes informations qu'au niveau système. Une distinction apparaît entre la route par défaut héritée du niveau système qui est repérée avec l'indicateur `K` et les autres routes qui correspondent aux réseaux IPv4 sur lesquels le routeur est directement connecté.

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
```

Routeur R2 - démon zebra

```
R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

C>* 10.1.12.0/26 is directly connected, eth0.12
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

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
```

Routeur R3 - démon zebra

```
R3-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

C>* 10.1.13.0/26 is directly connected, eth0.13
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

Q119. Quelle est l'opération à effectuer pour 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.

6.4. Configuration OSPF de base

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

Q120. Comment peut-on contrôler si le protocole OSPF est actif ou non sur le routeur ?

Une fois connecté au démon `ospfd`, lancer la commande de visualisation globale du protocole.

Cette commande est utilisable sur chacun des trois routeurs.

Dans les informations données dans la copie d'écran ci-dessous, il apparaît que l'algorithme de calcul de topologie n'a pas encore été exécuté. Le protocole OSPF n'est dononc pas encore actif.

```
etu@r1:~$ telnet localhost ospfd
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.24.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
R1-ospfd> en
Password:
R1-ospfd# sh ip ospf
 OSPF Routing Process, Router ID: 192.0.2.10
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm has not been run
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 0
```

Q121. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF et fixer manuellement l'identifiant du routeur ?

Avertissement

Les identifiants à utiliser lors de la séance de travaux pratiques sont donnés dans le Tableau 6.2, « Affectation des rôles, des numéros de VLANs et des adresses IP ».

Toujours à partir de la connexion au démon `ospfd`, on exécute les commandes suivantes sur chacun des trois routeurs en prenant soin d'implanter le bon identifiant.

```
R1-ospfd# conf t
R1-ospfd(config)# router ospf
R1-ospfd(config-router)# router-id 0.0.0.1
R1-ospfd(config-router)# ^Z
R1-ospfd# sh ip ospf
 OSPF Routing Process, Router ID: 0.0.0.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm has not been run
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 0
```

Le choix de codage des identifiants OSPF a pour but d'éviter une confusion avec les adresses des réseaux actifs sur chaque routeur. Dans l'exemple ci-dessus, on se réfère à la Section 6.8.3, « Choix des identifiants de routeur OSPF ».

Q122. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF pour les réseaux IPv4 connus de chaque routeur ?

Dans la configuration du démon `ospfd`, ajouter une entrée de réseau pour chaque lien connu du routeur. La liste des liens connus correspond aux entrées marquées `C` de la table de routage visualisée à partir du démon `zebra`.

Routeur R1

```
R1-ospfd# conf t
R1-ospfd(config)# router ospf
R1-ospfd(config-router)# router-id 0.0.0.1
R1-ospfd(config-router)# network 10.1.12.0/26 area 0
R1-ospfd(config-router)# network 10.1.13.0/26 area 0
```

Routeur R2

```
R2-ospfd# conf t
R2-ospfd(config)# router ospf
R2-ospfd(config-router)# router-id 0.0.0.2
R2-ospfd(config-router)# network 10.1.12.0/26 area 0
R2-ospfd(config-router)# network 10.1.23.0/26 area 0
```

Routeur R3

```
R3-ospfd# conf t
R3-ospfd(config)# router ospf
R3-ospfd(config-router)# router-id 0.0.0.3
R3-ospfd(config-router)# network 10.1.13.0/26 area 0
R3-ospfd(config-router)# network 10.1.23.0/26 area 0
```

Q123. Quelle est la commande qui permet de visualiser l'état des interfaces actives du routeur vis-à-vis du protocole de routage OSPF ?

Les interfaces sont automatiquement activées dès qu'une entrée de réseau est saisie au niveau du démon `ospfd` et que l'adresse IP de l'interface correspond à ce réseau.

C'est la commande **sh ip ospf interface** qui affiche l'état de chaque interface du routeur.

Exemple du routeur R1

```
R1-ospfd# sh ip ospf interface
eth0 is up
  ifindex 2, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth0.12 is up
  ifindex 11, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.12.1/26, Broadcast 10.1.12.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.0.1, Interface Address 10.1.12.1
  Backup Designated Router (ID) 0.0.0.2, Interface Address 10.1.12.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 1.379s
  Neighbor Count is 1, Adjacent neighbor count is 1
eth0.13 is up
  ifindex 12, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.13.1/26, Broadcast 10.1.13.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.0.1, Interface Address 10.1.13.1
  Backup Designated Router (ID) 0.0.0.3, Interface Address 10.1.13.3
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.490s
  Neighbor Count is 1, Adjacent neighbor count is 1
lo is up
  ifindex 1, MTU 65536 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  OSPF not enabled on this interface
```

Q124. À partir des informations affichées dans la question précédente, retrouver l'identifiant de routeur et le type de réseau, repérer et identifier la présence d'un autre routeur sur le même réseau.

Pour chaque interface vue du démon de routage OSPF, repérer les informations relatives au type de réseau et au décomptage des routeurs voisins.

En reprenant l'exemple du routeur `R1` et de son interface `eth0.12`, on relève les informations suivantes.

`Router ID 0.0.0.1, Network Type BROADCAST, Cost: 1`
: L'identifiant OSPF du routeur est `0.0.0.1` ; soit le routeur `R1`. L'interface est connectée sur un réseau Ethernet de type diffusion dans lequel `R1` est un «routeur désigné» (Designated Router) qui sert de référence aux autres routeurs du même périmètre de diffusion ; `R2` dans cet exemple.

`Neighbor Count is 1, Adjacent neighbor count is 1`
: Un routeur OSPF voisin a été reconnu sur ce lien et une adjacence a été établie. Il s'agit du routeur `R2` qui apparaît avec l'identifiant `0.0.0.2` sur une ligne au-dessus.

Q125. Comment peut-on vérifier que l'algorithme SPF du protocole OSPF à été correctement exécuté, que le protocole a convergé et que les entrées de table de routage ont été publiées ?

Visualiser les listes des routeurs voisins puis la liste des routes présentes dans la base de données du démon `ospfd`. Faire la correspondance entre les métriques affichées et les bandes passantes de chaque lien.

En reprenant l'exemple du routeur `R1`, on consulte la liste des voisins OSPF.

```
R1-ospfd# sh ip ospf neighbor

Neighbor ID Pri State        Dead Time Address      Interface         RXmtL RqstL DBsmL
0.0.0.2       1 Full/Backup    39.347s 10.1.12.2    eth0.12:10.1.12.1     0     0     0
0.0.0.3       1 Full/Backup    34.493s 10.1.13.3    eth0.13:10.1.13.1     0     0     0
```

Dans cette liste, on relève la présence des deux autres routeurs :

- `R2` avec l'identifiant `0.0.0.2`
- `R3` avec l'identifiant `0.0.0.3`

L'indicateur d'état `Full/Backup` montre que le dialogue entre les routeurs voisins à convergé vers l'état stable et que le routeur voisin de `R1` joue le rôle de «routeur désigné de secours» (Backup Designated Router).

Ensuite, on trouve de gauche à droite l'adresse IPv4 de l'interface réseau du routeur voisin et l'interface du routeur local ainsi que son adresse IP.

Toujours sur le routeur `R1`, la liste des routes publiées par le démon `ospfd` est donnée par la commande **sh ip ospf route**.

```
R1-ospfd# sh ip ospf route
============ OSPF network routing table ============
N    10.1.12.0/26          [100] area: 0.0.0.0
                           directly attached to eth0.12
N    10.1.13.0/26          [100] area: 0.0.0.0
                           directly attached to eth0.13
N    10.1.23.0/26          [200] area: 0.0.0.0
                           via 10.1.12.2, eth0.12
                           via 10.1.13.3, eth0.13

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

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

Les valeurs notées entre crochets correspondent à la métrique du lien pour joindre le réseau à gauche. Pour le protocole OSPF, le calcul de métrique se fait à partir de l'expression : 10^8 / Bande_Passante_du_lien.

La valeur du numérateur (10^8) correspond à un débit de 100Mbps. À l'époque de la rédaction du standard OSPFv2, ce débit a servi de référence. Aujourd'hui, cette valeur est complètement dépassée. C'est la raison pour laquelle on adapte le calcul de métrique en changeant le coefficient du numérateur. Voir la Section 6.7, « Adaptation de la métrique de lien au débit ».

Les deux premiers réseaux de la table sont joignable via un lien Ethernet à 100Mbps ; soit une métrique de `100`. Le troisième réseau est joignable via deux liens Ethernet à 100Mbps ; d'où la métrique de `200`.

Q126. Quel est le mode d'affichage de la table de routage du système qui offre le plus d'informations ?

Identifier le «lieu de la synthèse» de tous les canaux d'information sur la table de routage d'un routeur.

C'est le démon `zebra` qui offre le plus d'informations sur les entrées de table de routage de l'ensemble du système. Il faut donc se connecter à ce démon pour visualiser toutes les propositions ainsi que les choix retenus pour la table de routage.

```
R1-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.0.2.1, eth0
O   10.1.12.0/26 [110/100] is directly connected, eth0.12, 00:47:45
C>* 10.1.12.0/26 is directly connected, eth0.12
O   10.1.13.0/26 [110/100] is directly connected, eth0.13, 00:47:45
C>* 10.1.13.0/26 is directly connected, eth0.13
O>* 10.1.23.0/26 [110/200] via 10.1.12.2, eth0.12, 00:44:56
  *                        via 10.1.13.3, eth0.13, 00:44:56
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.0.2.0/26 is directly connected, eth0
```

- Les entrées avec le caractère `*` correspondent aux routes actives mémorisées dans la base de commutation des paquets IP ou Forward Information Base (FIB).
- L'entrée notée `K` correspond à une route apprise via la configuration système.
- Les entrées notées `C` correspondent à des routes pour lesquelles il existe une interface sur le routeur. Les métriques de ses routes ont la valeur `0`. Ce sont les routes les plus prioritaires.
- Les entrées notées `O` correspondent aux routes apprises via le protocole OSPF. La métrique de ces routes se décompose en deux parties. La valeur figée à `110` définit le niveau de priorité du protocole OSPF (Administrative Distance) relativement aux autres protocoles de routage. Les valeurs notées après le `/` sont les métriques de liens calculées comme indiqué ci-dessus.

Q127. Comment visualiser la table de routage au niveau système ?

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

Avec la commande **ip**, on voit apparaître les «sources» d'alimentation de la table de routage finale du système : `kernel` et `zebra`.

```
r1:~$ ip route ls
default via 192.0.2.1 dev eth0
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 zebra  metric 200
        nexthop via 10.1.12.2  dev eth0.12 weight 1
        nexthop via 10.1.13.3  dev eth0.13 weight 1
192.0.2.0/26 dev eth0  proto kernel  scope link  src 192.0.2.10
```

6.5. Publication d'une route par défaut via OSPF

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

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

Voici, pour mémoire, une copie de la base de données OSPF avant la mise en place de la publication de route par défaut. On reconnaît les LSAs (Link State Advertisement)) de type 1 et 2 qui correspondent respectivement aux annonces de routeurs et de réseaux.

```
R1-ospfd# sh ip ospf database

       OSPF Router with ID (0.0.0.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
0.0.0.1         0.0.0.1          210 0x80000021 0xebbe 2
0.0.0.2         0.0.0.2         1345 0x8000001e 0xe7ae 2
0.0.0.3         0.0.0.3         1654 0x8000001f 0xcf44 3

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.1.12.1       0.0.0.1         1670 0x80000018 0xda8b
10.1.13.1       0.0.0.1         1120 0x80000018 0xdd86
10.1.23.2       0.0.0.2         1405 0x80000018 0x69ed
```

Q128. Quelle est la condition préalable à respecter pour que le routeur `R1` soit en mesure de publier une route par défaut via son démon OSPF ?

Parmi toutes les méthodes de redistribution de routes dans la protocole OSPF, il en existe une dédiée à l'injection de route par défaut dans une aire normale : Voir Open Shortest Path First (OSPF).

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

Sur la maquette, on valide la présence de la route par défaut comme suit :

```
r1:~$ ip route ls | grep default
default via 192.0.2.1 dev eth0
```

Q129. Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPF ?

Rechercher le mot clé `default` dans la liste des commandes relatives au démon `ospfd`.

C'est la commande **default-information originate** que l'on doit placer dans la configuration de l'instance OSPF.

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

Une fois cette instruction exécutée, la base de données OSPF devient :

```
R1-ospfd# sh ip ospf database

       OSPF Router with ID (0.0.0.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
0.0.0.1         0.0.0.1          784 0x80000021 0xebbe 2
0.0.0.2         0.0.0.2           58 0x8000001f 0xe5af 2
0.0.0.3         0.0.0.3          463 0x80000020 0xcd45 3

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.1.12.1       0.0.0.1          394 0x80000019 0xd88c
10.1.13.1       0.0.0.1         1694 0x80000018 0xdd86
10.1.23.2       0.0.0.2          228 0x80000019 0x67ee

                AS External Link States

Link ID         ADV Router      Age  Seq#       CkSum  Route
0.0.0.0         0.0.0.1          514 0x80000003 0x3d88 E2 0.0.0.0/0 [0x0]
```

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

```
R1-ospfd# sh ip ospf
 OSPF Routing Process, Router ID: 0.0.0.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 39m29s ago
 Last SPF duration 148 usecs
 SPF timer is inactive
 Refresh timer 10 secs
 This router is an ASBR (injecting external routing information)
 Number of external LSA 1. Checksum Sum 0x00003d88
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.0 (Backbone)
   Number of interfaces in this area: Total: 2, Active: 2
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   SPF algorithm executed 16 times
   Number of LSA 6
   Number of router LSA 3. Checksum Sum 0x00029eb2
   Number of network LSA 3. Checksum Sum 0x00021e00
   Number of summary LSA 0. Checksum Sum 0x00000000
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000
```

Le routeur `R1`, est maintenant à la frontière entre deux systèmes autonomes. Il a le rôle ASBR (Autonmous System Border Router) et il est responsable de l'émission des LSAs de type 5 à destination des autres routeurs de l'aire.

Ici, `R1` utilise OSPF et possède une route statique définie au niveau système vers l'Internet. L'instruction donnée ci-dessus assure la redistribution de la route statique vers `R2` et `R3`. Cette route apparaît comme une entrée de type `E2` dans les tables de routage de ces routeurs.

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

Q130. Comment la publication de route par défaut apparaît-elle sur les différents routeurs OSPF ?

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

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

- Vu du démon `ospfd` :

```
R2-ospfd# sh ip ospf route
============ OSPF network routing table ============
N    10.1.3.0/29           [110] area: 0.0.0.0
                           via 10.1.23.3, eth0.23
N    10.1.12.0/26          [10] area: 0.0.0.0
                           directly attached to eth0.12
N    10.1.13.0/26          [20] area: 0.0.0.0
                           via 10.1.12.1, eth0.12
                           via 10.1.23.3, eth0.23
N    10.1.23.0/26          [10] area: 0.0.0.0
                           directly attached to eth0.23

============ OSPF router routing table =============
R    0.0.0.1               [10] area: 0.0.0.0, ASBR
                           via 10.1.12.1, eth0.12

============ OSPF external routing table ===========
N E2 0.0.0.0/0             [10/10] tag: 0
                           via 10.1.12.1, eth0.12
```

- Vu du démmon `zebra` :

```
R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.1.12.1, eth0.12, 2d03h12m
O>* 10.1.3.0/29 [110/110] via 10.1.23.3, eth0.23, 2d03h36m
O   10.1.12.0/26 [110/10] is directly connected, eth0.12, 2d03h32m
C>* 10.1.12.0/26 is directly connected, eth0.12
O>* 10.1.13.0/26 [110/20] via 10.1.12.1, eth0.12, 2d03h31m
  *                       via 10.1.23.3, eth0.23, 2d03h31m
O   10.1.23.0/26 [110/10] is directly connected, eth0.23, 2d03h32m
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

- Vu du niveau système :

```
r2:~$ ip route ls
default via 10.1.12.1 dev eth0.12  proto zebra  metric 10
10.1.3.0/29 via 10.1.23.3 dev eth0.23  proto zebra  metric 110
10.1.12.0/26 dev eth0.12  proto kernel  scope link  src 10.1.12.2
10.1.13.0/26  proto zebra  metric 20
        nexthop via 10.1.12.1  dev eth0.12 weight 1
        nexthop via 10.1.23.3  dev eth0.23 weight 1
10.1.23.0/26 dev eth0.23  proto kernel  scope link  src 10.1.23.2
```

La métrique relevée pour la route par défaut vaut `10`. Il s'agit d'un lien à 1Gbps. Voir la Section 6.7, « Adaptation de la métrique de lien au débit ». Cette métrique ne comprend que le coût du lien allant du routeur `R1` (avec le rôle ASBR) vers l'Internet.

6.6. Ajout de routes fictives

L'introduction de nouvelles entrées fictives dans les tables de routage est une pratique très répandue. Elle permet de qualifier le bon fonctionnement du filtrage réseau ou d'un service Internet sans ajouter de matériel. Cette section décrit justement la mise en place d'un service Web de test.

Q131. Quelles sont les opérations à effectuer pour pouvoir utiliser des interfaces réseau virtuelles de type boucle locale sur un système GNU/Linux ?

Avec un noyau Linux, il est conseillé d'utiliser les interfaces baptisées dummy pour ce genre d'usage. Les opérations à effectuer consistent à charger le module du même nom en mémoire et à appliquer une nouvelle configuration IP.

```
# modprobe -v dummy numdummies=2
insmod /lib/modules/4.2.0-1-686-pae/kernel/drivers/net/dummy.ko numdummies=2
# ip addr ls | grep dummy
5: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default
6: dummy1: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default
```

En prenant l'exemple du routeur `R3`, on créé une nouvelle interface avec l'adresse IP `10.1.3.3/30`. Bien sûr, cette adresse ne correspond à aucun réseau déjà connu des trois routeurs.

```
# ip addr add 10.1.3.3/29 brd + dev dummy0
# ip link set dev dummy0 up
# ip route ls
10.1.3.0/29 dev dummy0  proto kernel  scope link  src 10.1.3.3
10.1.12.0/26  proto zebra  metric 20
        nexthop via 10.1.13.1  dev eth0.13 weight 1
        nexthop via 10.1.23.2  dev eth0.23 weight 1
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
```

Cette nouvelle interface se retrouve automatiquement disponible dans la configuration les deux démons de routages. Dans `zebra` sur `R3`, on obtient les informations suivante sur l'interface `dummy0`.

```
R3-zebra# sh int dummy0
Interface dummy0 is up, line protocol detection is disabled
  index 5 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,NOARP>
  HWaddr: 12:08:a0:09:f3:7d
  bandwidth 100000 kbps
  inet 10.1.3.3/29 broadcast 10.1.3.7
  inet6 fe80::1008:a0ff:fe09:f37d/64
```

On a fixé manuellement à 100Mbps dans `zebra` le débit associé à l'interface `dummy0`. Ce paramètre influence le calcul de métrique dans les bases de données OSPF.

Q132. Quelles sont les opérations à effectuer pour installer un service Web en écoute exclusivement sur l'adresse IP de l'interface `dummy0` ?

Pour aller au plus court, on installe le paquet apache2 et on édite la configuration du service de façon à limiter l'accès à l'adresse IP voulue.

```
# aptitude install apache2
Les NOUVEAUX paquets suivants vont être installés :
  apache2 apache2-bin{a} apache2-data{a} apache2-utils{a} libapr1{a}
  libaprutil1{a} libaprutil1-dbd-sqlite3{a} libaprutil1-ldap{a} liblua5.1-0{a}
  ssl-cert{a}
  0 paquets mis à jour, 10 nouvellement installés, 0 à enlever et 0 non mis à jour.
  Il est nécessaire de télécharger 2 041 ko d'archives. Après dépaquetage,
  6 475 ko seront utilisés.
  Voulez-vous continuer ? [Y/n/?] Y
```

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

```
--- ports.conf.ucf-dist 2015-10-28 09:02:24.885709986 +0100
+++ ports.conf  2015-10-28 09:02:57.623486543 +0100
@@ -2,14 +2,14 @@
 # have to change the VirtualHost statement in
 # /etc/apache2/sites-enabled/000-default.conf

-Listen 80
+Listen 10.1.3.3:80

 <IfModule ssl_module>
-       Listen 443
+       Listen 10.1.3.3:443
 </IfModule>

 <IfModule mod_gnutls.c>
-       Listen 443
+       Listen 10.1.3.3:443
 </IfModule>
```

```
 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Après avoir redémarré l'instance de serveur Web, on vérifie que la nouvelle configuration est bien en place.

```
# /etc/init.d/apache2 restart
<snipped>
# # ss -nlt '( src :80 )'
State       Recv-Q Send-Q   Local Address:Port     Peer Address:Port
LISTEN      0      128           10.1.3.3:80                *:*
```

Q133. Comment ajouter la route correspondant au nouveau réseau `10.1.3.0/29` dans le domaine de routage OSPF ?

Comme dans le cas de la mise en place des autres routes dans la configuration du démon `ospfd`, on ajoute une entrée `network` dans l'instance OSPF du routeur.

Dans le cas du routeur `R3`, la syntaxe est la suivante :

```
R3-ospfd(config)# router ospf
R3-ospfd(config-router)# network 10.1.3.0/29 area 0
```

On vérifie ensuite que cette entrée est bien intégrée dans la base des routes OSPF.

```
R3-ospfd(config)# sh ip ospf route
```

Q134. Comment valider l'accès à ce service Web depuis les autres réseaux ?

En respectant l'ordre des protocoles de la pile TCP/IP, on commence par valider la connectivité au niveau réseau avant de passer à la couche application.

À partir du routeur `R1`, on utilise la séquence suivante :

- Test ICMP au niveau réseau :

```
r1:~$ ping -c2 10.1.3.3
PING 10.1.3.3 (10.1.3.3) 56(84) bytes of data.
64 bytes from 10.1.3.3: icmp_seq=1 ttl=64 time=0.939 ms
64 bytes from 10.1.3.3: icmp_seq=2 ttl=64 time=0.561 ms

--- 10.1.3.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.561/0.750/0.939/0.189 ms
```

- Test HTTP au niveau application :

```
r1:~$ wget -O /dev/null http://10.1.3.3
--2015-10-28 09:24:29--  http://10.1.3.3/
Connexion à 10.1.3.3:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 11104 (11K) [text/html]
```

6.7. Adaptation de la métrique de lien au débit

Par défaut, le calcul de métrique du le protocole OSPF se fait à partir de l'expression : 10^8 / Bande_Passante_du_lien.

Cette règle a été établie à une époque où l'utilisation d'un lien à 100Mbps devait être considéré comme une situation d'exploitation futuriste. Aujourd'hui, les liens à 100Mbps sont monnaie courante et les 100Gbps vont bientôt le devenir.

Cette section traite donc de la configuration des instances de protocole de routage OSPF utilisant des liens avec une capacité supérieure à 100Mbps.

Q135. Quelle est l'instruction à utiliser pour que le calcul de métrique OSPF se fasse sur la base d'un débit de lien à 10Gbps ?

Rechercher le mot clé bandwidth dans la liste des instructions de configuration du démon `ospfd`.

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

```
R2-ospfd(config-router)# auto-cost reference-bandwidth
  <1-4294967>  The reference bandwidth in terms of Mbits per second
```

Voici un extrait de la configuration du routeur `R2` après application d'une nouvelle référence à 10000Mbps soit 10Gbps.

```
router ospf
 ospf router-id 0.0.0.2
! Important: ensure reference bandwidth is consistent across all routers
 auto-cost reference-bandwidth 10000
 network 10.1.12.0/26 area 0.0.0.0
 network 10.1.23.0/26 area 0.0.0.0
```

Comme indiqué dans le commentaire ci-dessus, il est indispensable que tous les routeurs de l'aire OSPF aient la même référence de calcul de métrique. La même instruction doit donc être implantée dans la configuration des trois routeurs.

Q136. Comment modifier le débit d'un lien à 1Gbps ?

Normalement, le débit d'un lien est directement extrait des paramètres de l'interface connectée au lien. Dans le cas d'interface qui n'ont «aucune réalité physique», ce débit peut être attribué arbitrairement par configuration. On doit rechercher dans les options des démons `zebra` et `ospfd` le moyen d'attribuer un débit aux sous-interfaces de VLANs.

Comme indiqué dans la Section 6.2, « Préparation des routeurs », c'est dans la configuration du démon `zebra` que l'on attribue le débit binaire d'une interface réseau. Cette opération est nécessaire compte tenu du fait que le débit d'une même interface Ethernet `eth0` peut varier de 10Mbps à 100Gbps.

Dans toutes les sections précédentes, la bande passante des interfaces était fixée à 100Mbps. Pour tenir compte de la nouvelle référence de calcul de métrique, on applique une nouvelle valeur de bande passante fixée à 1Gbps sur les interfaces Ethernet des trois routeurs. Voici un extrait du fichier de configuration du démon `zebra` du routeur `R2`.

```
interface eth0
 bandwidth 1000000
 ipv6 nd suppress-ra
!
interface eth0.12
 bandwidth 1000000
 ipv6 nd suppress-ra
!
interface eth0.23
 bandwidth 1000000
 ipv6 nd suppress-ra
```

Comme la valeur de base de bande passante est le kilobit par seconde, il faut 1 million de Kbps pour faire 1Gbps ou 10^9bps.

Q137. Comment peut-on identifier le débit d'un lien dans la configuration OSPF ?

Visualiser les paramètres des interfaces réseau depuis la console du démon `ospfd`.

C'est l'instruction **sh ip ospf interface** qui permet d'afficher la valeur de débit attribuée aux interfaces avec le protocole de routage OSPF.

Voici le résultat obtenu sur le routeur `R2`.

```
R2-ospfd# sh ip ospf interface
eth0 is up
  ifindex 2, MTU 1500 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth0.12 is up
  ifindex 3, MTU 1496 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.12.2/26, Broadcast 10.1.12.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.0.2, Interface Address 10.1.12.2
  Backup Designated Router (ID) 0.0.0.1, Interface Address 10.1.12.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 8.604s
  Neighbor Count is 1, Adjacent neighbor count is 1
eth0.23 is up
  ifindex 4, MTU 1496 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.23.2/26, Broadcast 10.1.23.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 0.0.0.3, Interface Address 10.1.23.3
  Backup Designated Router (ID) 0.0.0.2, Interface Address 10.1.23.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 1.144s
  Neighbor Count is 1, Adjacent neighbor count is 1
lo is up
  ifindex 1, MTU 16436 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  OSPF not enabled on this interface
```

Q138. Quel est le coût d'accès au pseudo service Internet (réseau `10.1.3.0/29`) après modification de la référence de calcul de métrique ? Justifier la valeur de métrique obtenue.

À partir des informations de la base de données du démon `ospfd`, faire la somme des métriques de chaque lien entre les deux extrémités en communication.

Il est possible d'obtenir les informations de calcul de métrique soit à partir du démon de routage OSPF `ospfd`, soit à partir du démon de routage statique `zebra`.

- Démon de routage OSPF:

```
R2-ospfd# sh ip ospf route
============ OSPF network routing table ============
N    10.1.3.0/29           [110] area: 0.0.0.0
                           via 10.1.23.3, eth0.23
N    10.1.12.0/26          [10] area: 0.0.0.0
                           directly attached to eth0.12
N    10.1.13.0/26          [20] area: 0.0.0.0
                           via 10.1.12.1, eth0.12
                           via 10.1.23.3, eth0.23
N    10.1.23.0/26          [10] area: 0.0.0.0
                           directly attached to eth0.23

============ OSPF router routing table =============
R    0.0.0.1               [10] area: 0.0.0.0, ASBR
                           via 10.1.12.1, eth0.12

============ OSPF external routing table ===========
N E2 0.0.0.0/0             [10/10] tag: 0
                           via 10.1.12.1, eth0.12
```

- Démon de routage statique :

```
R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.1.12.1, eth0.12, 23:49:36
O>* 10.1.3.0/29 [110/110] via 10.1.23.3, eth0.23, 00:13:23
O   10.1.12.0/26 [110/10] is directly connected, eth0.12, 00:09:46
C>* 10.1.12.0/26 is directly connected, eth0.12
O>* 10.1.13.0/26 [110/20] via 10.1.12.1, eth0.12, 00:09:00
  *                       via 10.1.23.3, eth0.23, 00:09:00
O   10.1.23.0/26 [110/10] is directly connected, eth0.23, 00:09:42
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

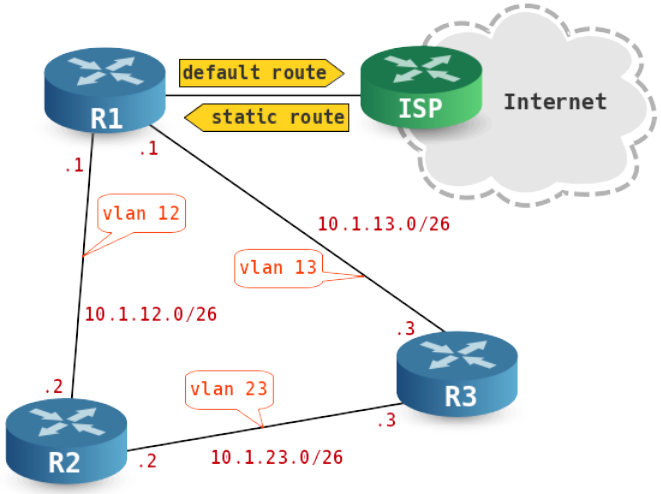
Prenons l'exemple de la métrique du réseau `10.1.3.0/29`. La valeur `110` correspond à la somme des coûts de deux liens : 1 lien à 1Gbps ($10^{10}/10^9 = 10$) + 1 lien à 100Mbps ($10^{10}/10^8 = 100$) = 110.

Le lien à 1Gbps correspond à la liaison entre les routeurs `R2` et `R3`. Le lien à 100Mbps correspond à la liaison entre le routeur `R3` et son interface virtuelle `dummy0` sur laquelle le service Web est en écoute.

6.8. Manipulations sur machines virtuelles

Il est possible de réaliser l'ensemble des manipulations de ce support à l'aide de trois instances de machines virtuelles et du commutateur Open vSwitch présenté dans le guide Virtualisation système et enseignement.

Voici quelques éléments sur la mise en œuvre de cette «infrastructure de travaux pratiques». Dans la figure ci-dessous, le routeur baptisé ISP correspond au système hôte sur lequel les systèmes virtuels sont exécutés.



6.8.1. Préparation du commutateur Open vSwitch

On se réfère au guide Virtualisation système et enseignement dans lequel le lancement du commutateur virtuel Open vSwitch est intégré à la configuration du système hôte. Voici un extrait du fichier `/etc/network/interfaces` relatif à la configuration du commutateur `swd-host`.

```
# The primary network interface
allow-ovs swd-host
iface swd-host inet manual
 ovs_type OVSBridge
 ovs_ports eth0 vlan1

allow-swd-host vlan1
iface vlan1 inet static
 address 192.0.2.29/26
 gateway 192.0.2.1
 ovs_type OVSIntPort
```

```
 ovs_bridge swd-host
 dns-nameservers 192.0.2.1
 up ip link set txqueuelen 10000 dev $IFACE
 up tc qdisc add dev $IFACE root pfifo_fast

iface vlan1 inet6 static
 address 2001:db8:fe00:8175::1d/64
 gateway 2001:db8:fe00:8175::1
 dns-nameservers 2001:db8:fe00:8175::1

allow-swd-host eth0
iface eth0 inet manual
 pre-up ip link set txqueuelen 10000 dev $IFACE
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down
 ovs_bridge swd-host
 ovs_type OVSPort
```

Dans cette configuration, `vlan1` est l'interface principale du système hôte. Elle sert à la fois au routage et à l'accès aux instances de machines virtuelles depuis les autres réseaux.

La configuration de départ du commutateur sur le système hôte est la suivante :

```
$ sudo ovs-vsctl show
0fb1e80b-37cf-4dce-9a19-17b9fb989610
    Bridge swd-host
        Port "eth0"
            Interface "eth0"
        Port "vlan1"
            Interface "vlan1"
                type: internal
        Port swd-host
            Interface swd-host
                type: internal
    ovs_version: "2.3.0"
```

Partant de cette configuration de départ, on créé 3 cordons de brassage ; un par routeur. On dispose d'un script pour répéter les opérations de base.

```
#!/bin/bash

for patchtap in $*
do
        if [ -z "$(ip addr ls | grep $patchtap)" ]; then
                echo setting $patchtap up
                sudo ip tuntap add mode tap dev $patchtap group kvm
                sudo ip link set dev $patchtap up
        else
                echo $patchtap is already there!
        fi
done
```

Avec cet outil, la création des cordons s'effectue comme suit :

```
$ sh ./patchcables.sh tap1 tap2 tap3
```

Maintenant que les cordons sont disponibles, il faut les brasser sur le commutateur `swd-host`. L'opération ci-dessous créé trois nouveaux ports sur le commutateur.

```
$ for port in tap1 tap2 tap3; do sudo ovs-vsctl add-port swd-host $port; done
```

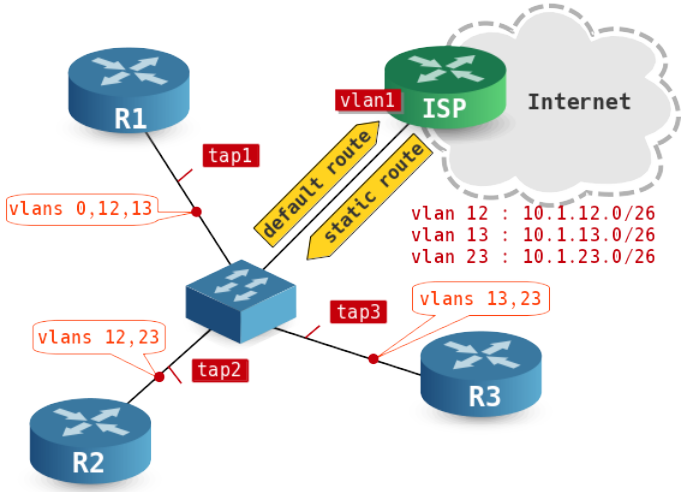
Pour respecter la topologie physique proposée dans l'énoncé, ces trois ports doivent être configurés en mode trunkde façon à ce que chaque routeur puisse router les VLANs qui le concernent. Avec Open vSwitch, les paramètres de configuration sont les suivants :

```
$ sudo ovs-vsctl set port tap1 vlan_mode=trunk trunks=0,12,13
$ sudo ovs-vsctl set port tap2 vlan_mode=trunk trunks=12,23
$ sudo ovs-vsctl set port tap3 vlan_mode=trunk trunks=13,23
```

Note

La particularité du commutateur Open vSwitch est d'utiliser le numéro 0 comme VLAN par défaut. Dans notre cas, ce VLAN 0 devient le VLAN natif dans lequel circulent les trames sans étiquettes IEEE802.1Q.

On aboutit à la représentation graphique suivante :



La table de correspondance entre VLAN et liaison devient :

VLAN numéro 0	Liaison R1 <-> ISP
VLAN numéro 12	Liaison R1 <-> R2
VLAN numéro 13	Liaison R1 <-> R3
VLAN numéro 23	Liaison R2 <-> R3

Une fois que les opérations des sections suivantes auront été effectuées, la consultation de la table CAM du commutateur donnera les informations suivantes :

```
$ sudo ovs-appctl fdb/show swd-host
[sudo] password for latu:
 port  VLAN  MAC                Age
   11    12  ba:ad:ca:fe:00:02   10
   12    23  ba:ad:ca:fe:00:03    9
   12    13  ba:ad:ca:fe:00:03    9
   10    13  ba:ad:ca:fe:0d:01    9
   10    12  ba:ad:ca:fe:0c:01    4
   11    23  ba:ad:ca:fe:00:02    1
    1     0  de:56:80:40:1d:ed    0
    2     0  42:d9:5d:6c:89:d8    0
```

6.8.2. Préparation des routeurs

Sachant que l'on dispose d'une image master sur laquelle est installée un système Debian GNU/Linux de base, on créé 3 images différentielles qui vont correspondre aux instances de routeurs. Voici un exemple de suite de commandes pour la mise en place des images.

```
:~$ mkdir -p ~/vm/ospf
:~$ cd vm
:~/vm$ wget http://www.stri/vm/vm0-debian-testing-i386-base.qed
<snipped/>
:~/vm$ cd ospf
:~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r1.qed
:~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r2.qed
:~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r3.qed
```

Le code du script de création de fichier image différentiel est fourni dans le guide Virtualisation système et enseignement.

Ensuite, on créé un nouveau script shell de lancement des instances de «routeurs» dans lequel on fixe les paramètres d'initialisation de ces mêmes «routeurs».

Avertissement

Ce script ne doit être lancé qu'après l'initialisation du commutateur virtuel pour que le brassage des routeurs sur les ports du commutateur puisse se faire correctement.

```
$ cd ~/vm/ospf
$ cat ospf-startup.sh
#!/bin/bash

../scripts/ovs-startup.sh r1.qed 512 1

../scripts/ovs-startup.sh r2.qed 512 2

../scripts/ovs-startup.sh r3.qed 512 3
```

6.8.3. Choix des identifiants de routeur OSPF

Pour toute instance du protocole de routage OSPF, le choix de l'identifiant se fait dans l'ordre suivant :

1. Adresse IPv4 définie à l'aide de l'instruction **router-id**.
2. Adresse IPv4 la plus élevée parmi toutes les interfaces de boucle locale
3. Adresse IPv4 la plus élevée parmi toutes les interfaces actives

La liste de identifiants utilisés pour les trois routeurs qui servent à la rédaction des réponses types est la suivante.

R1	`0.0.0.1`
R2	`0.0.0.2`
R3	`0.0.0.3`

6.8.4. Table de routage du système hôte

Pour que les réseaux de l'aire OSPF puissent communiquer avec le système hôte et l'Internet, il est nécessaire de compléter la table de routage du système hôte. Dans ce contexte le système hôte joue le rôle d'un routeur central et la technique usuelle employée pour répondre au besoin d'interconnexion consiste à implanter une route statique avec un «super réseau» qui rassemble tous les réseaux de l'aire en une seule entrée.

L'aire OSPF étudiée contient quatre réseaux :

- 10.1.3.0/29 - réseau fictif ajouté sur R3,
- 10.1.12.0/26 - réseau correspondant au lien entre R1 et R2,
- 10.1.13.0/26 - réseau correspondant au lien entre R1 et R3,
- 10.1.23.0/26 - réseau correspondant au lien entre R2 et R3,

L'utilisation de l'outil ipcalc permet de vérifier qu'un masque de 19 bits permet d'englober ces quatre réseaux en une seule entrée.

```
$ ipcalc 10.1.0.0/19
Address:   10.1.0.0             00001010.00000001.000 00000.00000000
Netmask:   255.255.224.0 = 19   11111111.11111111.111 00000.00000000
Wildcard:  0.0.31.255           00000000.00000000.000 11111.11111111
=>
Network:   10.1.0.0/19          00001010.00000001.000 00000.00000000
HostMin:   10.1.0.1             00001010.00000001.000 00000.00000001
HostMax:   10.1.31.254          00001010.00000001.000 11111.11111110
Broadcast: 10.1.31.255          00001010.00000001.000 11111.11111111
Hosts/Net: 8190                  Class A, Private Internet
```

On complète donc la table de routage du système hôte avec l'instruction suivante :

```
$ sudo ip route add 10.1.0.0/19 via 192.0.2.10
```

Une fois cette nouvelle entrée de la table de routage du système hôte en place, on peut valider l'accessibilité des réseaux de l'aire en testant le service Web factice implanté sur le routeur R3 depuis le système hôte.

```
$ wget -O /dev/null http://10.1.3.3
--2015-10-30 15:35:37--  http://10.1.3.3/
Connexion à 10.1.3.3:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 11104 (11K) [text/html]
Sauvegarde en : « /dev/null »
```

Le chargement de la page depuis le serveur web factice fonctionne bien !

6.9. 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

Initiation au routage, 1ère partie
: L'article Initiation au routage, 1ère partie introduit l'utilisation de quagga et de son premier démon baptisé `zebra`. Ce démon permet de mettre en place un routage statique associé à la table de routage définie dans la configuration du système.

Initiation au routage, 3ème partie
: L'article Initiation au routage, 3ème partie introduit l'utilisation du protocole OSPF sur plusieurs aires. Ce n'est pas l'objectif de ce support qui se limite au routage dynamique dans un système autonome unique ; donc une aire unique.

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.

Open Shortest Path First (OSPF)
: La page consacrée au protocole sur le site Cisco™ : Open Shortest Path First (OSPF) regroupe des ressources importantes sur la conception d'architecture réseau utilisant ce protocole.

  On peut aussi citer les supports de formation Cisco Networking Academy qui sont d'une grande qualité sur l'initiation à l'utilisation des protocoles de routage. Malheureusement, ce ne sont pas des documents libres d'utilisation.

Virtualisation système et enseignement
: Le support Virtualisation système et enseignement présente la solution de virtualisation intégrée au noyau Linux : KVM. Associée au commutateur Open vSwitch, cette solution permet de construire des maquettes de travaux pratiques très complètes en offrant de nombreuses fonctions réseau «réelles» dont une table Content-addressable memory.

CHAPITRE 7

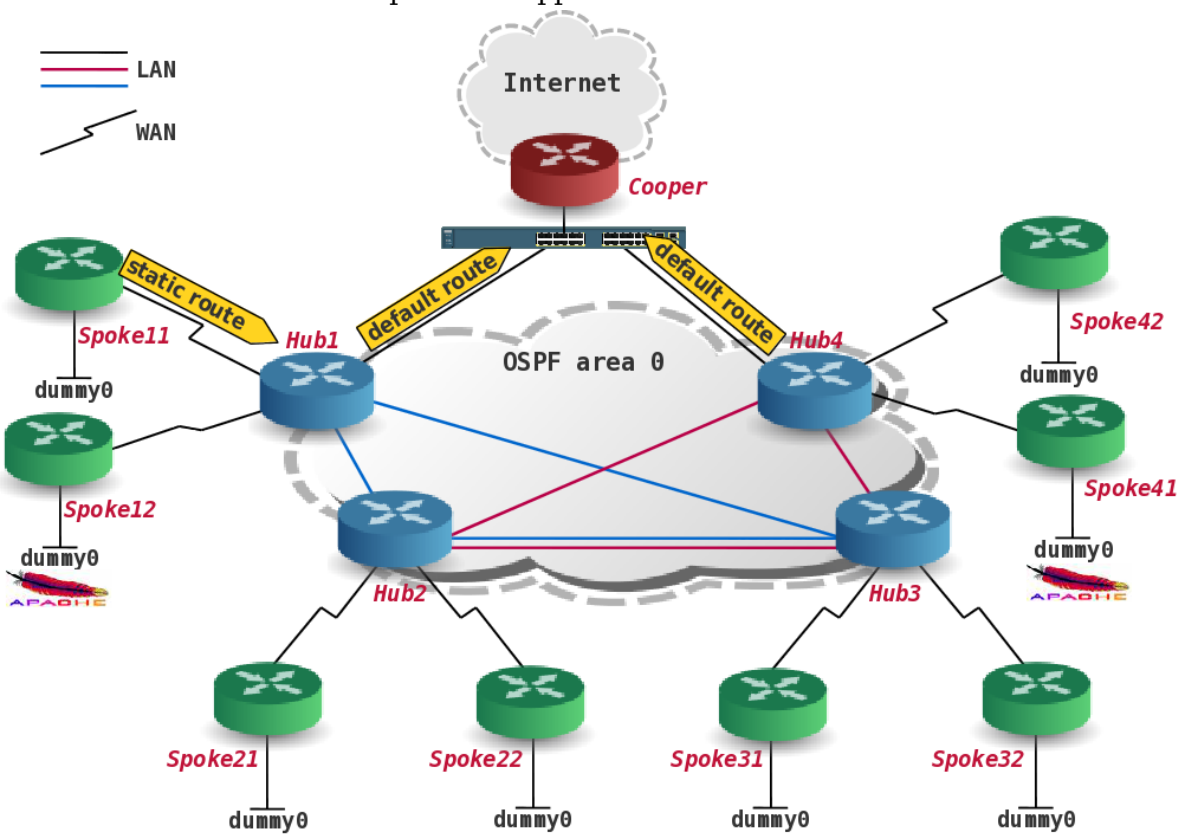
Étude de cas de synthèse sur l'interconnexion LAN/WAN

Résumé

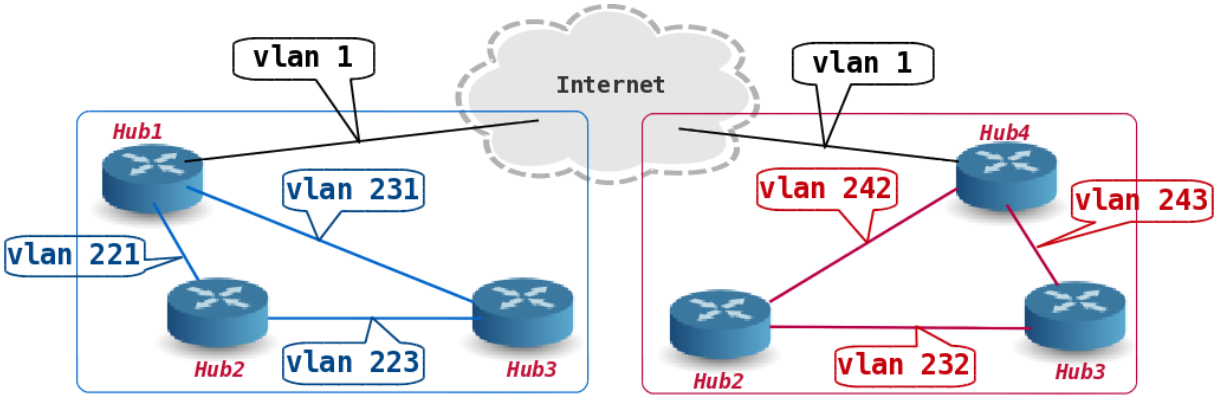
L'objectif de cette étude de cas est de faire la synthèse sur l'ensemble du cycle de travaux pratiques sur le thème de l'interconnexion réseau LAN/WAN. Côté réseaux étendus, on retrouve la configuration des accès via PPP sur trames «HDLC synchrones» (RNIS) et le filtrage avec et sans traduction d'adresses. Côté réseaux locaux, on reprend le routage inter-VLAN avec le protocole de routage dynamique OSPF.

7.1. Topologies réseaux

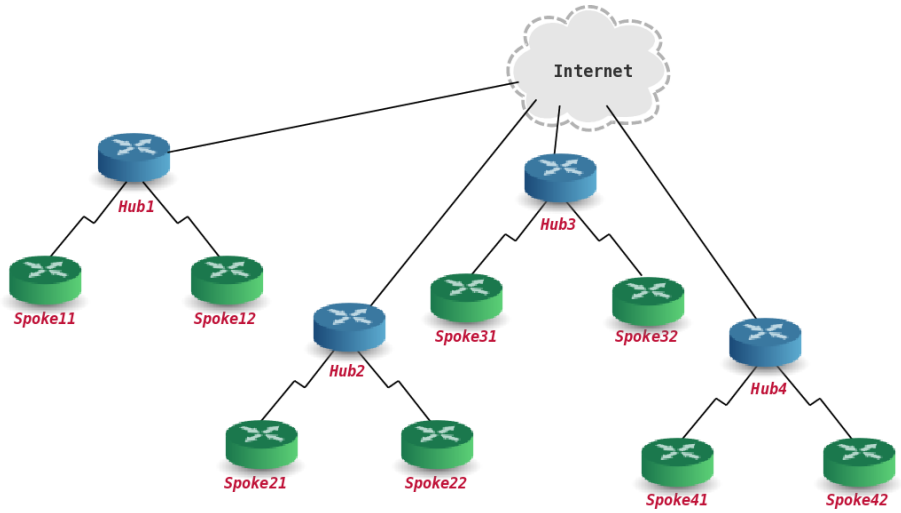
La topologie logique globale se présente comme une associations de topologies triangulaires LAN et WAN. L'ensemble des routeurs présentés appartiennent à une seule aire OSPF.



Les deux sous-topologies LAN reprennent l'architecture étudiée dans le support Routage dynamique avec OSPF. 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 le support Topologie Hub & Spoke avec le protocole PPP. 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 Web sur les interfaces fictives (`dummy`) depuis un routeur Spoke vers n'importe quel autre routeur Spoke sans recourir une seule fois à la traduction d'adresses IP.

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.

Tableau 7.1. Affectation des rôles, des numéros de bus S0 et des adresses IP

Groupe	Poste	Rôle	Bus S0	N° Tél.	Interface	Réseau/Authentification
1	centares	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`
			-	-	dummy0	`10.106.0.1/29`
	alderaan	Spoke12	S0.2	107	ippp0	`etu_s12 / Sp0k3.12`
			-	-	dummy0	`10.107.0.1/29`
2	endor	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`
			-	-	dummy0	`10.110.0.1/29`
	coruscant	Spoke22	S0.4	111	ippp0	`etu_s22 / Sp0k3.22`
			-	-	dummy0	`10.111.0.1/29`
3	hoth	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`
			-	-	dummy0	`10.114.0.1/29`
	felucia	Spoke32	S0.6	115	ippp0	`etu_s32 / Sp0k3.32`

Groupe	Poste	Rôle	Bus S0	N° Tél.	Interface	Réseau/Authentification
			-	-	dummy0	`10.115.0.1/29`
4	naboo	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`
	mustafar	Spoke41	S0.8	118	ippp0	`etu_s41 / Sp0k3.41`
			-	-	dummy0	`10.118.0.1/29`
	tatooine	Spoke42	S0.8	119	ippp0	`etu_s42 / Sp0k3.42`
			-	-	dummy0	`10.119.0.1/29`

7.3. Plan d'adressage LAN

Les seules connexions physiques imposées les liens montants vers l'infrastructure de travaux pratiques. Ces connexions utilisent les ports `fa0/24` des commutateurs indiqués dans le tableau ci-dessous. Les ports doivent être configurés en mode trunk avec le VLAN natif numéro 1. Le réseau IP. Le réseau IP correspondant au VLAN numéro 1 a l'adresse : `172.16.0.0/20`.

Comme indiqué dans la section «Plan d'adressage» du document Architecture réseau des travaux pratiques, l'adresse du routeur Cooper est `172.16.0.4/20`. Le brassage peut se faire sur un même commutateur : `sw5.infra.stri` par exemple.

Tableau 7.2. Affectation des rôles, des numéros de VLANs et des adresses IP

Groupe	Poste	Rôle	router-id OSPF	VLAN	Interface	Réseau
1	centares	Hub1	`0.0.0.10`	1 (natif)	eth0	`172.16.1.1/20`
				221	eth0.221	`10.0.21.1/26`
				231	eth0.231	`10.0.31.1/26`
2	endor	Hub2	`0.0.0.20`	221	eth0.221	`10.0.21.2/26`
				242	eth0.242	`10.0.42.2/26`
				223	eth0.223	`10.0.23.2/26`
				232	eth0.232	`10.0.32.2/26`
3	hoth	Hub3	`0.0.0.30`	243	eth0.243	`10.0.43.3/26`
				231	eth0.231	`10.0.31.3/26`
				223	eth0.223	`10.0.23.3/26`
				232	eth0.232	`10.0.32.3/26`
4	naboo	Hub4	`0.0.0.40`	1 (natif)	eth0	`172.16.4.1/20`
				242	eth0.242	`10.0.42.4/26`
				243	eth0.243	`10.0.43.4/26`

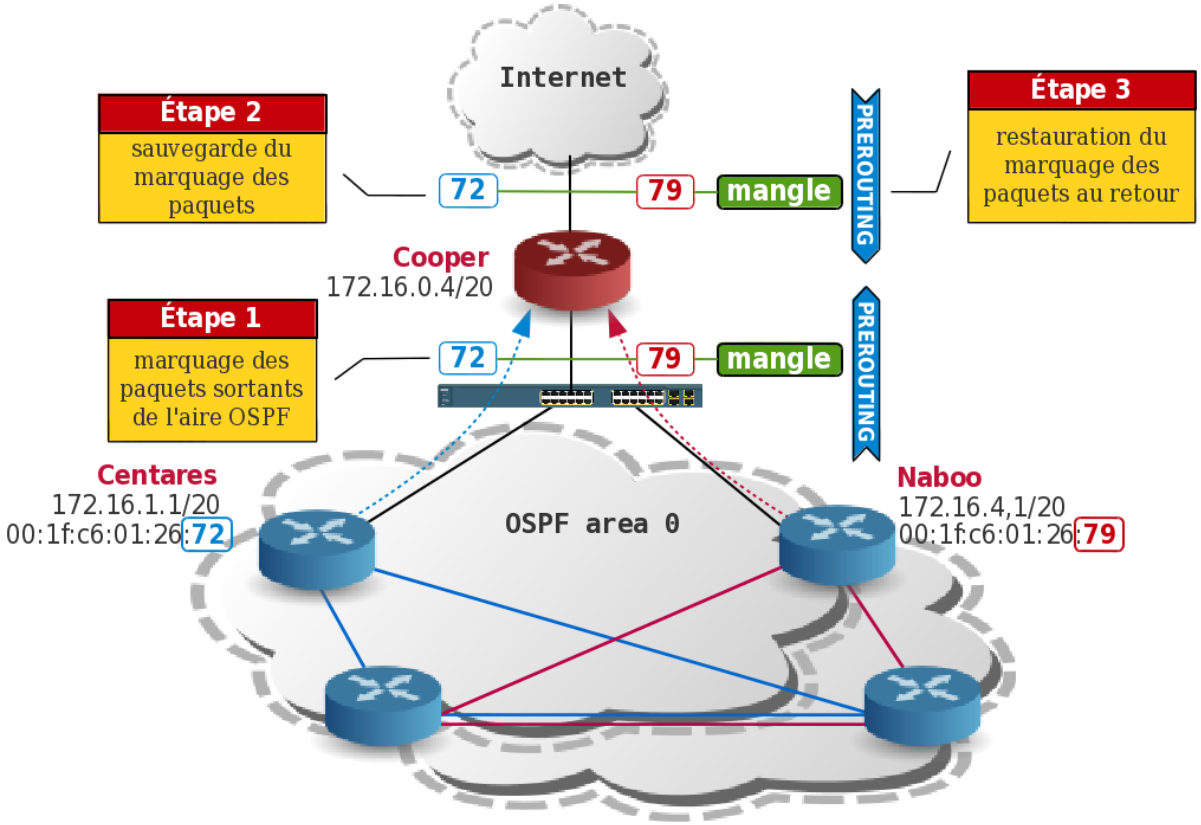
7.4. Interconnexion avec deux routeurs de bordure OSPF

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 172.16.1.1 table centares
# ip route add 10.0.32.0/20 via 172.16.1.1 table centares
# ip route add default dev bond0 table centares
```

```
# ip route add 10.0.16.0/20 via 172.16.4.1 table naboo
# ip route add 10.0.32.0/20 via 172.16.4.1 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'une marque.

```
# iptables -t mangle -A PREROUTING -i bond0.1 -m mac --mac-source 00:1f:c6:01:26:72 -j MARK --set-ma
# iptables -t mangle -A PREROUTING -i bond0.1 -m mac --mac-source 00:1f:c6:01:26:79 -j MARK --set-ma
# iptables -t mangle -A PREROUTING -i bond0.1 -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 la marque 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.