Technologie de l'Internet
Module M2103

Travaux pratiques

IUT de Villetaneuse — R&T — DUT R&T

Camille Coti
camille.coti@iutv.univ-paris13.fr

1 Routage statique sur un réseau simple

Le but de ce TP est de mettre en place un réseau constitué de plusieurs sous-réseaux et d'un point de sortie vers Internet, et d'assurer le routage des paquets entre ces réseaux et vers l'extérieur.

Ce TP se déroule grâce au simulateur de réseaux marionnet disponible gratuitement sous licence GPL, voir `www.marionnet.org`. Rappel : le mot de passe de root d'une machine virtuelle est `root`.

1 Plan d'adressage et routage statique entre deux machines

Créez le réseau suivant :
- Un routeur
- Deux machines reliées directement à ce routeur
- Une *gateway* entre le routeur et le monde extérieur

Lors de la création des machines, choisissez l'image Debian. Dans la configuration du routeur, cochez la case "Show Unix terminal".

Le réseau que vous obtenez ressemble à la figure 1.

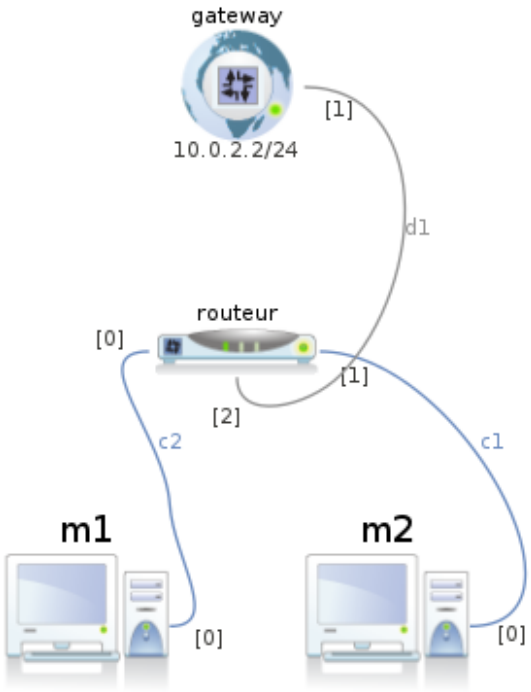


FIGURE 1 – Réseau à mettre en place pour le début du TP

Nous allons partager notre réseau en deux sous-réseau :
- Le réseau 10.0.10.0/24, qui contient une machine
- Le réseau 10.0.20.0/24, qui contient l'autre machine

1. Proposez une adresse pour les interfaces du routeur (y compris celle qui est reliée à la gateway) et celles des machines.

2. Quels sont les masques des trois réseaux (en comptant celui de la gateway) de notre TP ? Pourquoi ce choix assure-t-il l'indépendance de nos trois sous-réseaux ?
3. Démarrez le routeur et la gateway. Configurez l'interface réseau du routeur qui est reliée à la gateway. Vous pourrez utiliser `ping` pour tester si votre configuration est correcte.
4. Démarrez les deux machines. Configurez leurs interfaces réseaux ainsi que les interfaces correspondantes sur le routeur.
5. Affichez la table de routage du routeur avec la commande `route`. Pourquoi n'est-elle pas correcte ?
6. Supprimez les entrées incorrectes de la table de routage et ajoutez les entrées correspondant à vos trois sous-réseaux ainsi qu'une route par défaut (bien sûr, vers la gateway).
7. Par quelle passerelle devra passer la machine `m1` pour communiquer avec la machine `m2` ? Avec l'extérieur ? Quelle est alors l'adresse à indiquer dans la table de routage comme passerelle par défaut ?
8. Lorsque la machine `m1` communique avec l'extérieur, par quels éléments du réseau transitent ses paquets ?
9. Configurez les tables de routage des deux machines pour qu'elles puissent communiquer entre elles et avec l'extérieur.

   Le réseau local est bien là (avec le bon masque), seule la passerelle par défaut a besoin d'être ajoutée.
10. Testez votre configuration avec l'utilitaire `ping` entre tous les éléments de votre réseau. La passerelle ne laisse pas passer les messages ICMP vers l'extérieur : vous ne pourrez donc pas l'utiliser pour tester si les paquets sont bien routés vers l'extérieur. Vous pouvez utiliser `wget lipn.fr` en remplaçant le contenu du fichier `/etc/resolv.conf` par la ligne suivante :

```
nameserver 10.0.2.3
```

2 Extension du réseau

Modifiez votre réseau de la façon suivante :
- Ajoutez un deuxième routeur
- Les deux routeurs sont interconnectés par un switch
- Une nouvelle machine est connectée directement au deuxième routeur
- Une machine est connectée au switch

Le réseau que vous obtenez ressemble à la figure 2.

On ajoute ici deux sous-réseaux :
- 10.0.30.0/24 pour le réseau relié au switch (auquel sont reliés une machine et les deux routeurs)
- 10.0.40.0/24 pour la machine directement reliée au nouveau routeur.

1. Proposez des adresses pour les interfaces des routeurs et celles des machines ajoutées.
2. Allumez les nouveaux éléments du réseau et configurez les interfaces réseaux des deux nouvelles machines, du nouveau routeur et la quatrième interface du premier routeur.
3. Lorsque les machines des réseaux 10.0.10.0 et 10.0.20.0 communiquent avec la machine du réseau 10.0.30.0, par quels élements du réseau transitent les paquets ?
4. Lorsque les machines des réseaux 10.0.10.0 et 10.0.20.0 communiquent avec la machine du réseau 10.0.40.0, par quels élements du réseau transitent les paquets ?
5. Lorsque la machine du réseau 10.0.30.0 communique avec la machine du réseau 10.0.40.0, par quels élements du réseau transitent les paquets ?

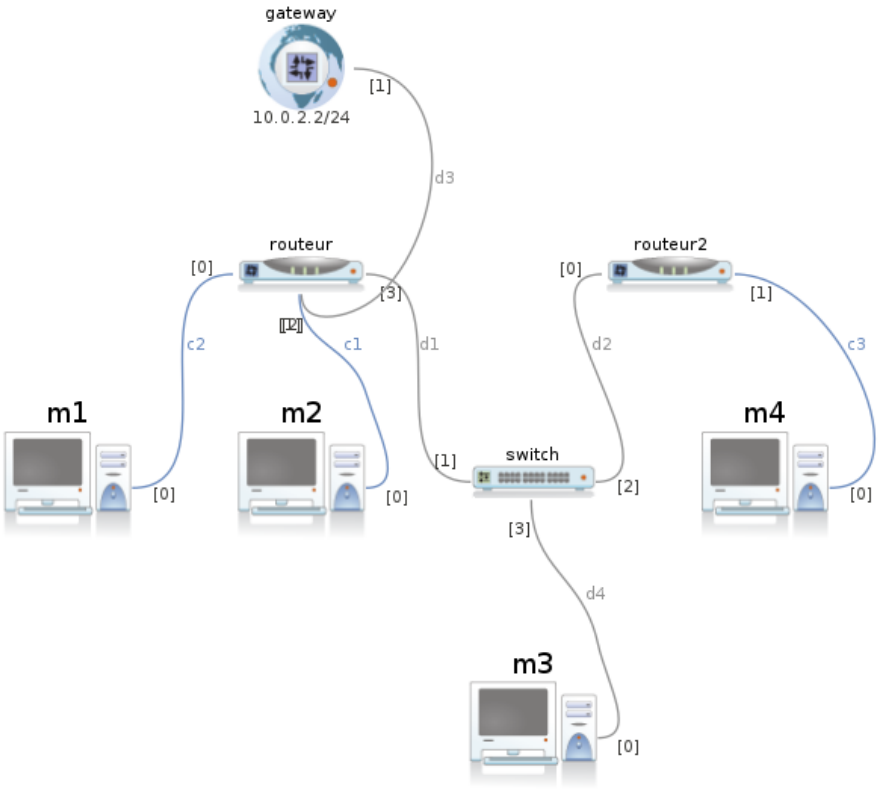


FIGURE 2 – Réseau à mettre en place pour la deuxième partie du TP

6. Lorsque la machine du réseau 10.0.30.0 communique avec les machines des réseaux 10.0.10.0 et 10.0.20.0, par quels élements du réseau transitent les paquets ?
7. Lorsque la machine du réseau 10.0.40.0 communique avec l'extérieur, par quels élements du réseau transitent les paquets ?
8. Configurez les tables de routage des routeurs pour prendre en compte les nouveaux réseaux.
9. Configurez l'interface réseau et la table de routage de la machine `m4`.
10. Configurez l'interface réseau et la table de routage de la machine `m3`.
11. Avec `ping`, vérifiez que toutes les machines peuvent communiquer les unes avec les autres. En modifiant le fichier `/etc/resolv.conf` sur `m3` et `m4`, vérifiez que les nouvelles machines peuvent communiquer avec l'extérieur.
12. Avec l'utilitaire `traceroute`, on obtient la liste des hôtes (routeurs, par exemple) traversés lorsqu'un paquet transite entre deux hôtes. Utilisez-le pour examiner le chemin parcouru entre `m4` et `m1`.

2 Routage statique sur un réseau comportant plusieurs routeurs

Le but de ce TP est de mettre en place un réseau constitué de plusieurs sous-réseaux et d'un point de sortie vers Internet, et d'assurer le routage des paquets entre ces réseaux et vers l'extérieur.

Ce TP se déroule grâce au simulateur de réseaux marionnet disponible gratuitement sous licence GPL, voir `www.marionnet.org`. Rappel : le mot de passe de root d'une machine virtuelle est `root`.

1 Plan d'adressage

Le réseau que vous allez construire au cours de ce TP ressemble à la figure 3.

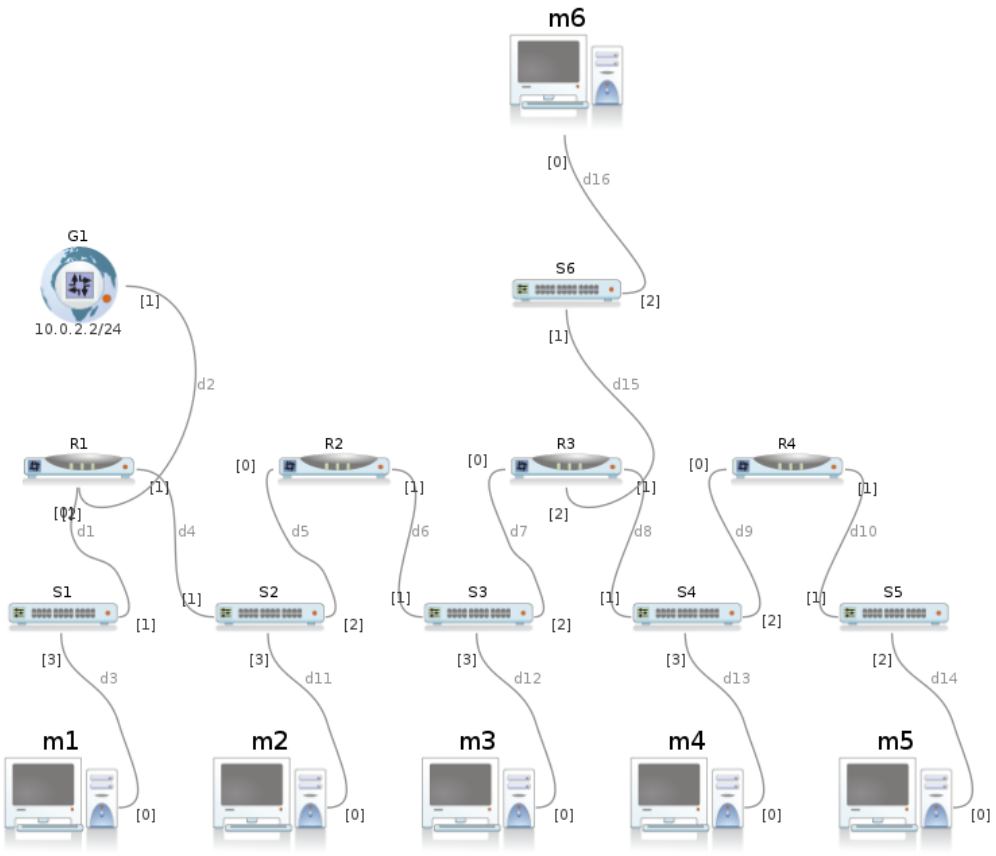


FIGURE 3 – Réseau à mettre en place au cours du TP

La *gateway* sert à sortir vers le réseau extérieur. À chaque switch correspond un sous-réseau :
- 10.0.10.0/24 sur le switch 1
- 10.0.20.0/24 sur le switch 2
- 10.0.30.0/24 sur le switch 3
- 10.0.40.0/24 sur le switch 4
- 10.0.50.0/24 sur le switch 5
- 10.0.60.0/24 sur le switch 6

1. À quels sous-réseaux chaque routeur est-il relié ?
2. Proposez une adresse pour les interfaces réseaux de chaque routeur et de chaque machine.

1.1 Tables de routage des routeurs

1. Depuis le routeur R1, par quelles routes doivent transiter les paquets pour atteindre chacun des autres réseaux ? Pour atteindre l'extérieur du réseau ?
2. Déduisez-en la table de routage de R1 et configurez R1.
3. Depuis le routeur R2, par quelles routes doivent transiter les paquets pour atteindre chacun des autres réseaux ? Pour atteindre l'extérieur du réseau ?
4. Déduisez-en la table de routage de R2 et configurez R2.
5. Depuis le routeur R3, par quelles routes doivent transiter les paquets pour atteindre chacun des autres réseaux ? Pour atteindre l'extérieur du réseau ?
6. Déduisez-en la table de routage de R3 et configurez R3.
7. Depuis le routeur R4, par quelles routes doivent transiter les paquets pour atteindre chacun des autres réseaux ? Pour atteindre l'extérieur du réseau ?
8. Déduisez-en la table de routage de R4 et configurez R4.
9. En utilisant `ping`, assurez-vous que les routeurs arrivent à communiquer les uns avec les autres.

1.2 Tables de routage des machines

1. Quels sont les réseaux accessibles directement par la machine M1 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
2. Déduisez-en la table de routage de M1 et configurez-la.
3. Quels sont les réseaux accessibles directement par la machine M2 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
4. Déduisez-en la table de routage de M2 et configurez-la.
5. Quels sont les réseaux accessibles directement par la machine M3 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
6. Déduisez-en la table de routage de M3 et configurez-la.
7. Quels sont les réseaux accessibles directement par la machine M4 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
8. Déduisez-en la table de routage de M4 et configurez-la.
9. Quels sont les réseaux accessibles directement par la machine M5 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
10. Déduisez-en la table de routage de M5 et configurez-la.
11. Quels sont les réseaux accessibles directement par la machine M6 ? Par quelles routes doivent transiter les paquets qu'elle envoie pour atteindre chacun des autres réseaux et l'extérieur ?
12. Déduisez-en la table de routage de M6 et configurez-la.
13. En utilisant `ping`, assurez-vous que les machiness arrivent toutes à communiquer les unes avec les autres.

2 Utilisation de liaisons directes entre les routeurs

Modifiez le réseau afin d'obtenir une configuration équivalente à celle étudiée dans le TP 3. Les quatre routeurs sont reliés les uns autres autres avec une liaison directe, et les commutateurs ne sont reliés qu'à un seul routeur à la fois.

- R1 est relié à R2, à la gateway et à S1.
- R2 est relié à R1, R3 et à S2.
- R3 est relié à R2, R4, S3 et à S6.
- R4 est relié à R3, S4 et à S5.

Les commutateurs S1, S2, S3, S4, S5 et S6 relient les mêmes sous-réseaux qu'à la question précédente.

Le réseau que vous allez construire dans cette partie ressemble à la figure 4.

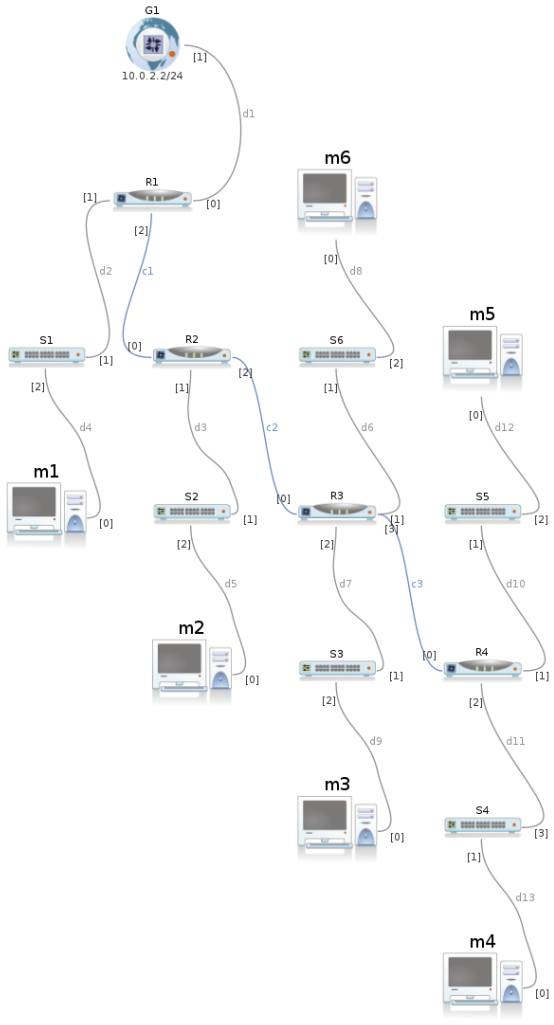


FIGURE 4 – Réseau à mettre en place dans la deuxième partie du TP

1. Pour chaque liaison directe entre deux routeurs, vous allez devoir utiliser un petit réseau qui ne contiendra que les deux routeurs concernés. Vous pourrez utiliser :
   - 10.0.3.0/24 entre R1 et R2
   - 10.0.4.0/24 entre R2 et R3

– 10.0.5.0/24 entre R3 et R4
À quels réseaux sont reliés les routeurs ?

2. Proposez une adresse pour chaque interface réseau des routeurs.
3. Comme vu dans le TD 3, configurez les tables de routage des routeurs et des machines de ce réseau.
4. En utilisant `ping`, assurez-vous que les routeurs et les machines arrivent à communiquer les uns avec les autres.
5. Comparez cette configuration avec la précédente : est-elle plus simple à administrer ? Plus simple à mettre en œuvre ? Quelle est la différence au point de vue de l'administration des machines et des équipements réseaux ?

3 Utilisation d'un backbone

Modifiez le réseau afin d'utiliser un backbone : vos quatre routeurs doivent désormais être reliés entre eux par un commutateur. Vous devez introduire un commutateur S0 qui relie R1, R2, R3 et R4. Les commutateurs S1 à S6 restent, eux, reliés au même routeur que dans le réseau précédent. On introduit alors un nouveau sous-réseau pour le backbone : vous pourrez prendre l'adresse réseau 10.0.1.0/24.

Le réseau que vous allez construire dans cette partie ressemble à la figure 5.

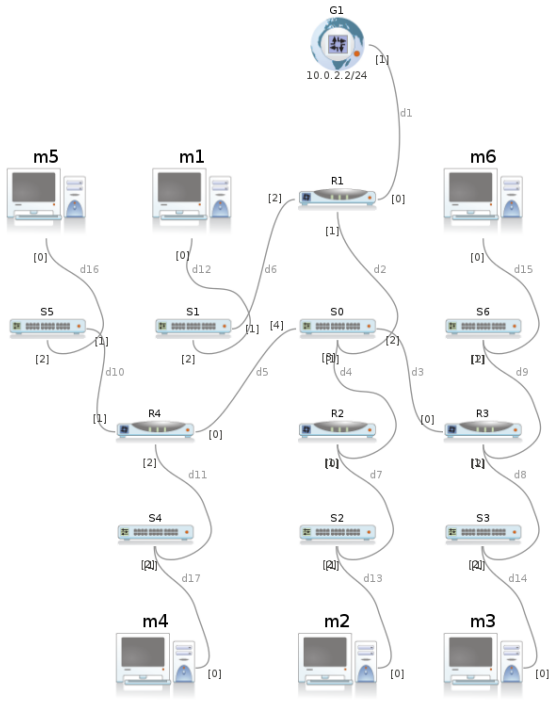


FIGURE 5 – Réseau à mettre en place dans la troisième partie du TP

1. À quels réseaux sont reliés les routeurs ?
2. Proposez une adresse pour chaque interface réseau des routeurs.
3. Comme vu dans le TD 3, configurez les tables de routage des routeurs et des machines de ce réseau.

4. En utilisant `ping`, assurez-vous que les routeurs et les machines arrivent à communiquer les uns avec les autres.
5. Comparez cette configuration avec les deux précédentes : est-elle plus simple à administrer ? Plus simple à mettre en œuvre ? Quelle est la différence au point de vue de l'administration des machines et des équipements réseaux ?

3 Routage inter-VLANs et filtrage en DMZ

Le but de ce TP est de segmenter un réseau local en deux sous-réseaux virtuels ou VLANs dont les machines ne sont pas contiguës physiquement, et de mettre en place un filtrage entre ces deux sous-réseaux afin de placer des serveurs en zone démilitarisée dans l'un des deux.

Ce TP se déroule grâce au simulateur de réseaux marionnet disponible gratuitement sous licence GPL, voir `www.marionnet.org`. Rappel : le mot de passe de root d'une machine virtuelle est `root`.

1 Mise en place du réseau et des VLANs

Créez le réseau suivant :
- Un routeur
- Deux commutateurs : l'un comportant quatre ports et l'autre comportant six ports
- Six machines

Le routeur est connecté au commutateur disposant de six ports avec deux câbles. Les deux commutateurs sont reliés par un câble. Les machines A1, A2 et B1 sont reliées au commutateur disposant de six ports ; les machines A3, B2 et B3 sont reliées à l'autre commutateur.

Attention, lorsque vous créerez vos commutateurs et votre routeur, cochez la case "show Unix terminal" et la case "show VDE terminal". Soyez très attentifs aux numéros de ports utilisés par les câbles branchés à vos commutateurs.

Le réseau que vous obtenez ressemble à la figure 6.

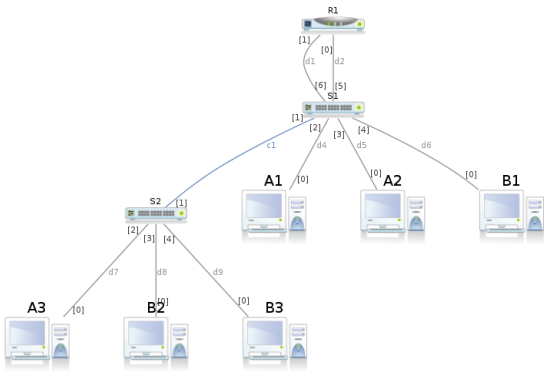


FIGURE 6 – Réseau à mettre en place pour le TP

Les machines A1, A2 et A3 sont dans le VLAN 1, d'adresse 192.168.10.0/24, et les machines B1, B2 et B3 sont dans le VLAN 2, d'adresse 192.168.20.0/24.

1. Configurez les interfaces réseaux des machines et du routeur
2. En utilisant `ping`, vérifiez que A1 et A2 peuvent communiquer entre elles. Peuvent-elles communiquer avec A3 ? pourquoi ?
3. Dans votre configuration, quels sont les ports de chaque commutateur faisant partie du VLAN 1 ? Quels sont ceux qui font partie du VLAN 2 ?
4. Les commutateurs utilisent l'interface VDE. Avec la commande `help` vous pouvez obtenir la liste des commandes disponibles et une description de chacune. Avec la commande `vlan/create`, créez deux VLANs 1 et 2 sur vos deux commutateurs.

5. On ajoute un port dans un VLAN avec la commande `port/setvlan`. D'après la liste des ports que vous avez établie, configurez vos commutateurs pour mettre les ports dans les VLANs idoines (sauf la liaison entre les deux commutateurs).
6. Vous pouvez afficher l'état de la configuration de vos VLANs avec la commande `vlan/print`.
7. La mise en place d'un lien entre les deux commutateurs dans le trunk se fait en deux étapes :
   – On met le port dans un VLAN avec la commande `port/setvlan`
   – On définit un port inter-VLAN avec la commande `vlan/addport`
   Configurez la liaison entre vos deux commutateurs afin de rendre possible les communications entre machines au sein d'un même VLAN mais n'étant pas reliées au même commutateur.
8. Sur chaque machine, configurez les tables de routage pour utiliser le routeur comme route par défaut.
9. Affichez l'état des VLANs configurés sur vos commutateurs et vérifiez que toutes les machines et le routeur peuvent communiquer les uns avec les avec les autres.

2 Mise en place de serveurs en DMZ

Dans cette partie du TP, vous allez mettre en place des serveurs dans le VLAN 2 (machines dont le nom commence par B). Vous allez ensuite configurer des règles de filtrage au niveau du routeur afin de limiter les possibilités de communications entre les deux VLANs : ainsi, le VLAN 2 peut être considéré comme une zone démilitarisée dans laquelle vous avez des serveurs.

2.1 Démarrage des serveurs

Vous pourrez démarrer ces serveurs avec les commandes suivantes :
– Serveur Apache (serveur HTTP) :
```
B1:~# /etc/init.d/apache2 start
```

– Serveur SSH :
```
B1:~# /etc/init.d/ssh start
```

1. Démarrez le serveur Apache (serveur HTTP) sur la machine B1. Ignorez le message d'avertissement.
2. Démarrez le serveur SSH sur les machines B1 et B3.
3. La commande `netstat` permet d'afficher les communications réseaux en cours (ouvertes, établies ou en cours de fermeture) sur la machine. On l'utilise souvent avec la combinaison d'options `-lapute`. À quoi correspond la commande `netstat -lapute` ? Exécutez-la sur la machine B1 et vérifiez que les serveurs Apache et SSH sont bien en attente de connexions.
4. Sur votre serveur Apache, les fichiers mis à disposition par le serveur HTTP sont situés dans le répertoire `/var/www`. Créez un fichier texte dans ce répertoire et écrivez quelque chose dedans.
5. Sur la machine A1, utilisez `wget` pour télécharger ce fichier. Vérifiez également que vous pouvez vous connecter sur la machine B3 en utilisant `ssh`. Vous veillerez à vous déconnecter de votre session SSH avec Ctrl+D.

```
A1:~# wget 192.168.20.1/monfichier
A1:~# ssh 192.168.20.3
```

2.2 Filtrage entre les VLANs

Sur le routeur, vous allez utiliser le pare-feu `iptables` pour filtrer le communications entre les deux VLANs.

1. Vous pouvez à tout moment utiliser `iptables -L` pour afficher les règles que vous avez établies. Quelles sont les règles définies pour le moment ?
2. Il existe trois chaînes :
   - `INPUT` : concerne les paquets entrant dans le pare-feu (qui lui sont adressés)
   - `OUTPUT` : concerne les paquets sortant du pare-feu
   - `FORWARD` : concerne les paquets passant d'une interface à l'autre du pare-feu

   C'est cette dernière chaîne qui va vous intéresser particulièrement dans ce TP. Pourquoi ?
3. En ajoutant l'option `--line-numbers`, vous pouvez afficher les numéros de règles. C'est utile notamment en cas d'erreur, pour pouvoir supprimer une règle avec la commande `iptables -D <chaine> <numero>`.
4. On souhaite appliquer un filtrage qui, par défaut, jette les paquets sans envoyer de message d'erreur (politique DROP). La politique par défaut d'une chaîne se définit de la façon suivante :
   `/sbin/iptables -P <chaine> <politique>`

   Appliquez cette politique par défaut aux trois chaînes et vérifiez qu'elle est bien prise en compte en affichant les règles en place sur votre pare-feu.
5. Pouvez-vous toujours vos connecter en SSH de la machine A1 sur la machine B1 ? Pouvez-vous toujours récupérer des fichiers avec `wget` ? Pouvez-vous envoyer des messages ICMP avec `ping` ? Que se passe-t-il quand vous essayez de le faire ?
6. Le filtrage va autoriser toutes les communications TCP sur le port 80 entre les deux VLANs. Cette autorisation se fait en deux temps :
   – Autorisation de toutes les communications ayant le port 80 pour port de destination
   – Autorisation de toutes les communications ayant le port 80 pour port source

   On utilise `iptables` avec les options suivantes :
   - `-A <chaine>` : spécifie la chaîne concernée
   - `-p <protocole>` : spécifie le protocole de communications
   - `--sport <port>` ou `--dport <port>` : spécifie le port source ou destination
   - `-J ACCEPT` : spécifie que ces communications sont acceptées

   Autorisez les communications avec le port 80.
7. Vérifiez que vous pouvez maintenant télécharger votre fichier depuis une machine du VLAN 1.
8. Vous allez maintenant autoriser la machine B2 (et uniquement la machine B2) à communiquer avec le protocole ICMP. Elle sera donc utilisée pour répondre à `ping`. Adaptez la commande précédentes en sachant que :
   - Le protocole concerné est `icmp`
   - `-d <adresse ip>` : spécifie l'adresse IP de destination des paquets concernés
   - `-s <adresse ip>` : spécifie l'adresse IP source des paquets concernés

   Vérifiez ensuite que la machine A1 peut bien communiquer avec la machine B2 en utilisant `ping`.
9. On souhaite que seul le serveur SSH de B3 soit accessible depuis l'extérieur du VLAN. Pour cela, vous allez autoriser les communications TCP utilisant le port 22 uniquement si elles impliquent la machine B3. Quelles commandes `iptables` utilisez-vous ? Vérifiez que vous pouvez bien vous connecter en SSH depuis A1 sur B3 mais pas sur B1.
10. D'après `iptables -L`, quelles sont les règles en place sur votre pare-feu ?

4 Routage dynamique avec RIP

Le but de ce TP est de mettre en place un réseau composé de plusieurs routeurs et d'observer la configuration automatique de leurs tables de routage avec le protocole de routage dynamique RIP au cours de l'évolution du réseau.

Ce TP se déroule grâce au simulateur de réseaux marionnet disponible gratuitement sous licence GPL, voir `www.marionnet.org`. Rappel : le mot de passe de root d'une machine virtuelle est `root`.

1 Mise en place du réseau et configuration des sous-réseaux locaux

Créez le réseau suivant :

- Trois routeur : R1, R2 et R3
- Quatre commutateurs : S1 et S2 connectés à R1, S3 connecté à R2 et S4 connecté à R3
- Quatre machines, chacune connectée à un commutateur
- R1 et R3 sont chacun reliés à R2.

Attention, lorsque vous créerez vos routeurs, cochez la case "show Unix terminal". Laissez les commutateurs tels qu'ils sont.

Le réseau que vous obtenez ressemble à la figure 7.

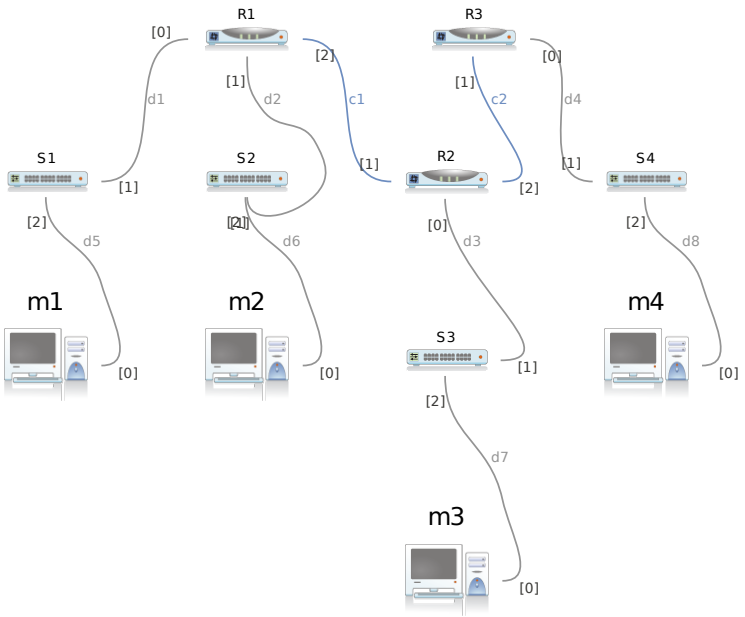


FIGURE 7 – Réseau à mettre en place pour le début du TP

Les sous-réseaux utilisés sont les suivants :

- La machine m1 est dans le sous-réseau 192.168.10.0/24
- La machine m2 est dans le sous-réseau 192.168.20.0/24
- La machine m3 est dans le sous-réseau 192.168.30.0/24
- La machine m4 est dans le sous-réseau 192.168.40.0/24
- R1 et R2 sont dans le sous-réseau 192.168.1.0/24
- R2 et R3 sont dans le sous-réseau 192.168.2.0/24

1. Configurez les interfaces réseaux des machines et leur passerelle par défaut.
2. Configurez les interfaces réseaux des routeurs. Ne touchez pas à leurs tables de routage !
3. Les protocoles de routage dynamique sont mis en place sur les routeurs par un programme appelé Quagga. La liste des protocoles qui doivent être activés est définie dans le fichier `/etc/quagga/daemons`. Éditez ce fichier et n'activez que `zebra` et `ripd`. Effectuez cette modification sur les trois routeurs.
4. Àchaque fois que vous effectuez une modification dans la configuration de Quagga (ou d'un de ses composants), vous devez impérativement le redémarrer afin que ces modifications soient prises en compte. Sur les trois routeurs, redémarrez Quagga et observez les affichages.
5. Avec la commande `ps -ef | grep quagga`, observez quels programmes tournent pour Quagga.
   La configuration de chaque démon en charge d'un protocole est située dans un fichier spécifique qui porte son nom situé dans le répertoire `/etc/quagga`. Quels sont les fichiers qui vont nous intéresser ici ?

2 Routes statiques avec Zebra

Les routes statiques de chaque routeur, c'est-à-dire les réseaux auxquel il est directement, sont découvertes par Zebra. Dans cette partie, nous allons observer ce que Zebra voit sur nos trois routeurs.

1. Vous avez déterminé dans la partie précédente dans quel fichier est configuré le démon en charge de Zebra. Éditez ce fichier : il contient un exemple de fichier de configuration. Modifiez-le pour utiliser le vrai hostname des routeurs. Effectuez cette opération sur les trois routeurs.
2. Pour configurer le démon Zebra, on s'y connecte avec l'utilitaire Telnet. Il écoute sur le port 2601. Utilisez le mot de passe que vous avez défini dans le fichier de configuration.

   ```
   root@R1:~# telnet localhost 2601
   Trying 127.0.0.1...
   Connected to localhost.
   Escape character is '^]
   '.

   Hello, this is Quagga (version 0.99.5).
   Copyright 1996-2005 Kunihiro Ishiguro, et al.


   User Access Verification

   Password:
   R1>
   ```

   Une invite de commande s'affiche. La commande `enable` permet de passer en mode administrateur :

```
R1> enable
Password:
R1#
```

Lorsque l'invite de commande se termine par un dièse (`#`), cela signifie que vous êtes en mode administrateur.
Vous pouvez afficher la liste des commandes disponibles en tapant un point d'interrogation (`?`).

```
Afficher la liste des commandes disponibles : ?

R1# ?
  configure  Configuration from vty interface
  copy       Copy configuration
  debug      Debugging functions (see also 'undebug')
  disable    Turn off privileged mode command
  echo       Echo a message back to the vty
  end        End current mode and change to enable mode.
  exit       Exit current mode and down to previous mode
  help       Description of the interactive help system
  list       Print command list
  logmsg     Send a message to enabled logging destinations
  no         Negate a command or set its defaults
  quit       Exit current mode and down to previous mode
  show       Show running system information
  terminal   Set terminal line parameters
  who        Display who is on vty
  write      Write running configuration to memory, network, or terminal
```

La commande `show interface ethX`, avec *ethX* une interface réseau de votre routeur, vous pouvez visualiser la configuration et l'état d'une interface :

```
R1# show interface eth0
Interface eth0 is up, line protocol detection is disabled
  index 1 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 02:04:06:5b:68:5b
  inet 192.168.10.254/24 broadcast 192.168.10.255
  inet6 fe80::4:6ff:fe5b:685b/64
    37 input packets (0 multicast), 942 bytes, 0 dropped
    0 input errors, 0 length, 0 overrun, 0 CRC, 0 frame
    0 fifo, 0 missed
    43 output packets, 1652 bytes, 0 dropped
    0 output errors, 0 aborted, 0 carrier, 0 fifo, 0 heartbeat
    0 window, 0 collisions
```

La commande `show ip route` permet d'afficher les routes statiques reconnues par Zebra. Ne tenez pas compte de l'interface `eth42` et du réseau 172.23.0.0/16 : il s'agit d'une interface utilisée en interne par Marionnet. Sur chacun de vos trois routeurs, quelles sont les routes vues par Zebra ?

3. Déconnectez-vous de Zebra avec Ctrl+D ou `quit`.

3 Routage dynamique avec RIP

1. De la même manière que pour configurer Zebra, éditez le fichier de configuration de RIP afin d'attribuer le bon hostname sur chacun de vos routeurs. N'oubliez pas de redémarrer

Quagga après votre modification. Effectuez cette opération sur les trois routeurs.

2. Pour configurer le démon RIP, on s'y connecte également avec l'utilitaire `telnet`, cette fois sur le porte 2602. Utilisez le mot de passe que vous avez défini dans le fichier de configuration.

```
root@R1:~# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.5).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
R1>
```

   On passe en mode administrateur avec la commande `enable`. Puis on passe en mode configuration avec `configure terminal`. Enfin, on accède à la configuration spécifique de RIP avec `router rip` :

```
R1> enable
Password:
R1# configure terminal
R1(config)#
R1(config)# router rip
R1(config-router)#
```

3. À quels réseaux est relié directement chacun de vos routeurs ?
4. Vous devez déclarer chaque réseau relié directement au routeur avec la commande `network` suivie de l'adresse du réseau (en notation CIDR). Effectuez cette opération sur le routeur R1 uniquement.
5. Sortez du mode configuration avec Ctrl+D jusqu'à avoir juste l'invite de commande administrateur de Quagga : `R1#`
6. Avec la commande `show ip rip`, affichez les réseaux connus de R1. Lesquels voit-on ? Ici encore, vous allez voir le réseau interne de Marionnet 172.23.0.0/16 : ne vous en préoccupez toujours pas.
7. Effectuez la même manipulation sur les deux autres routeurs. Une fois le routeur R2 configuré, attendez 30 secondes et tapez à nouveau la commande `show ip rip` sur R1. Que constatz-vous ?
8. Configurez les trois routeurs et assurez-vous avec `show ip rip` que tous les routeurs voient bien tous les sous-réseaux.
9. Sortez de l interface de configuration de Quagga et affichez la table de routage de chaque routeur. Que constatez-vous ?
10. Assurez-vous avec ping que les machines peuvent s contacter entre elles.

4 Ajout d'un routeur à chaud

Ajoutez un routeur R4 dans le systèmeme, relié à R1 et R3. Le réseau que vous obtenez ressemble à la figure 8.

Ce routeur est relié à R1 par le sous-réseau 192.168.3.0/24 et à R3 par le sous-réseau 192.168.4.0/24.

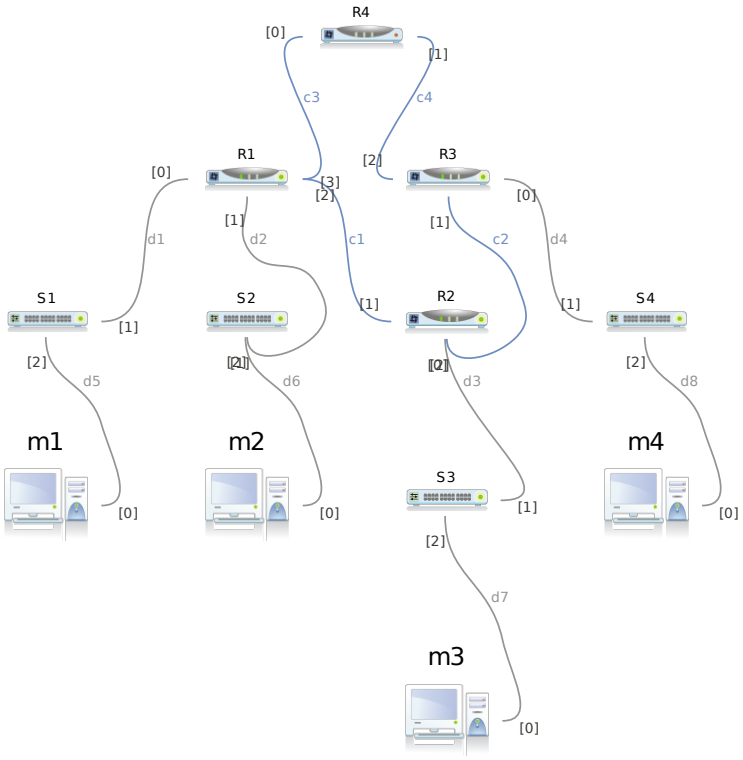


FIGURE 8 – Réseau à mettre en place pour la suite du TP

1. Configurez Zebra et RIP sur R4 en ajoutant dans RIP les deux nouveaux sous-réseaux qui le relient à R1 et à R3.
2. Attendez quelques minutes (2 ou 3 suffiront) et affichez la table de routage de tous les routeurs. Que constatez-vous ? Pourquoi ?
3. Sur R1 et R3 (les deux routeurs reliés aux sous-réseaux qui connectent R4), ajoutez les nouveaux sous-réseaux dans les réseaux diffusés par RIP.
4. Affichez à nouveau les tables de routage des quatre routeurs. Que constatez-vous ? Pourquoi ?
5. Dans les tables de routage, vous voyez deux colonnes que vous n'avez pas vues jusqu'à maintenant avec le routage statique : il s'agit de "Metric"et "Ref". À votre avis, à quoi correspond la colonne "metric" ?

5 Tolérance aux pannes et résilience

1. R3 dispose d adresses IP sur deux réseaux entre routeurs : une adresse sur le réseau entre R2 et R3 et une adresse sur le réseau entre R4 et R3. Depuis R1, effectuez un `traceroute` vers R3 en utilisant chacune de ces deux adresses. Que constatez-vous ?
2. Débranchez R2 du réseau. Vous pouvez le faire de façon "pas trop propre" : commencez par débrancher les câbles qui le relient à R1 et à R3, puis éteignez-le. En exécutant la commande `route` sur les routeurs restants, observez l'évolution des tables de routage de ceux-ci. Par

exemple, vous pouvez exécuter la boucle suivante qui effectue un appel à `route` toutes les 20 secondes :

```
for t in ‘seq 1 10 ‘ ; do route ; sleep 20 ; done
```

Quelles modifications sont faites dans les tables de routage ?

3. Depuis R1, essayez à nouveau d'exécuter les deux `traceroute` vers R3. Que remarquez-vous ?

5 Initiation à IPv6

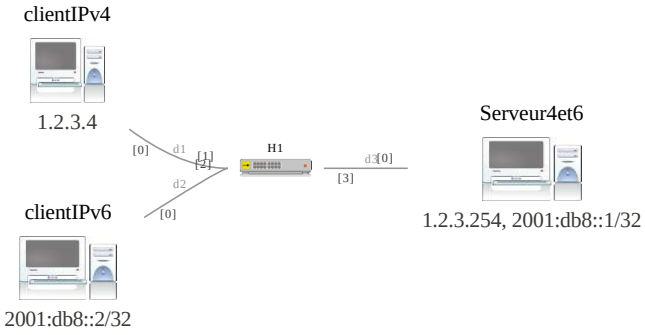
(Franck Butelle : `franck.butelle@iutv.univ-paris13.fr`)

Ce TP utilise marionnet, voir `http://www.marionnet.org` avec deux types de machine virtuelle : Debian (image standard pour marionnet) et Mandriva 2010 (disponible en téléchargement sur `http://www.marionnet.org`).

1 Réaliser un réseau mixte IPv4/IPv6

1.1 Mise en place du réseau

D'abord réalisez le réseau suivant :



Sachant que le `clientIPv4` doit utiliser l'image mandriva (l'image debian par défaut de marionnet n'a pas les commandes nécessaires), les autres postes auront la configuration par défaut (debian lenny).

Constatez que tous les postes après démarrage ont au moins une adresse IPv6 (commande `ifconfig eth0`) de "scope" Link (lien) sauf pour mandriva qui n'a pas été configuré pour démarrer le service réseau. Pour cette dernière il faut faire `ifconfig eth0 up`.

Pour désactiver les fonctionnalités IPv6 d'un système linux, il est préférable de ne pas charger le module `ipv6`, on ne peut faire ainsi avec marionnet, on va se contenter de supprimer les adresses IPv6 associées à eth0. Supprimez sur le clientIPv4 les adresses IPv6 de la carte eth0 : `ip -6 addr flush dev eth0 scope link`

Puis fixez les adresses comme spécifié sur le schéma.

Rappel : configuration des adresses IPv4 : `ifconfig eth0 <adresseIPv4>` et pour les adresses IPv6 : `ifconfig eth0 inet6 add <adresseIPv6>` ou `ip addr add <adresseIPv6> dev eth0`. Pensez à bien fixer les deux adresses pour eth0 de `serveur4et6`.

1.2 Vérifications de base

Observez les tables de routage IPv4 par la commande `route` et IPv6 par `route -A inet6`. Cherchez dans le cours à quoi correspondent les adresses commençant par `ff00` et `fe80` et l'adresse `::1/128`.

Vérifiez que vous pouvez utiliser `ping` pour la boucle locale (et `ping6` pour IPv6). Quelles sont les commandes correspondantes ?

Testez par ping la connectivité IPv4 et (indépendamment) IPv6.

Remarque : on peut forcer ping6 à utiliser une interface ou une autre pour pouvoir utiliser un adresse de type lien (link) : `ping6 -I eth0 <adresseIPv6Link>` .

Lancez wireshark sur le serveur et lancez depuis `clientIPv6` un ping vers l'adresse de scope global du serveur. Vous devriez observer le protocole ICMPv6 d'abord Neighbor advertisement (en multicast au niveau Ethernet) suivi de Neighbor advertisement (l'équivalent de ARP-request/ARP-reply en IPv4). puis les echo request/echo reply classiques mais en IPv6.

Comment wireshark sait-il que c'est IPv6 qui est encapsulé dans Ethernet ? Et que IPv6 encapsule ICMPv6 ?

On peut aussi faire de la découverte de voisins IPv6 par `ping6 -I eth0 ff02::1`. Par l'analyseur vous pouvez vérifier que, là encore, ce n'est pas du broadcast mais du multicast.

1.3 Serveur Web double pile TCP/IP

Notre objectif est maintenant d'avoir un serveur web qui tourne sur la machine serveur et qui puisse répondre de façon différente aux clients IPv4 et aux clients IPv6.

(re)démarrez le service apache2 sur le serveur par `/etc/init.d/apache2 restart`.

Depuis le client IPv4 vous pouvez utiliser lynx <adresseIPv4serveur> pour afficher la page d'accueil (*It works !*). Malheureusement lynx ne comprends pas les URL en IPv6. Vous pouvez tester que le serveur répond en IPv6 par firefox (avec comme URL `http://[<adresseIPv6>]` ou par une simple connexion sur son port http (80/TCP) (on se fait passer pour un navigateur respectant le procotole HTTP/1.0) :
`telnet <adresseIPv6> http`

Supposons que la connexion soit acceptée (sinon vous avez un problème !), dans la connexion tapez :
`GET / HTTP/1.0`

puis deux retour chariot ("Carriage return" ou CR). Normalement vous devriez avoir le contenu de la réponse HTTP qui doit être traitée par le navigateur.

Pour que le serveur web (apache2) sur `serveur4et6` réponde de façon différente suivant le client, il faut utiliser les virtualhost de apache2. Supprimez le fichier `/etc/apache2/sites-enabled/000-default` (c'est un lien symbolique en fait), puis créez les fichiers V4 et V6 dans `/etc/apache2/sites-enabled` avec respectivement comme contenu les éléments suivants :

```
#fichier V4
<VirtualHost 1.2.3.254>
DocumentRoot /var/www
</VirtualHost>
```

```
#fichier V6
<VirtualHost [2001:db8::1]>
DocumentRoot /var/www6
</VirtualHost>
```

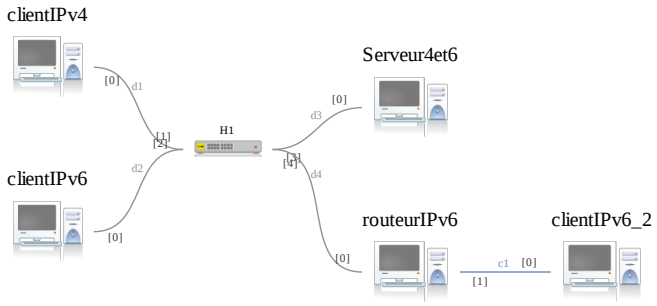
Les adresses IPv6 doivent être entourées de crochets car dans le cas contraire, un éventuel port optionnel ne pourrait pas être déterminé.

Pensez à créer `/var/www6` et mettre un fichier `index.html` dedans par exemple avec "Bienvenue client IPv6 !" Relancez le service web par `/etc/init.d/apache2 restart`.

Vous pouvez vérifier que votre serveur web est en fonction avec `netstat -atlp` (tcp6 chez Debian siginifie TCP sur IPv6).

2 Routage statique en IPv6

Ajoutez deux machines virtuelle Mandriva, `RouteurIPv6` avec **2** cartes réseaux et `clientIPv6_2` et réalisez le schéma suivant :



Attention, le câble qui relie routeurIPv6 et clientIPv6_2 est un câble croisé. L'objectif est de faire en sorte que clientIPv6_2 puisse se connecter à `serveur4et6`.

On demande à ce que la carte réseau du côté droit (eth1) du routeurIPv6 soit en `2001:db9::ff/32`. Choisissez une adresse cohérente pour sa carte eth0 et de même pour la carte réseau clientIPv6_2.

Le routage statique en IPv6 se configure presque comme celui d'IPv4, à ceci près qu'il ne faut pas oublier l'option `-A inet6` à la commande `route` (voir man route). Comment s'écrit l'ajout d'une route par défaut ?

L'activation du routage IPv6 au niveau du noyau se fait par `sysctl -w net.ipv6.conf.all.forwarding=1`. Testez le routage statique de bout en bout avec ping6.

3 Autoconfiguration

Un des apports importants de IPv6, c'est l'autoconfiguration sans état (stateless, c'est à dire sans trace, sans DHCP). Normalement un vrai routeur moderne est pré-équipé pour cela, mais ici, pour une machine Linux il faut ajouter le démon radvd.

Connectez temporairement la carte réseau eth0 du routeurIPv6 à une "real world gateway" avec les réglages par défaut. Faire `dhclient eth0` pour récupérer une adresse IPv4 et une route vers l'extérieur par DHCP. Ensuite faites `urpmi radvd` pour aller chercher le package sur internet et l'installer (les dépôts par défaut ont déjà été configurés).

Une fois le démon installé, supprimez la "real world gateway" et reliez à nouveau le routeur à H1.

Lors de son installation, il est possible que le démon donne plusieurs avertissements, en particulier que le réseau ne semble pas démarré par défaut — ce qui est exact. Son fichier de configuration est `/etc/radvd.conf` : modifiez-le pour qu'il fournisse des adresses en `2001:8::/64` (et non `2001:db8::/32` pour voir la différence) sur eth0 et `2001:9::/64` sur eth1.

Démarrez wireshark sur un client IPv6 avant de redémarrer le démon par `/etc/init.d/radvd restart`. Vous devriez observer les annonces, lancées de temps en temps, par le routeur. Les cartes réseaux des clients seront autoconfigurées si vous faites `ifconfig eth0 down` puis `ifconfig eth0 up`. Donnez les grandes étapes de ce protocole.